MODUL
PROGRAMMING II

FAKULTAS TEKNIK DAN INFORMATIKA

UNIVERSITAS BINA SARANA INFORMATI

KATA PENGANTAR

Segala puji dan syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa, karena berkat rahmat-Nya penulisan modul Mata Kuliah Web Programming II ini dapat terselesaikan dengan baik. Modul ini disusun untuk memenuhi kebutuhan mahasiswa dalam mata kuliah Web Programming II, yang disajikan dalam bentuk praktikum dan diharapkan dapat membekali mahasiswa dalam memahami pembuatan web menggunakan Framework Laravel.

Modul Web Programming II ini membahas materi yang dibatasi hingga pembuatan halaman Administrator (Back-End). Di akhir perkuliahan, mahasiswa diharapkan mampu mengimplementasikan materi yang telah dipelajari dalam bentuk final project yang harus dipresentasikan sebagai syarat kelulusan mata kuliah Web Programming II. Teknik penyajian dalam modul ini dilakukan secara terpadu dan sistematis.

Sebagai sebuah modul pembelajaran, pembahasan diawali dengan menjelaskan target pembelajaran yang hendak dicapai. Dengan demikian, pengguna modul ini dapat secara mandiri mengukur tingkat pemahaman dan ketuntasan yang telah dicapai.

Kami menyadari bahwa modul ini masih memiliki banyak kekurangan. Oleh karena itu, kami dengan lapang dada menerima segala masukan dan kritik yang konstruktif dari berbagai pihak demi kesempurnaan modul ini di masa yang akan datang. Semoga modul ini dapat bermanfaat bagi para pembaca.

Jakarta, September 2024

Tim Penulis

DAFTAR ISI

Minggu Ke-1
Instalasi & Konfigurasi Pada Laravel

Laravel adalah salah satu framework PHP yang paling populer dan banyak digunakan untuk pengembangan aplikasi web. Diciptakan oleh Taylor Otwell, Laravel dirancang untuk membuat pengembangan web menjadi lebih mudah dan lebih cepat dengan menyediakan berbagai alat dan fitur bawaan.
Fitur Utama Laravel:

1. **Eloquent ORM**: Laravel memiliki Object-Relational Mapping (ORM) yang sangat kuat bernama Eloquent. Ini memungkinkan pengembang untuk berinteraksi dengan basis data menggunakan model dan relasi, serta menyediakan metode yang mudah digunakan untuk melakukan operasi CRUD.
2. **Blade Templating Engine**: Laravel memiliki sistem template yang disebut Blade. Blade menggunakan kode PHP di dalam template dengan sintaks yang sederhana dan rapi, serta mendukung pewarisan template dan komponen.
3. **Routing**: Laravel menyediakan sistem routing yang fleksibel dan mudah digunakan. Kita dapat menentukan rute untuk aplikasi dengan cara yang sangat sederhana dan membaca URL.
4. **Middleware**: Middleware di Laravel memungkinkan memfilter permintaan HTTP yang masuk ke aplikasi. Ini sangat berguna untuk memeriksa autentikasi, logging, dan berbagai kebutuhan lainnya.
5. **Artisan CLI**: Laravel menyediakan alat baris perintah yang kuat bernama Artisan. Artisan membantu dengan berbagai tugas pengembangan seperti pembuatan model, migrasi, dan kontroler, serta memungkinkan untuk membuat perintah kustom.
6. **Migration dan Seeding**: Laravel menyediakan sistem migrasi basis data yang dapat melacak dan mengelola perubahan skema basis data dengan mudah. Seeding dapat mengisi basis data dengan data dummy untuk pengujian.
7. **Autentikasi dan Otentikasi**: Laravel menyediakan sistem autentikasi dan otorisasi bawaan yang memudahkan pengembang untuk menyiapkan login, registrasi, dan kontrol akses pengguna.
8. **Event Broadcasting**: Laravel memungkinkan untuk menyiarkan event aplikasi ke klien waktu nyata menggunakan WebSocket dengan driver seperti Pusher dan Redis.
9. **Testing**: Laravel dibangun dengan testing dalam pikiran dan mendukung PHPUnit secara bawaan. Ini memungkinkan kita dapat menulis tes unit dan fitur untuk aplikasi.

Software Development adalah proses pembuatan, desain, deployment, dan pemeliharaan aplikasi perangkat lunak. Ini melibatkan berbagai langkah dan metodologi untuk memastikan bahwa perangkat lunak yang dibangun memenuhi kebutuhan pengguna dan standar kualitas yang diinginkan.

Sertifikasi Kompetensi adalah proses pengakuan formal yang diberikan kepada individu yang telah menunjukkan kemampuan, pengetahuan, dan keterampilan dalam bidang tertentu sesuai dengan standar yang ditetapkan. Sertifikasi ini biasanya diberikan oleh lembaga atau badan sertifikasi yang diakui secara nasional atau internasional. Salah satu manfaat sertifikasi kompetensi bagi Developer memberikan pengakuan resmi terhadap keahlian dan pengetahuan

teknis seorang developer, yang dapat meningkatkan kredibilitas mereka di mata perusahaan atau klien.

Sertifikasi kompetensi diselenggarakan oleh Lembaga Sertifikasi (LSP) yang telah mendapat lisensi dari Badan Nasional Sertifikasi Profesi (BNSP), seperti **LSP Universitas Bina Sarana Informatika**. Berikut adalah salah satu skema yaitu **skema programmer** pada tabel 1.1 & **skema Analis Program** pada tabel 1.2

Tabel I. 1
Skema Programmer

No	Kode Unit	Judul Unit	Jenis Standar (Standar Khusus/Standar Internasional/SKKNI)
1.	J. 620100.017.02	Mengimplementasikan Pemrograman Terstruktur	**SKKNI**
2.	J. 620100.016.01	Menulis Kode Dengan Prinsip Sesuai Guidelines dan Best Practice	**SKKNI**
3.	J. 620100.025.02	Melakukan Debugging	**SKKNI**
4.	J. 620100.023.02	Membuat Dokumen Kode Program	**SKKNI**
5.	J.620100.009.01	Menggunakan Spesifikasi Program	**SKKNI**
6.	J. 620100.018.02	Mengimplementasikan pemrograman berorientasi objek	**SKKNI**
7.	J. 620100.019.02	Menggunakan Library atau Komponen Pre-Existing	**SKKNI**
8.	J. 620100.021.02	Menerapkan Akses Basis Data	**SKKNI**
9.	**J. 620100.033.02**	**Melaksanakan Pengujian Unit Program**	**SKKNI**

Tabel I. 2
Skema Analis Program

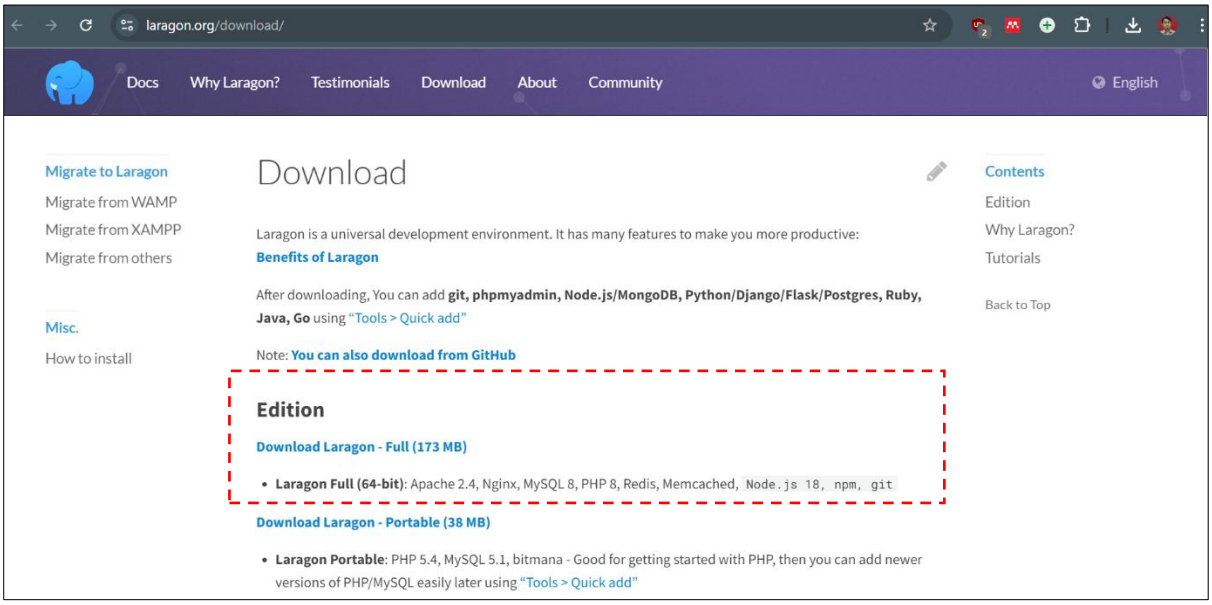
No	Kode Unit	Judul Unit	Jenis Standar (Standar Khusus/Standar Internasional/SKKNI)
1.	J.620100.002.01	Menganalisis skalabilitas perangkat lunak	**SKKNI**
2.	J.620100.020.02	Menggunakan sql	**SKKNI**
3.	J.620100.021.02	Menerapkan akses basis data	**SKKNI**
4.	J.620100.022.02	Mengimplementasikan algoritma pemrograman	**SKKNI**
5.	J.620100.023.02	Membuat dokumen kode program	**SKKNI**
6.	J.620100.025.02	Melakukan debugging	**SKKNI**
7.	J.620100.032.01	Menerapkan code review	**SKKNI**

8.	J.620100.033.02	Melaksanakan pengujian unit program	**SKKNI**
9.	J.620100.034.02	Melaksanakan pengujian integrasi program	**SKKNI**

Keunggunalan lain dengan kita menggunakan *Framework Laravel* pada situs resmi *Laravel* https://laravel.com/docs/10.x kita sudah dapat memenuhi salah satu unit pada **Skema Programmer** (contoh lengkap pengerjaan ***Unit Kerja Programmer*** dapat dilihat di https://bit.ly/LaravelWebPro2 yakni **Menulis Kode Dengan Prinsip Sesuai Guidelines dan Best Practice**, sehingga kita dapat merujuk ke dokuemtasi

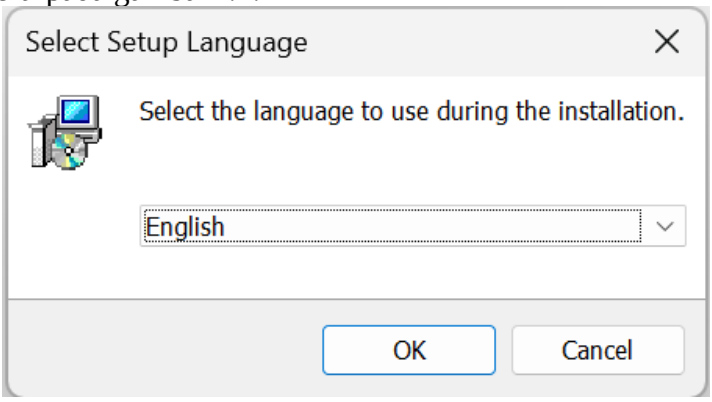
1.1. Instalasi laragon

1. Untuk mendapatkan **laragon-wamp.exe** Download Laragon di website resminya https://laragon.org/download/ atau https://bit.ly/LaravelWebPro2
2. Pilih bagia *Edition*, lalu klik Download Laragon – Full, sperti pada gambar I.1



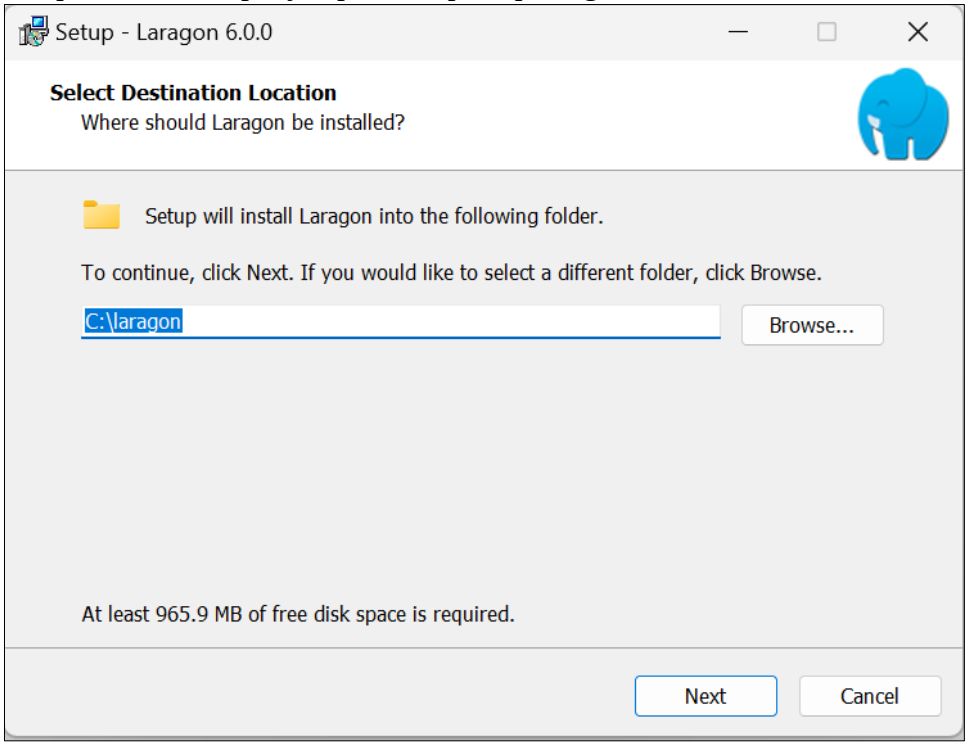
Gambar I. 1
Download Laragon

3. Setelah *download* selesai, buka folder di mana file Laragon disimpan. Double klik file **laragon-wamp.exe** pada installer yang telah anda download, kemudian pilih bahasa jika sudah klik **OK** seperti pada gambar I.2.



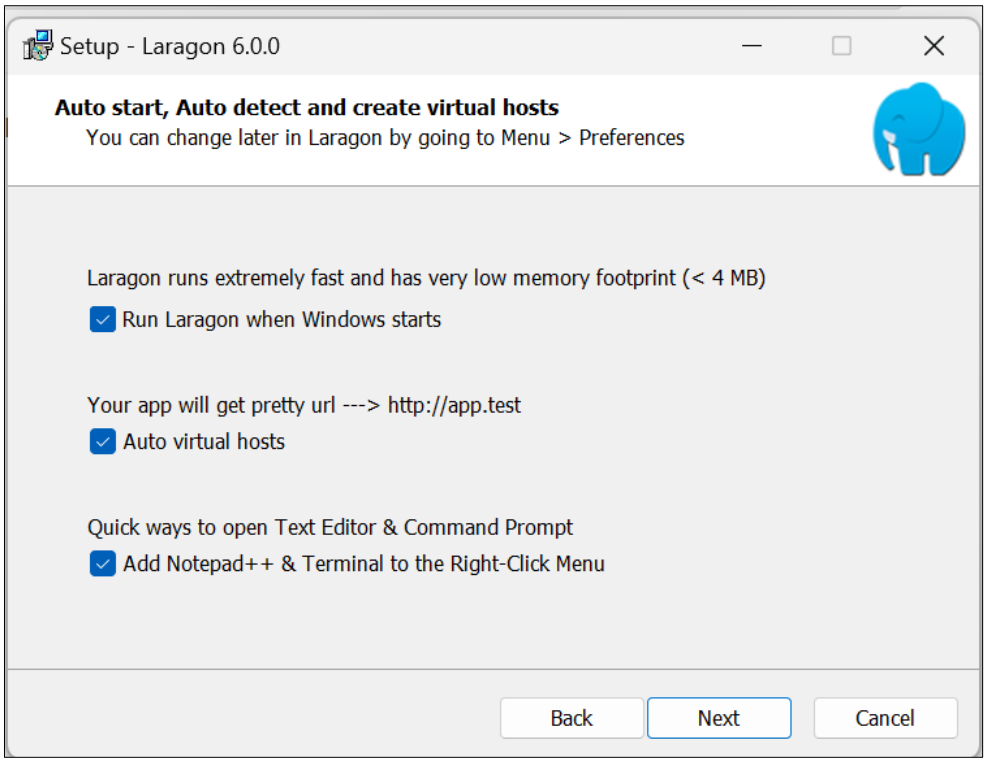
Gambar I. 2
Pilih Bahasa Pada saat *Install*

3. Kemudian pilih direktori penyimpanan seperti pada gambar I.3

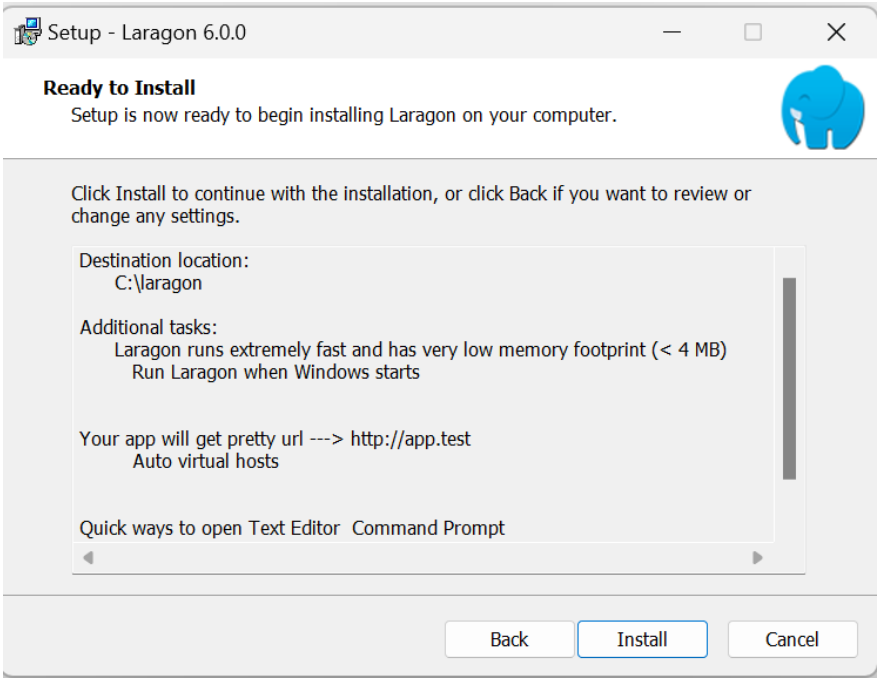


Gambar I. 3
Pilih Lokasi Direktori

4. Centang beberapa "option" sesuai kebutuhan seperti pada gambar I.4, lalu Klik **Next**. Kemudian muncul konfirmasi untuk instal klik **Install** seperti gambar I.5 dan tunggu hingga proses install selesai.

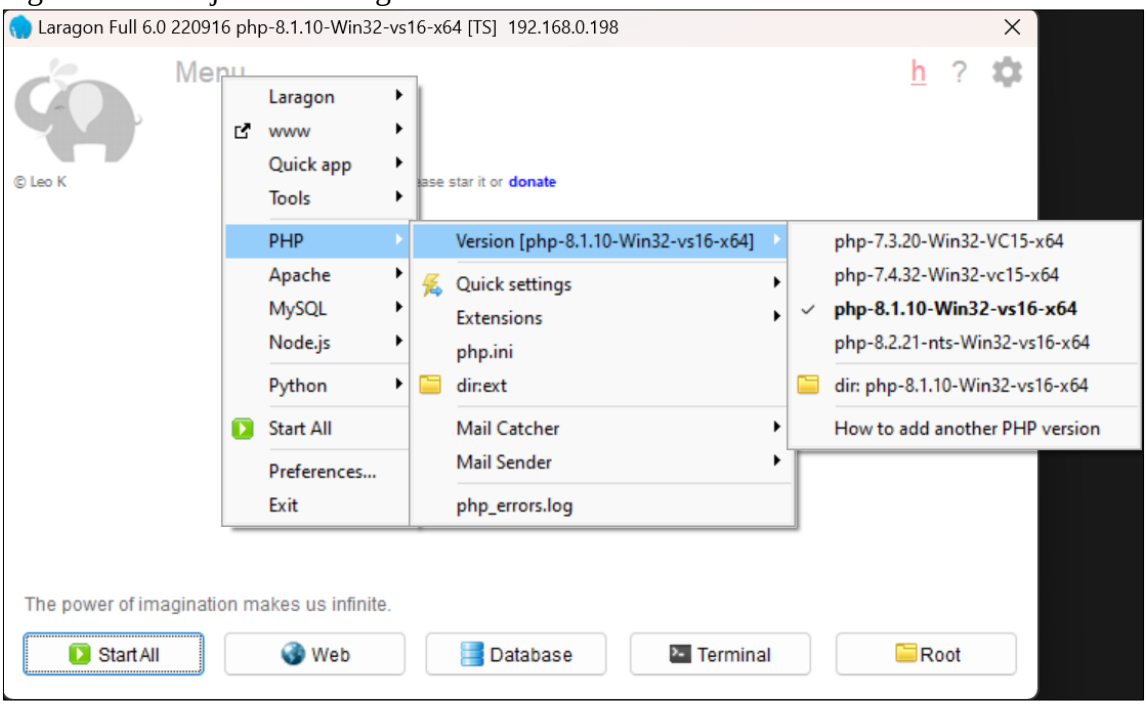


Gambar I. 4
Centang Option Sesuai Kebutuhan

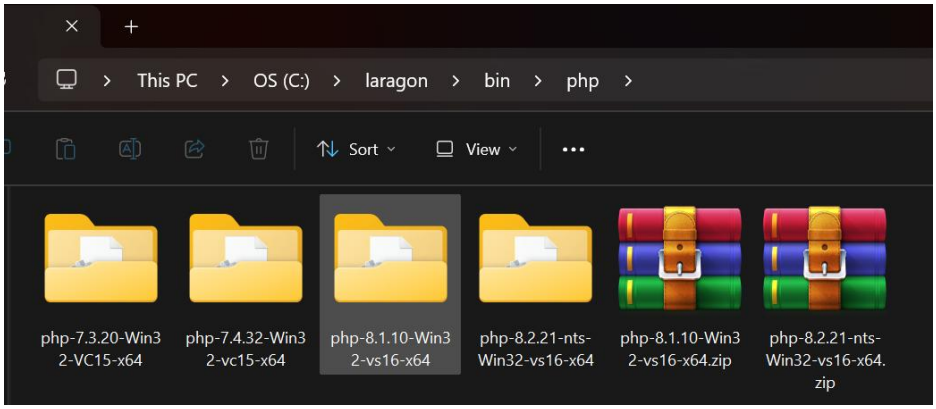


Gambar I. 5
Konfirmasi *Insatall*

5. Jika sudah berhasil berikut maka tampil seperti gambar I.6 pada Menu kita dapat memilih versi laragon sesaui dengan kebutuhan((jika instalasi awal hanya memiliki satu versi PHP bawaan dari Laragon) & untuk menambahkan versi PHP bisa klik ***How to add another PHP*** version atau mengunjungi https://www.php.net/downloads.php atau bisa mengakses versi sebelumnya https://windows.php.net/downloads/releases/archives/. Setelah berhasil download versi PHP yang digunakan simpan pada direktori c:\laragon\bin\php seperti pada gambar I.7 jika hasil download dalam bentuk kompresi, pastikan telah mengekstraknya. Dan untuk mengaktifkan menjalankan laragon klik **Start All**

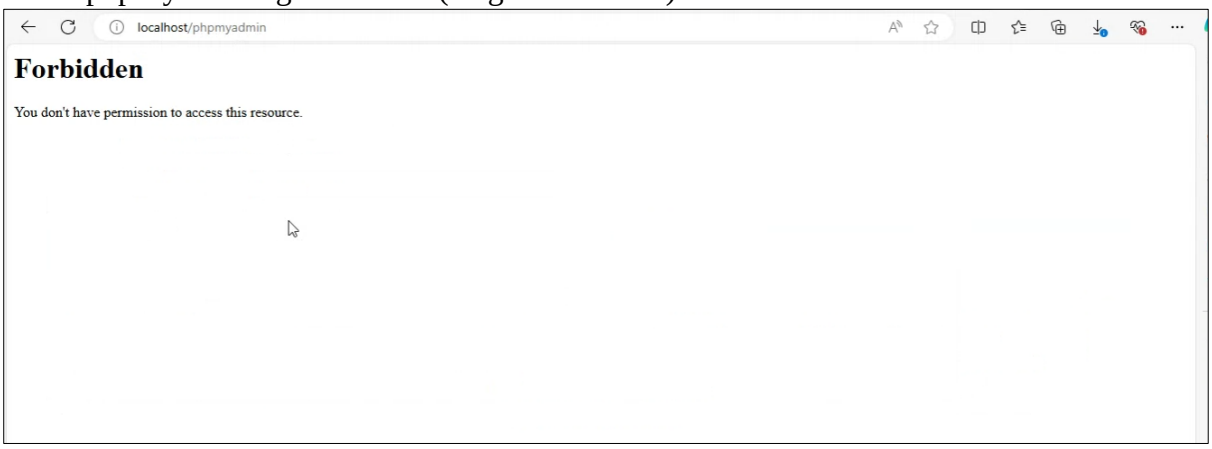


Gambar I. 6
Version Laragon

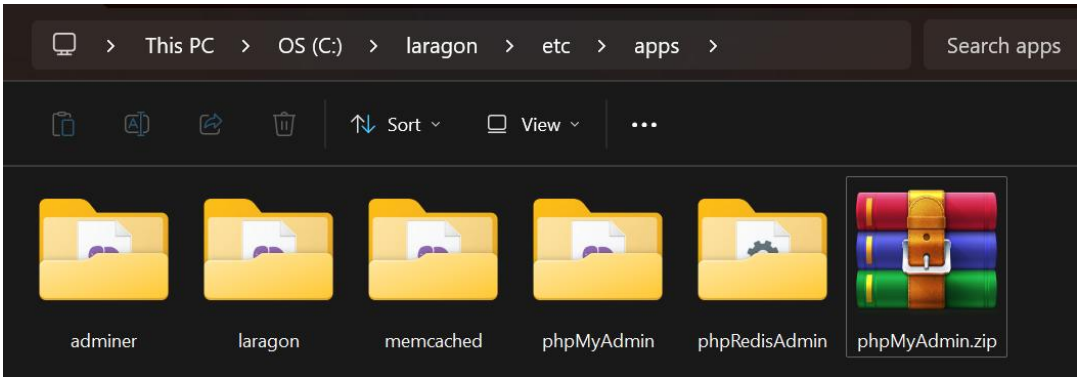


Gambar I. 7
Menambahan versi PHP Pada Laragon

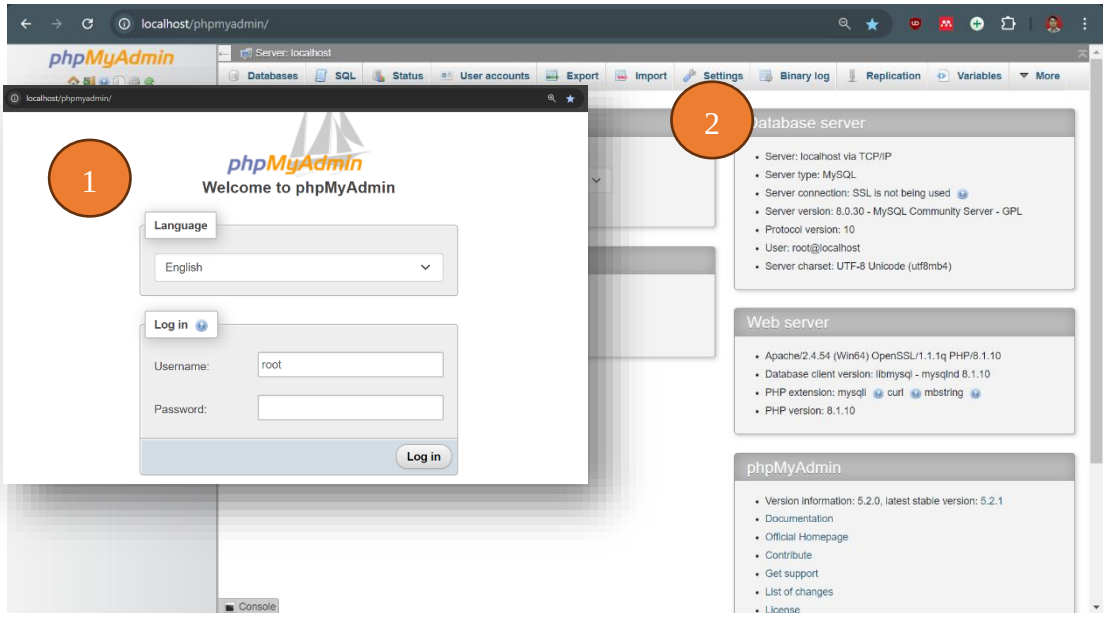
6. Kemudian kita tambahkan phpmyadmin pada https://www.phpmyadmin.net/ atau https://bit.ly/LaravelWebPro2, Jika kita buka *browser* dan akses ke alamat http://localhost/phpmyadmin/ akan tampil seperti pada gambar I.8. Setelah berhasil mendownload **phpmyadmin** biasanya dalam bentuk **kompresi** kemudian **extract** file ke folder c:laragon/etc/apps seperti pada gambar I.9 kemudian hasil ekstrak dan beri nama **phpMyAdmin,** jika berhasil ketika kita akses kembali pada alamat http://localhost/phpmyadmin/ maka akan diminta *username* isikan **root** dan *password* **kosongkan** seperti pada gambar I.10 (langkah nomor 1), kemudian akan tampil kehalaman utama phpMyAdmin gambar I.10 (langkah nomor 2)



Gambar I. 8
phpMyAdmin Belum Tersedia Pada Laragon



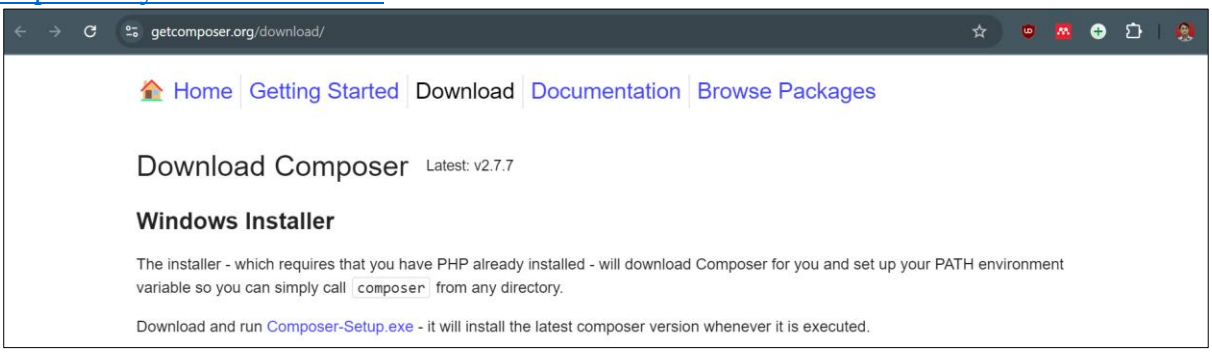
Gambar I. 9
Menambahkan phpMyAdmin Pada Laragon



Gambar I. 10
Halaman Utama phpMyAdmin

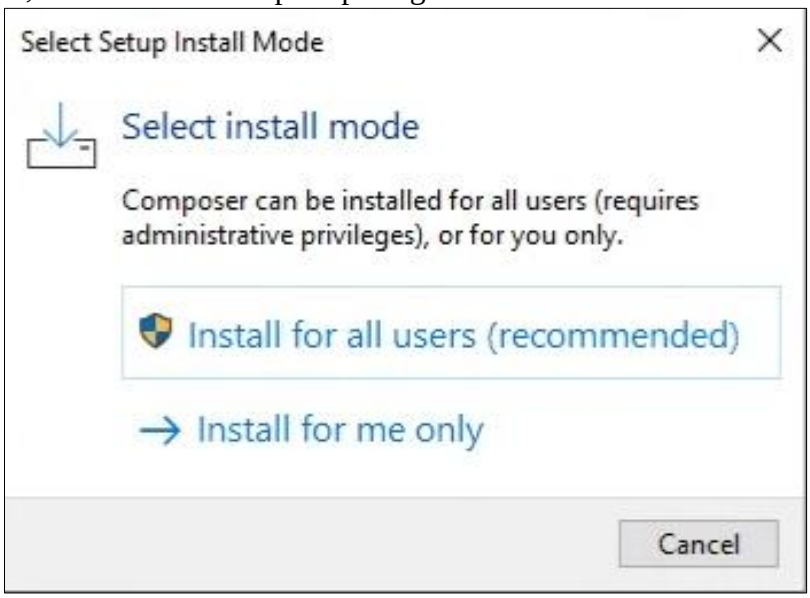
1.2. Instalasi Composer Pada Laragon

1. Untuk mendapatkan **composer-setup.exe** dapat didownload pada situs resmi https://getcomposer.org/download/ seperti pada gambar I.11 atau https://bit.ly/LaravelWebPro2

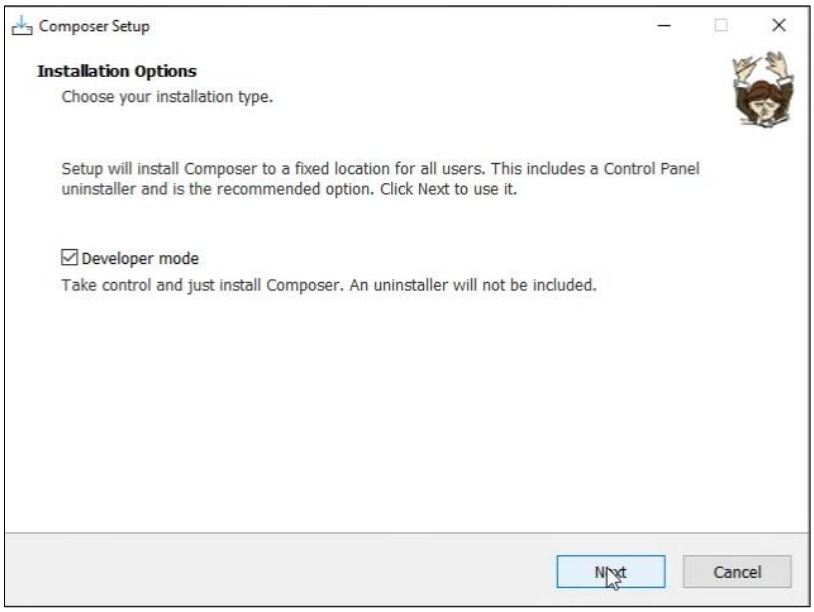


Gambar I. 11
Download Composer

2. Double klik file **.exe** kemudian pilih model instal apakah untuk semua user atau tidak seperti pada gambar I.12, kemudian **Next** seperti pada gambar GI.13

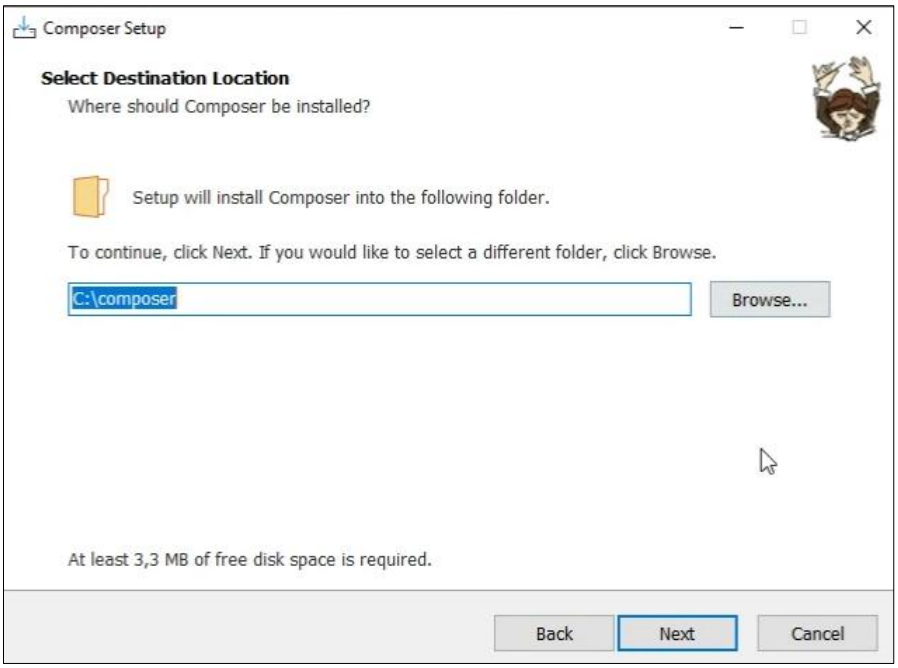


Gambar I. 12
Pilih *Mode Install*

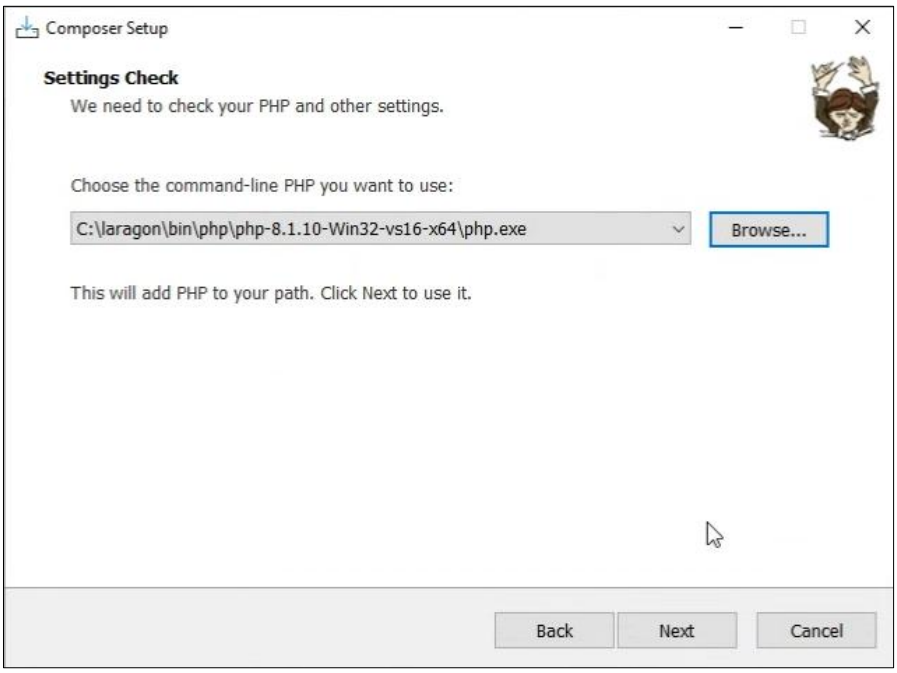


Gambar I. 13
Installation type

3. Pilih direktori penyimpanan laragon seperti pada gambar I.14 kemudian kemudian pilih **Next**, berikutnya tampil pilihan untuk memilih php.exe seperti pada gambar I.15, misalnya pada **php-8.1.10-Win32-vs16-x64** maka direktori pada C:\laragon\bin\php\php-8.1.10-Win32-vs16-x64 jika kita memiliki lebih dari satu PHP pada laragon maka lakukan langkah nomor 1 sampe nomor 3 ini untuk setiap versi PHP dengan php.exe berikutnya hingga instalasi selesai

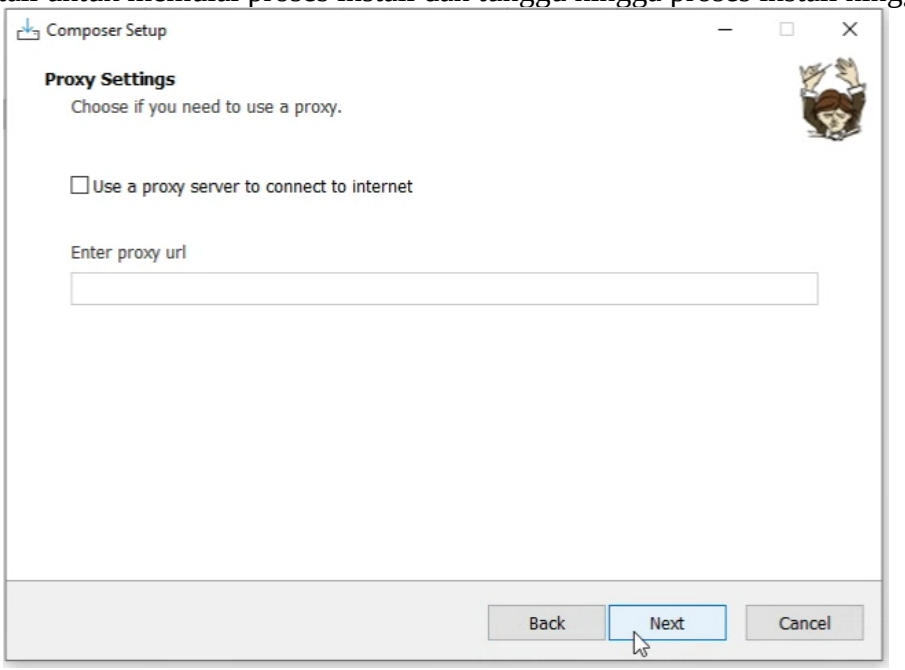


Gambar I. 14
Direktori Laragon

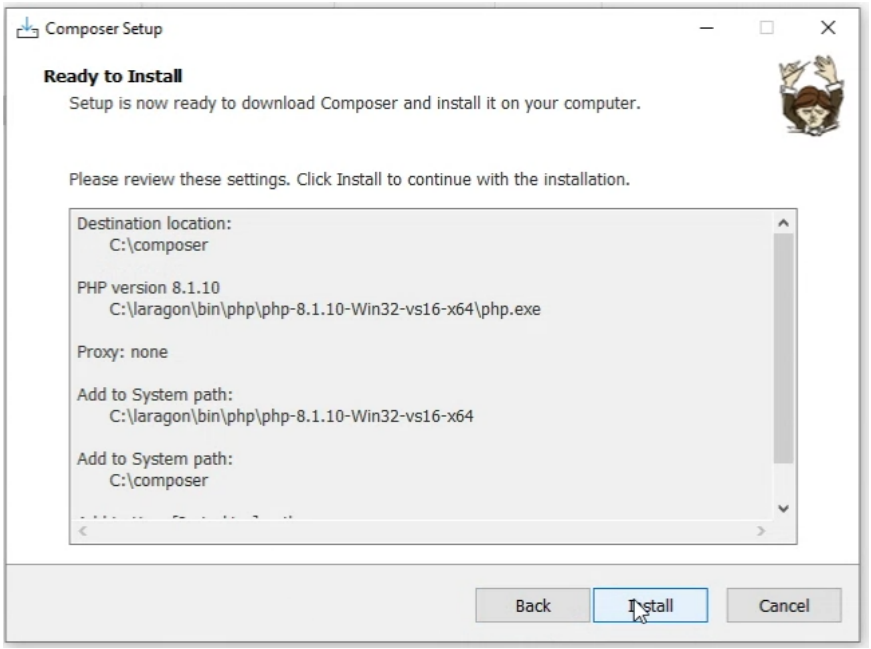


Gambar I. 15
Pilih PHP.exe

4. Pada halaman ini *proxy settings* pilih **Next** seperti pada gambar I.16. Kemudian pada gambar I.17 klik install untuk memulai proses install dan tunggu hingga proses install hingga selesai

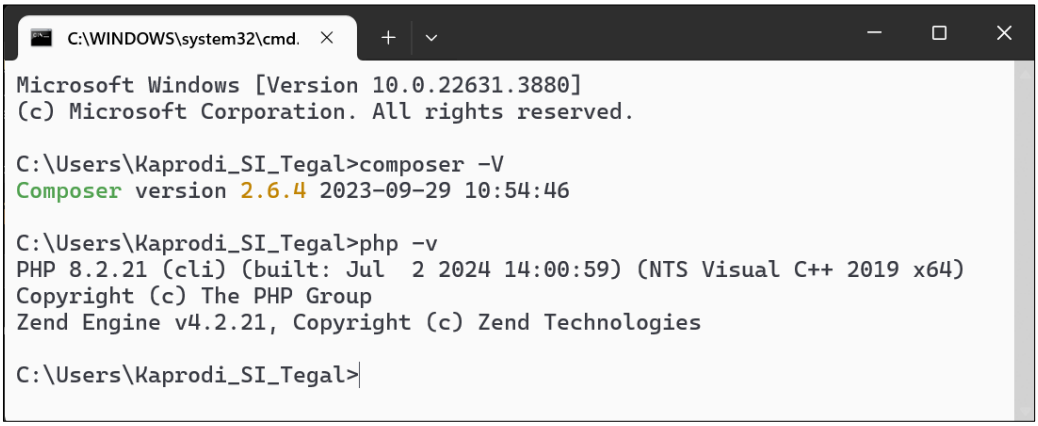


Gambar I. 16
Proxy Settings



Gambar I. 17
Memulai *Istall*

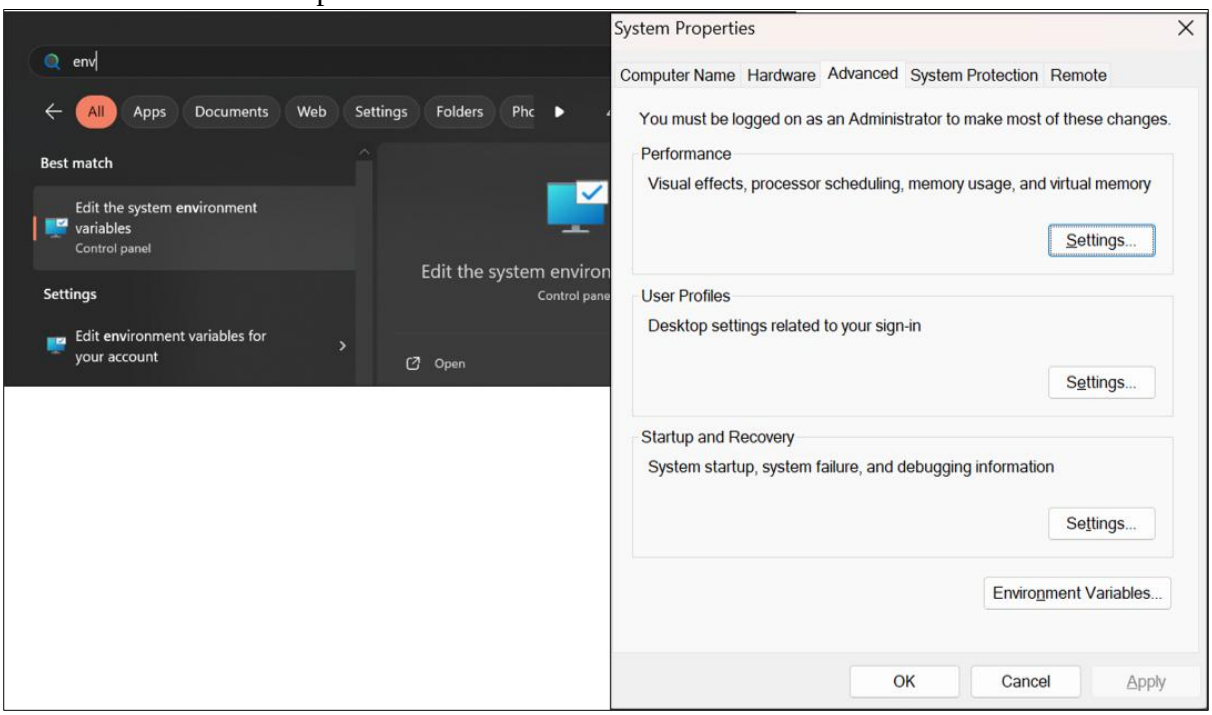
5. Untuk memastikan bahwa insatalasi berhasil dilakukkan maka, buka **Command Prompt** atau tekan tombol **Windows + R** dan ketikkan **cmd**. Kita dapat melihat versi composer yang kita gunakan dengan cara mengetikkan composer -V sedangkan versi php php -v seperti pada gambar I.18



Gambar I. 18
Melihat Versi Composer atau PHP Pada Command Prompt

1.3. Konfigurasi environment

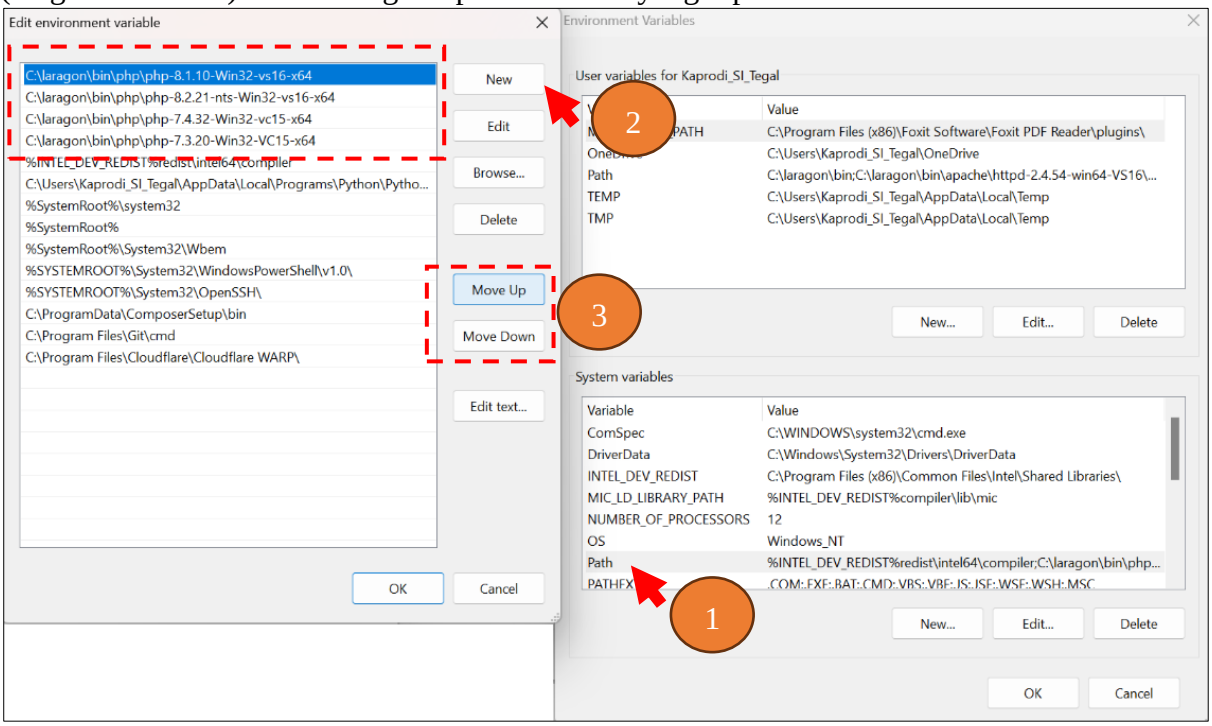
1. Langkah ini hanya dilakukakan jika versi PHP lebih dari satu pada Windows 11 seperti pada Gambar I.6 dimana yang diaktifkan PHP 8.1 tetapi yang terbaca PHP 8.2 dan kita menginginkan versi PHP 8.1. Untuk mengatur ini pada start lakukan pencarian kemudian cari **environment** kemudian pilih **Environment Variables**



Gambar I. 19
Environment Pada Windows 11

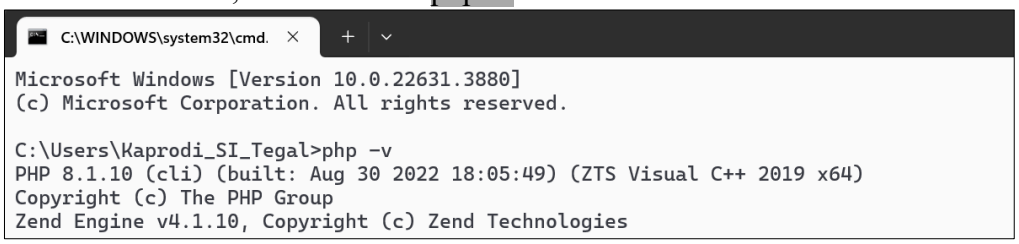
2. Pada Environment Variables, pilih **Path** (langkah nomor 1). Maka akan muncul jendela **Edit Environment Variable**. Klik tombol **New** (langkah nomor 2), kemudian ketikkan direktori PHP yang akan digunakan. Klik tombol **New** lagi jika ingin menambahkan versi PHP lainnya. Agar PHP yang terbaca sesuai dengan keinginan kita, pastikan posisi direktori PHP yang

diinginkan berada di bagian teratas. Kita dapat menggeser posisi direktori ke atas atau ke bawah (langkah nomor 3) untuk mengatur prioritas PHP yang dipilih.



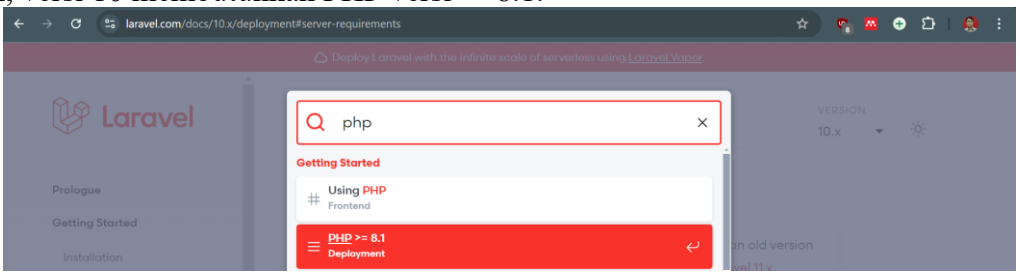
Gambar I. 20
Edit Environment Variable

3. Setelah melakukan **konfigurasi Environment Variable**, pastikan untuk membuka jendela **Command Prompt** yang baru. Jangan mengecek versi PHP menggunakan **Command Prompt** yang sudah dibuka sebelum konfigurasi Environment Variable. **Command Prompt** tidak memiliki sistem **refresh**, jadi pastikan Anda menggunakan jendela **Command Prompt** yang **terbaru**. Kemudian, ketik kembali php -v untuk memeriksa versi PHP.



Gambar I. 21
Command Prompt setalah konfigurasi Environment Variable

4. Kemudian, versi PHP apa yang dibutuhkan oleh Laravel? Jika kita cek di dokumentasi Laravel, versi 10 membutuhkan PHP versi >=8.1.

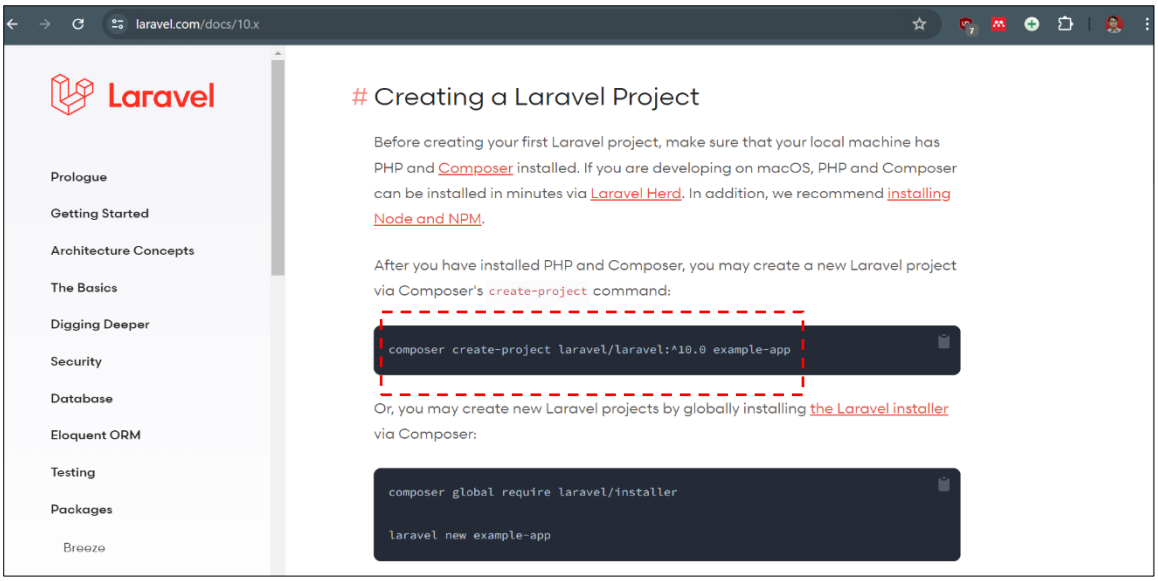


Gambar I. 22
Dokumentasi versi PHP Pada Laravel 10

1.4. Konfigurasi Laravel

1. Pada gambar I.23 bagamana cara mendownload *Project* baru pada *Laravel* 10 dengan *composer* dengan tautan https://laravel.com/docs/10.x

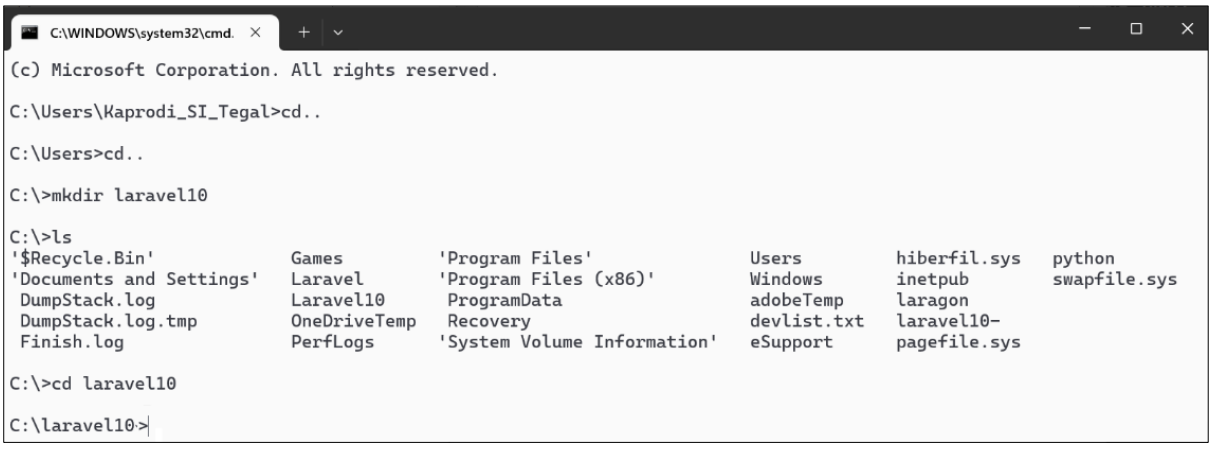
```
composer create-project laravel/laravel:^10.0 example-app
```



Gambar I. 23
Membuat *Project* Baru Pada Laravel 10

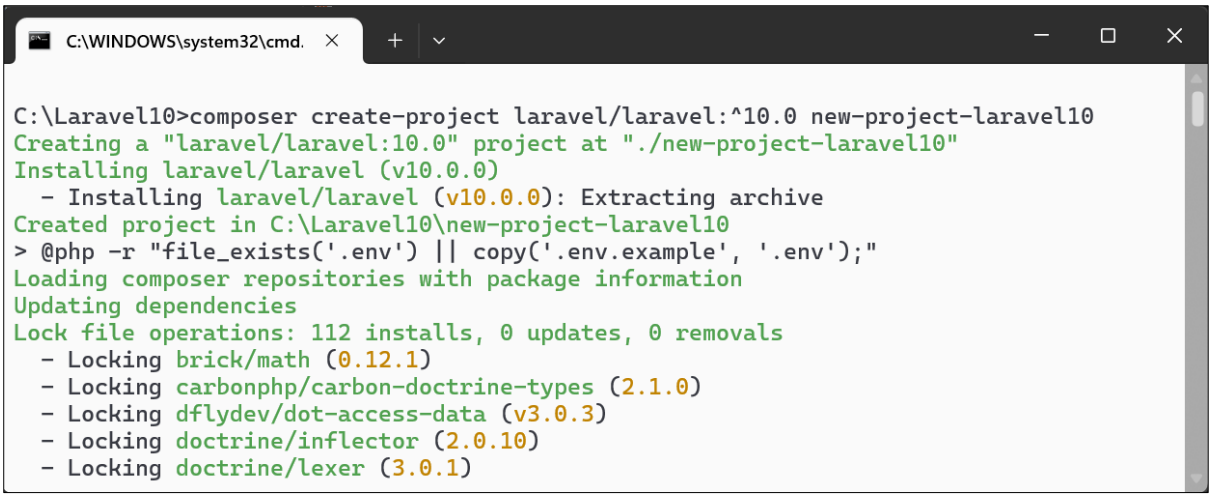
2. Untuk memulai download *Project* baru pada *Laravel* 10 menggunakan *Composer* kita bisa membukan ***Command Prompt*** dengan kombinasi keyboard **Windows+R** kemudian ketikan **cmd** atau bisa menggunakan ***Git Bash*** atau ***Terminal*** (Linux/Mac) atau Terminal yang disedian oleh **Laragon**, dimana *Project* akan kita simpan pada direktori **C:\Laravel10** sehingga pada *Terminal* akan terlihat sebagai berikut:

a. Membuat folder Laravel10 pada localdisk C dan buka direktori **C:\Laravel10** ikuti langkah dibawah ini



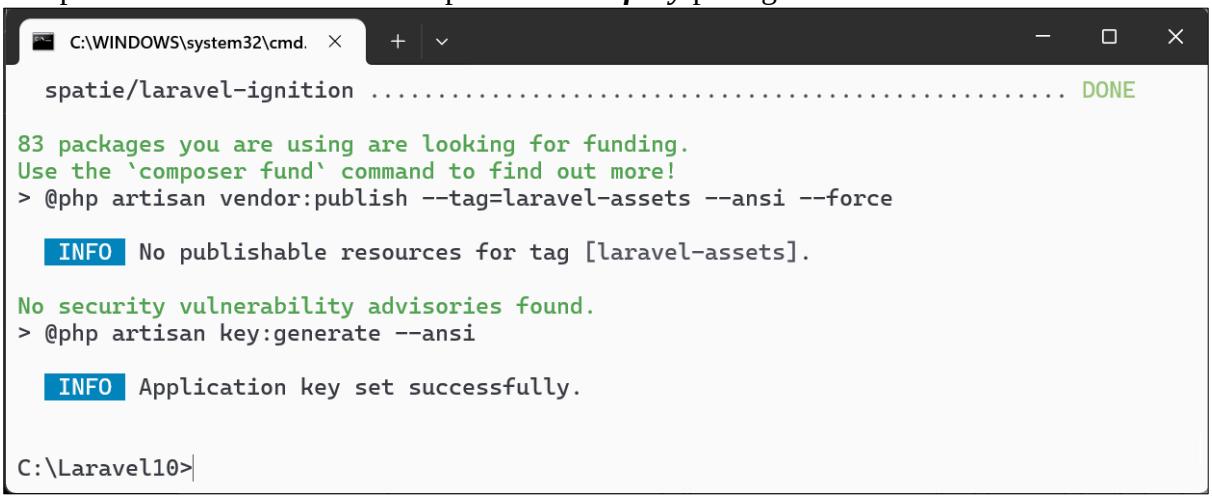
Gambar I. 24
Membuat Folder Dan Masuk Folder Pada Command Prompt

2. Copy composer create-project laravel/laravel:^10.0 **example-app** kemudian paste pada **Command Prompt** dan ganti **example-app** misalnya menjadi **new-project-laravel10**, kemudian tekan enter pada keyboard untuk proses *download Project* Laravel dan pastikan PC terkoneksi dengan internet seperti pada gambar I.25



Gambar I. 25
Proses *Download Project* Baru

Jika proses *download* berhasil ada pesan ***successfully*** pada gambar I.26

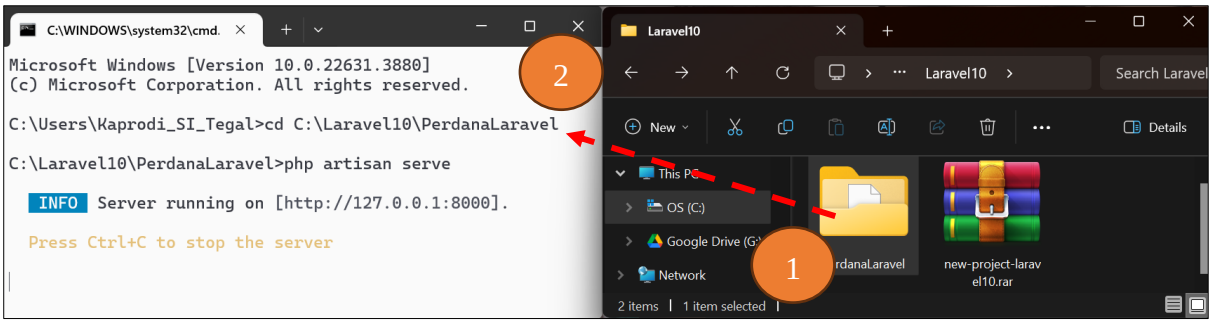


Gambar I. 26
Project Baru Berhasil di *Download*

Dan jangan lupa untuk membackup **new-project-laravel10** dengan mengompresnya, misalnya menggunakan format **.zip** atau **.rar.** Dengan demikian, kita memiliki salinan master Project baru. Jika kita membutuhkan Project tersebut di masa mendatang, kita tidak perlu mendwonload kembali; cukup ekstrak file kompresi dari **new-project-laravel10**.

3. Ganti *Project* **new-project-laravel10** menjadi **PerdanaLaravel**. Kali ini, kita akan buka direktori **C:\Laravel10\PerdanaLaravel**, dengan cara di drag ke **Command Prompt.** Pada **Command Prompt** atau Terminal ketikan **cd** kemudian drag folder **PerdanaLaravel** (langkah nomor 1) ke terminal (langkah nomor 2), jika direktori sesuai akan tampil pada Terminal **cd /c/Laravel10/PerdanaLaravel**, untuk melanjukan tekan **Enter** seperti pada gambar I.27 & berikutnya ketikkan perintah untuk menjalankan *Project* pada terminal

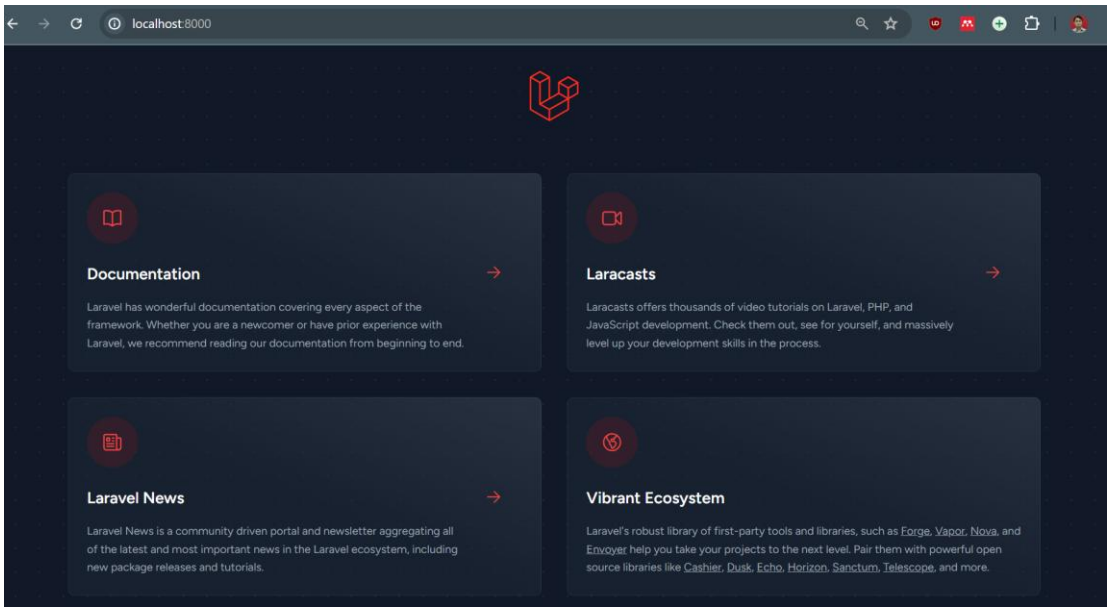
```
php artisan serve
```



Gambar I. 27
Menjalankan Project Laravel

4. Pada *browser* dengan mengetikkan alamat http://localhost:8000 seperti pada gambar I.28

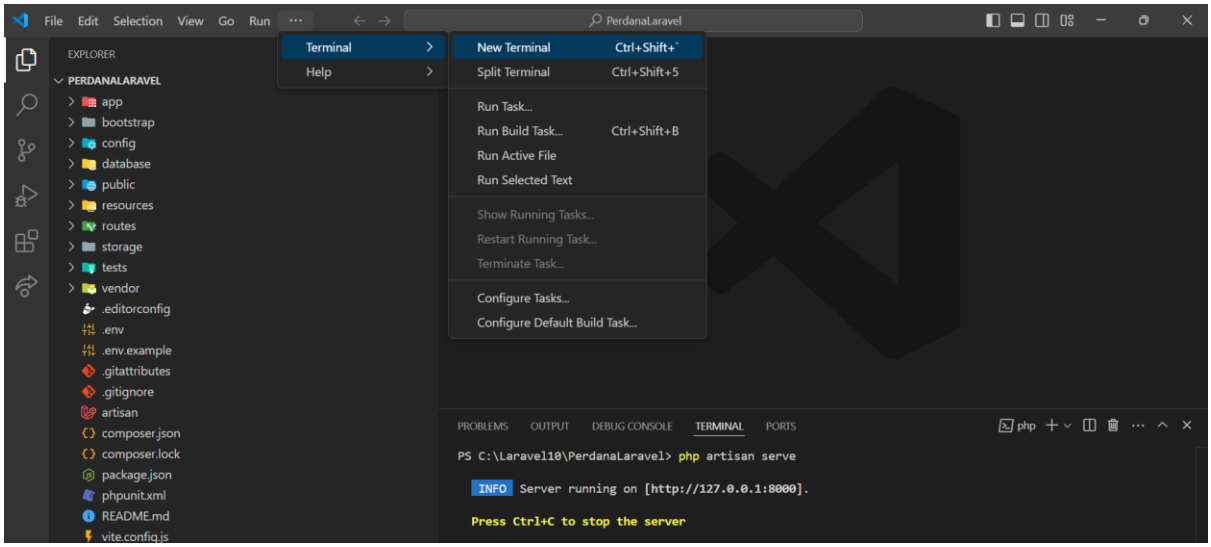
```
http://localhost:8000
```



Gambar I. 28
Melihat Hasil *Project* Pada *browser*

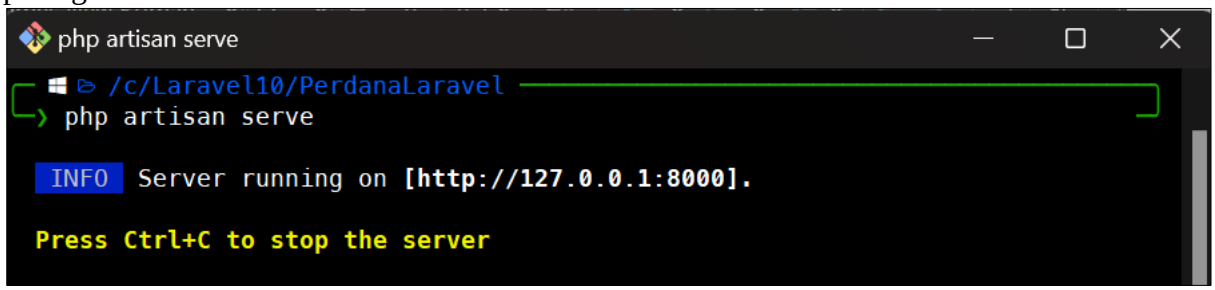
5. Sedangkan untuk menghentikan *Project* yang sedang berjalan dengan cara pada keyboard kombinasi **Ctrl+C**

Gambar I. 29
Service Dalam Keadaan Berhenti



Gambar I. 30
Menjalankan program melalui Terminal VSC

Untuk menjalan perintah **artisan** kita bisa menggunakan terminal Visual Studio Code (VSC) pertama buka *project* dengan cara **File -> Open Folder** yakni pada direktori **C:/Laravel10/ PerdanaLaravel**, kemudian buka Terminal dengan cara **Terminal->New Terminal** dan pada terminal ketikan perinah Artisan seperti pada gambar I.30, atau menggunakan **git best** seperti pada gambar I.31



Gambar I. 31
Menjalankan program melalui Terminal Git Bash

6. untuk mengetahui versi laravel yang kita gunakan dengan mengetikan perintah pada terminal **php artisan --version** seperti pada gambar 1.6 dan pastikan pengecekan sudah sesuai dengan direktori *Project* laravel yang akan kita lihat. Saat pembuatan modul ini versi yang digunakan *Laravel Framework 10.48.16*

Gambar I. 32
Melihat Versi Project Laravel Yang Digunakan

Latihan Mandiri 1:
Portofolio sertifikasi kompetensi**,** Impentasikan Unit Kompetensi Software Development pada **Mengimplementasikan Pemrograman Terstruktur**. Studi kasus dapat diambil dari *Project* kelompok semester sebelumnya dimana tugas yang memiliki laporan lengkap. Misalnya *Project* kelompok, pada Web Programming I (WP1) Semester 2 atau Dasar Pemrograman (Python) Semester 1. Dengan memenuhi kriteria sebagai berikut:

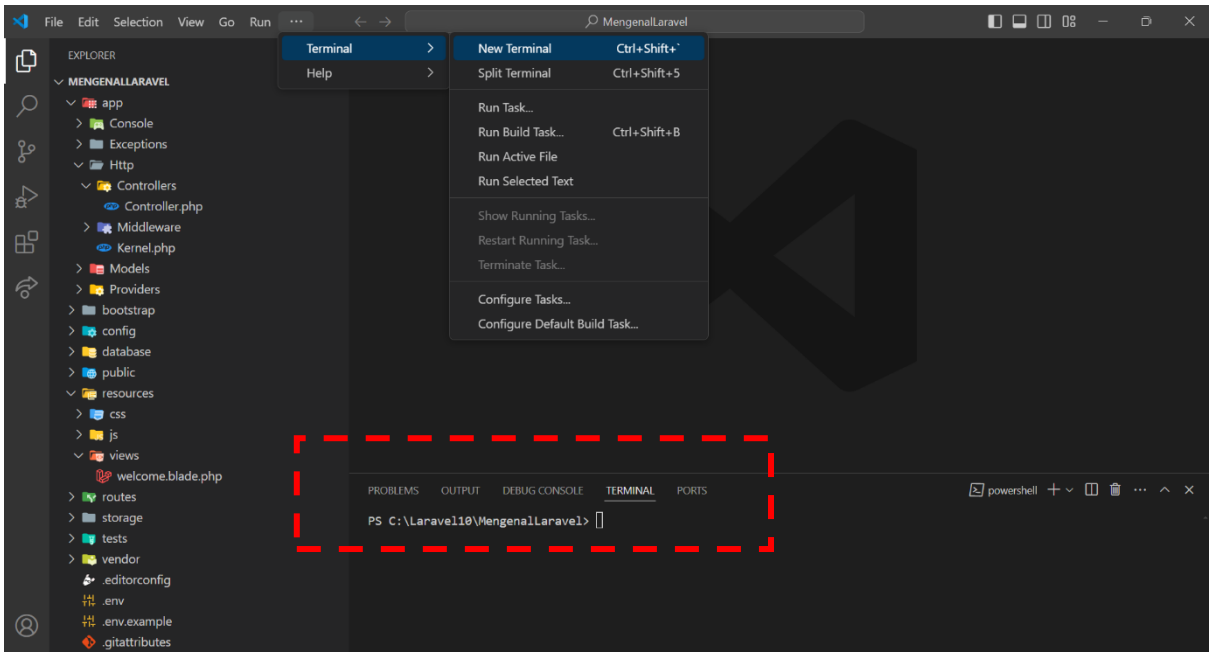
- Tampilkan input dan output pada Project.
- Ketentuan nama variabel, seperti pembuatan nama variabel dan ketentuan saat pembuatan variabel.
- Operator yang digunakan.
- Percabangan.
- Perulangan.
- Penggunaan *function*.
- Array.
- Library.
- Membuat dokumentasi *script*.

Minggu Ke-2
Mengenal Controller, *Function*, Route & View

Controller adalah kelas yang mengatur logika aplikasi yang bertanggung jawab untuk menerima input dari pengguna, memprosesnya, dan mengembalikan respons yang sesuai. Dengan fungsi utama dari pada controller untuk menghubungkan model (yang mewakili data) dengan tampilan (*views*) yang menampilkan informasi kepada pengguna dan mengelola logika atau untuk dikirimkan kembali ke pengguna atau tampilan yang sesuai. Sedangkan Route adalah mekanisme yang menentukan bagaimana URL atau rute tertentu dalam aplikasi web akan diproses. Route menghubungkan URL yang diminta oleh pengguna dengan *function* atau *Controller* yang akan menanganinya.

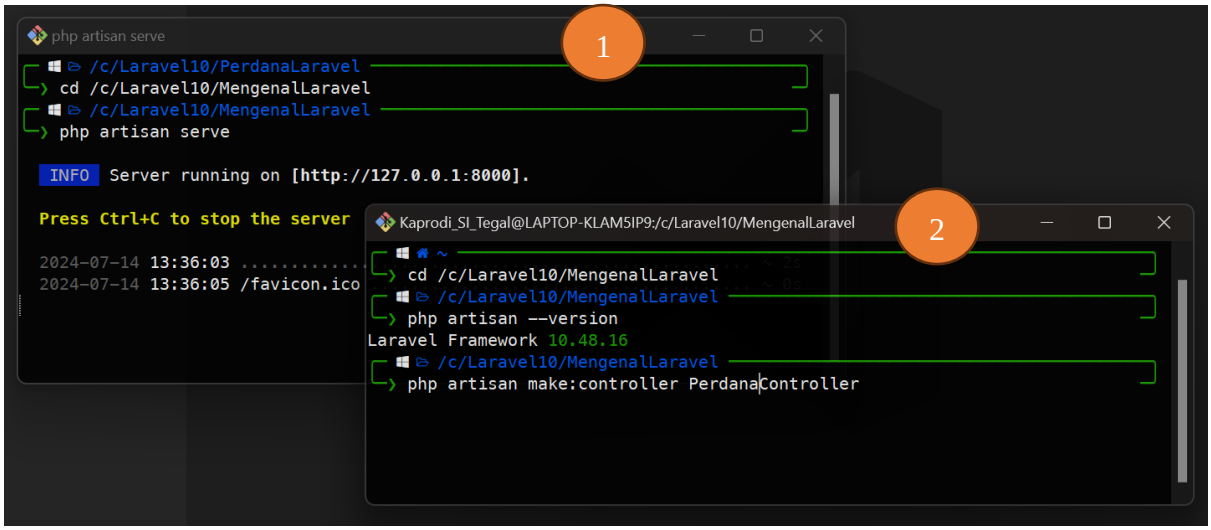
2.1. Persiapan Project Baru

1. Extrak **new-project-laravel10** & ganti nama *Project* menjadi **MengenalLaravel** Kemudian buka *Project*, pada *Visual Studio Code* pada menu bar **File -> Open Folder** yakni pada direktori **c:/Laravel10/MengenalLaravel**
2. Untuk terminal kita juga bisa menggunakan terminal yang disediakan oleh *Visual Studio Code* pada **Terminal->New Terminal** disinilah kita mengetikan perintah-perintak seperti membuat *Controller*, *Model* dan lainnya.



Gambar II. 1
Membuka Project Pada Visual Studio Code

3. Namun pada modul ini penulis untuk menjalan perinta menggunakan terminal pada ***Git Bash*** misal saya gunakan 2 terminal**,** dengan demikan tidak memberatkan *Visual Studio Code* serta lembar kerja lebih luas. [1] Terminal untuk menjalankan program *php artisan serve* [2] menjalankan perintah-perintah *artisan* terlihat pada gambar II.2

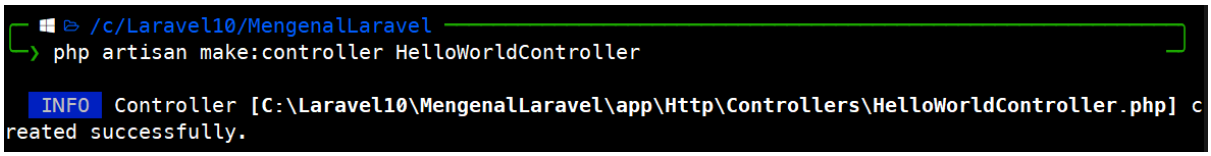


Gambar II. 2
Membukan 2 Terminal

2.2. Membuat Controller

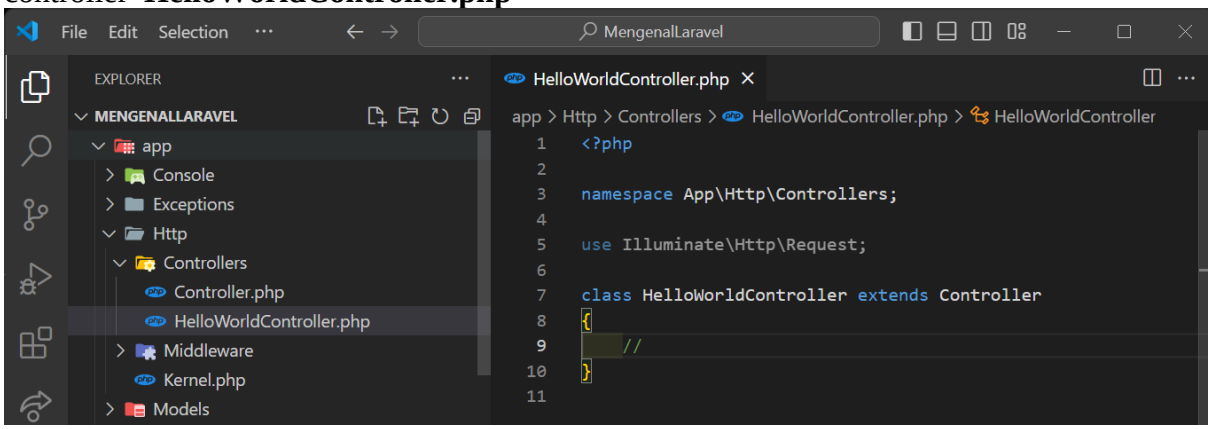
1. Mari kita mulai pertama kali membuat Controller pada terminal, dengan nama controller **HelloWorldController** dengan perintah sebagai berikut, jika berhasil seperti gambar II.3

```
php artisan make:controller HelloWorldController
```



Gambar II. 3
Membuat Controller

2. Sehingga pada direktori **App\Http\Controllers** bertambah controller dengan nama controller **HelloWorldController.php**



Gambar II. 4
Direktori Controller

2.3. Membuat Function

Pada **HelloWorldController.php** kita akan tambahkan *function index()* sehingga perubahan *Controller* sebagai berikut:

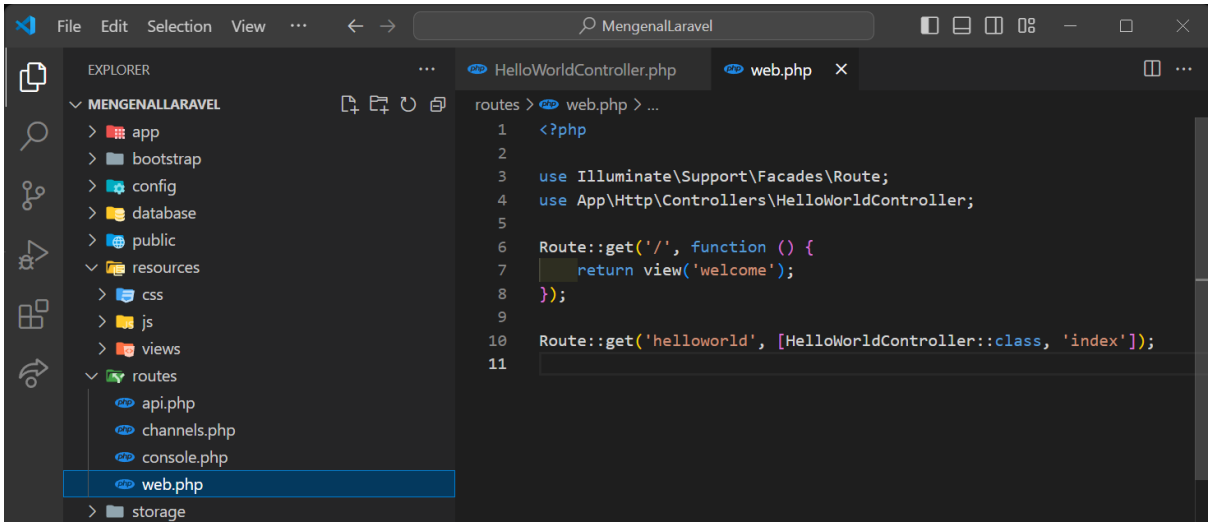
```
<?php
namespace App\Http\Controllers;
```

```php
use Illuminate\Http\Request;

class HelloWorldController extends Controller
{
    public function index()
    {
        return "Selamat Belajar Framework Laravel 10";
    }
}
```

2.4. Konfigurasi Route

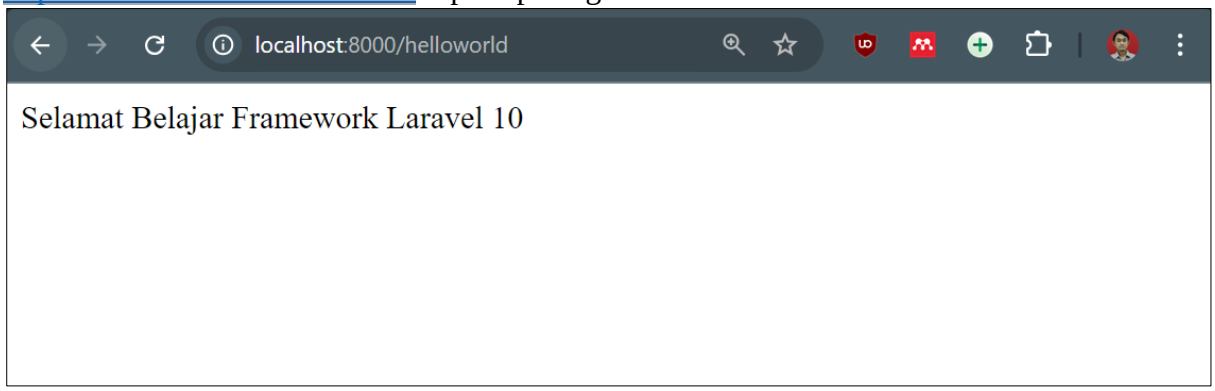
1. Kemudian pada routes**web.php** kita tambahkan *script* sebagai berikut sehingga terlihat seperti pada gambar II.5

```php
<?php
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\HelloWorldController;

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

Route::get('helloworld', [HelloWorldController::class, 'index']);
```



Gambar II. 5
Mengenal Route

2. Dengan demikan kita dapat melihat hasilnya pada *browser* dengan mengetikkan alamat http://localhost:8000/helloworld seperti pada gambar II.6



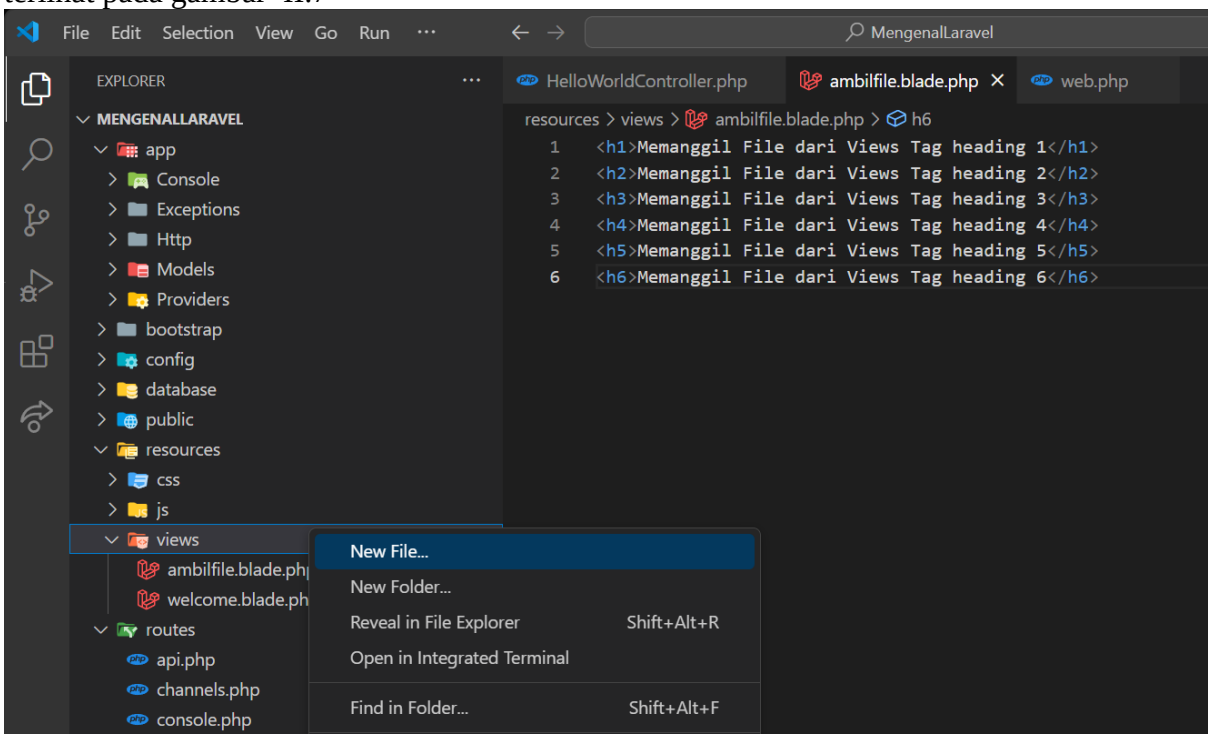
Gambar II. 6
Melihat Hasil Helloword

2.5. Membuat Views

1. Kali ini kita akan memanggil file pada direktori *views*, Pertama kita tambahkan *function ambilFile()*

```php
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;

class HelloWorldController extends Controller
{
    public function index()
    {
        return "Selamat Belajar Framework Laravel 10";
    }
    public function ambilFile()
    {
        return view('ambilfile');
    }
}
```

2. Maka pada direktori *views* kita tambahkan file dengan cara klik kanan pada *resources/views* pilih *New File* kemudian beri nama file **ambilfile.blade.php.** Tambahkan *script html* yang terlihat pada gambar II.7



Gambar II. 7
Menambahkan File pada Views

3. sehingga pada routes**web.php** kita tambahankan *script* sebagai berikut:

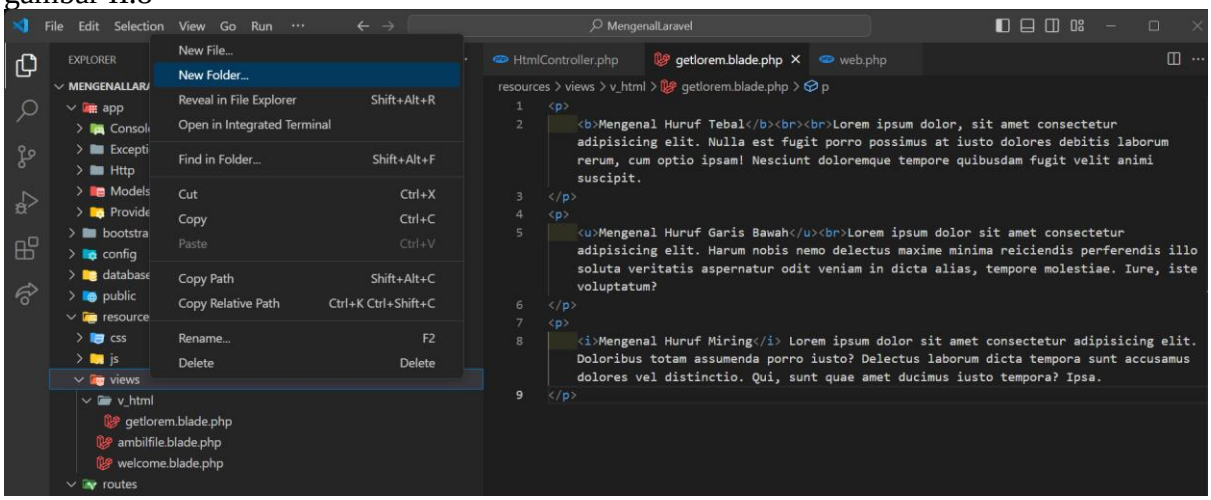
```php
<?php
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\HelloWorldController;

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

```
Route::get('helloworld', [HelloWorldController::class, 'index']);
Route::get('ambilfile', [HelloWorldController::class, 'ambilFile']);
```

4. Kali ini, kita akan membuat *controller* baru dengan nama **HtmlController**. Kita juga akan membuat struktur folder yang lebih rapi, misalnya dengan menempatkan view dalam folder **v_html**. Dengan demikian, semua file view yang berhubungan dengan **HelloWorldController** akan berada dalam folder **v_html**. *Script* HelloWorldController sebagai berikut:

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;

class HtmlController extends Controller
{
    public function getLorem()
    {
        return view('v_html.getlorem');
    }
}
```

Dengan baris *script* seperti ini return view('v_html.getlorem'); artinya kita diminta untuk membuat folder **v_html** dengan file **getlorem.blade.php** pada *resources/views* seperti pada gambar II.8

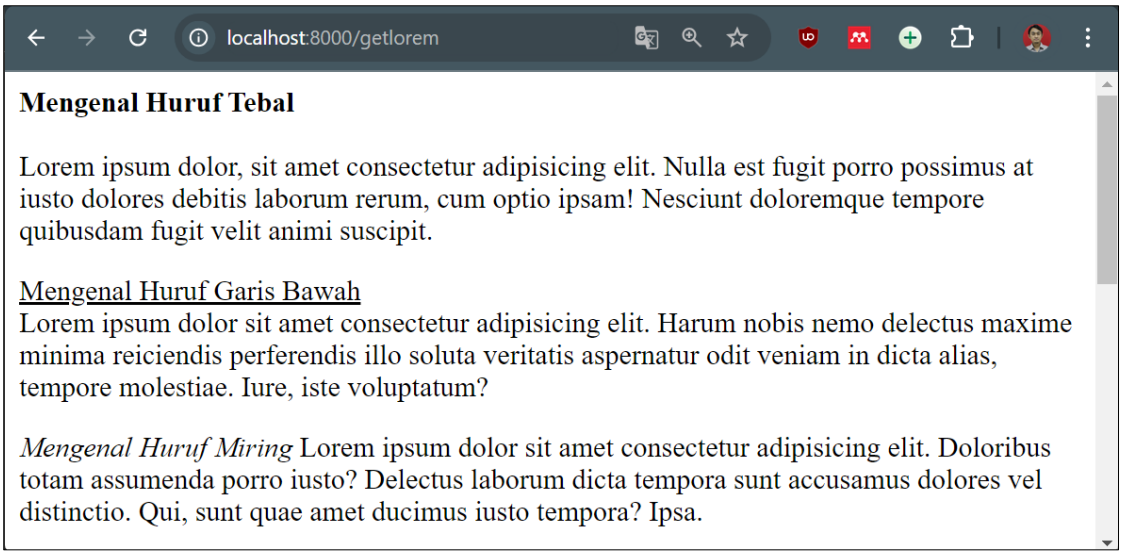


Gambar II. 8
Menambahkan Folder pada Views

5. Pada routes**web.php** kita tambahkan script sebagai berikut:

```
<?php
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\HelloWorldController;
use App\Http\Controllers\HtmlController;

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

Route::get('helloworld', [HelloWorldController::class, 'index']);
Route::get('ambilfile', [HelloWorldController::class, 'ambilFile']);
Route::get('getlorem', [HtmlController::class, 'getLorem']);
```

6. Dengan demikan kita pada melihat hasilnya pada *browser* dengan mengetikkan alamat http://localhost:8000/getlorem seperti pada gambar II.9



Gambar II. 9
Mengenal View getlorem Pada browser

Studi Kasus:
Buat *Contoller* baru dengan nama **LatihanController.** Pada **LatihanController terdapat 2** *function* **:**

1. *Function* **getTabel,** menampilkan disain tabel dengan tampilan sebagai berikut:

Data Mahasiswa

No	NIM	Nama	Kelas
1	NIM 1	Nama Lengkap 1	Kelas 1
2	NIM 2	Nama Lengkap 2	Kelas 2

2. *Function* **getForm,** menampilkan disain form dengan tampilan sebagai berikut:

NIM

Nama Lengkap

Kelas

Simpan

Latihan Mandiri 2:
Portofolio sertifikasi kompetensi, Impentasikan Unit Kompetensi Software Development pada **Menulis Kode Dengan Prinsip Sesuai Guidelines dan Best Practice**. Dengan minimal memenuhi kriteria berikut:

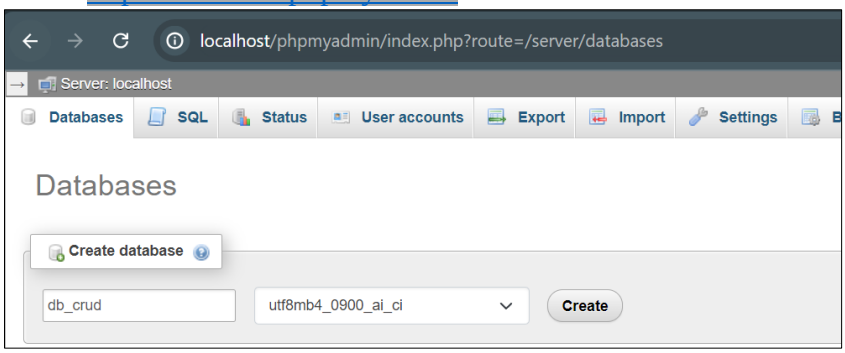
- Instalasi *Project*
- Menjalankan Program.
- *Route*.
- *Contoller*.

Minggu Ke-3
Manajemen Basis Data: Konfigurasi Databases, Migration, dan Seeders

Salah satu Fitur Utama Laravel yakni **Migration dan Seeding** Laravel menyediakan sistem migrasi basis data yang memungkinkan kita dapat melacak dan mengelola perubahan skema basis data dengan mudah. Seeding dapat mengisi basis data dengan data dummy untuk pengujian.

3.1. Membuat Database

1. Buat *Project* Baru dengan nama **BelajarCRUD**
2. Buat *database* baru di *phpmyadmin* dengan nama **db_crud** pada *browser* dengan mengetikkan alamat http://localhost/phpmyadmin

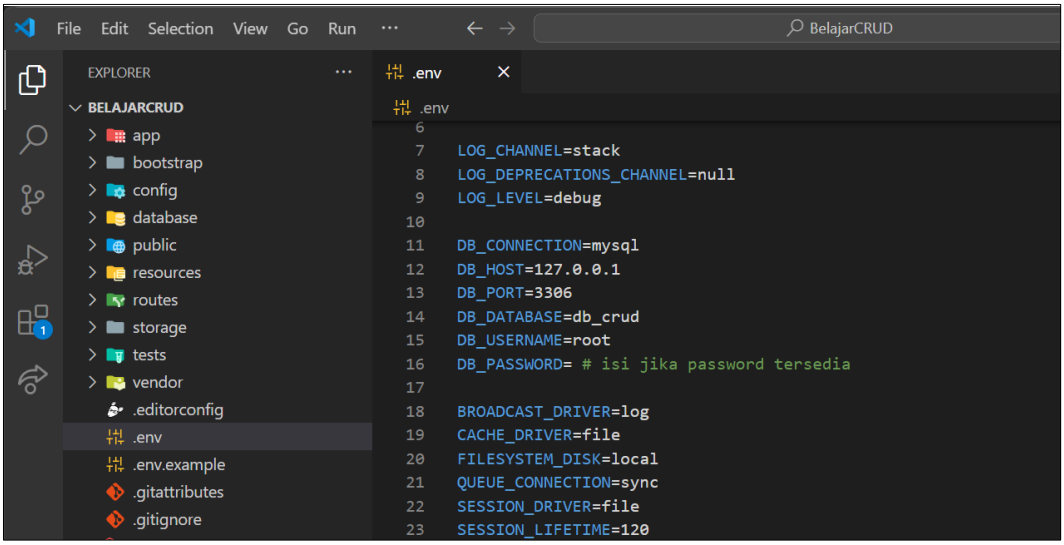


Gambar III. 1
Membuat Database

3.2. Konfigurasi Database

Selanjutnya, buka *Project* **BelajarCRUD** dan sesuaikan konfigurasi database pada file **.env** dengan server yang kita gunakan seperti pada gambar III.2, Jika menggunakan *Laragon*, konfigurasi sebagai berikut:

```
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=db_crud
DB_USERNAME=root
DB_PASSWORD= # isi jika password tersedia
```



Gambar III. 2
Konfigurasi Database

3.3. Migration

1. Membuat migration, pada terminal ketikan perintah berikut, seperti pada gambar III.3

```
php artisan make:migration create_anggota_table
```

```
/c/Laravel10/BelajarCRUD
php artisan make:migration create_anggota_table

  INFO  Migration [C:\Laravel10\BelajarCRUD\database\migrations/2024_07_14_133429_create_anggota_table.php] created successfully.
```

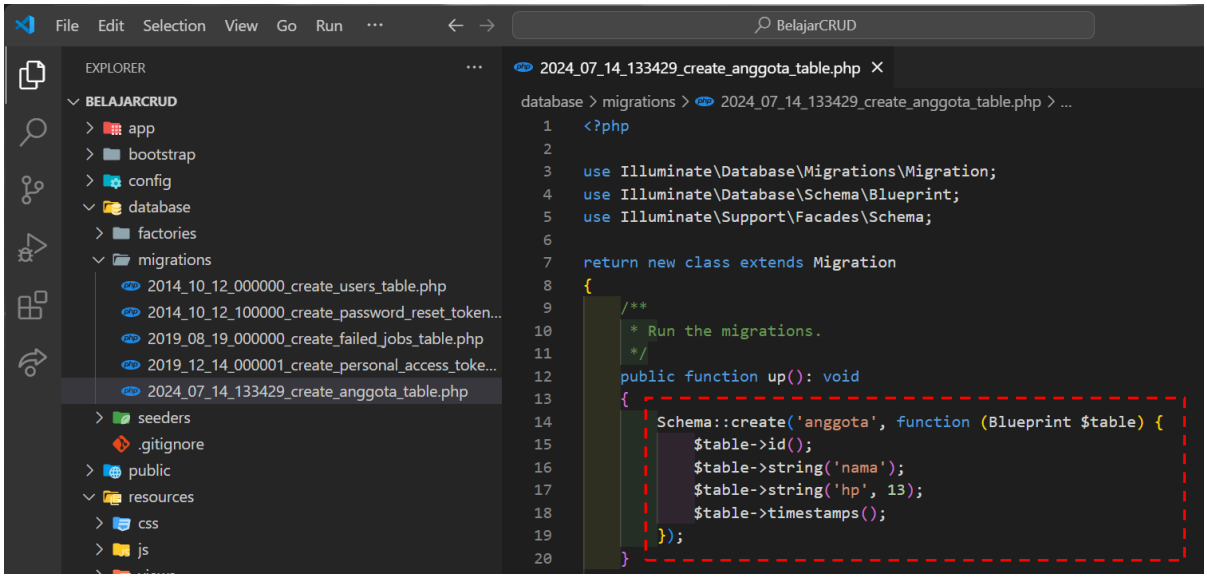
Gambar III. 3
Membuat Migration

2. Rancangan tabel yang digunakan dapat dilihat pada Tabel III.1. Selanjutnya, buka *blueprint* tabel anggota seperti yang ditunjukkan pada Gambar III.4, yang terdapat pada direktori database/migrations. Tambahkan *script* berikut:

```php
<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    public function up(): void
    {
        Schema::create('anggota', function (Blueprint $table) {
            $table->id();
            $table->string('nama');
            $table->string('hp', 13);
            $table->timestamps();
        });
    }

    public function down(): void
    {
        //
    }
};
```

Tabel III. 1
Tabel Anggota

Fild	Tipe Data	Keterangan
id	Bigint	Primary Key
nama	Varchar(255)	
hp	Varchar(13)	
created_at	Timestamp	
updated_at	Timestamp	



Gambar III. 4
Blueprint Table

3. Untuk menjalankan *migrations* pada terminal sebagai berikut, seperti pada gambar III.5

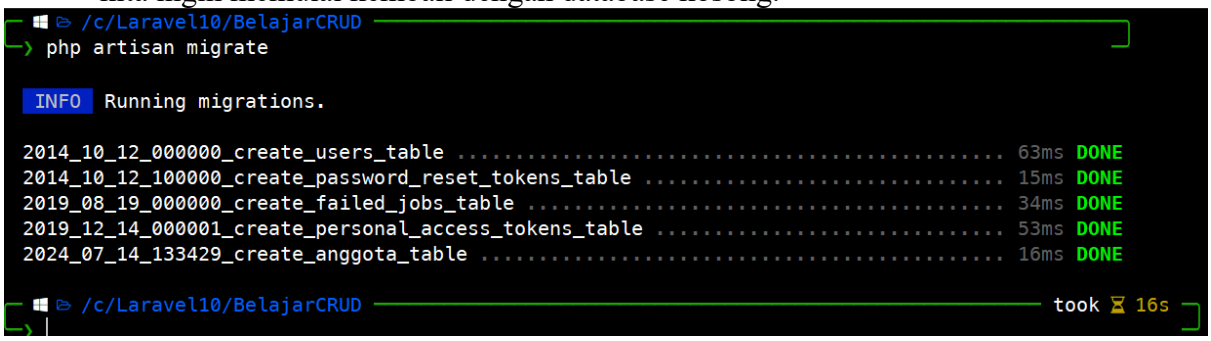
```
php artisan migrate
```

atau

```
php artisan migrate:fresh
```

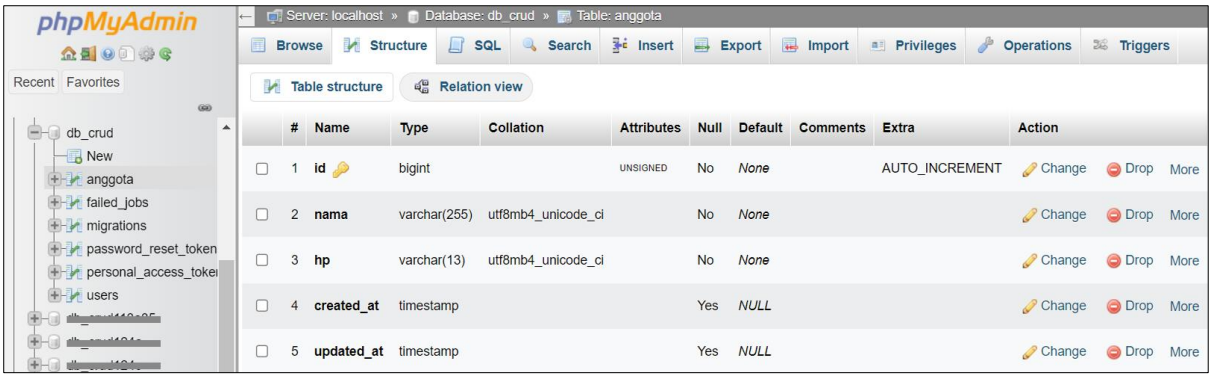
Kapan Menggunakan:

- php artisan migrate digunakan saat hanya ingin menjalankan migration baru tanpa mengganggu data yang sudah ada.
- php artisan migrate:fresh digunakan saat mengembangkan aplikasi dan perlu mengatur ulang database untuk memastikan semua migration berfungsi dengan benar atau ketika kita ingin memulai kembali dengan database kosong.



Gambar III. 5
Menjalankan *Migrations*

4. Maka pada saat kita cek di *Apache* terdapat beberapa tabel lainya yang dimana bawaan dari laravel & terdapat tabel anggota sesuai dengan rancangan yang kita rancanng gambar III.4, berikut tampilan *Structure table* anggota seperti terlihat pada gambar III.6



Gambar III. 6
Structure Table

3.4. Seeders

Seeders dalam Laravel digunakan untuk mengisi database dengan data awal dengan tujuan mempermudah pembuatan data ini secara otomatis. Pada saat ingin menjalankan *seeders* maka kita harus menyiapkan **Model** terlebih dulu.

1. Maka kita buat *model* dengan nama **Anggota** pada terminal:

```
php artisan make:model Anggota
```

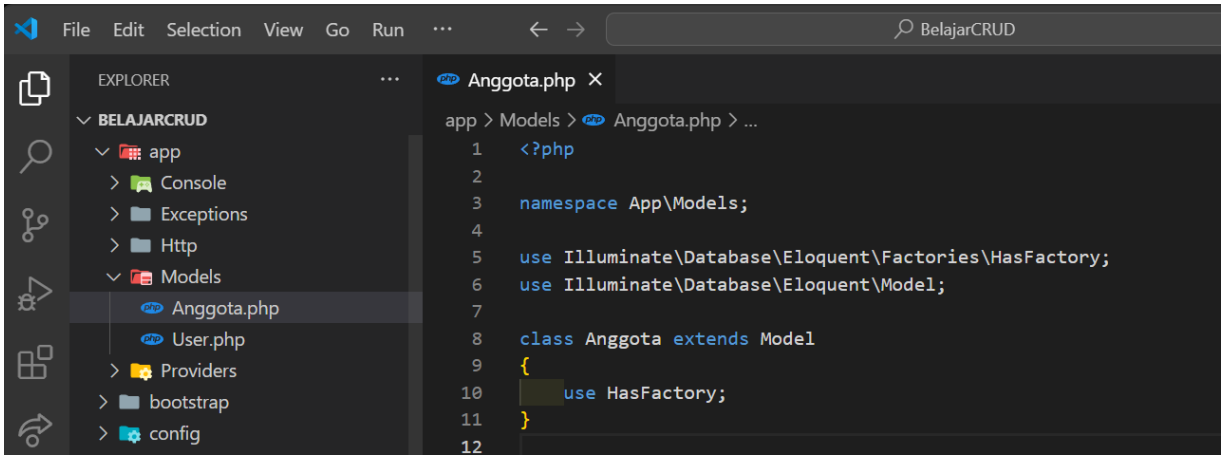
```
/c/Laravel10/BelajarCRUD took 16s
❯ php artisan make:model Anggota

  INFO  Model [C:\Laravel10\BelajarCRUD\app\Models\Anggota.php] created successfully.
```

Gambar III. 7
Membuat Model

2. Model dapat kita buka pada direktori *app/Models/anggota.php* seperti pada gambar III.8 & ubah *script model* **Anggota** sebagai berikut:

```php
<?php
namespace App\Models;

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

class Anggota extends Model
{
    public $timestamps = true;
    protected $table = "anggota";
    // protected $fillable = ['nama','hp'];
    protected $guarded = ['id'];
}
```

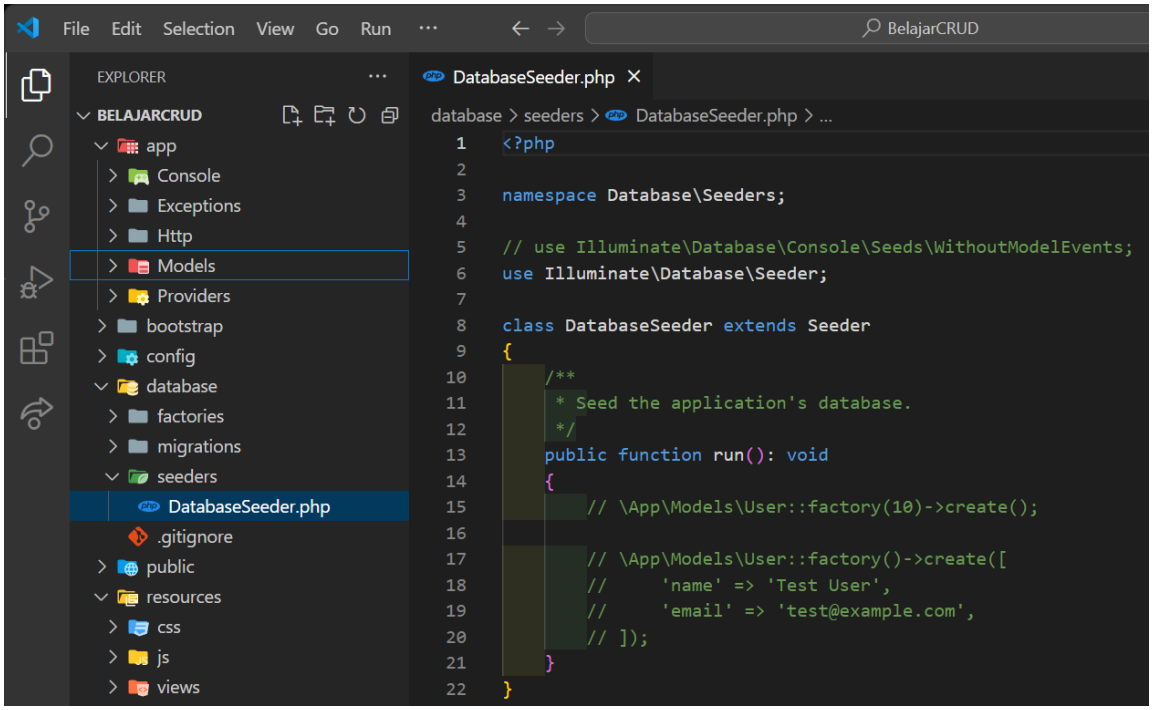
- Gunakan $fillable jika kita ingin secara eksplisit mengizinkan atribut tertentu untuk diisi. dengan fillable, semua atribut yang ingin diizinkan harus disebutkan satu per satu.
- Gunakan $guarded sebagai kebalikan dari fillable. Dalam guarded, hanya atribut yang disebutkan yang tidak dapat diisi, sedangkan semua atribut lainnya dapat diisi.



Gambar III. 8
Model Angggota

3. Buka file **DatabaseSeeder.php** pada direktori *database/seeders* seperti pada gambar III.9 & ubah *script DatabaseSeeder.php* sebagai berikut:

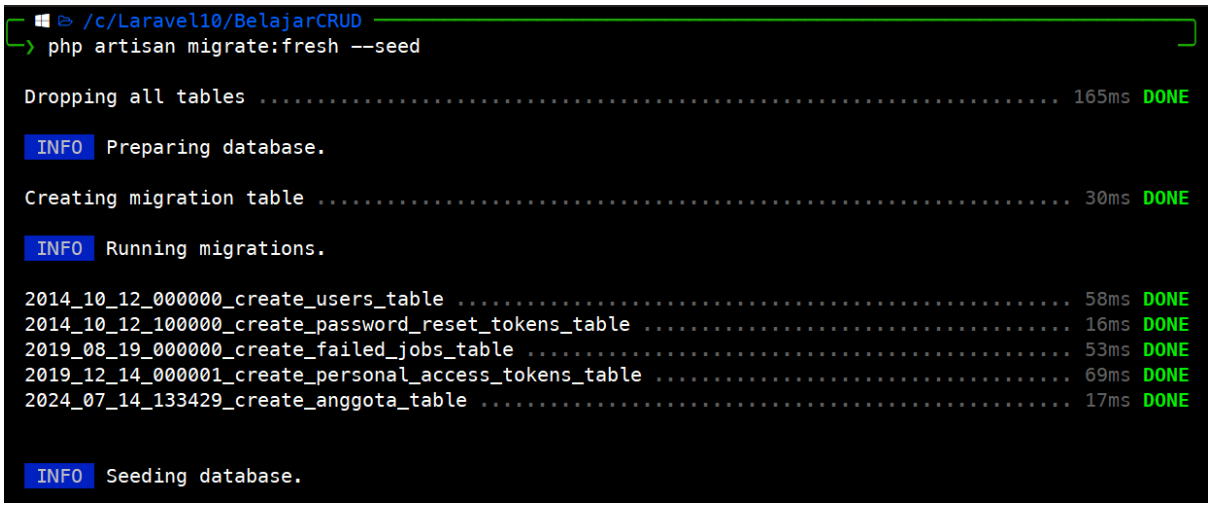
```php
<?php
namespace Database\Seeders;
use Illuminate\Database\Seeder;
use App\Models\Anggota;

class DatabaseSeeder extends Seeder
{
    public function run(): void
    {
        Anggota::create([
            'nama' => 'Sopian Aji',
            'hp' => '085123456781',
        ]);
        Anggota::create([
            'nama' => 'Husni Faqih',
            'hp' => '085123456782',
        ]);
        Anggota::create([
            'nama' => 'Rousyati',
            'hp' => '085123456783',
        ]);
    }
}
```



Gambar III. 9
DatabaseSeeder

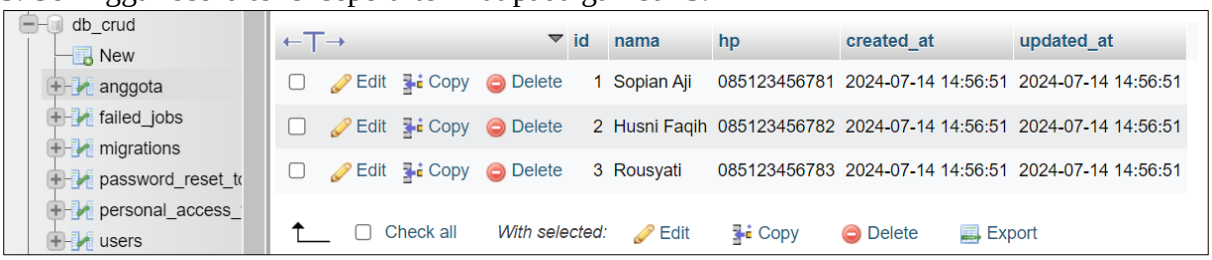
4. Untuk menjalankan *seeders* pada terminal seperti terlihat pada gambar III.10:

```
php artisan migrate:fresh --seed
```



Gambar III. 10
Menjalankan *seeders*

5. Sehingga record terisi seperti terlihat pada gambar 3.11



Gambar III. 11
Record Terisi Melalui *seeders*

Latihan Mandiri 3:
Portofolio sertifikasi kompetensi, Impentasikan Unit Kompetensi Software Development pada **Menerapkan Akses Basis Data**, dengan minimal memenuhi kriteria berikut. Tugas dikirim di LSM **MyBest**:

- server yang digunakan
- Konfigurasi Database.

Minggu Ke-4
CRUD (Create, Read, Update, Delete)

MVC (*Model, View, Controller*) adalah pendekatan perangkat lunak yang memisahkan logika aplikasi dari presentasinya. Pendekatan ini membagi aplikasi menjadi beberapa komponen, seperti manipulasi data, controller, dan antarmuka pengguna. MVC sangat berkaitan dengan *Object-Oriented Programming* (OOP) karena setiap komponen dapat diimplementasikan sebagai kelas dan objek, yang memanfaatkan prinsip-prinsip OOP seperti *enkapsulasi*, *pewarisan*, dan *polimorfisme*. Dengan menggunakan prinsip-prinsip OOP, MVC membantu dalam menciptakan aplikasi yang terstruktur, mudah dikelola, dan dapat diperluas.

- **Model**: Model mewakili struktur data dan biasanya berisi fungsi-fungsi yang membantu pengelolaan basis data, seperti memasukkan data, memperbarui data, dan sebagainya.
- **View**: View adalah bagian yang bertanggung jawab untuk menampilkan informasi kepada pengguna, biasanya dalam bentuk halaman web.
- **Controller**: Controller adalah komponen yang menghubungkan model dan view, bertindak sebagai perantara antara keduanya.

4.1. Penggunaan Resource Controller di Laravel

Untuk studi kasus masih melanjutkan pada *Project* **BelajarCRUD** dengan *Model* **Anggota** langkah berikutnya kita akan membuat *Controller* dengan nama **AnggotaController** namun pada CRUD kali ini kita akan menggunakan **–resource** seperti pada gambar IV.1. sehingga pada terminal sebagai berikut:

```
php artisan make:controller AnggotaController --resource
```

```
/c/Laravel10/BelajarCRUD
php artisan make:controller AnggotaController --resource

  INFO  Controller [C:\Laravel10\BelajarCRUD\app\Http\Controllers\AnggotaController.php] created succes
sfully.
```

Gambar IV. 1
Controller --resource

Berikut adalah kegunaan masing-masing *function* pada *Controller* yang dihasilkan dengan opsi *--resource* dalam Laravel:

- **index(),** digunakan menampilkan semua entri dari suatu model.
- **create(),** digunakan untuk menampilkan halaman input data baru.
- **store(Request $request),** digunakan untuk memproses dan menyimpan data yang dikirim dari *form create*.
- **show($id),** digunakan untuk menampilkan informasi lengkap dari satu entri data.
- **edit($id),** Digunakan untuk menampilkan halaman pengeditan data.
- **update(Request $request, $id),** Digunakan untuk memproses dan menyimpan perubahan data yang dikirim dari *form edit*.
- **destroy($id),** digunakan untuk menghapus satu entri data.

Setiap *function* ini sesuai dengan operasi CRUD dan mempermudah dalam pengelolaan data di aplikasi Laravel.

4.2. Menampilkan Data Pada Laravel

1. Pertama kita akan menampilkan data anggota secara keselurahan berdasarkan ID. Maka pada *AnggotaController function* ***index()*** berikut perubahannya:

```php
<?php
namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Anggota;

class AnggotaController extends Controller
{
    public function index()
    {
        $anggota = Anggota::orderBy('id', 'desc')->get();
        return view('v_anggota.index', [
            'judul' => 'Data Anggota',
            'index' => $anggota
        ]);
    }
    /**
     * dan berikutnya terdapat function  lainnya
     */
}
```

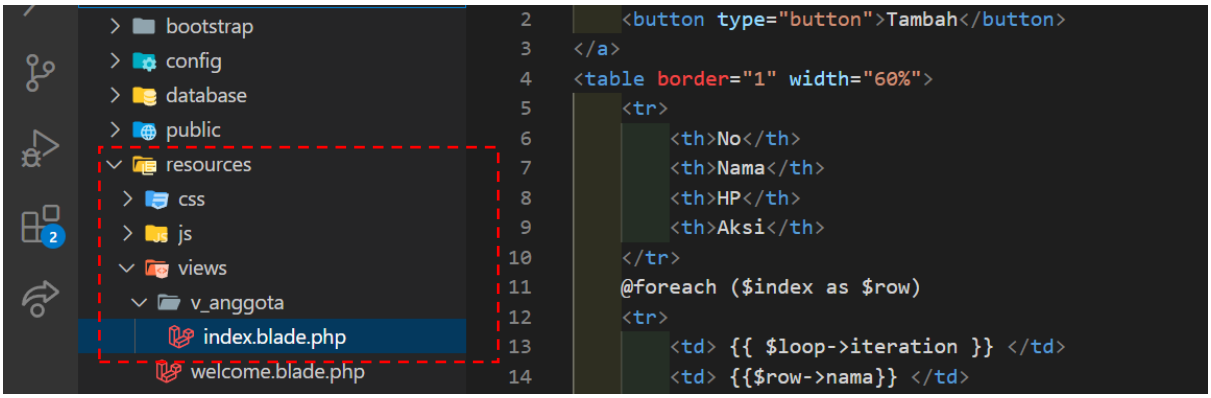
2. Maka pada *resources/views* kita tambahkan folder **v_anggota** & file dengan nama **index.blade.php** seperti yang terlihat pada gambar Gambar IV.2

```php
<h3> {{$judul}} </h3>
<a href="{{ route('anggota.create') }}">
    <button type="button">Tambah</button>
</a>
<table border="1" width="60%">
    <tr>
        <th>No</th>
        <th>Nama</th>
        <th>HP</th>
        <th>Aksi</th>
    </tr>
    @foreach ($index as $row)
    <tr>
        <td> {{ $loop->iteration }} </td>
        <td> {{$row->nama}} </td>
        <td> {{$row->hp}} </td>
        <td>
            <a href="{{ route('anggota.edit', $row->id) }}">
                <button type="button">Ubah</button>
            </a>
            <form action="{{ route('anggota.destroy', $row->id) }}" method="POST">
                @method('delete')
                @csrf
                <button type="submit">Hapus</button>
            </form>
        </td>
    </tr>
    @endforeach
</table>
```

Tambahkan style="display: inline-block;" pada aksi **Ubah** dan **Hapus** digunakan untuk memastikan bahwa elemen <a> dan <form> ditampilkan sebagai elemen blok yang tetap berada dalam satu baris dengan elemen lainnya, menjaga keselarasan dan tata letak yang rapi pada halaman web. Berikut perubahan setelah ditambahan style="display: inline-block;"

```php
<h3> {{$judul}} </h3>
<a href="{{ route('anggota.create') }}">
```

```
    <button type="button">Tambah</button>
</a>
<table border="1" width="60%">
    <tr>
        <th>No</th>
        <th>Nama</th>
        <th>HP</th>
        <th>Aksi</th>
    </tr>
    @foreach ($index as $row)
    <tr>
        <td> {{ $loop->iteration }} </td>
        <td> {{$row->nama}} </td>
        <td> {{$row->hp}} </td>
        <td>
            <a href="{{ route('anggota.edit', $row->id) }}" style="display: inline-block;">
                <button type="button">Ubah</button>
            </a>
            <form action="{{ route('anggota.destroy', $row->id) }}" method="POST"
style="display: inline-block;">
                @method('delete')
                @csrf
                <button type="submit">Hapus</button>
            </form>
        </td>
    </tr>
    @endforeach
</table>
```



Gambar IV. 2
Index.blade.php()

3. Pada routes**web.php** sebagai berikut:

```
<?php
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\AnggotaController;

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

Route::resource('anggota', AnggotaController::class);
```

4. Dengan demikan kita pada melihat hasilnya pada *browser* dengan mengetikkan alamat http://localhost:8000/*anggota*

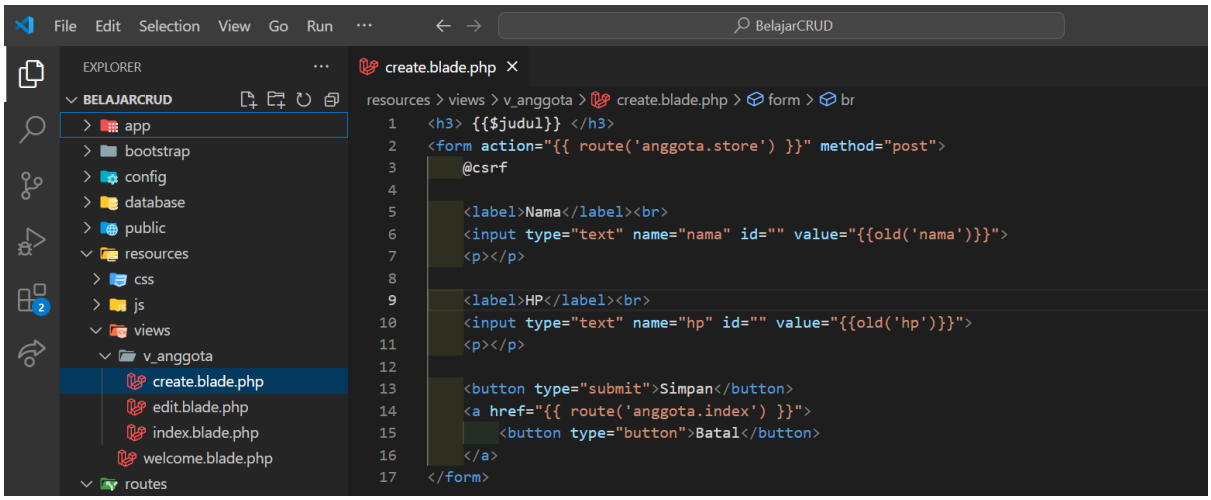
4.3. Simpan Data Pada Laravel

1. Selanjutnya aksi **Tambah**, ketika tombol tambah diklik maka akan ditampilkan form tambah dengan url http://localhost:8000/anggota/create. Berikut perubahan *Controller* yakni *AnggotaController* pada *function* ***create()***:

```
    public function create()
    {
        return view('v_anggota.create', [
            'judul' => 'Tambah Anggota'
        ]);
    }
```

2. Sehingga, pada *resources/views/v_anggota* kita tambahkan file dengan nama **create.blade.php** seperti yang terlihat pada gambar Gambar IV.3

```
<h3> {{$judul}} </h3>
<form action="{{ route(anggota.store') }}" method="post">
    @csrf
    <label>Nama</label><br>
    <input type="text" name="nama" id="">
    <p></p>
    <label>HP</label><br>
    <input type="text" name="hp" id="">
    <p></p>
    <button type="submit">Simpan</button>
    <a href="{{ route('anggota.index') }}">
        <button type="button">Batal</button>
    </a>
</form>
```



Gambar IV. 3
Create.blade.php

Pada *Input Text* tambahkan value="{{old('nama_text')}}"> sangat berguna untuk menjaga data input pengguna tetap ada selama siklus validasi form, sehingga data yang telah di-input tidak hilang. Berikut setelah form ditambahakan value="{{old('nama_text')}}">. Tambahan lainnya yakni kita juga dapat menambahan placeholder yang berfungsi untuk memudahkan pengguna dalam mengisi formulir, terutama ketika informasi yang diminta tidak langsung jelas dari konteks form.

```
<h3> {{$judul}} </h3>
<form action="{{ route('anggota.store') }}" method="post">
    @csrf
    <label>Nama</label><br>
    <input type="text" name="nama" id="" value="{{old('nama')}}" placeholder="Masukkan Nama
Lengkap">
```

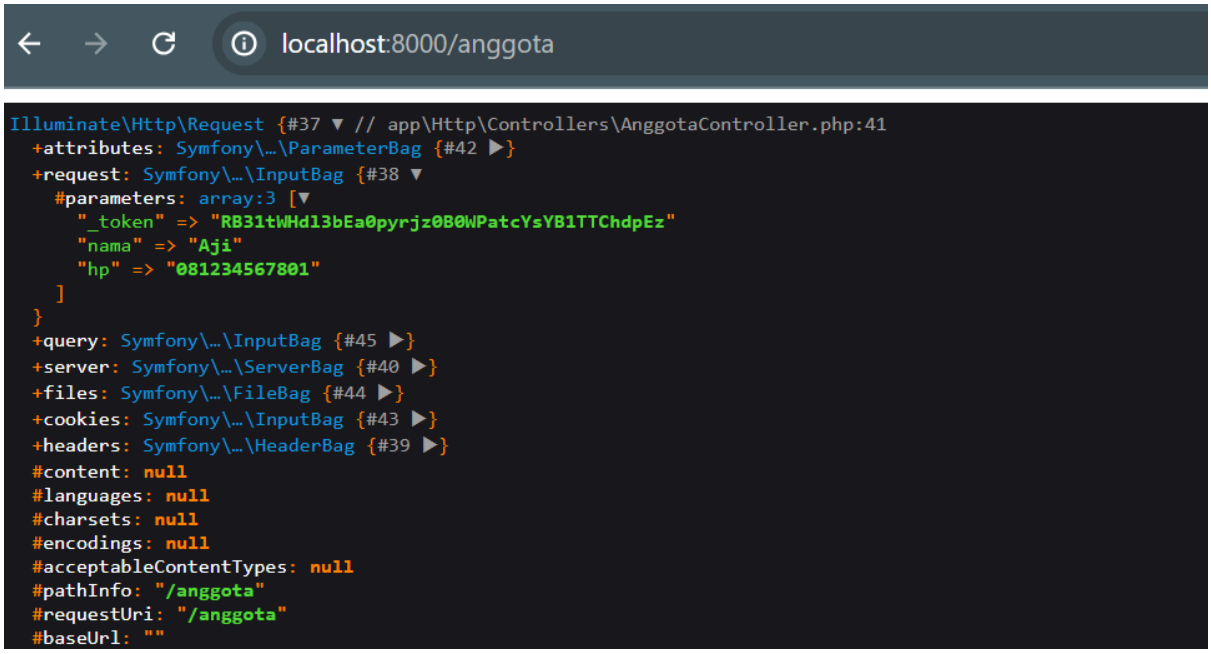
```
    <p></p>
    <label>HP</label><br>
    <input type="text" name="hp" id="" value="{{old('hp')}}" placeholder="Masukkan Nomor
HP">
    <p></p>
    <button type="submit">Simpan</button>
    <a href="{{ route('anggota.index') }}">
        <button type="button">Batal</button>
    </a>
</form>
```

3. Sebelum kita mengklik tombol **Simpan**, periksa data yang dikirim dengan cara pada *Controller* di *function store* dengan menggunakan ddd($request) atau dd($request). Dengan demikian, kita dapat memastikan bahwa *form* berfungsi dengan baik. Berikut adalah perubahan *AnggotaController* pada *function* store:

```
    public function store(Request $request)
    {
        // Debugging request data
        dd($request);
        // atau untuk lebih banyak informasi debugging
        // ddd($request);
    }
```

Penjelasan ddd($request) atau dd($request):

- **dd($request)** singkatan dari "dump and die". Ini akan menampilkan informasi tentang variabel $request dan menghentikan eksekusi *script*. Sangat berguna untuk debugging. Sehingga saat data dikirim ke *function store* terlihat seperti gambar IV.4
- **ddd($request)** singkatan dari "dump, die, and debug". Ini melakukan hal yang sama seperti dd($request), tetapi juga menampilkan lebih banyak informasi debugging yang berguna.

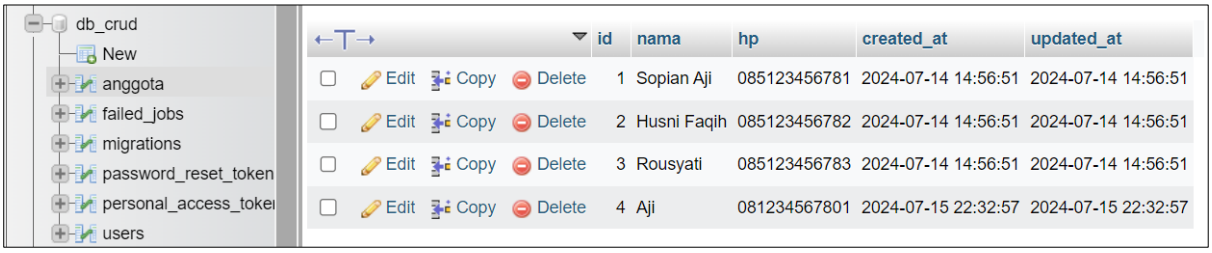


Gambar IV. 4
dd($request)

4. Jika *request* sudah sesuai dengan yang diharapkan, tambahkan *script* untuk menyimpan data ke *database* pada *function store()*. Jika data berhasil tersimpan, kita akan kembali ke halaman

yang menggunakan *fungsi index()*, sehingga record baru akan masuk dan seluruh data anggota akan ditampilkan. Kita juga dapat memeriksa di database apakah record telah bertambah, seperti yang terlihat pada gambar IV.5

```
    public function store(Request $request)
    {
        // Debugging request data
        // dd($request);
        // atau untuk lebih banyak informasi debugging
        // ddd($request);
        $validatedData = $request->validate([
            'nama' => 'required|max:255',
            'hp' => 'required|min:10|max:13',
        ]);
        Anggota::create($validatedData);
        return redirect('/anggota');
    }
```



Gambar IV. 5
Record Baru Berhasil Tersimpan

4.4. Hapus Data Pada Laravel

1. Berikutnya pada aksi **Hapus**, dimana saat aksi Hapus diklik data akan terhapus. Tambahkan *script* pada *function destroy* pada *AnggotaController* sebagai berikut:

```
    public function destroy(string $id)
    {
        $anggota = Anggota::findOrFail($id);
        $anggota->delete();
        return redirect('/anggota');
    }
```

2. Pada **index.blade.php** yakni pada folder **v_anggota**, pastikan aksi hapus telah tersedia

```
<form action="{{ route('anggota.destroy', $row->id) }}" method="POST" style="display:
inline-block;">
                @method('delete')
                @csrf
                <button type="submit">Hapus</button>
</form>
```

4.5. Ubah Data Pada Laravel

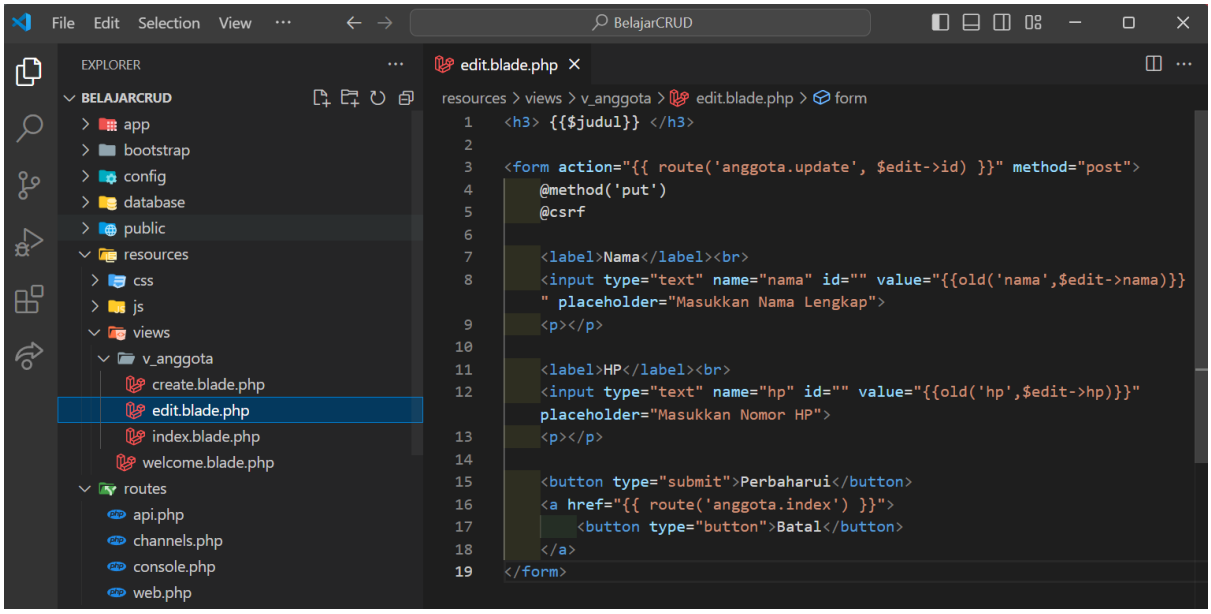
1. Berikutnya pada aksi **Ubah,** dimana saat aksi Ubah diklik akan tampil form ubah data. Tambahkan *script* pada *function* ***edit()*** *&* *function* ***Update ()*** pada *AnggotaController* sebagai berikut:

```
    public function edit(string $id)
    {
        $anggota = Anggota::find($id);
        return view('v_anggota.edit', [
            'judul' => 'Ubah Anggota',
            'edit' => $anggota
        ]);
    }

    /**
```

```
     * Update the specified resource in storage.
     */
    public function update(Request $request, string $id)
    {
        $rules = [
            'nama' => 'required|max:100',
            'hp' => 'required|min:10|max:13',
        ];
        $validatedData = $request->validate($rules);
        Anggota::where('id', $id)->update($validatedData);
        return redirect('/anggota');
    }
```

2. Dengan *view* pada *resources/views/v_anggota* kita tambahkan file dengan nama ***edit.blade.php*** seperti yang terlihat pada gambar Gambar IV.6

```
<h3> {{$judul}} </h3>
<form action="{{ route('anggota.update', $edit->id) }}" method="post">
    @method('put')
    @csrf
    <label>Nama</label><br>
    <input type="text" name="nama" id="" value="{{old('nama',$edit->nama)}}"
placeholder="Masukkan Nama Lengkap">
    <p></p>
    <label>HP</label><br>
    <input type="text" name="hp" id="" value="{{old('hp',$edit->hp)}}"
placeholder="Masukkan Nomor HP">
    <p></p>
    <button type="submit">Perbaharui</button>
    <a href="{{ route('anggota.index') }}">
        <button type="button">Batal</button>
    </a>
</form>
```



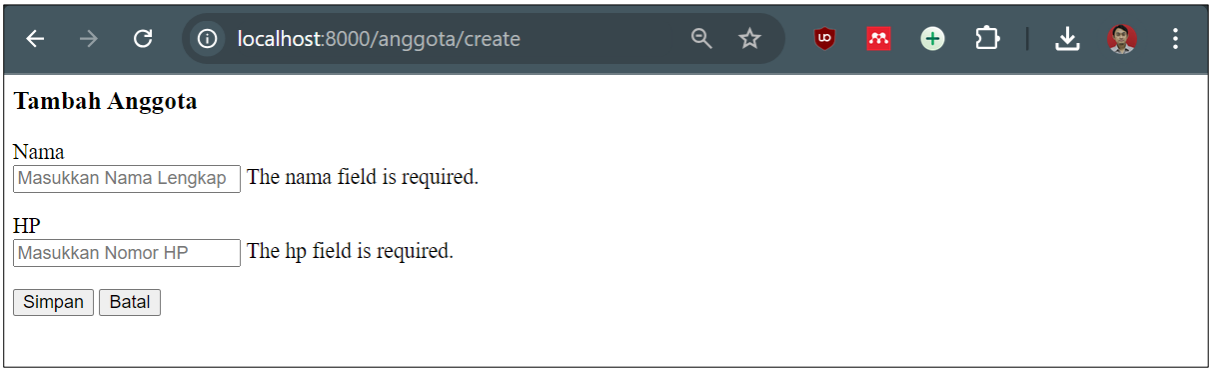
Gambar IV. 6
edit.blade.php

4.6. Menerapkan *invalid-feedback*

Pada *create.blade.php* dan *edit.blade.php*, kita bisa menambahkan ***invalid-feedback***. Dengan menambahkan ***invalid-feedback***, kita dapat memberikan umpan balik yang jelas dan langsung kepada pengguna ketika mereka mengisi form dengan data yang tidak valid.

a. Perubahaan ***create.blade.php*** setelah ditambahan *invalid-feedback*, sehingga saat form disini kosong terdapat pesan bahwa data tidak boleh kosong seperti pada gambar IV.7

```
<h3> {{$judul}} </h3>
<form action="{{ route('anggota.store') }}" method="post">
    @csrf
    <label>Nama</label><br>
    <input type="text" name="nama" id="" value="{{old('nama')}}" placeholder="Masukkan Nama
Lengkap" class="form-control @error('nama') is-invalid @enderror">
    @error('nama')
    <span class="invalid-feedback alert-danger" role="alert">
        {{$message}}
    </span>
    @enderror
    <p></p>
    <label>HP</label><br>
    <input type="text" name="hp" id="" value="{{old('hp')}}" placeholder="Masukkan Nomor
HP" class="form-control @error('hp') is-invalid @enderror">
    @error('hp')
    <span class="invalid-feedback alert-danger" role="alert">
        {{$message}}
    </span>
    @enderror
    <p></p>
    <button type="submit">Simpan</button>
    <a href="{{ route('anggota.index') }}">
        <button type="button">Batal</button>
    </a>
</form>
```

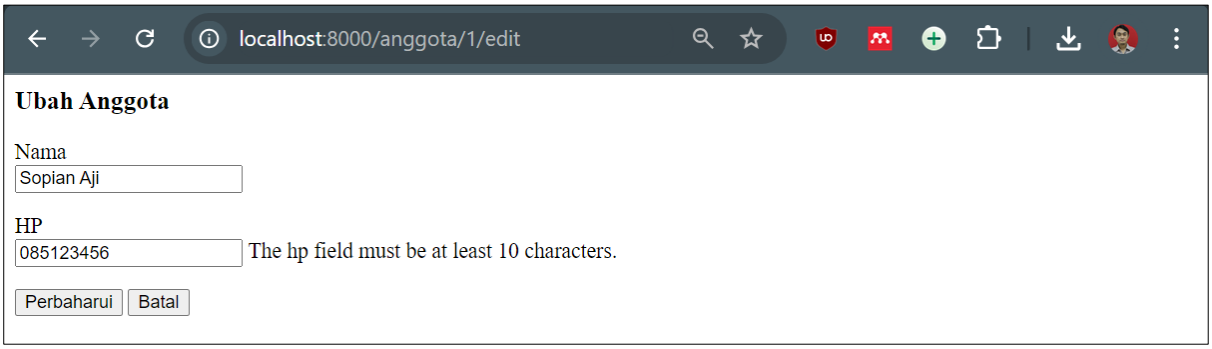


Gambar IV. 7
View Create Setelah di Tambahkan **invalid-feedback**

b. Perubahaan ***edit.blade.php*** setelah ditambahan *invalid-feedback*, misalnya, jika kolom HP diisi dengan kurang dari 10 karakter, maka akan muncul pesan yang menyatakan bahwa data tidak boleh kurang dari 10 karakter seperti pada gambar IV.8

```
<h3> {{$judul}} </h3>
<form action="{{ route('anggota.update', $edit->id) }}" method="post">
    @method('put')
    @csrf
    <label>Nama</label><br>
    <input type="text" name="nama" id="" value="{{old('nama',$edit->nama)}}"
placeholder="Masukkan Nama Lengkap" class="form-control @error('nama') is-invalid
@enderror">
    @error('nama')
    <span class="invalid-feedback alert-danger" role="alert">
        {{$message}}
    </span>
```

```
    @enderror
    <p></p>
    <label>HP</label><br>
    <input type="text" name="hp" id="" value="{{old('hp',$edit->hp)}}"
placeholder="Masukkan Nomor HP" class="form-control @error('hp') is-invalid @enderror">
    @error('hp')
    <span class="invalid-feedback alert-danger" role="alert">
        {{$message}}
    </span>
    @enderror
    <p></p>
    <button type="submit">Perbaharui</button>
    <a href="{{ route('anggota.index') }}">
        <button type="button">Batal</button>
    </a>
</form>
```



Gambar IV. 8
View Create Setelah di Tambahkan invalid-feedback

Berikul tampilan lengkap *script Controller* dengan nama **AnggotaController**:

```
<?php

namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Models\Anggota;

class AnggotaController extends Controller
{

   public function index()
   {
      $anggota = Anggota::orderBy('id', 'desc')->get();
      return view('v_anggota.index', [
         'judul' => 'Data Anggota',
         'index' => $anggota
      ]);
   }

   public function create()
   {
      return view('v_anggota.create', [
         'judul' => 'Tambah Anggota'
      ]);
   }

   public function store(Request $request)
   {
      // Debugging request data
      // dd($request);
      // atau untuk lebih banyak informasi debugging
```

```
        // ddd($request);
        $validatedData = $request->validate([
            'nama' => 'required|max:255',
            'hp' => 'required|min:10|max:13',
        ]);
        Anggota::create($validatedData);
        return redirect('/anggota');
    }

    public function show(string $id)
    {
        //
    }

    public function edit(string $id)
    {
        $anggota = Anggota::find($id);
        return view('v_anggota.edit', [
            'judul' => 'Ubah Anggota',
            'edit' => $anggota
        ]);
    }

    public function update(Request $request, string $id)
    {
        $rules = [
            'nama' => 'required|max:100',
            'hp' => 'required|min:10|max:13',
        ];
        $validatedData = $request->validate($rules);
        Anggota::where('id', $id)->update($validatedData);
        return redirect('/anggota');
    }

    public function destroy(string $id)
    {
        $anggota = Anggota::findOrFail($id);
        $anggota->delete();
        return redirect('/anggota');
    }
}
```

Latihan Mandiri 4:

Portofolion sertifikasi kompetensi, Impentasikan Unit Kompetensi Software Development pada **Melakukan Debugging**, Dengan minimal memenuhi kriteria berikut:

- penggunaan ddd($request) atau dd($request).
- invalid-feedback
- Penanganan pesan kesalahan pada *browser* dan cara penyelesaiannya

Minggu Ke-5
Authenticate (Logika Login) & Pengujian Unit

Pada studi kasus kali ini adalah *Project* berkelompok, dimana pengerjaannya dapat dilakukan bersama dengan berkelompok yang sudah ditentukan. Pada pembahasan modul ini studi kasus yang akan dibangun yakni aplikasi toko online. *Authenticate* adalah proses verifikasi identitas pengguna untuk mengakses aplikasi atau sistem. Untuk membuat *Authenticate* pada Laravel 10, kita dapat membuat *Authenticate* secara otomatis misalnya menggunakan *Laravel Breeze*, *Laravel Jetstream*, *Laravel Fortify* dll, atau kita juga dapat membuat *Authenticate* secara manual. Pada studi kasus aplikasi toko online yang akan dibahas dalam modul pembelajaran ini, autentikasi akan dibuat secara manual.

5.1. Mempersiapkan *Project* Pada Studi Kasus

1. Buat *Project* baru dengan nama **TokoOnline** dan buat database dengan nama **db_tokoonline** (nama *Project* dan database dapat diganti sesuai kesepakatan kelompok).
2. konfigurasi database pada file **.env** sebagai berikut:

```
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=db_tokoonline
DB_USERNAME=root
DB_PASSWORD= # isi jika password tersedia
```

3. Pastikan terminal sudah berada di direktori *Project* TokoOnline. Kemudian Jalankan migrasi pada terminal, dan secara otomatis akan dibuatkan tabel yang disediakan oleh Laravel

```
php artisan migrate
```

5.2. *Blueprint* Tabel User

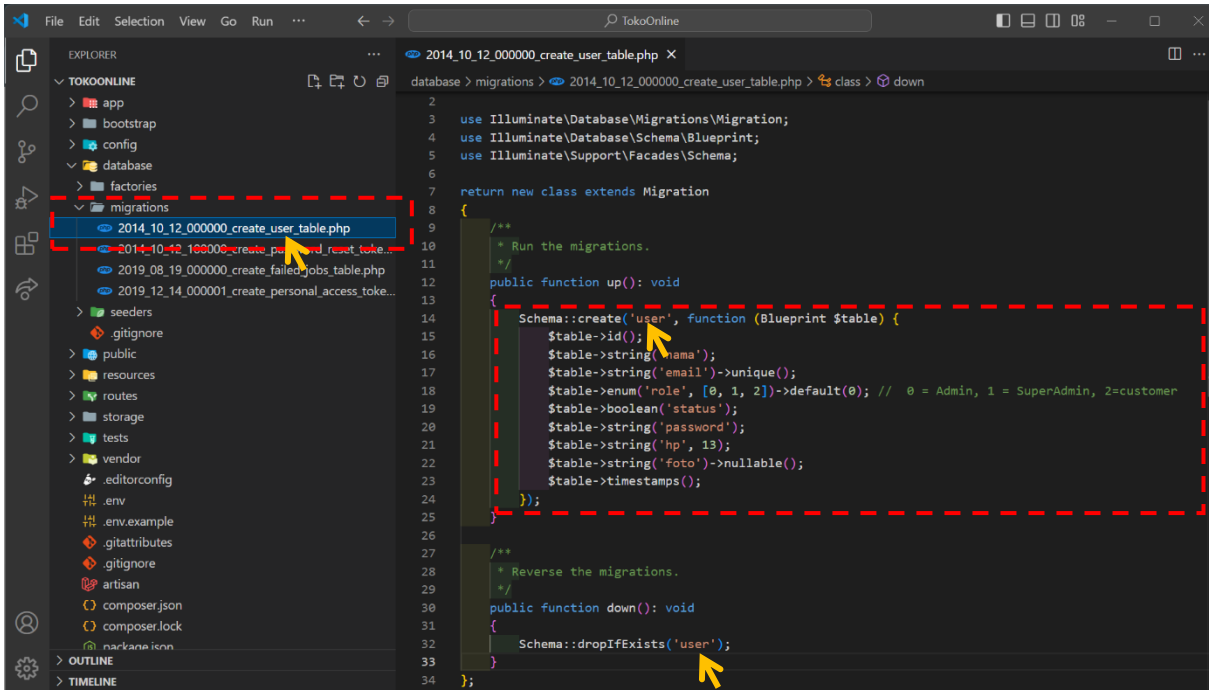
1. Kita ubah tabel migrations **users** menjadi **user** demikan juga pada *blueprint* seperti pada gambar V.1, dengan rancangan tabel V.1 & *blueprint* user sebagai berikut:

```
Schema::create('user', function (Blueprint $table) {
            $table->id();
            $table->string('nama');
            $table->string('email')->unique();
            $table->enum('role', [0, 1, 2])->default(0); //  0 = Admin, 1 = SuperAdmin, 2=customer
            $table->boolean('status'); //  0 = Belum aktif, 1=Aktif
            $table->string('password');
            $table->string('hp', 13);
            $table->string('foto')->nullable();
            $table->timestamps();
        });
```

Tabel V. 1
Tabel User

Fild	Tipe Data	Keterangan
id	Bigint	Primary Key
nama	Varchar(255)	
email	Varchar(255)	
role	Enum	
status	Boolean	
password	Varchar(255)	
hp	Varchar(13)	
foto	Varchar(255)	Nullable

created_at	Timestamp	
updated_at	Timestamp	



Gambar V. 1
Blueprint Tabel User

Peran (*role*) di tabel user menggunakan tipe data **enum** dengan nilai:

- **0 untuk Admin**: Pengguna dengan hak akses administratif yang dapat mengelola sistem.
- **1 untuk SuperAdmin**: Pengguna dengan hak akses tertinggi yang memiliki kontrol penuh atas sistem.
- **2 untuk Customer**: Peran ini dapat diubah sesuai dengan studi kasus yang diambil, misalnya diganti dengan peran sebagai Member, Student, Teacher, dll.

Fungsi dari **Status** menggunakan tipe data **boolean** dengan nilai:

- **0** : Status user belum aktif atau dinonaktifkan sehingga user tidak diberikan akses untuk login.
- **1** : User dapat mengakses dan login ke sistem

Sedangkan untuk **foto**, kita beri atribut **nullable** sehingga record dapat dikosongkan dan tidak wajib diisi.

5.3. Model User Untuk Manajemen User

1. Kemudian kita ubah model **User**, berikut perubahan *script* lengkap pada model **User**:

```php
<?php

namespace App\Models;

// use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Laravel\Sanctum\HasApiTokens;
```

```php
class User extends Authenticatable
{
    use HasApiTokens, HasFactory, Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    protected $table = "user";
    protected $fillable = [
        'nama',
        'email',
        'role',
        'status',
        'password',
        'hp',
        'foto',
    ];

    /**
     * The attributes that should be hidden for serialization.
     *
     * @var array<int, string>
     */
    protected $hidden = [
        'password',
        'remember_token',
    ];

    /**
     * The attributes that should be cast.
     *
     * @var array<string, string>
     */
    protected $casts = [
        'email_verified_at' => 'datetime',
        'password' => 'hashed',
    ];
}
```

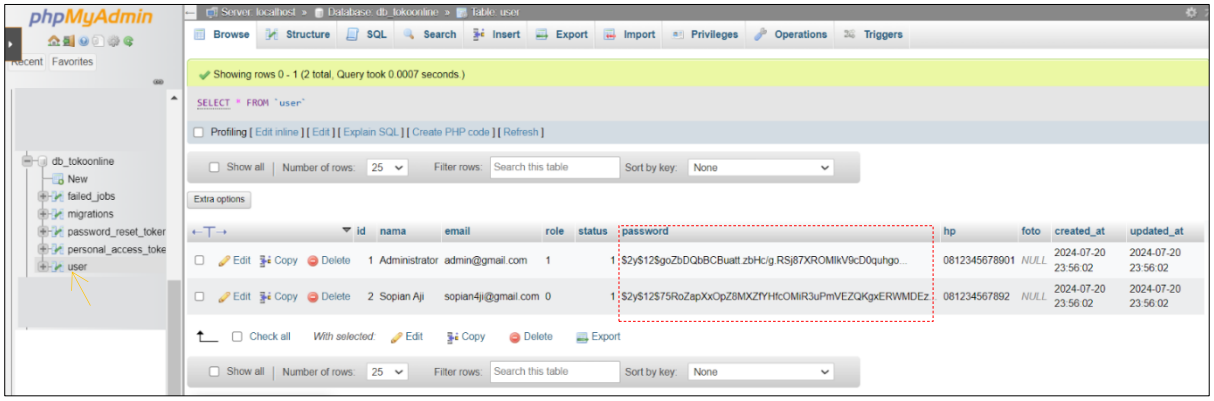
2. Pada *seeder* kita tentukan nilai tabel dari user, sebagai berikut:

```php
<?php

namespace Database\Seeders;

// use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;
use App\Models\User;

class DatabaseSeeder extends Seeder
{
    /**
     * Seed the application's database.
     */
    public function run(): void
    {
        User::create([
            'nama' => 'Administrator',
            'email' => 'admin@gmail.com',
            'role' => '1',
```

```
            'status' => 1,
            'hp' => '0812345678901',
            'password' => bcrypt('P@55word'),
        ]);
        #untuk record berikutnya silahkan, beri nilai berbeda pada nilai: nama, email, hp dengan
nilai masing-masing anggota kelompok
       User::create([
            'nama' => 'Sopian Aji',
            'email' => 'sopian4ji@gmail.com',
            'role' => '0',
            'status' => 1,
            'hp' => '081234567892',
            'password' => bcrypt('P@55word'),
        ]);
    }
}
```

3. Kemudian jalankan *seeder* pada terminal sehingga tabel kini menjadi user dan record terisi sesuai dengan *seeder* yang kita buat. Untuk *field* password akan terenkripsi (jika lupa password maka bisa dicek kembali pada *seeder*) periksa di *phpmyadmin* seperti yang terlihat pada gambar V.2

```
php artisan migrate:fresh --seed
```



Gambar V. 2
Tabel User Pada phpmyadmin

5.4. Mempersiapkan Halaman Utama

1. Sebelum kita membuat *Authenticate* dan Logika Login, kita siapkan halaman utama jika login berhasil dilakukan. Kita akan buat membuat controller dengan nama **BerandaController**

```
php artisan make:controller BerandaController
```

script lengkap **BerandaController** sebagai berikut:

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class BerandaController extends Controller
{
    public function berandaBackend()
    {
        return view('backend.v_beranda.index', [
            'judul' => 'Halaman Beranda',
```

```
        ]);
    }
}
```

Dengan view *function berandaBackend()* pada direktori resources\views\backend\v_beranda\ **index.blade.php** sebagai berikut:

```
<h3> {{$judul}} </h3>
<p>
    Selamat Datang, <b>Nama_User</b> pada aplikasi Toko Online dengan hak akses yang anda
miliki sebagai <b>Role_User</b> ini adalah halaman utama dari aplikasi ini.
</p>
```

2. Kemudian pada routes**web.php** sebagai berikut:

```
<?php
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\BerandaController;

Route::get('/', function () {
    return view('welcome');
});
Route::get('backend/beranda', [BerandaController::class, 'berandaBackend'])-
>name('backend.beranda');
```

Sehingga pada *browser* kita dapat mengetikkan alamat http://localhost:8000/backend/beranda

5.5. Menerapkan Yield

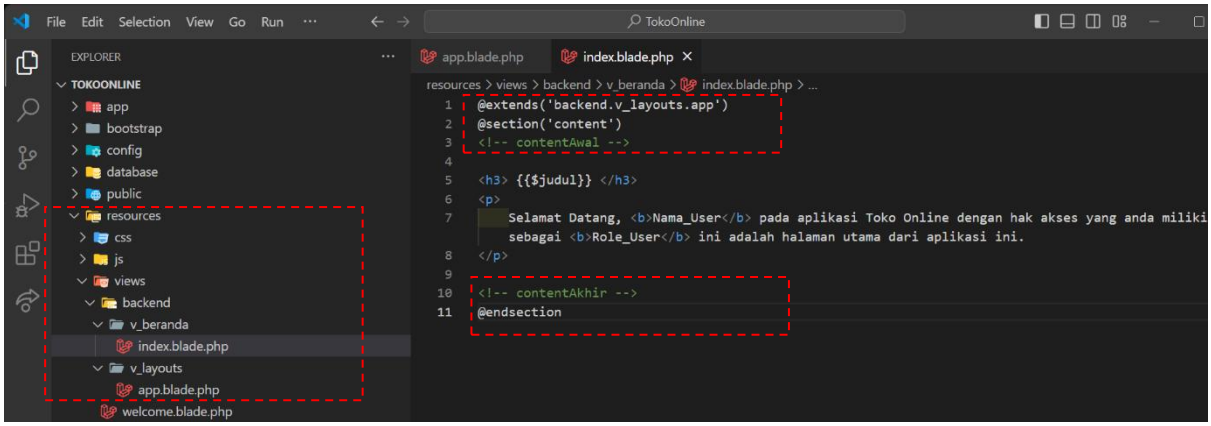
Kita akan buat layout utama dengan menambahkan **yield** pada *Project* kita. **Yield** adalah salah satu fitur yang disediakan oleh Laravel untuk mengelola *template blade*. Fitur ini memungkinkan kita untuk mendefinisikan bagian yang dinamis dalam layout utama, yang kemudian dapat diisi oleh konten spesifik dari halaman-halaman lain. Pada direktori resources\views\backend\v_layouts**app.blade.php** sehingga berikut *script* lengkap pada **app.blade.php** terlihat seperti gambar V.3

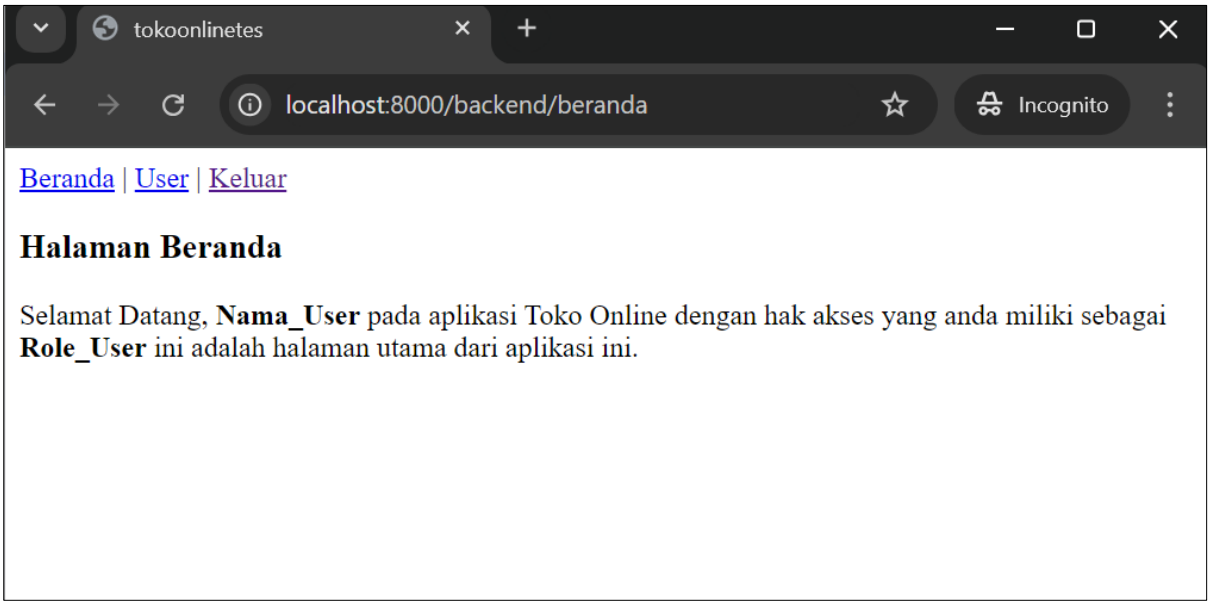
```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>tokoonline</title>
</head>
<body>
```

```
    <a href="{{ route('backend.beranda') }}">Beranda</a> |
    <a href="#">User</a> |
    <a href="#">Keluar</a>
    <p></p>

    <!-- @yieldAwal -->
    @yield('content')
    <!-- @yieldAkhir-->
</body>
</html>
```

Sehingga kita lakukkan perubahan pada resources\views\backend\v_beranda**index.blade.php**

```
@extends('backend.v_layouts.app')
@section('content')
<!-- contentAwal -->

<h3> {{$judul}} </h3>
<p>
    Selamat Datang, <b>Nama_User</b> pada aplikasi Toko Online dengan hak akses yang anda
miliki sebagai <b>Role_User</b> ini adalah halaman utama dari aplikasi ini.
</p>

<!-- contentAkhir -->
@endsection
```





Gambar V. 3
Penerapan Yield

5.6. Membut Login Pada Laravel

1. Setelah kita konfigurasi database, menyiapkan data **user** & Halaman Utama, langkah selanjutnya adalah membuat *controller* login dengan nama **LoginController**

```
php artisan make:controller LoginController
```

script lengkap untuk **LoginController** sebagai berikut:

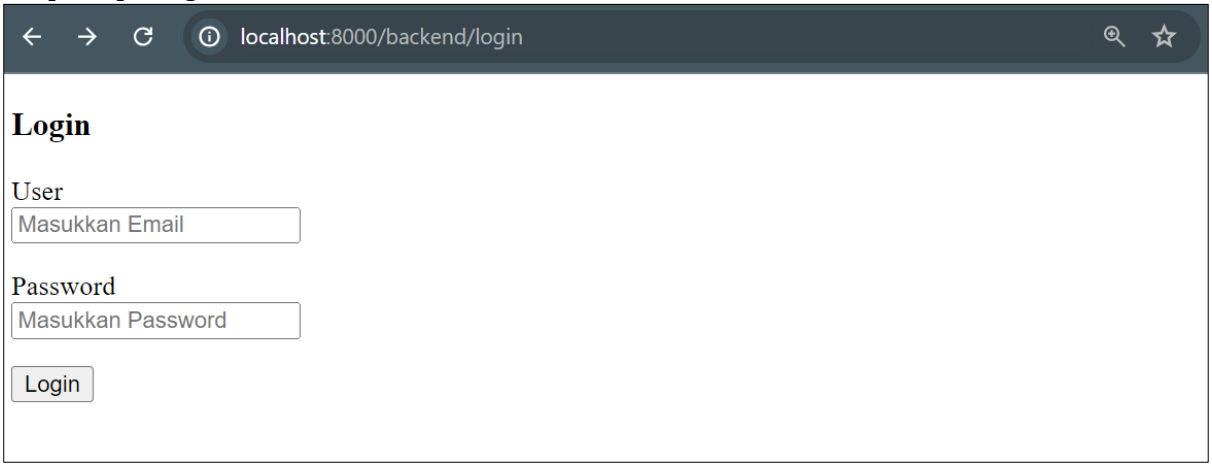
```php
<?php
namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class LoginController extends Controller
{
    public function loginBackend()
    {
        return view('backend.v_login.login', [
            'judul' => 'Login',
        ]);
    }

    public function authenticateBackend(Request $request)
    {
        $credentials = $request->validate([
            'email' => 'required|email',
            'password' => 'required'
        ]);

        if (Auth::attempt($credentials)) {
            if (Auth::user()->status == 0) {
                Auth::logout();
                return back()->with('error', 'User belum aktif');
            }
            $request->session()->regenerate();
            return redirect()->intended(route('backend.beranda'));
        }
        return back()->with('error', 'Login Gagal');
    }

    public function logoutBackend()
    {
        Auth::logout();
        request()->session()->invalidate();
        request()->session()->regenerateToken();
        return redirect(route('backend.login'));
    }
}
```

2. Sehingga view pada **LoginController** *function loginBackend()* pada direktori resources\views\backend\v_login**login.blade.php** dengan *scrip* lengkap sebagai berikut:

```php
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>tokoonline</title>
</head>

<body>

    <h3> {{$judul}}</h3>
    <!-- error -->
```

```
    @if(session()->has('error'))
    <div class="alert alert-danger alert-dismissible" role="alert">
        <button type="button" class="close" data-dismiss="alert" aria-label="Close"><span
aria-hidden="true">&times;</span></button>
        <strong>{{ session('error')}} </strong>
    </div>
    @endif
    <!-- errorEnd -->

    <form action="{{ route('backend.login') }}" method="post">
        @csrf
        <label>User</label><br>
        <input type="text" name="email" id="" value="{{old('email')}}" class="form-control
@error('email') is-invalid @enderror" placeholder="Masukkan Email">
        @error('email')
        <span class="invalid-feedback alert-danger" role="alert">
            {{$message}}
        </span>
        @enderror
        <p></p>

        <label>Password</label><br>
        <input type="password" name="password" id="" value="{{old('password')}}"
class="form-control @error('password') is-invalid @enderror" placeholder="Masukkan
Password">
        @error('password')
        <span class="invalid-feedback alert-danger" role="alert">
            {{$message}}
        </span>
        @enderror
        <p></p>

        <button type="submit">Login</button>
    </form>

</body>
</html>
```

3. Untuk melihat hasilnya ketikan alamat http://localhost:8000/backend/login di *browser* akan tampil seperti gambar V.4



Gambar V. 4
Halaman Login

4. Kita lakukan perubahan pada routes**web.php** sehingga ketika kita mengakses http://localhost:8000/ di *browser* tidak lagi http://localhost:8000/backend/login. kita akan diarahkan langsung ke halaman login gambar V.4. Selain itu, kita tambahkan *middleware* pada

BerandaController sehingga controller hanya dapat diakses oleh pengguna yang sudah berhasil login:

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\BerandaController;
use App\Http\Controllers\LoginController;

Route::get('/', function () {
    // return view('welcome');
    return redirect()->route('backend.login');
});
Route::get('backend/beranda', [BerandaController::class, 'berandaBackend'])-
>name('backend.beranda')->middleware('auth');

Route::get('backend/login', [LoginController::class, 'loginBackend'])-
>name('backend.login');
Route::post('backend/login', [LoginController::class, 'authenticateBackend'])-
>name('backend.login');
Route::post('backend/logout', [LoginController::class, 'logoutBackend'])-
>name('backend.logout');
```

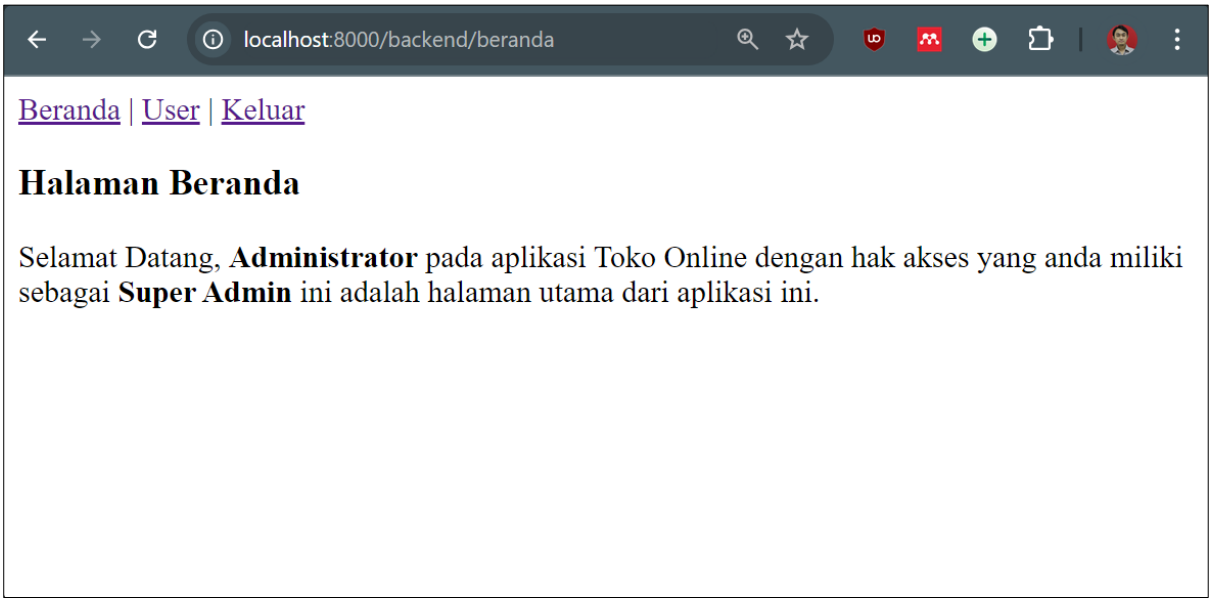
5. Pada direktori resources\views\backend\v_layouts**app.blade.php** kita ubah kembali sehingga berikut *script* lengkap:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>tokoonline</title>
</head>

<body>
    <a href="{{ route('backend.beranda') }}">Beranda</a> |
    <a href="#">User</a> |
    <a href="" onclick="event.preventDefault(); document.getElementById('keluar-
app').submit();">Keluar</a>
    <p></p>

    <!-- @yieldAwal -->
    @yield('content')
    <!-- @yieldAkhir-->

    <!-- keluarApp -->
    <form id="keluar-app" action="{{ route('backend.logout') }}" method="POST" class="d-
none">
        @csrf
    </form>
    <!-- keluarAppEnd -->
</body>

</html>
```

6. Sehingga kita lakukkan perubahan kembali pada resources\views\backend\v_beranda**index.blade.php** sehingga halaman beranda dapat menggetahui nama user yang login dan juga *role* dari user.

```
@extends('backend.v_layouts.app')
@section('content')
<!-- contentAwal -->

<h3> {{$judul}} </h3>
<p>
```

```
    Selamat Datang, <b>{{ Auth::user()->nama }}</b> pada aplikasi Toko Online dengan hak
akses yang anda miliki sebagai
    <b>
        @if (Auth::user()->role ==1)
        Super Admin
        @elseif(Auth::user()->role ==0)
        Admin
        @endif
    </b>
    ini adalah halaman utama dari aplikasi ini.
</p>

<!-- contentAkhir -->
@endsection
```

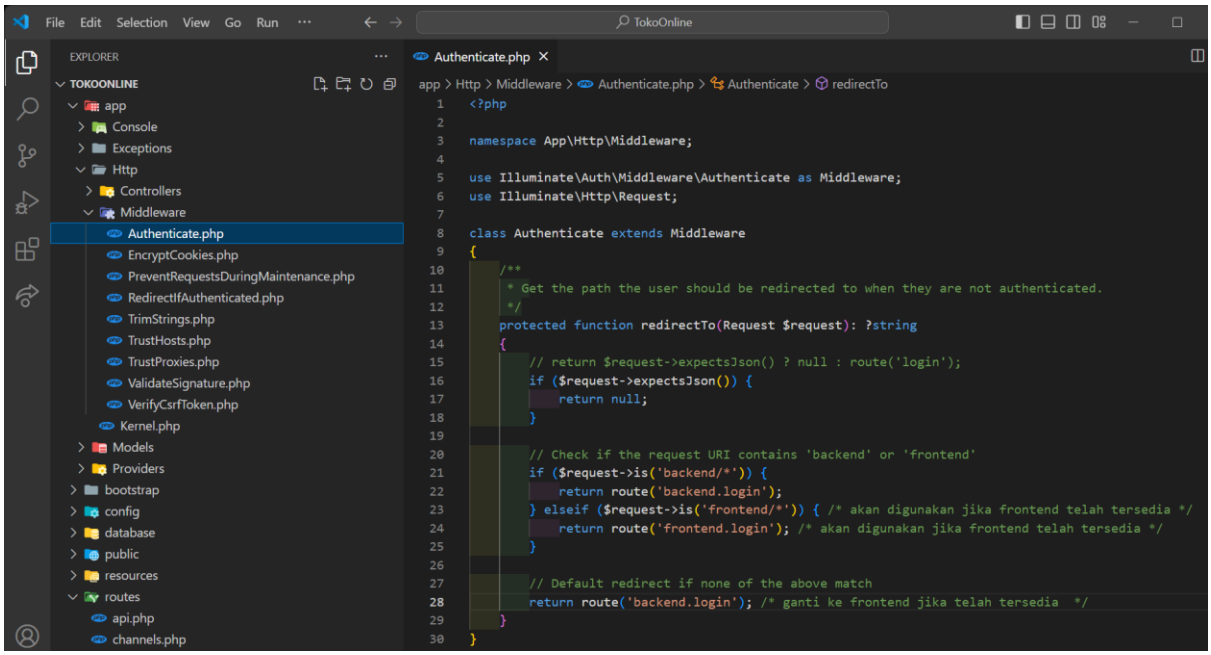


Gambar V. 5
Login sebagai Role Super Admin

Gambar V. 6
Login sebagai Role Admin

7. Konfigurasi Http\Middleware**Authenticate.php** seperti pada gambar V.6, dengan *script* sebagai berikut:

```php
<?php

namespace App\Http\Middleware;

use Illuminate\Auth\Middleware\Authenticate as Middleware;
use Illuminate\Http\Request;

class Authenticate extends Middleware
{
    /**
     * Get the path the user should be redirected to when they are not authenticated.
     */
    protected function redirectTo(Request $request): ?string
    {
        // return $request->expectsJson() ? null : route('login');
        if ($request->expectsJson()) {
            return null;
        }

        // Check if the request URI contains 'backend' or 'frontend'
        if ($request->is('backend/*')) {
            return route('backend.login');
        } elseif ($request->is('frontend/*')) { /* akan digunakan jika frontend telah tersedia */
            return route('frontend.login'); /* akan digunakan jika frontend telah tersedia */
        }

        // Default redirect if none of the above match
        return route('backend.login'); /* ganti ke frontend jika telah tersedia  */
    }
}
```



Gambar V. 7
Authenticate.php

5.7. Pengujian Terhadap Form Login

Pengujian Black Box adalah metode pengujian perangkat lunak di mana penguji mengevaluasi fungsionalitas suatu aplikasi tanpa melihat atau mempertimbangkan kode sumber internalnya. Pengujian ini fokus pada input yang diberikan ke sistem dan output yang dihasilkan,

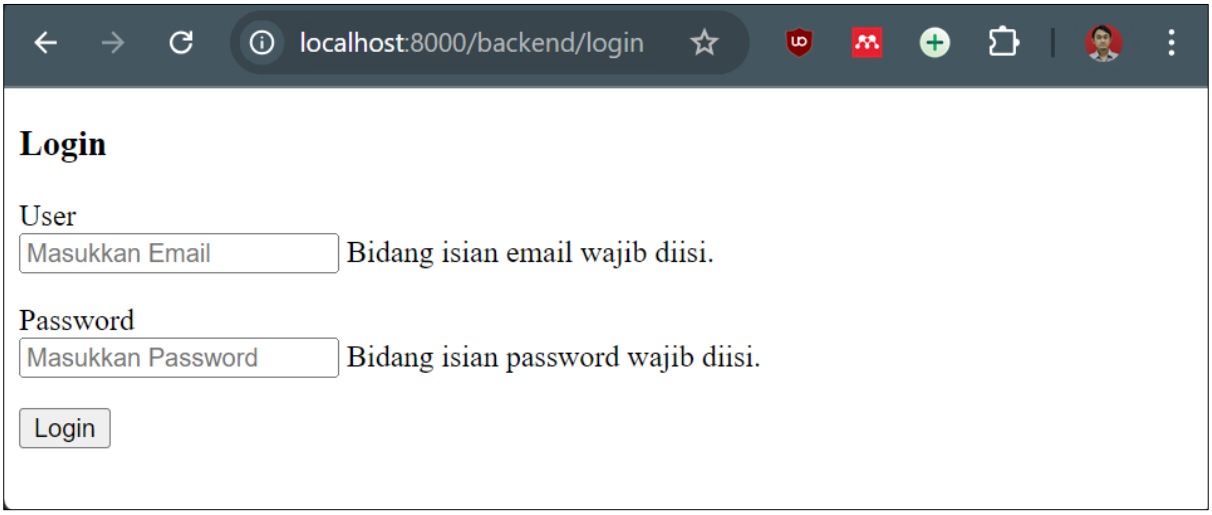
memastikan bahwa sistem berfungsi sesuai dengan spesifikasi dan persyaratan yang ditetapkan.

Pengujian Black Box pada Halaman Login melibatkan pengujian berbagai skenario penggunaan yang mungkin dilakukan oleh pengguna. Contoh pengujian yang dapat dilakukan pada Form Login:

Tabel V. 2
Hasil Pengujian Black Box Testing Halaman Login

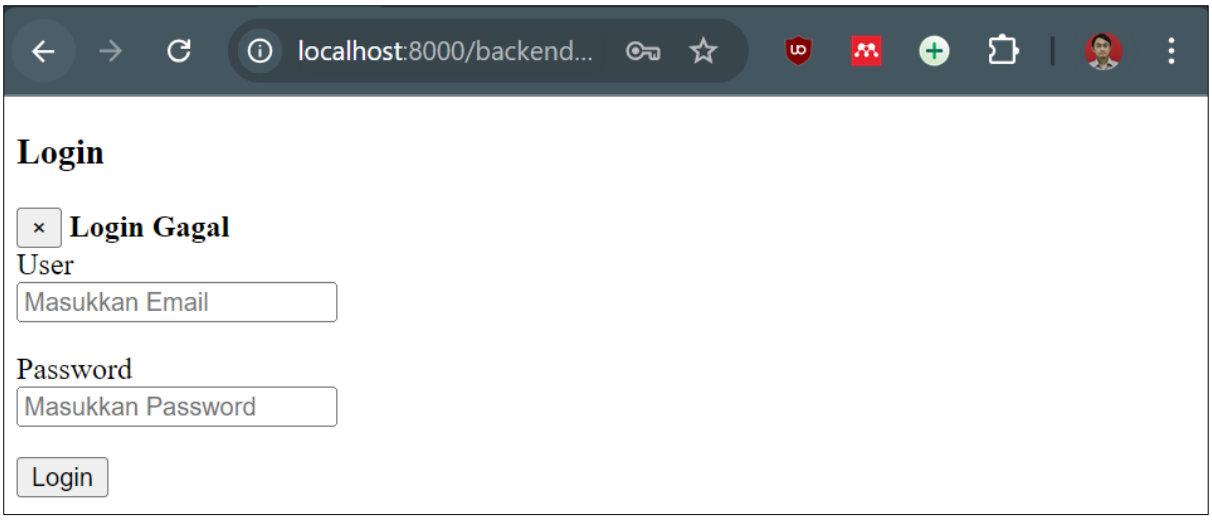
No	Skenario pengujian	Test case	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	Email User dan Password tidak diisi kemudian klik tombol login	Email User:(kosong) Password:(kosong)	Sistem akan menolak, pada Email User dan menampilkan "Bidang isian email wajib diisi". Dan pada Password dan menampilkan "Bidang isian password wajib diisi"	Sesuai harapan	Valid
2	Email User tidak menggunakan format email dan Password tidak diisi kemudian klik tombol login	Email User: admin Password:(kosong)	Sistem akan menolak, pada Email User dan menampilkan "Isian email harus berupa alamat surel yang valid". Dan pada Password dan menampilkan "Bidang isian password wajib diisi"	Sesuai harapan	Valid
3	Mengetikkan semua kondisi salah, baik pada Email User atau Password	Email User: user@gmail.com Password:S@n41	Sistem akan menolak, pada Email User dan Password kosong, dengan menampilkan pesan "Login Gagal".	Sesuai harapan	Valid
4	Mengetikkan salah satu kondisi, salah pada Email User atau Password kemudian klik tombol login	Email User: admin@gmail.com Password:S@n41	Sistem akan menolak, pada Email User dan Password kosong, dengan menampilkan pesan "Login Gagal".	Sesuai harapan	Valid
5	Mengetikkan Email User atau Password dengan data yang benar, tetapi status kondisi 0 (status akun tidak aktif)	Email User: sopian4ji@gmail.com Password:P@55word	Sistem akan menolak, pada Email User dan Password kosong, dengan menampilkan pesan "Status User tidak aktif".	Sesuai harapan	Valid

6	Mengetikkan Email User atau Password dengan data yang benar dan status kondisi 1 (status akun aktif)	Email User: admin@gmail.com Password:P@55word	Sistem menerima akses login dan kemudian langsung menampilkan menu utama.	Sesuai harapan	Valid

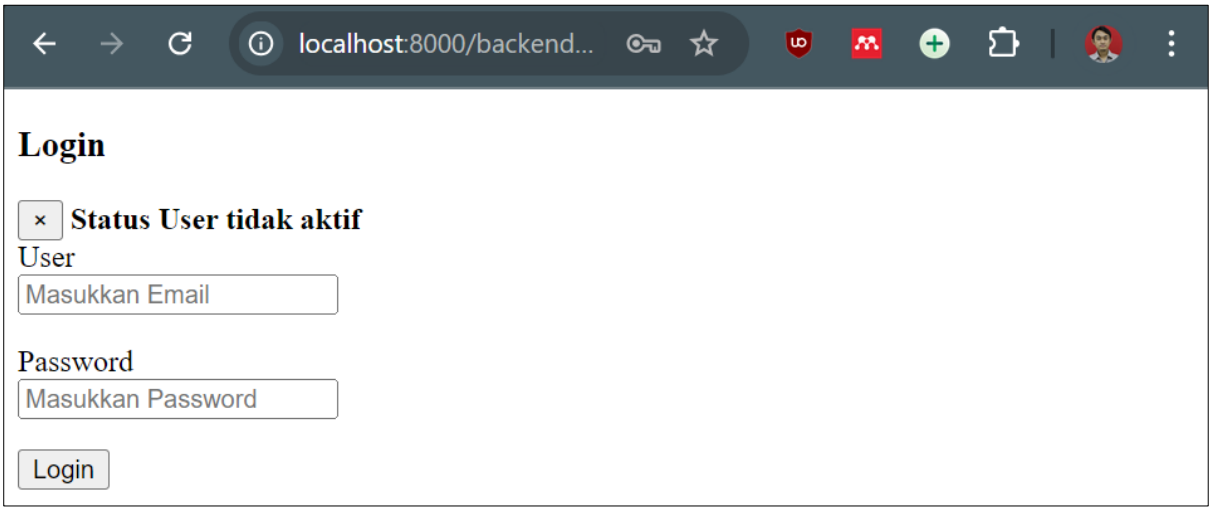


Gambar V. 8
Pengujian Pada Nomor 1

Gambar V. 9
Pengujian Pada Nomor 2



Gambar V. 10
Pengujian Pada Nomor 3 & 4



Gambar V. 11
Pengujian Pada Nomor 5

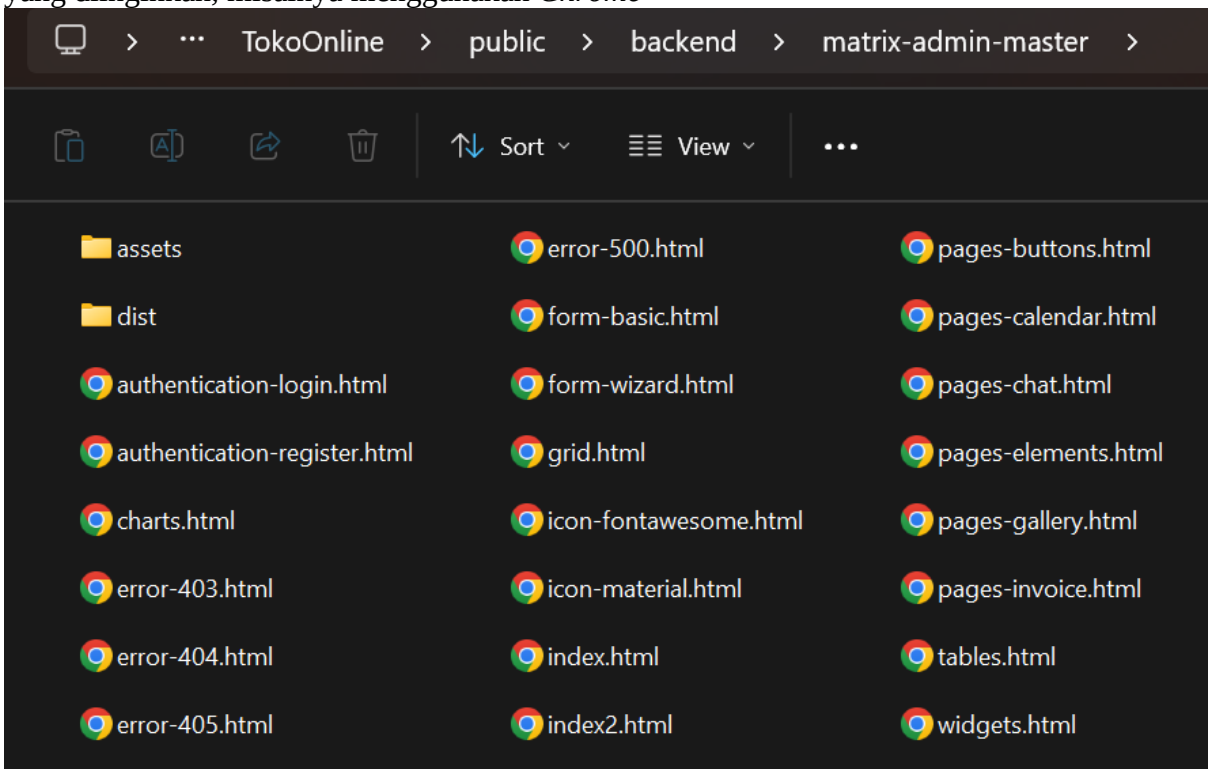
Latihan Mandiri 5:
Portofolion sertifikasi kompetensi, Impentasikan Unit Kompetensi Software Development pada **Menggunakan Library atau Komponen Pre-Existing**.

Minggu Ke-6
Template HTML, CSS, Bootstrap, dan JavaScript

Untuk template pada studi kasus toko online menggunakan template dengan nama Matrix Admin – dengan menggunakan Free HTML5 Bootstrap 4, untuk template dapat di download pada https://bit.ly/LaravelWebPro2.

6.1. Mengenal Template Admin

1. Setelah template berhasil diunduh, simpan pada direktori TokoOnline\public\backend dan ekstrak file tersebut (hasil ekstraksi untuk demo template) seperti pada gambar VI.1. Untuk menjalankan demo, dapat langsung menggunakan *browser* atau klik kanan dan pilih *browser* yang diinginkan, misalnya menggunakan *Chrome*



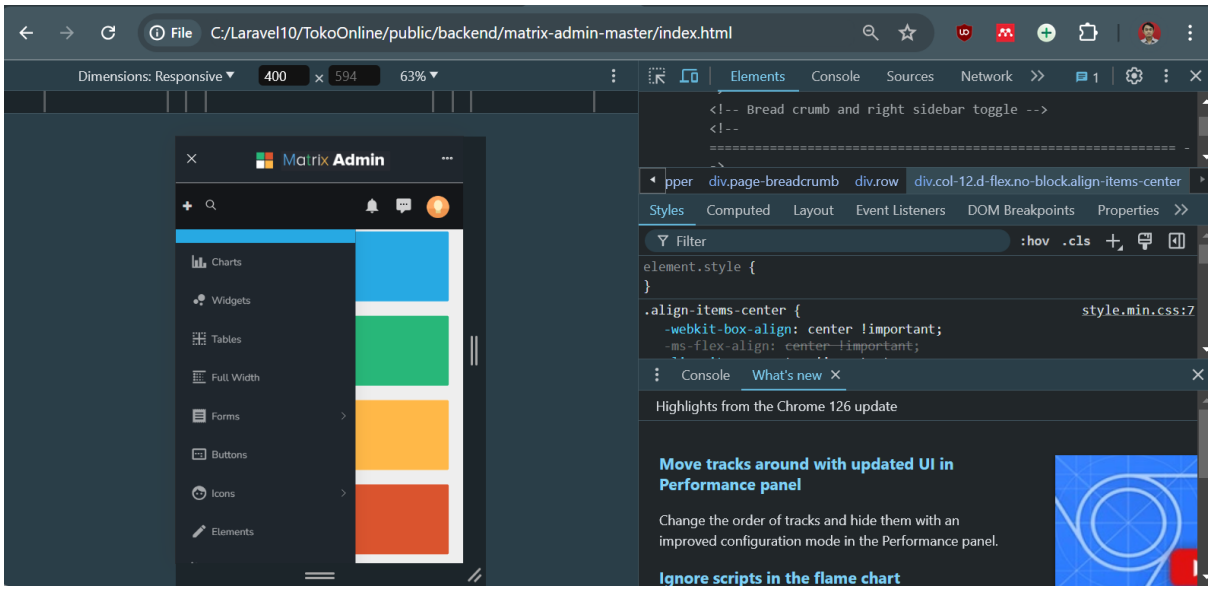
Gambar VI. 1
Template - Matrix Admin

2. Berikut demo ***Template - Matrix Admin*** dimana fitur utama yang digunakan sudah sesuai dengan kebutuhan *Project* toko online:

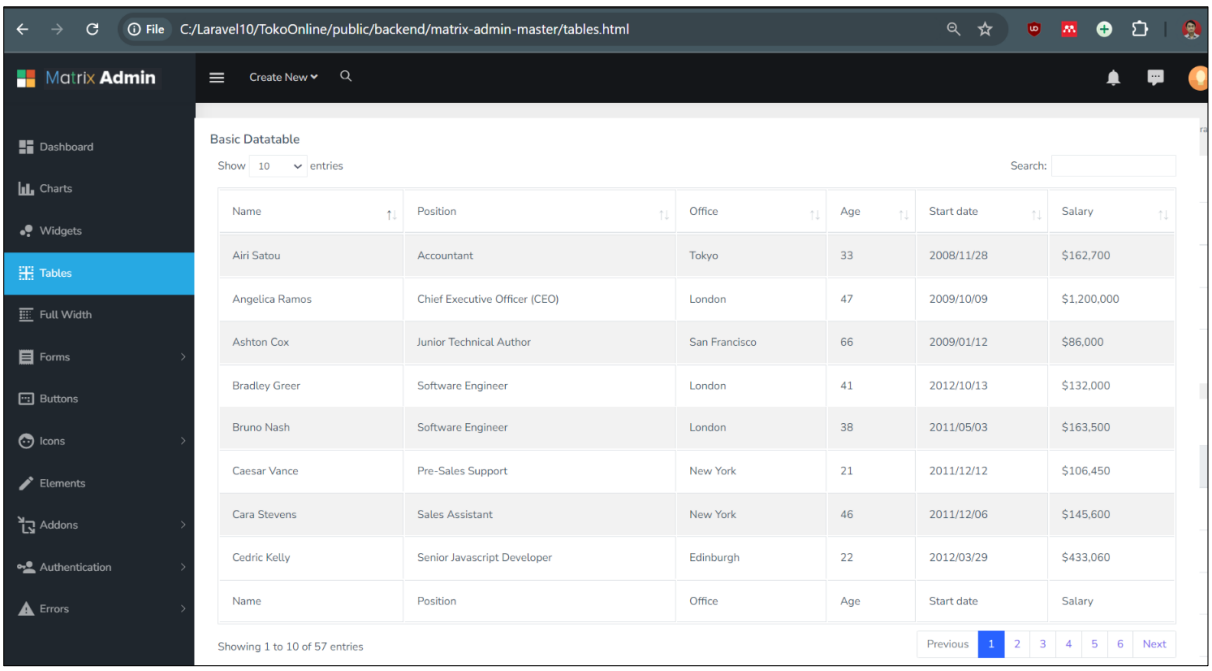
Tabel 6.1 Tabel VI. 1
Key Features

Key Features	Keterangan
Fully responsive	Dapat diakses baik menggunakan PC ataupun mobile secara responsif, dapat dicek dengan cara klik kanan pada browser pilih **inspect** pada gambar VI.2
Basic data tables	Memudahkan untuk mengelompokkan data, pencarian data, atau paginasi, yakni pada file **tables.html** seperti pada gambar VI.3.
chart	Untuk menampilkan grafik pada gambar VI.4

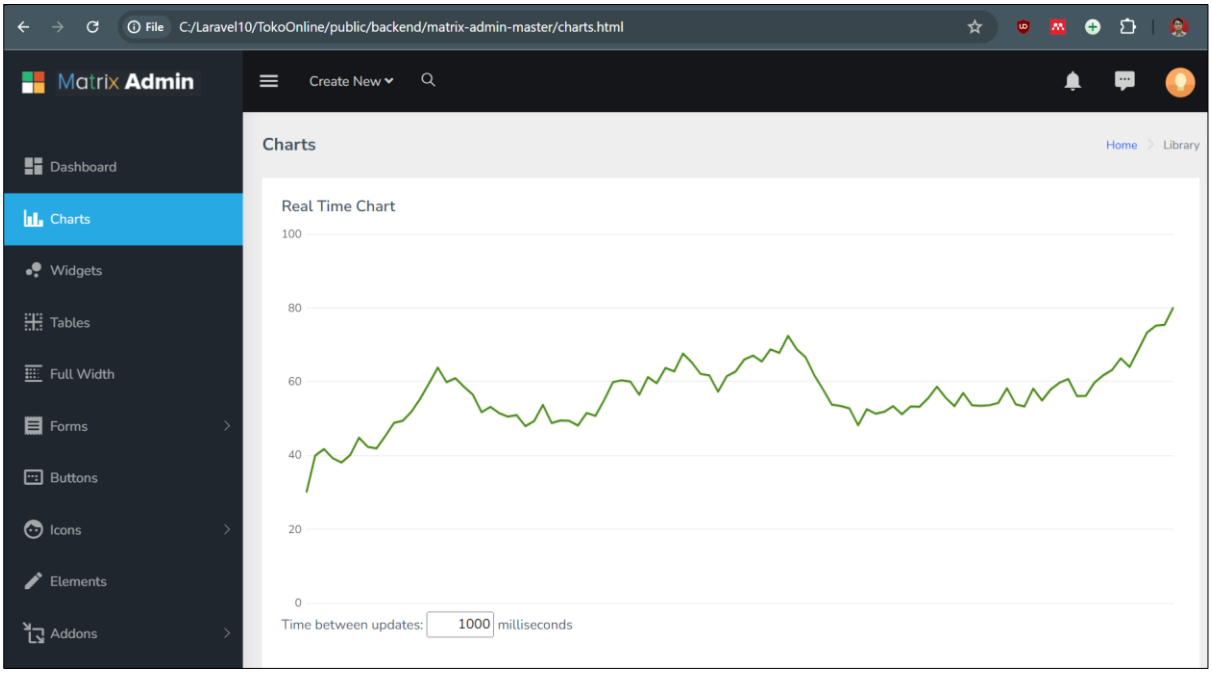
Drop-down menu	Sidebar dimana menu didalam menu, seperti pada gambar VI.5 (garis merah putus-putus)
FontAwesome atau *Material icon font icons*	Untuk memudahkan dalam menggati icon pada gambar VI.5
Buttons	Untuk mengetahui jenis *Buttons* yang tersedia pada gambar VI.6
Authentication pages	Halaman login seperti pada gambar VI.7
Forms	Halaman Form seperti pada gambar VI.8
elements	Untuk mengetahui jenis Badges, Notificaion, Additional Content dan lainnya pada gambar VI.9



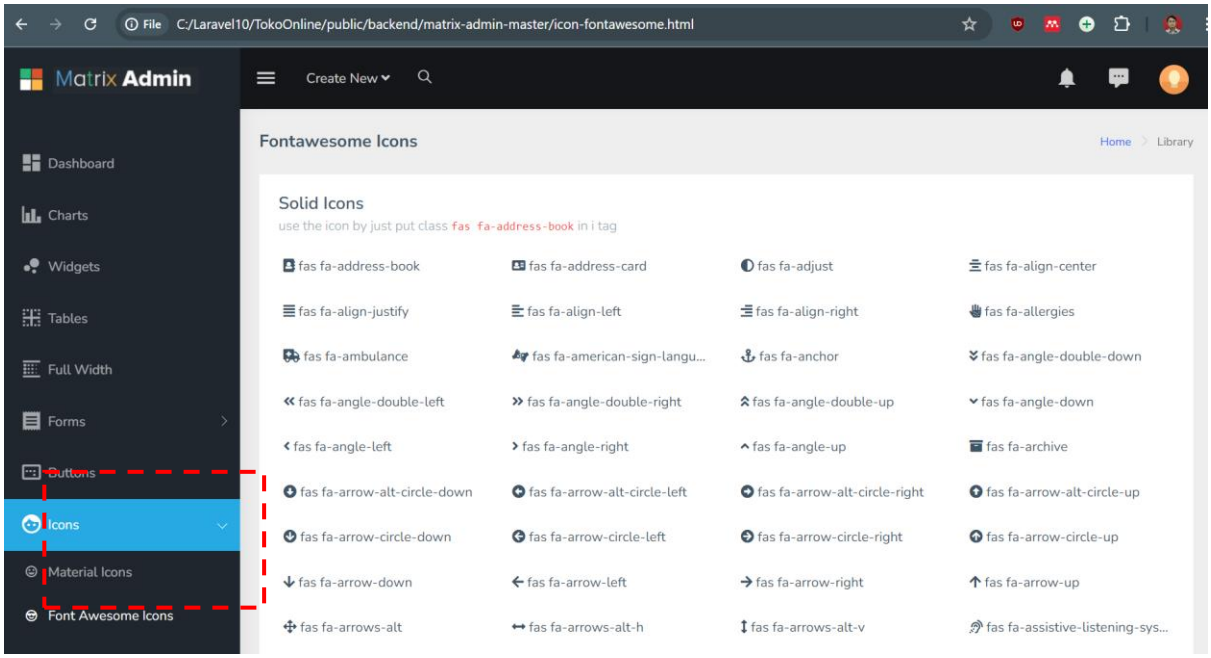
Gambar VI. 2
Fully Responsive



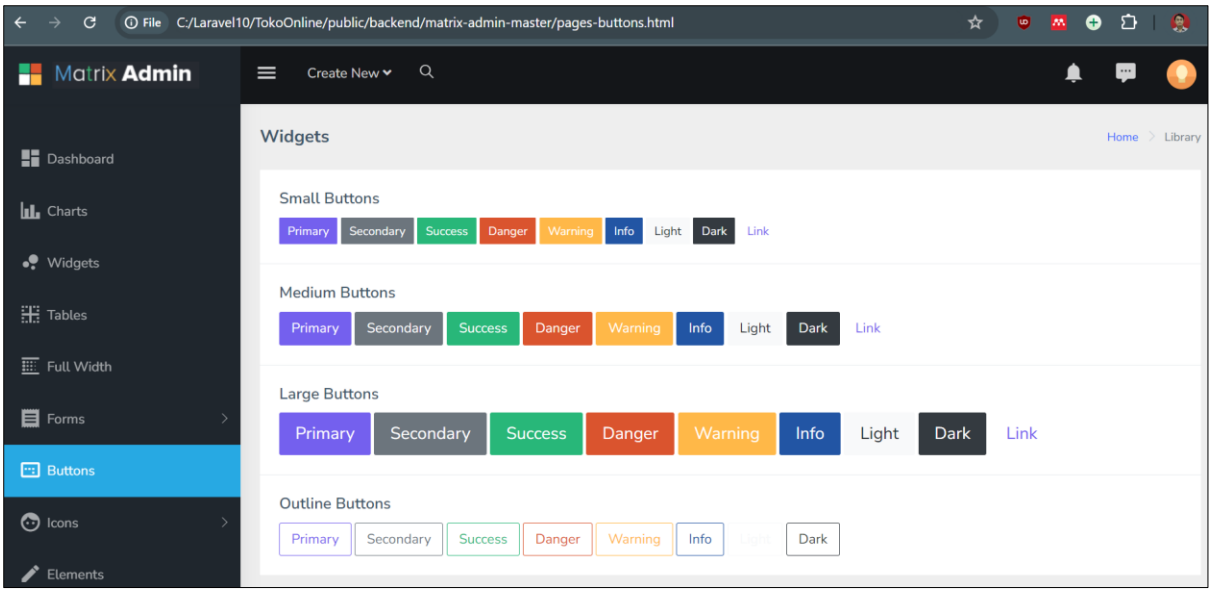
Gambar VI. 3
Basic Data Tables



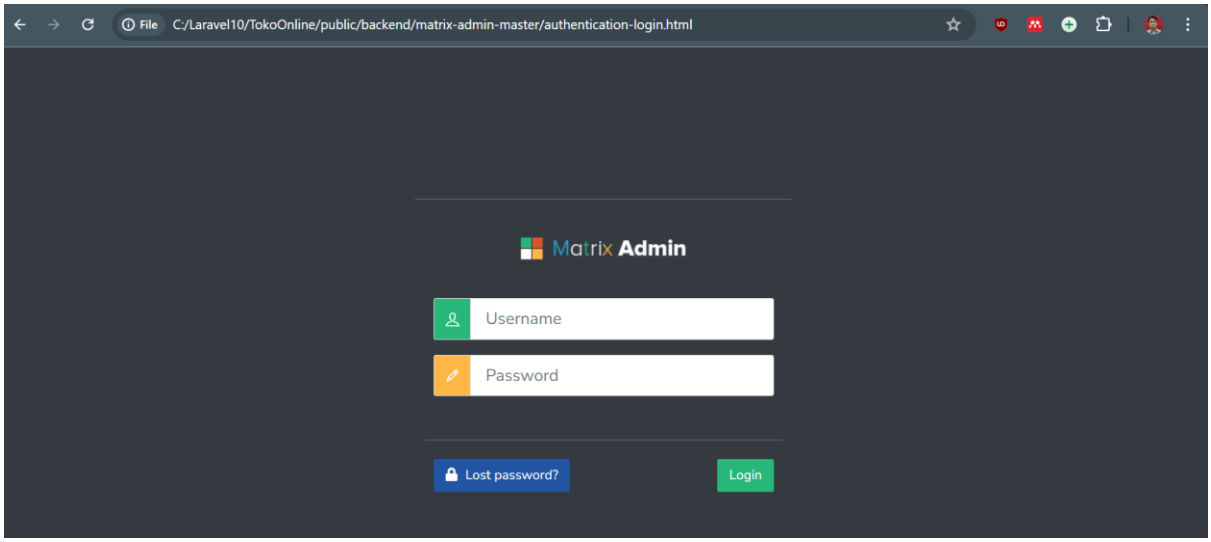
Gambar VI. 4
Charts



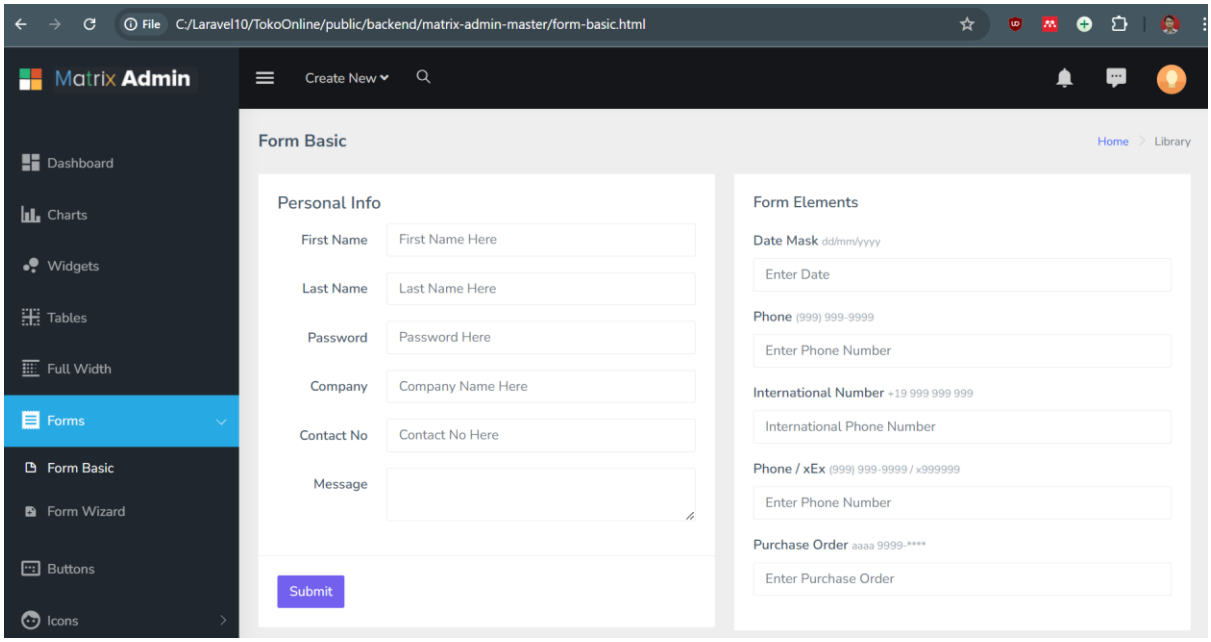
Gambar VI. 5
Icons



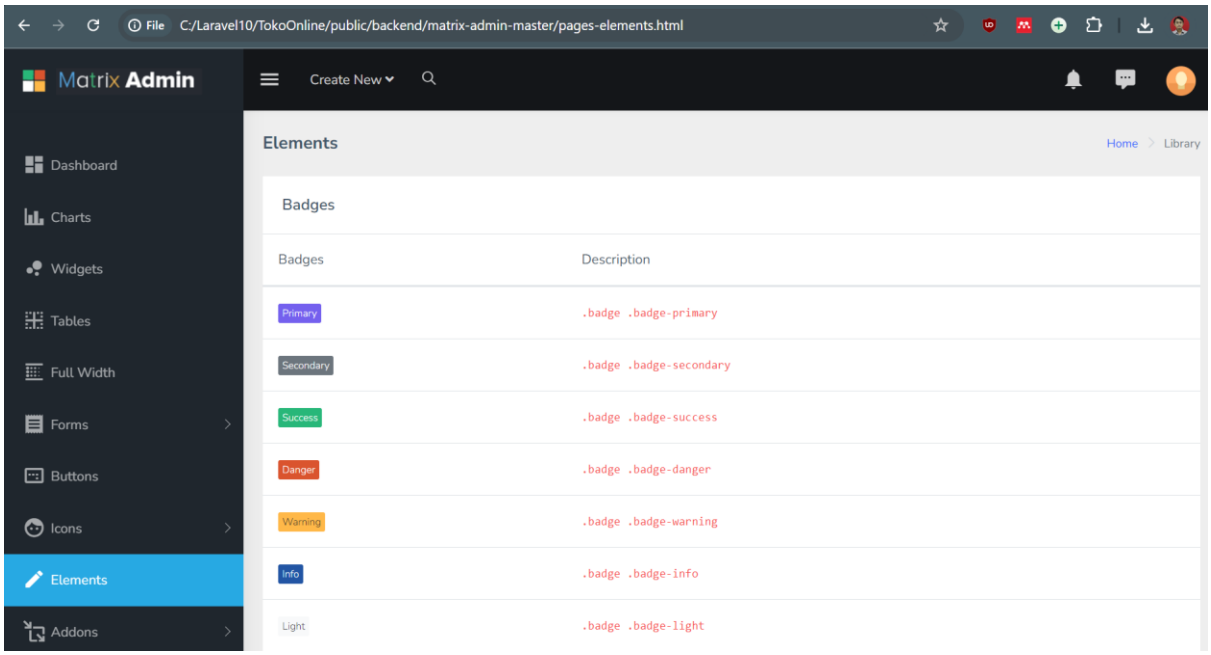
Gambar VI. 6
Buttons



Gambar VI. 7
Authentication Pages



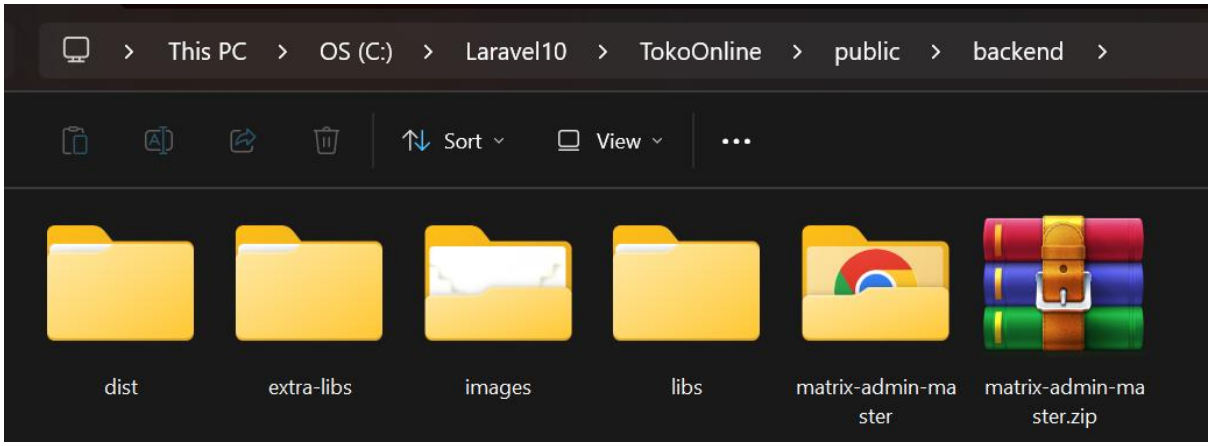
Gambar VI. 8
Forms



Gambar VI. 9
Elements

6.2. Implementasi Template Admin

1. copy folder **dist** dan isi folder **assets** maka terdapat 3 folder yakni **extra-libs, images** dan **libs** pada direktori TokoOnline\public\backend sehingga terlihat pada gambar VI.10

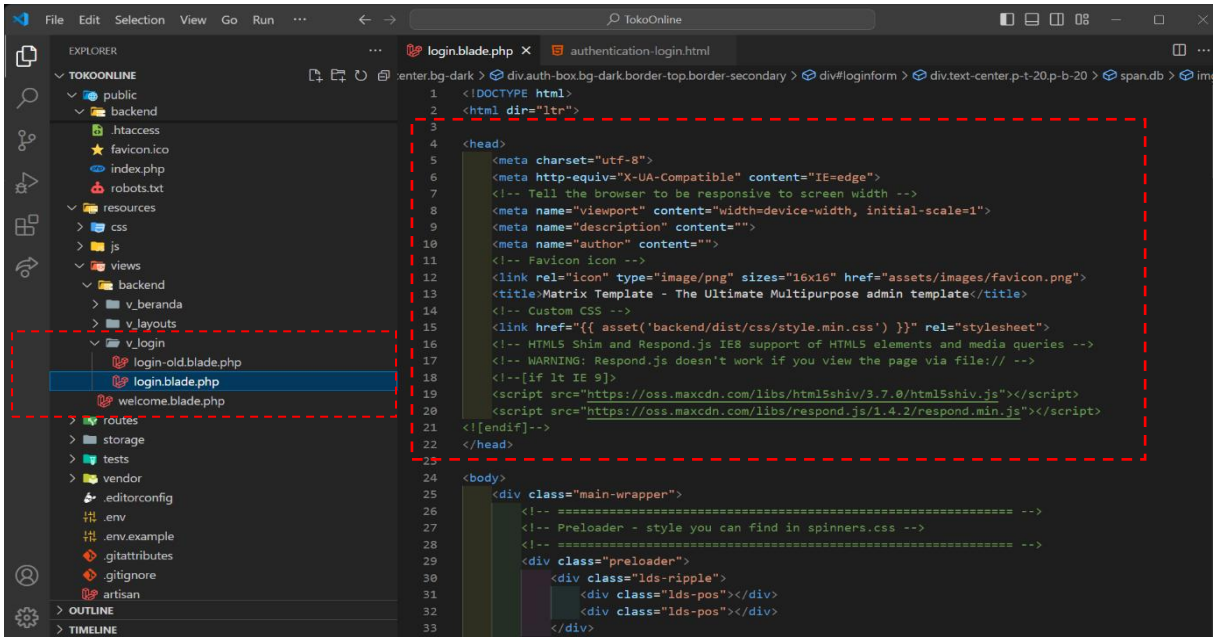


Gambar VI. 10
Komponen Template Backend - Matrix Admin

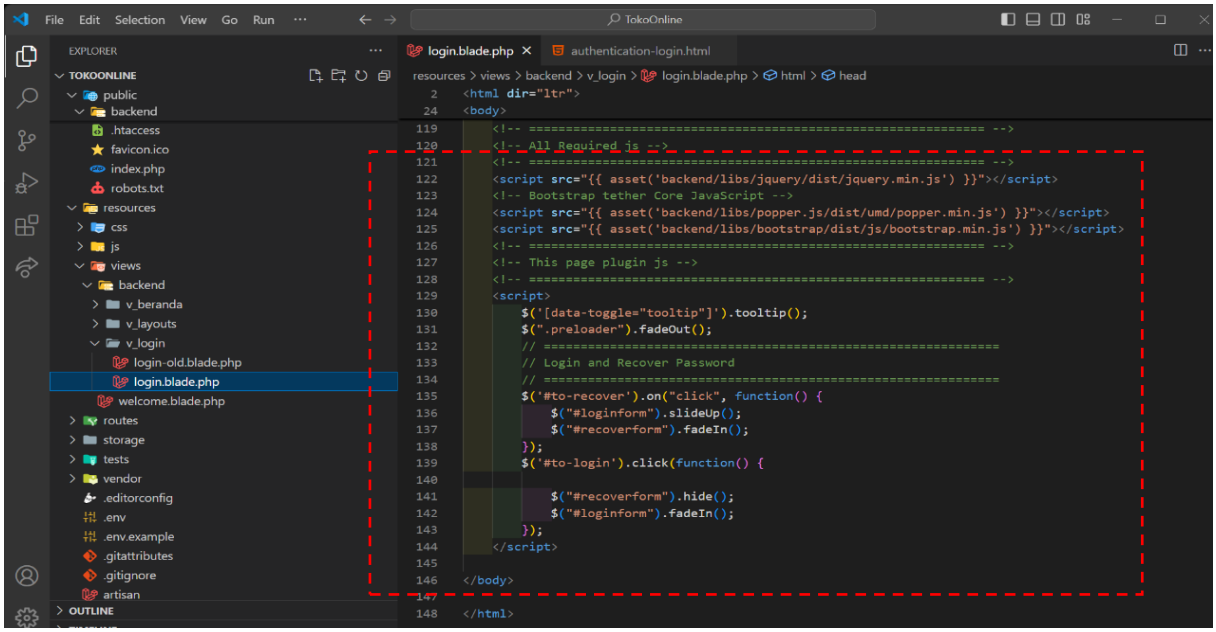
2. Menerapkan Template Login:

- Duplikat view login yang terletak pada direktori resources\views\backend\v_login\login.blade.php. Beri nama hasil duplikat tersebut **login-old.blade.php** seperti pada gambar VI.11.
- Kemudian, buka file seperti pada gambar VI.1 dengan nama file **authentication-login.html** yang terletak pada direktori TokoOnline\public\backend\matrix-admin-master menggunakan *Visual Studio Code*,
- Ganti seluruh *script* di **login.blade.php** dengan *script* dari **authentication-login.html**.

3. Ubah direktori komponen dengan menggunakan **Asset** <link href="gantidirektori" rel="stylesheet"> menjadi {{ asset('asset') }} umumnya script pada blok kode **<head></head>** seperti pada gambar VI.11, dan pada blok kode akhir sebelum **</body>** yakni **<script src=""> </script>,** seperti pada gambar VI.12.



Gambar VI. 11
Asset Pada Blok Kode<head></head>



Gambar VI. 12
Asset Pada Akhir Sebelum </body>

Berikut adalah *script* lengkap pada login.blade.php:

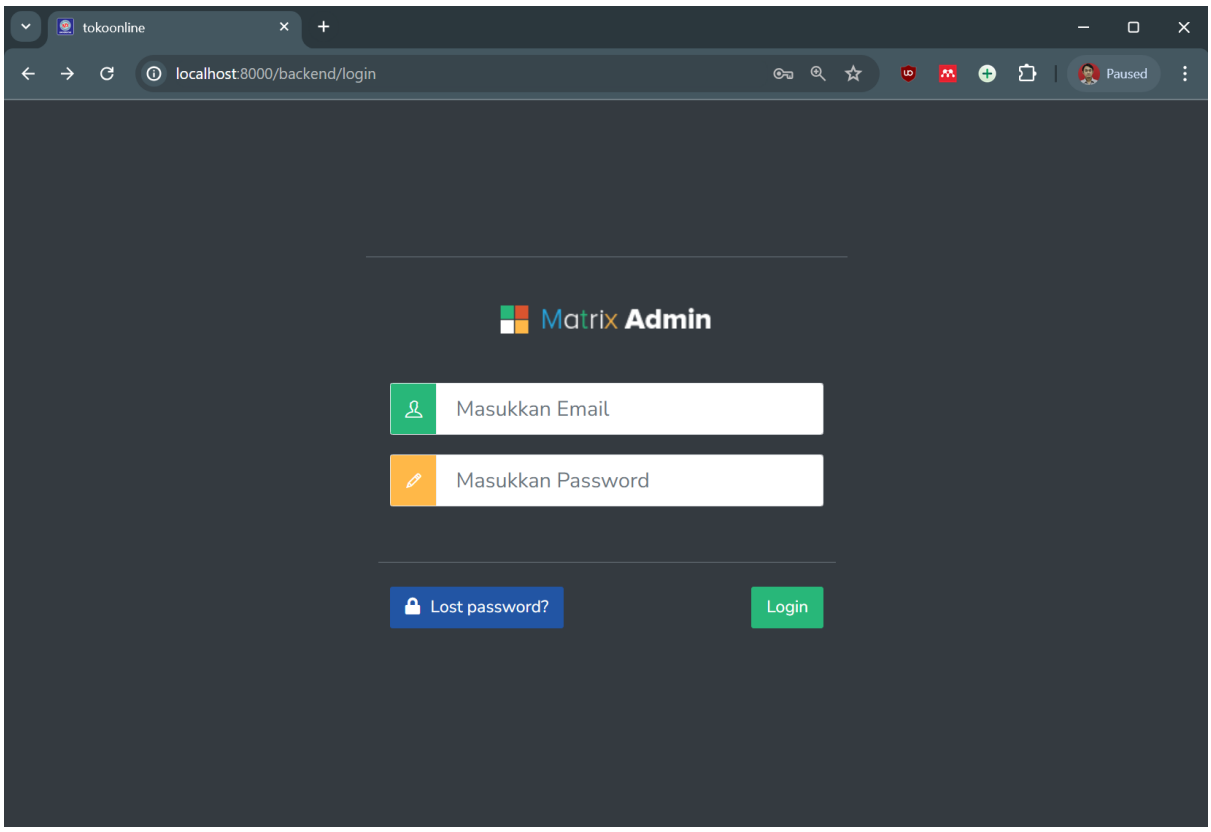
- **Favicon icon,** untuk mendapatkan **icon_univ_bsi.png** dapat didownload https://bit.ly/LaravelWebPro2 dengan direktori **TokoOnline\public\image** kemudian direktori menggunakan **aset**. dimana gambar yang digunakan **icon_univ_bsi.png**
- **Title,** diubah menjadi tokoonline
- **Logo template,** dimana gambar yang digunakan **icon_univ_bsi.png** telah menggunakan **asset**
- **Form,** sudah disesuiakan dimana penggunaan text email & password dimana kode referensi dari **login-old.blade.php**

```
<!DOCTYPE html>
<html dir="ltr">

<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <!-- Tell the browser to be responsive to screen width -->
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="">
    <meta name="author" content="">
    <!-- Favicon icon -->
    <link rel="icon" type="image/png" sizes="16x16" href="{{
asset('image/icon_univ_bsi.png') }}">
    <title>tokoonline</title>
    <!-- Custom CSS -->
    <link href="{{ asset('backend/dist/css/style.min.css') }}" rel="stylesheet">
    <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
    <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
    <script src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>
<![endif]-->
</head>

<body>
    <div class="main-wrapper">
```

```
        <!-- ============================================================== -->
        <!-- Preloader - style you can find in spinners.css -->
        <!-- ============================================================== -->
        <div class="preloader">
            <div class="lds-ripple">
                <div class="lds-pos"></div>
                <div class="lds-pos"></div>
            </div>
        </div>
        <!-- ============================================================== -->
        <!-- Preloader - style you can find in spinners.css -->
        <!-- ============================================================== -->
        <!-- ============================================================== -->
        <!-- Login box.scss -->
        <!-- ============================================================== -->
        <div class="auth-wrapper d-flex no-block justify-content-center align-items-center
bg-dark">
            <div class="auth-box bg-dark border-top border-secondary">
                <div id="loginform">
                    <div class="text-center p-t-20 p-b-20">
                        <span class="db"><img src="{{ asset('backend/images/logo.png') }}"
alt="logo" /></span>
                    </div>
                    <!-- Form -->
                    <!-- error -->
                    @if(session()->has('error'))
                    <div class="alert alert-danger alert-dismissible" role="alert">
                        <button type="button" class="close" data-dismiss="alert" aria-
label="Close"><span aria-hidden="true">&times;</span></button>
                        <strong>{{ session('error')}} </strong>
                    </div>
                    @endif
                    <!-- errorEnd -->
                    <form class="form-horizontal m-t-20" id="loginform" action="{{
route('backend.login') }}" method="post">
                        @csrf
                        <div class="row p-b-30">
                            <div class="col-12">
                                <div class="input-group mb-3">
                                    <div class="input-group-prepend">
                                        <span class="input-group-text bg-success text-
white" id="basic-addon1"><i class="ti-user"></i></span>
                                    </div>
                                    <input type="text" name="email"
value="{{old('email')}}" class="form-control form-control-lg @error('email') is-invalid
@enderror" placeholder="Masukkan Email" aria-label="Username" aria-describedby="basic-
addon1">
                                    @error('email')
                                    <span class="invalid-feedback alert-danger"
role="alert">
                                        {{$message}}
                                    </span>
                                    @enderror
                                </div>
                                <div class="input-group mb-3">
                                    <div class="input-group-prepend">
                                        <span class="input-group-text bg-warning text-
white" id="basic-addon2"><i class="ti-pencil"></i></span>
                                    </div>
                                    <input type="password" name="password" class="form-
control form-control-lg @error('password') is-invalid @enderror" placeholder="Masukkan
Password" aria-label="Password" aria-describedby="basic-addon1">
                                    @error('password')
                                    <span class="invalid-feedback alert-danger"
role="alert">
                                        {{$message}}
```

```
                                    </span>
                                    @enderror
                                </div>
                            </div>
                        </div>
                        <div class="row border-top border-secondary">
                            <div class="col-12">
                                <div class="form-group">
                                    <div class="p-t-20">
                                        <button class="btn btn-info" id="to-recover"
type="button"><i class="fa fa-lock m-r-5"></i> Lost password?</button>
                                        <button class="btn btn-success float-right"
type="submit">Login</button>
                                    </div>
                                </div>
                            </div>
                        </div>
                    </form>
                </div>
                <div id="recoverform">
                    <div class="text-center">
                        <span class="text-white">Enter your e-mail address below and we
will send you instructions how to recover a password.</span>
                    </div>
                    <div class="row m-t-20">
                        <!-- Form -->
                        <form class="col-12" action="index.html">
                            <!-- email -->
                            <div class="input-group mb-3">
                                <div class="input-group-prepend">
                                    <span class="input-group-text bg-danger text-white"
id="basic-addon1"><i class="ti-email"></i></span>
                                </div>
                                <input type="text" class="form-control form-control-lg"
placeholder="Email Address" aria-label="Username" aria-describedby="basic-addon1">
                            </div>
                            <!-- pwd -->
                            <div class="row m-t-20 p-t-20 border-top border-secondary">
                                <div class="col-12">
                                    <a class="btn btn-success" href="#" id="to-login"
name="action">Back To Login</a>
                                    <button class="btn btn-info float-right" type="button"
name="action">Recover</button>
                                </div>
                            </div>
                        </form>
                    </div>
                </div>
            </div>
        </div>
        <!-- ============================================================== -->
        <!-- Login box.scss -->
        <!-- ============================================================== -->
        <!-- ============================================================== -->
        <!-- Page wrapper scss in scafholding.scss -->
        <!-- ============================================================== -->
        <!-- ============================================================== -->
        <!-- Page wrapper scss in scafholding.scss -->
        <!-- ============================================================== -->
        <!-- ============================================================== -->
        <!-- Right Sidebar -->
        <!-- ============================================================== -->
        <!-- ============================================================== -->
        <!-- Right Sidebar -->
        <!-- ============================================================== -->
    </div>
```

```
    <!-- ============================================================== -->
    <!-- All Required js -->
    <!-- ============================================================== -->
    <script src="{{ asset('backend/libs/jquery/dist/jquery.min.js') }}"></script>
    <!-- Bootstrap tether Core JavaScript -->
    <script src="{{ asset('backend/libs/popper.js/dist/umd/popper.min.js') }}"></script>
    <script src="{{ asset('backend/libs/bootstrap/dist/js/bootstrap.min.js') }}"></script>
    <!-- ============================================================== -->
    <!-- This page plugin js -->
    <!-- ============================================================== -->
    <script>
        $('[data-toggle="tooltip"]').tooltip();
        $(".preloader").fadeOut();
        // ==============================================================
        // Login and Recover Password
        // ==============================================================
        $('#to-recover').on("click", function() {
            $("#loginform").slideUp();
            $("#recoverform").fadeIn();
        });
        $('#to-login').click(function() {

            $("#recoverform").hide();
            $("#loginform").fadeIn();
        });
    </script>

</body>

</html>
```

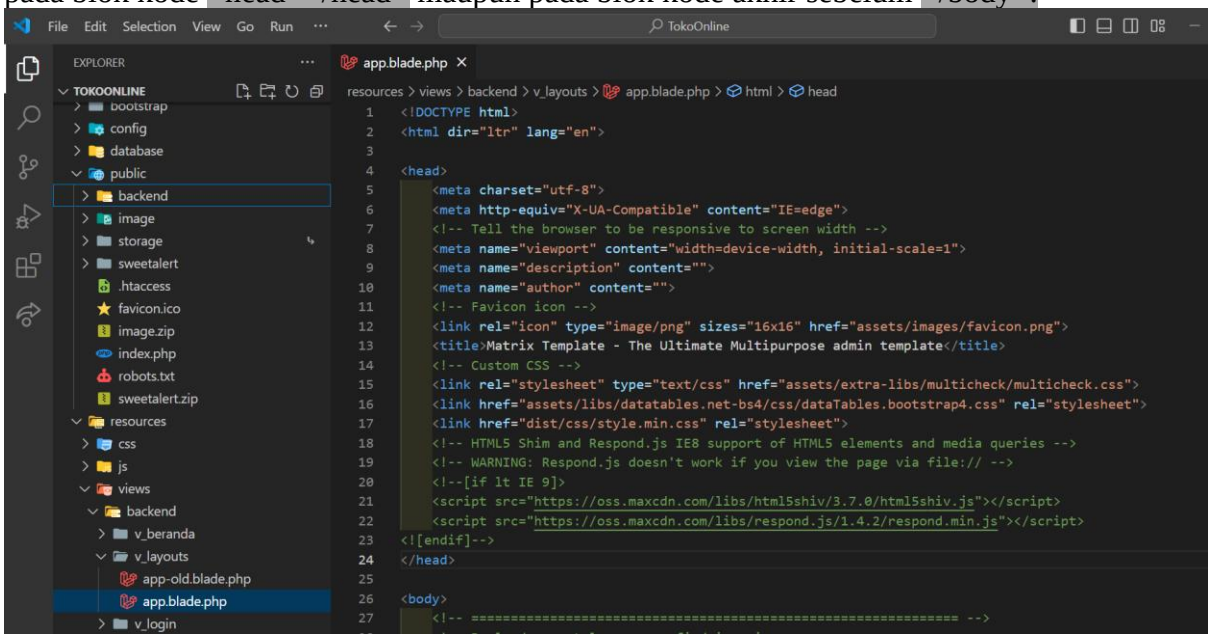


Gambar VI. 13
Login setelah Menggunakan Template

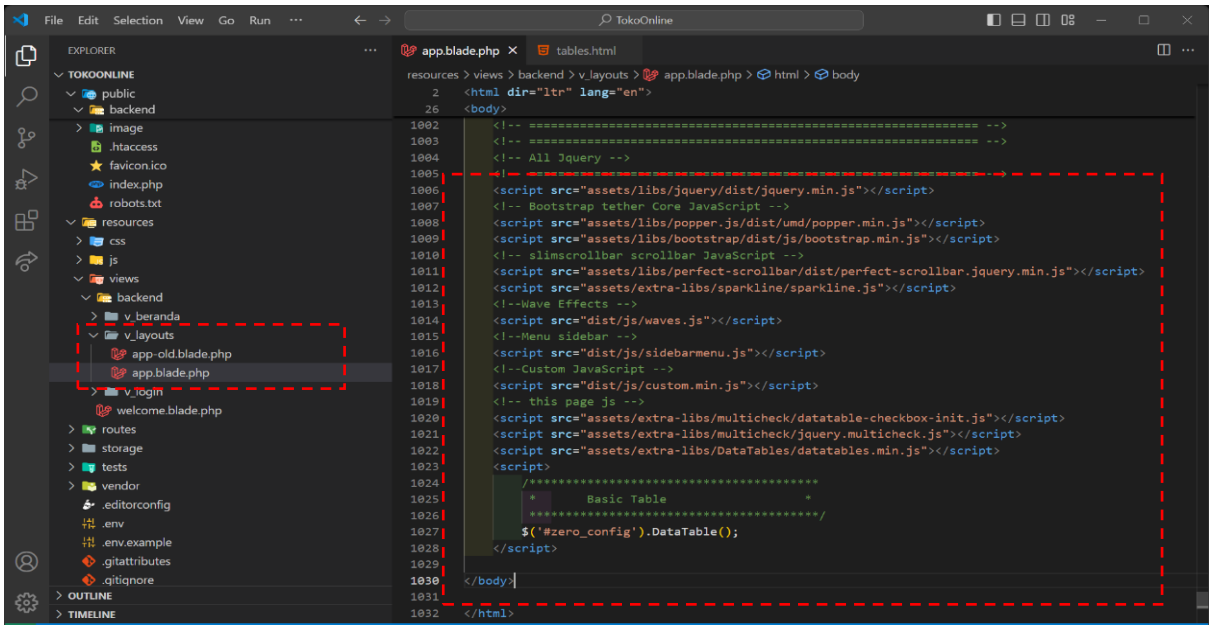
4. Menerapkan Template Menu Utama, Langkah-langkahnya sama seperti menerapkan template pada login:

- Duplikat view login yang terletak pada direktori resources\views\backend\v_layouts**app.blade.php**. Beri nama hasil duplikat tersebut **app-old.blade.php** seperti pada gambar VI.11.
- Kemudian, buka file seperti pada gambar VI.1 dengan nama file **tables.html** yang terletak pada direktori TokoOnline\public\backend\matrix-admin-master menggunakan *Visual Studio Code*
- Ganti seluruh *script* di **app.blade.php** dengan *script* dari **tables.html**. Dengan menggunakan master template dari **tables.html** dengan demikian kita akan lebih mudah dalam menggunakan *Datatable*.

5. Ubah direktori komponen dengan menggunakan **{{ asset('gantidengandirektori') }}** baik pada blok kode <head></head> maupun pada blok kode akhir sebelum </body>.



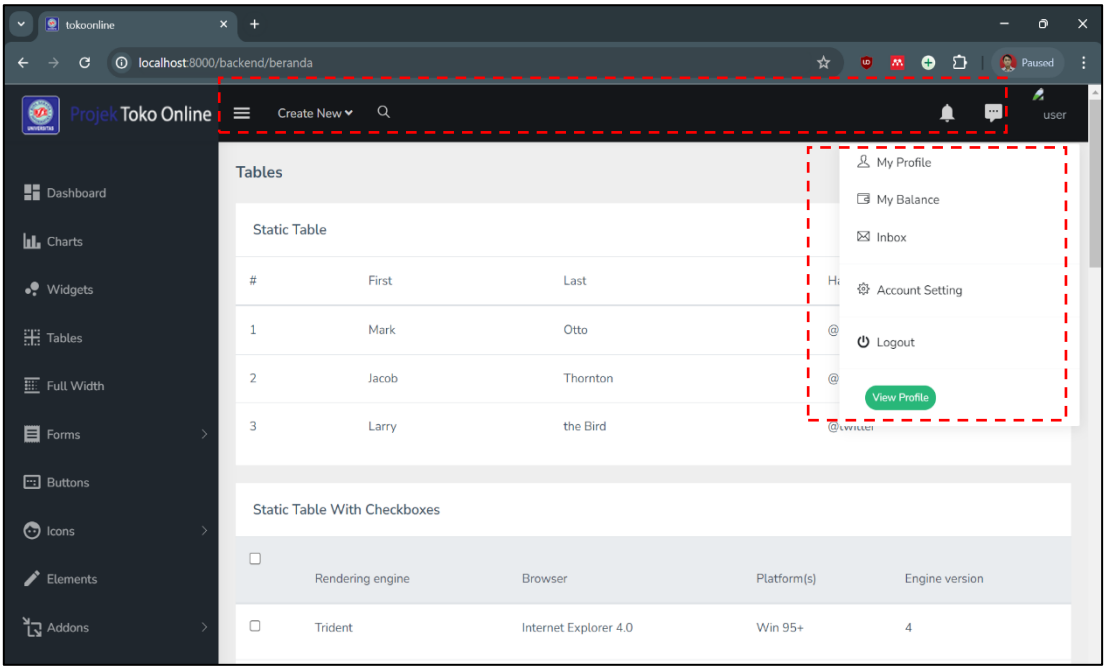
Gambar VI. 14
Asset Blok Kode <head></head> Pada Menu Utama



Gambar VI. 15
Blok Kode Akhir Sebelum </body> Pada Menu Utama

Berikut adalah *script* lengkap pada **app.blade.php** dan beberapa perubahan *script* lainnya sebagai berikut:

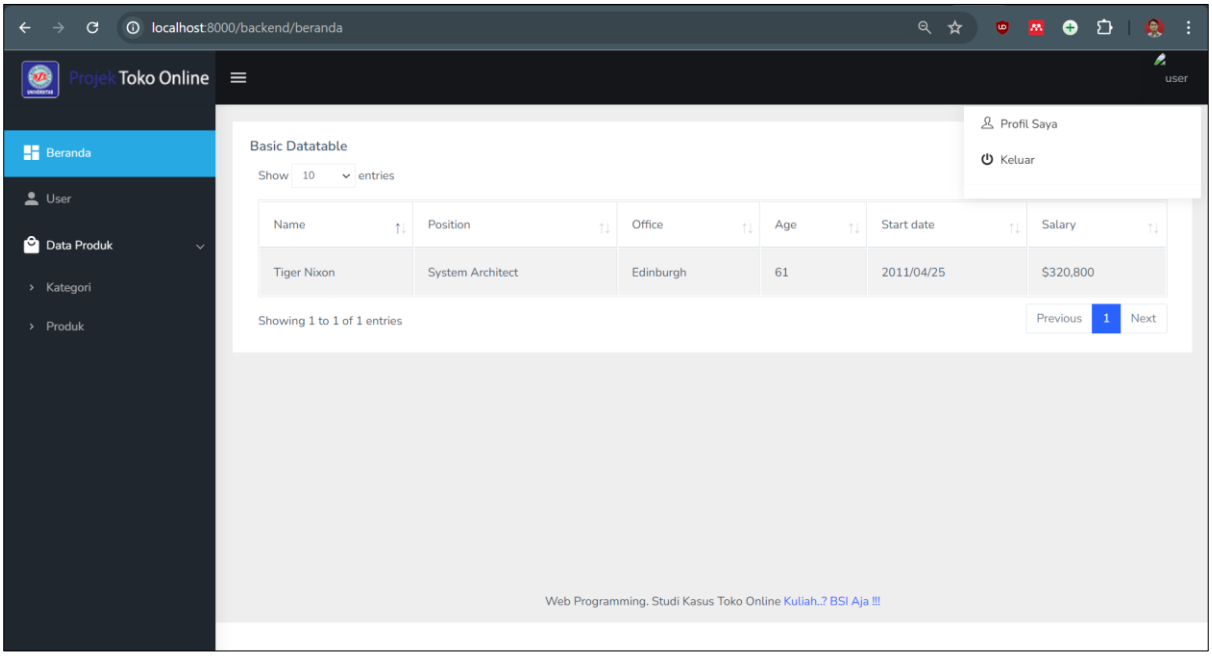
- **Favicon icon,** dimana gambar yang digunakan **icon_univ_bsi.png** telah menggunakan **asset**
- **Title,** diubah menjadi tokoonline
- **Logo template,** dimana gambar yang digunakan **icon_univ_bsi.png** & **logo_text.png** telah menggunakan **asset** hasil akan terlihat seperti pada gambar VI.16



Gambar VI. 16
Penyesuaian Pada Menu Utama

Berikutnya kembali kita sesuaikan pada **app.blade.php** hasilnya seperti gambar VI.16 dan beberapa perubahan *script* lainnya sebagai berikut:

- Hapus **Topbar header** seperti pada gambar gambar VI.6 (garis merah putus-putus)
- **Sidebar,** sudah disesuiakan dengan kebutuhan studi kasus toko online dengan refensi dari **app-old.blade.php** yakni tombol Beranda {{ route('backend.beranda') }} dan Tombol Keluar untuk mengakhiri *session()* dengan memanggil <form id="keluar-app">. Serta ada penambahan icon (referensi dari Gambar VI.5 Icons) & dropdwon.
- **Datatable,** ambil tabel yang berformat Datatable kemudian contoh record hapus sisahkan 1 record untuk master Datatable
- **Avatar,** menu yang digunakan *My Profile* dan *Logout.* & ubah *footer* sesuai dengan kesepakatan kelompok masing-masing



Gambar VI. 17
Penyesuaian Pada Topbar header, Sidebar, Datatable, Avatar & footer

Kita tambahkan **@yield('content')** pada **app.blade.php**. Cari baris *script* dengan kata kunci pada dokumentasi <!-- Start Page Content --> seperti pada gambar VI.18, sehingga saat tombol Beranda diklik, halaman akan tampil seperti pada gambar VI.19

```
<!DOCTYPE html>
<html dir="ltr" lang="en">

<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <!-- Tell the browser to be responsive to screen width -->
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="">
    <meta name="author" content="">
    <!-- Favicon icon -->
    <link rel="icon" type="image/png" sizes="16x16" href="{{
asset('image/icon_univ_bsi.png') }}">
    <title>tokoonline</title>
    <!-- Custom CSS -->
    <link rel="stylesheet" type="text/css" href="{{ asset('backend/extra-
libs/multicheck/multicheck.css') }}">
```

```
    <link href="{{ asset('backend/libs/datatables.net-bs4/css/dataTables.bootstrap4.css')
}}" rel="stylesheet">
    <link href="{{ asset('backend/dist/css/style.min.css') }}" rel="stylesheet">
    <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
    <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
    <script src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>
<![endif]-->
</head>

<body>
    <!-- ============================================================== -->
    <!-- Preloader - style you can find in spinners.css -->
    <!-- ============================================================== -->
    <div class="preloader">
        <div class="lds-ripple">
            <div class="lds-pos"></div>
            <div class="lds-pos"></div>
        </div>
    </div>
    <!-- ============================================================== -->
    <!-- Main wrapper - style you can find in pages.scss -->
    <!-- ============================================================== -->
    <div id="main-wrapper">
        <!-- ============================================================== -->
        <!-- Topbar header - style you can find in pages.scss -->
        <!-- ============================================================== -->
        <header class="topbar" data-navbarbg="skin5">
            <nav class="navbar top-navbar navbar-expand-md navbar-dark">
                <div class="navbar-header" data-logobg="skin5">
                    <!-- This is for the sidebar toggle which is visible on mobile only -->
                    <a class="nav-toggler waves-effect waves-light d-block d-md-none"
href="javascript:void(0)"><i class="ti-menu ti-close"></i></a>
                    <!-- ============================================================== -->
                    <!-- Logo -->
                    <!-- ============================================================== -->
                    <a class="navbar-brand" href="index.html">
                        <!-- Logo icon -->
                        <b class="logo-icon p-l-10">
                            <!--You can put here icon as well // <i class="wi wi-
sunset"></i> //-->
                            <!-- Dark Logo icon -->
                            <img src="{{ asset('image/icon_univ_bsi.png') }}"
alt="homepage" class="light-logo" />

                        </b>
                        <!--End Logo icon -->
                        <!-- Logo text -->
                        <span class="logo-text">
                            <!-- dark Logo text -->
                            <img src="{{ asset('image/logo_text.png') }}" alt="homepage"
class="light-logo" />

                        </span>
                        <!-- Logo icon -->
                        <!-- <b class="logo-icon"> -->
                        <!--You can put here icon as well // <i class="wi wi-sunset"></i>
//-->
                        <!-- Dark Logo icon -->
                        <!-- <img src="assets/images/logo-text.png" alt="homepage"
class="light-logo" /> -->

                        <!-- </b> -->
                        <!--End Logo icon -->
                    </a>
```

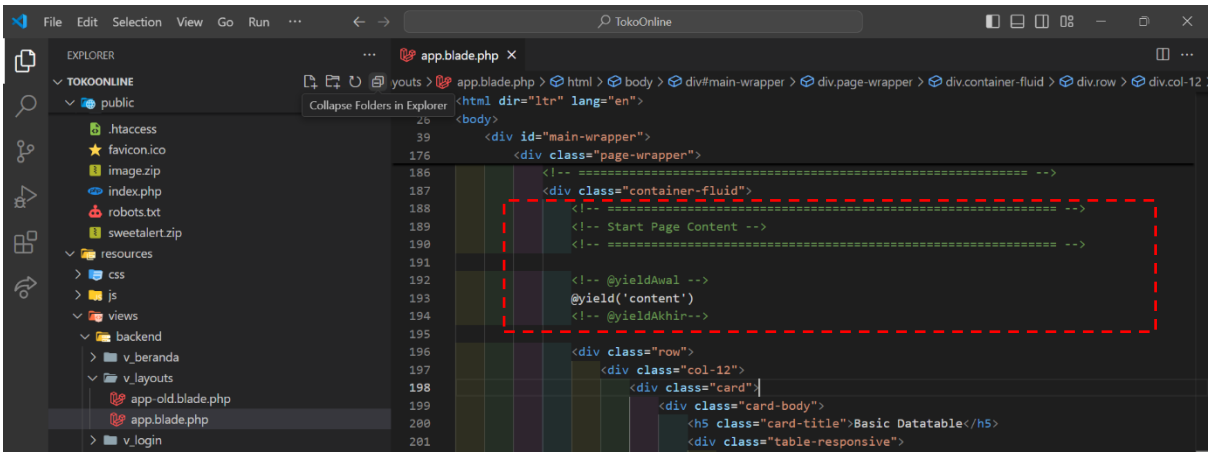
```
                    <!-- ============================================================== -->
                    <!-- End Logo -->
                    <!-- ============================================================== -->
                    <!-- ============================================================== -->
                    <!-- Toggle which is visible on mobile only -->
                    <!-- ============================================================== -->
                    <a class="topbartoggler d-block d-md-none waves-effect waves-light"
href="javascript:void(0)" data-toggle="collapse" data-target="#navbarSupportedContent"
aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation"><i class="ti-more"></i></a>
                </div>
                <!-- ============================================================== -->
                <!-- End Logo -->
                <!-- ============================================================== -->
                <div class="navbar-collapse collapse" id="navbarSupportedContent" data-
navbarbg="skin5">
                    <!-- ============================================================== -->
                    <!-- toggle and nav items -->
                    <!-- ============================================================== -->
                    <ul class="navbar-nav float-left mr-auto">
                        <li class="nav-item d-none d-md-block"><a class="nav-link
sidebartoggler waves-effect waves-light" href="javascript:void(0)" data-sidebartype="mini-
sidebar"><i class="mdi mdi-menu font-24"></i></a></li>
                        <!-- ============================================================
-->
                        <!-- create new -->
                        <!-- ============================================================
-->

                        <!-- ============================================================
-->
                        <!-- Search -->
                        <!-- ============================================================
-->

                    </ul>
                    <!-- ============================================================== -->
                    <!-- Right side toggle and nav items -->
                    <!-- ============================================================== -->
                    <ul class="navbar-nav float-right">
                        <!-- ============================================================
-->
                        <!-- Comment -->
                        <!-- ============================================================
-->

                        <!-- ============================================================
-->
                        <!-- End Comment -->
                        <!-- ============================================================
-->
                        <!-- ============================================================
-->
                        <!-- Messages -->
                        <!-- ============================================================
-->

                        <!-- ============================================================
-->
                        <!-- End Messages -->
                        <!-- ============================================================
-->

                        <!-- ============================================================
-->
                        <!-- User profile and search -->
```

```
                        <!-- ============================================================
-->
                         <li class="nav-item dropdown">
                            <a class="nav-link dropdown-toggle text-muted waves-effect
waves-dark pro-pic" href="" data-toggle="dropdown" aria-haspopup="true" aria-
expanded="false"><img src="assets/images/users/1.jpg" alt="user" class="rounded-circle"
width="31"></a>
                            <div class="dropdown-menu dropdown-menu-right user-dd
animated">
                                <a class="dropdown-item" href="javascript:void(0)"><i
class="ti-user m-r-5 m-l-5"></i> Profil Saya</a>
                                <a class="dropdown-item" href=""
onclick="event.preventDefault(); document.getElementById('keluar-app').submit();"><i
class="fa fa-power-off m-r-5 m-l-5"></i> Keluar</a>
                                <div class="dropdown-divider"></div>

                            </div>
                        </li>
                        <!-- ============================================================
-->
                        <!-- User profile and search -->
                        <!-- ============================================================
-->
                    </ul>
                </div>
            </nav>
        </header>
        <!-- ============================================================== -->
        <!-- End Topbar header -->
        <!-- ============================================================== -->
        <!-- ============================================================== -->
        <!-- Left Sidebar - style you can find in sidebar.scss  -->
        <!-- ============================================================== -->
        <aside class="left-sidebar" data-sidebarbg="skin5">
            <!-- Sidebar scroll-->
            <div class="scroll-sidebar">
                <!-- Sidebar navigation-->
                <nav class="sidebar-nav">
                    <ul id="sidebarnav" class="p-t-30">
                        <li class="sidebar-item"> <a class="sidebar-link waves-effect
waves-dark sidebar-link" href="{{ route('backend.beranda') }}" aria-expanded="false"><i
class="mdi mdi-view-dashboard"></i><span class="hide-menu">Beranda</span></a>
                        </li>
                        <li class="sidebar-item"> <a class="sidebar-link waves-effect
waves-dark sidebar-link" href="charts.html" aria-expanded="false"><i class="mdi mdi-
account"></i><span class="hide-menu">User</span></a>
                        </li>
                        <li class="sidebar-item"> <a class="sidebar-link has-arrow waves-
effect waves-dark" href="javascript:void(0)" aria-expanded="false"><i class="mdi mdi-
shopping"></i><span class="hide-menu">Data Produk </span></a>
                            <ul aria-expanded="false" class="collapse  first-level">
                                <li class="sidebar-item"><a href="icon-material.html"
class="sidebar-link"><i class="mdi mdi-chevron-right"></i><span class="hide-menu"> Kategori
</span></a>
                                </li>
                                <li class="sidebar-item"><a href="icon-fontawesome.html"
class="sidebar-link"><i class="mdi mdi-chevron-right"></i><span class="hide-menu"> Produk
</span></a>
                                </li>
                            </ul>
                        </li>
                    </ul>
                </nav>
                <!-- End Sidebar navigation -->
            </div>
            <!-- End Sidebar scroll-->
```

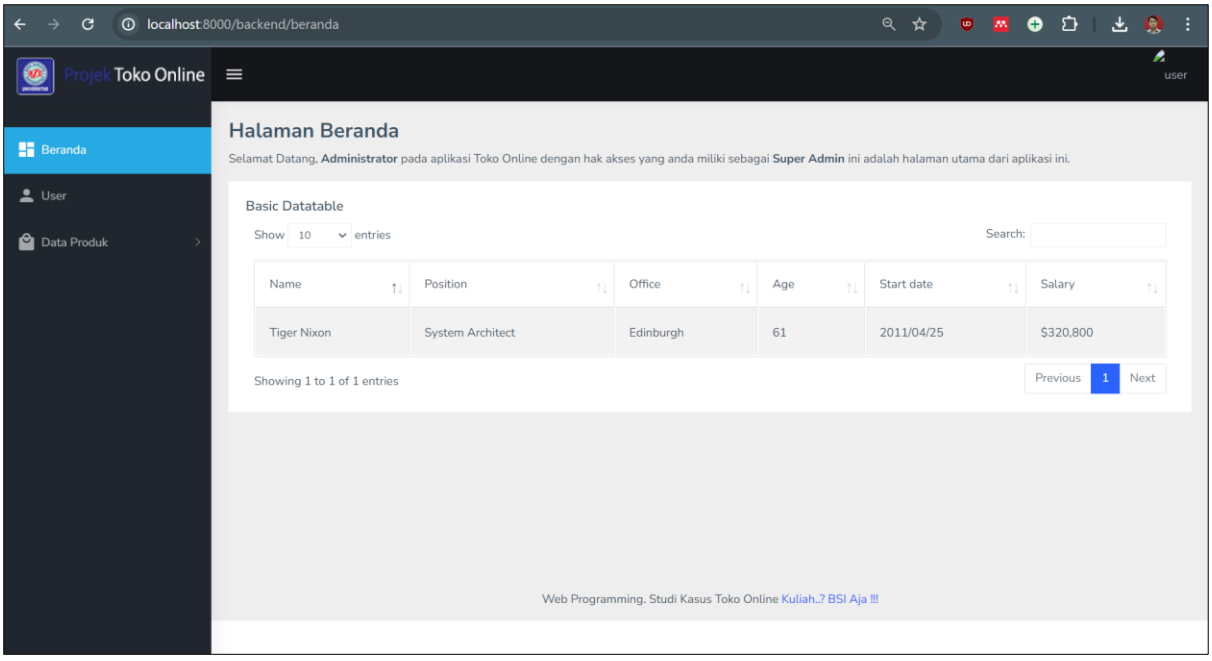
```
        </aside>
        <!-- ============================================================== -->
        <!-- End Left Sidebar - style you can find in sidebar.scss  -->
        <!-- ============================================================== -->
        <!-- ============================================================== -->
        <!-- Page wrapper  -->
        <!-- ============================================================== -->
        <div class="page-wrapper">
            <!-- ============================================================== -->
            <!-- Bread crumb and right sidebar toggle -->
            <!-- ============================================================== -->
            <div class="page-breadcrumb">
                <div class="row">
                    <div class="col-12 d-flex no-block align-items-center">
                        <h4 class="page-title">Tables</h4>
                        <div class="ml-auto text-right">
                            <nav aria-label="breadcrumb">
                                <ol class="breadcrumb">
                                    <li class="breadcrumb-item"><a href="#">Home</a></li>
                                    <li class="breadcrumb-item active" aria-
current="page">Library</li>
                                </ol>
                            </nav>
                        </div>
                    </div>
                </div>
            </div>
            <!-- ============================================================== -->
            <!-- End Bread crumb and right sidebar toggle -->
            <!-- ============================================================== -->
            <!-- ============================================================== -->
            <!-- Container fluid  -->
            <!-- ============================================================== -->
            <div class="container-fluid">
                <!-- ============================================================== -->
                <!-- Start Page Content -->
                <!-- ============================================================== -->

                <!-- @yieldAwal -->
                @yield('content')
                <!-- @yieldAkhir-->

                <div class="row">
                    <div class="col-12">
                        <div class="card">
                            <div class="card-body">
                                <h5 class="card-title">Basic Datatable</h5>
                                <div class="table-responsive">
                                    <table id="zero_config" class="table table-striped
table-bordered">
                                        <thead>
                                            <tr>
                                                <th>Name</th>
                                                <th>Position</th>
                                                <th>Office</th>
                                                <th>Age</th>
                                                <th>Start date</th>
                                                <th>Salary</th>
                                            </tr>
                                        </thead>

                                        <tbody>
                                            <tr>
                                                <td>Tiger Nixon</td>
                                                <td>System Architect</td>
```

```
                                        <td>Edinburgh</td>
                                        <td>61</td>
                                        <td>2011/04/25</td>
                                        <td>$320,800</td>
                                    </tr>
                                </tbody>
                            </table>
                        </div>

                    </div>
                </div>
            </div>
            <!-- ============================================================== -->
            <!-- End PAge Content -->
            <!-- ============================================================== -->
            <!-- ============================================================== -->
            <!-- Right sidebar -->
            <!-- ============================================================== -->
            <!-- .right-sidebar -->
            <!-- ============================================================== -->
            <!-- End Right sidebar -->
            <!-- ============================================================== -->
        </div>
        <!-- ============================================================== -->
        <!-- End Container fluid  -->
        <!-- ============================================================== -->
        <!-- ============================================================== -->
        <!-- footer -->
        <!-- ============================================================== -->
        <footer class="footer text-center">
            Web Programming. Studi Kasus Toko Online <a
href="https://bsi.ac.id/">Kuliah..? BSI Aja !!!</a>
        </footer>
        <!-- ============================================================== -->
        <!-- End footer -->
        <!-- ============================================================== -->
    </div>
    <!-- ============================================================== -->
    <!-- End Page wrapper  -->
    <!-- ============================================================== -->
</div>
<!-- ============================================================== -->
<!-- End Wrapper -->
<!-- ============================================================== -->
<!-- ============================================================== -->
<!-- All Jquery -->
<!-- ============================================================== -->
<script src="{{ asset('backend/libs/jquery/dist/jquery.min.js') }}"></script>
<!-- Bootstrap tether Core JavaScript -->
<script src="{{ asset('backend/libs/popper.js/dist/umd/popper.min.js') }}"></script>
<script src="{{ asset('backend/libs/bootstrap/dist/js/bootstrap.min.js') }}"></script>
<!-- slimscrollbar scrollbar JavaScript -->
<script src="{{ asset('backend/libs/perfect-scrollbar/dist/perfect-
scrollbar.jquery.min.js') }}"></script>
<script src="{{ asset('backend/extra-libs/sparkline/sparkline.js') }}"></script>
<!--Wave Effects -->
<script src="{{ asset('backend/dist/js/waves.js') }}"></script>
<!--Menu sidebar -->
<script src="{{ asset('backend/dist/js/sidebarmenu.js') }}"></script>
<!--Custom JavaScript -->
<script src="{{ asset('backend/dist/js/custom.min.js') }}"></script>
<!-- this page js -->
<script src="{{ asset('backend/extra-libs/multicheck/datatable-checkbox-init.js')
}}"></script>
```

```
    <script src="{{ asset('backend/extra-libs/multicheck/jquery.multicheck.js')
}}"></script>
    <script src="{{ asset('backend/extra-libs/DataTables/datatables.min.js') }}"></script>
    <script>
        /***********************************
         *        Basic Table                  *
         ***********************************/
        $('#zero_config').DataTable();
    </script>
    <!-- form keluar app -->
    <form id="keluar-app" action="{{ route('backend.logout') }}" method="POST" class="d-
none">
        @csrf
    </form>
    <!-- form keluar app end -->

</body>

</html>
```



Gambar VI. 18
Menambahakan @yield('content') pada App.blade.php



Gambar VI. 19
Konten Pada Halaman Beranda

6.2. Membuat Master DataTable

Sehingga **Beranda** dan **DataTable** tampil, namun seharusnya **DataTable** ini tidak muncul di halaman Beranda seperti pada Gambar VI.19. Selanjutnya, kita pindahkan **DataTable** sesuai dengan **view** yang menggunakan **DataTable**. Maka kita siapkan terlebih dulu data yang akan ditampilkan dengan menggunakan **DataTable**. Misalnya kita akan menampilkan data user, maka kita buat **Controller** terlebih dulu dengan nama **UserController**.

1. Buat controller dengan nama **UserController** pada terminal

```
php artisan make:controller UserController --resource
```

2. Pada *function index()* sebagai berikut:

```
    public function index()
    {
        $user = User::orderBy('updated_at', 'desc')->get();
        return view('backend.v_user.index', [
            'judul' => 'Data User',
            'index' => $user
        ]);
    }
```

3. view dengan direktori resources\views\backend\v_user\ **index.blade.php**

```
@extends('backend.v_layouts.app')
@section('content')
<!-- contentAwal -->
<h3> {{$judul}} </h3>
<a href="{{ route('backend.user.create') }}">
    <button type="button">Tambah</button>
</a>
<table border="1" width="80%">
    <tr>
        <th>No</th>
        <th>Email</th>
        <th>Nama</th>
        <th>Role</th>
        <th>Status</th>
        <th>Aksi</th>
    </tr>
    @foreach ($index as $row)
    <tr>
        <td> {{ $loop->iteration }} </td>
        <td> {{$row->nama}} </td>
        <td> {{$row->email}} </td>
        <td> {{$row->role}} </td>
        <td> {{$row->status}} </td>
        <td>
            <a href="{{ route('backend.user.edit', $row->id) }}">
                <button type="button">Ubah</button>
            </a>
            <form action="{{ route('backend.user.destroy', $row->id) }}" method="POST">
                @method('delete')
                @csrf
                <button type="submit">Hapus</button>
            </form>

        </td>
    </tr>
    @endforeach
</table>

<!-- contentAkhir -->
```

```
@endsection
</table>
```

4. pada routes**web.php** sebagai berikut:

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\BerandaController;
use App\Http\Controllers\LoginController;
use App\Http\Controllers\UserController;

Route::get('/', function () {
    // return view('welcome');
    return redirect()->route('backend.login');
});
Route::get('backend/beranda', [BerandaController::class, 'berandaBackend'])-
>name('backend.beranda')->middleware('auth');

Route::get('backend/login', [LoginController::class, 'loginBackend'])-
>name('backend.login');
Route::post('backend/login', [LoginController::class, 'authenticateBackend'])-
>name('backend.login');
Route::post('backend/logout', [LoginController::class, 'logoutBackend'])-
>name('backend.logout');

// Route::resource('backend/user', UserController::class)->middleware('auth');
Route::resource('backend/user', UserController::class, ['as' => 'backend'])-
>middleware('auth');
```

as sebagai Alias

as digunakan untuk memberikan alias atau prefix ke nama route yang dihasilkan oleh metode routing seperti **Route::resource**. Ini digunakan untuk mengelompokkan nama-nama route atau menambahkan namespace ke dalamnya.

Tanpa as:

```
Route::resource('user', UserController::class);
```

Route yang dihasilkan:

- user.index
- user.create
- user.store
- user.show
- user.edit
- user.update
- user.destroy

Dengan as:

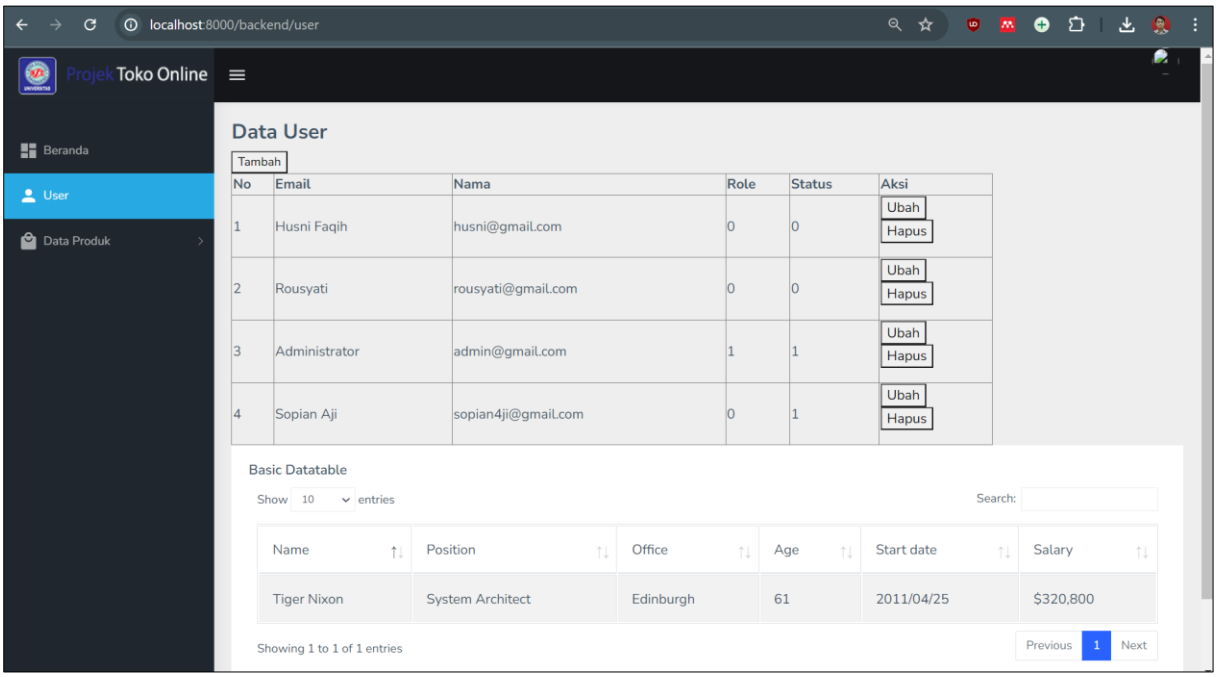
```
Route::resource('user', UserController::class, ['as' => 'backend']);
```

Route yang dihasilkan:

- backend.user.index
- backend.user.create
- backend.user.store
- backend.user.show
- backend.user.edit
- backend.user.update
- backend.user.destroy

5. Pada **sidebar** di resources\views\backend\v_layouts**app.blade.php**, kita tambahkan href="panggilRoute" pada menu User. Perubahan pada **app.blade.php** dilakukan pada bagian sidebar dengan kata kunci pada dokumentasi <!-- Sidebar navigation--> sehingga saat tombol User diklik, akan sesuai dengan href="panggilRoute" yang akan dipanggil, seperti pada gambar VI.20.

```
<div class="scroll-sidebar">
                <!-- Sidebar navigation-->
                <nav class="sidebar-nav">
                    <ul id="sidebarnav" class="p-t-30">
                        <li class="sidebar-item"> <a class="sidebar-link waves-effect
waves-dark sidebar-link" href="{{ route('backend.beranda') }}" aria-expanded="false"><i
class="mdi mdi-view-dashboard"></i><span class="hide-menu">Beranda</span></a>
                        </li>
                        <li class="sidebar-item"> <a class="sidebar-link waves-effect
waves-dark sidebar-link" href="{{ route('backend.user.index') }}" aria-expanded="false"><i
class="mdi mdi-account"></i><span class="hide-menu">User</span></a>
                        </li>
                        <li class="sidebar-item"> <a class="sidebar-link has-arrow waves-
effect waves-dark" href="javascript:void(0)" aria-expanded="false"><i class="mdi mdi-
shopping"></i><span class="hide-menu">Data Produk </span></a>
                            <ul aria-expanded="false" class="collapse  first-level">
                                <li class="sidebar-item"><a href="icon-material.html"
class="sidebar-link"><i class="mdi mdi-chevron-right"></i><span class="hide-menu"> Kategori
</span></a>
                                </li>
                                <li class="sidebar-item"><a href="icon-fontawesome.html"
class="sidebar-link"><i class="mdi mdi-chevron-right"></i><span class="hide-menu"> Produk
</span></a>
                                </li>
                            </ul>
                        </li>

                    </ul>
                </nav>
                <!-- End Sidebar navigation -->
</div>
```



Gambar VI. 20
Konten Pada Halaman User

Latihan Mandiri 6:
Portofolion sertifikasi kompetensi, Impentasikan Unit Kompetensi Software Development pada **Membuat Dokumen Kode Program**.

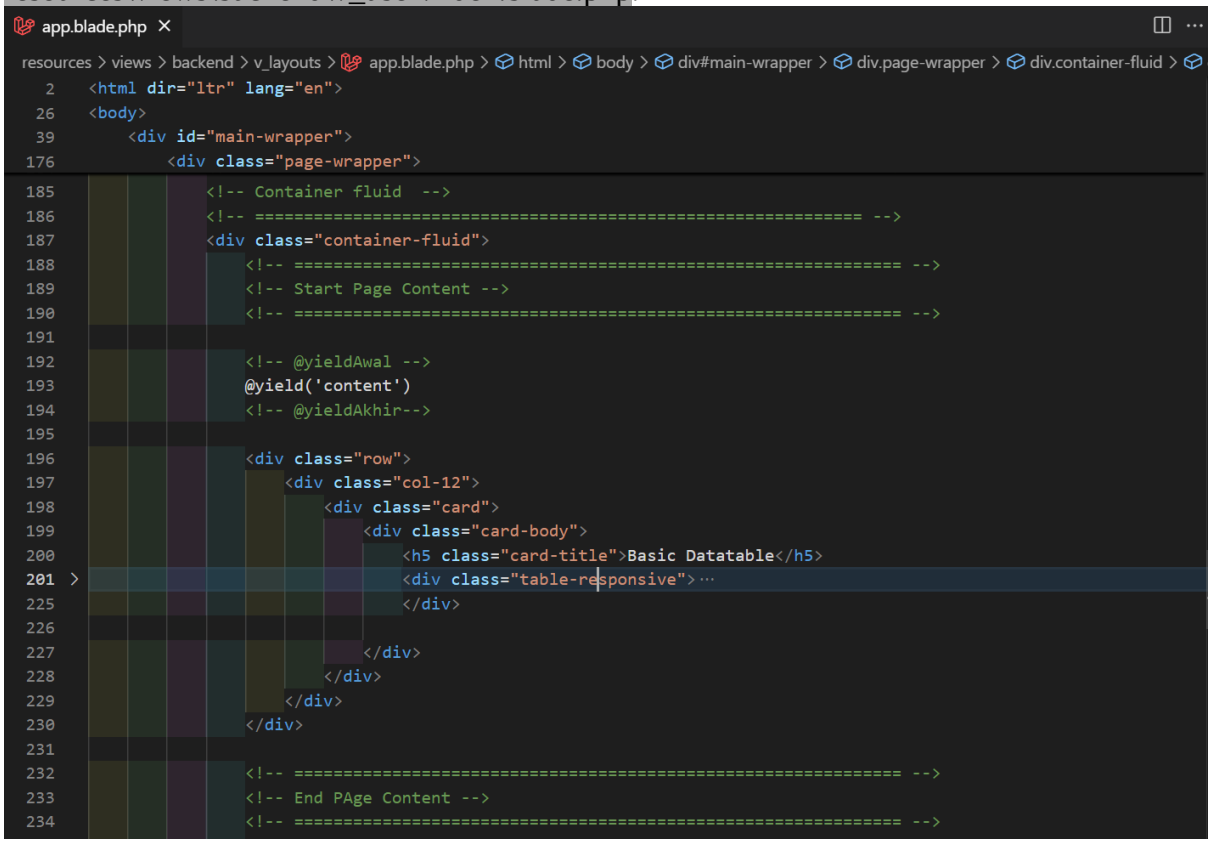
Minggu Ke-7
Implementasi DataTable dan Form dengan Template

DataTable adalah plugin jQuery yang sangat berguna untuk memanipulasi dan menampilkan data dalam bentuk tabel HTML. Plugin ini menyediakan berbagai fitur yang meningkatkan kemampuan tabel HTML biasa menjadi lebih interaktif dan mudah digunakan.
Fitur Utama DataTable

- Pencarian: Memungkinkan pengguna untuk mencari data dalam tabel secara real-time.
- Pagination: Membagi data dalam tabel menjadi beberapa halaman, memudahkan navigasi ketika data yang ditampilkan sangat banyak.
- Sortir Kolom: Memungkinkan pengguna untuk mengurutkan data berdasarkan kolom tertentu secara ascending atau descending.
- Pengaturan Tampilan Kolom: Memungkinkan pengguna untuk memilih kolom mana yang akan ditampilkan atau disembunyikan.
- AJAX: Mendukung pengambilan data secara dinamis melalui AJAX, sehingga data dapat dimuat tanpa perlu memuat ulang halaman.

7.1. Implementasi DataTable

1. Melanjutkan pertemuan sebelumnya, kita akan lakukan perubahan pada resources\views\backend\v_layouts**app.blade.php** dengan kata kunci pada dokumentasi <!-- Start Page Content --> atau lebih detail lagi dengan kata kunci pada dokumentasi **Basic Datatable** kemudian ambil baris *script* dari *line* 196 sampai dengan *line* 230 dengan cara dipotong (cut atau ctrl+x). Kemudian pindahkan kode tersebut ke views pada direktori resources\views\backend\v_user\index.blade.php.

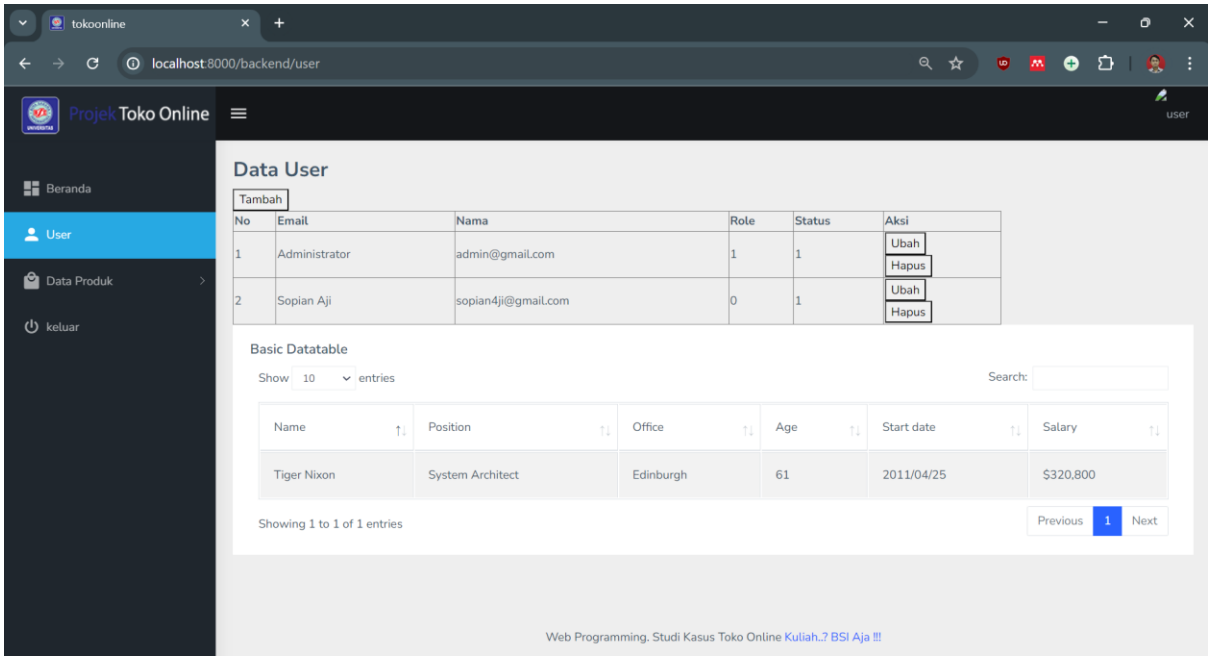


Gambar VII. 1
Implementasi @yield('content') pada app.blade.php

2. Sehingga view pada resources\views\backend\v_user\ **index.blade.php** bertumpuk dengan datatable, seperti pada gambar 7.2 sebagai berikut:

```
@extends('backend.v_layouts.app')
@section('content')
<!-- contentAwal -->
<h3> {{$judul}} </h3>
<a href="{{ route('backend.user.create') }}">
    <button type="button">Tambah</button>
</a>
<table border="1" width="80%">
    <tr>
        <th>No</th>
        <th>Email</th>
        <th>Nama</th>
        <th>Role</th>
        <th>Status</th>
        <th>Aksi</th>
    </tr>
    @foreach ($index as $row)
    <tr>
        <td> {{ $loop->iteration }} </td>
        <td> {{$row->nama}} </td>
        <td> {{$row->email}} </td>
        <td> {{$row->role}} </td>
        <td> {{$row->status}} </td>
        <td>
            <a href="{{ route('backend.user.edit', $row->id) }}">
                <button type="button">Ubah</button>
            </a>
            <form action="{{ route('backend.user.destroy', $row->id) }}" method="POST">
                @method('delete')
                @csrf
                <button type="submit">Hapus</button>
            </form>

        </td>
    </tr>
    @endforeach
</table>

<div class="row">
    <div class="col-12">
        <div class="card">
            <div class="card-body">
                <h5 class="card-title">Basic Datatable</h5>
                <div class="table-responsive">
                    <table id="zero_config" class="table table-striped table-bordered">
                        <thead>
                            <tr>
                                <th>Name</th>
                                <th>Position</th>
                                <th>Office</th>
                                <th>Age</th>
                                <th>Start date</th>
                                <th>Salary</th>
                            </tr>
                        </thead>

                        <tbody>
                            <tr>
                                <td>Tiger Nixon</td>
                                <td>System Architect</td>
                                <td>Edinburgh</td>
                                <td>61</td>
                                <td>2011/04/25</td>
```

```
                            <td>$320,800</td>
                        </tr>
                    </tbody>
                </table>
            </div>

        </div>
    </div>
</div>

<!-- contentAkhir -->
@endsection
```

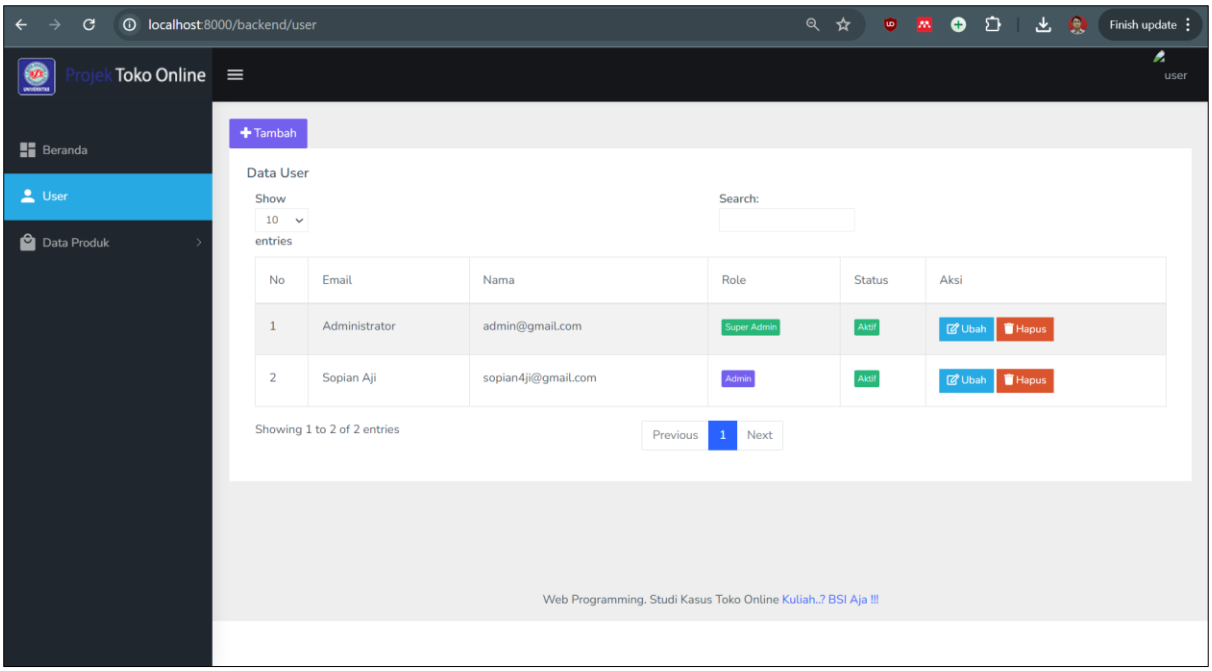


Gambar VII. 2
Konten User & DataTabel

3. Sehingga, pada view di resources\views\backend\v_user\index.blade.php, tabel yang digunakan hanya satu yaitu menggunakan DataTable. Berikut adalah *script* lengkap pada **index.blade.php**, yang sudah ditambahkan aksi Tambah, Ubah, dan Hapus (referensi dari Gambar VI.6 Buttons) pada DataTable. Kami ini **index.blade.php** seperti pada gambar VII.3

```
@extends('backend.v_layouts.app')
@section('content')
<!-- contentAwal -->

<div class="row">

    <div class="col-12">
        <a href="{{ route('backend.user.create') }}">
            <button type="button" class="btn btn-primary"><i class="fas fa-plus"></i>
Tambah</button>
        </a>
        <div class="card">
            <div class="card-body">
                <h5 class="card-title"> {{$judul}} </h5>
                <div class="table-responsive">
                    <table id="zero_config" class="table table-striped table-bordered">
                        <thead>
                            <tr>
                                <th>No</th>
```

```
                                <th>Email</th>
                                <th>Nama</th>
                                <th>Role</th>
                                <th>Status</th>
                                <th>Aksi</th>
                            </tr>
                        </thead>

                        <tbody>
                            @foreach ($index as $row)
                            <tr>
                                <td> {{ $loop->iteration }} </td>
                                <td> {{$row->nama}} </td>
                                <td> {{$row->email}} </td>
                                <td>
                                    @if ($row->role == 1)
                                    <span class="badge badge-success"></i>
                                        Super Admin</span>
                                    @elseif($row->role == 0)
                                    <span class="badge badge-primary"></i>
                                        Admin</span>
                                    @endif
                                </td>
                                <td>
                                    @if ($row->status ==1)
                                    <span class="badge badge-success"></i>
                                        Aktif</span>
                                    @elseif($row->status ==0)
                                    <span class="badge badge-secondary"></i>
                                        NonAktif</span>
                                    @endif
                                </td>
                                <td>
                                    <a href="{{ route('backend.user.edit', $row->id) }}"
title="Ubah Data">
                                        <button type="button" class="btn btn-cyan btn-
sm"><i class="far fa-edit"></i> Ubah</button>
                                    </a>

                                    <form method="POST" action="{{
route('backend.user.destroy', $row->id) }}" style="display: inline-block;">
                                        @method('delete')
                                        @csrf
                                        <button type="submit" class="btn btn-danger btn-sm"
title='Hapus Data'>
                                            <i class="fas fa-trash"></i> Hapus</button>
                                    </form>

                                </td>
                            </tr>
                            @endforeach
                        </tbody>
                    </table>
                </div>

            </div>
        </div>
    </div>
</div>

<!-- contentAkhir -->
@endsection
```

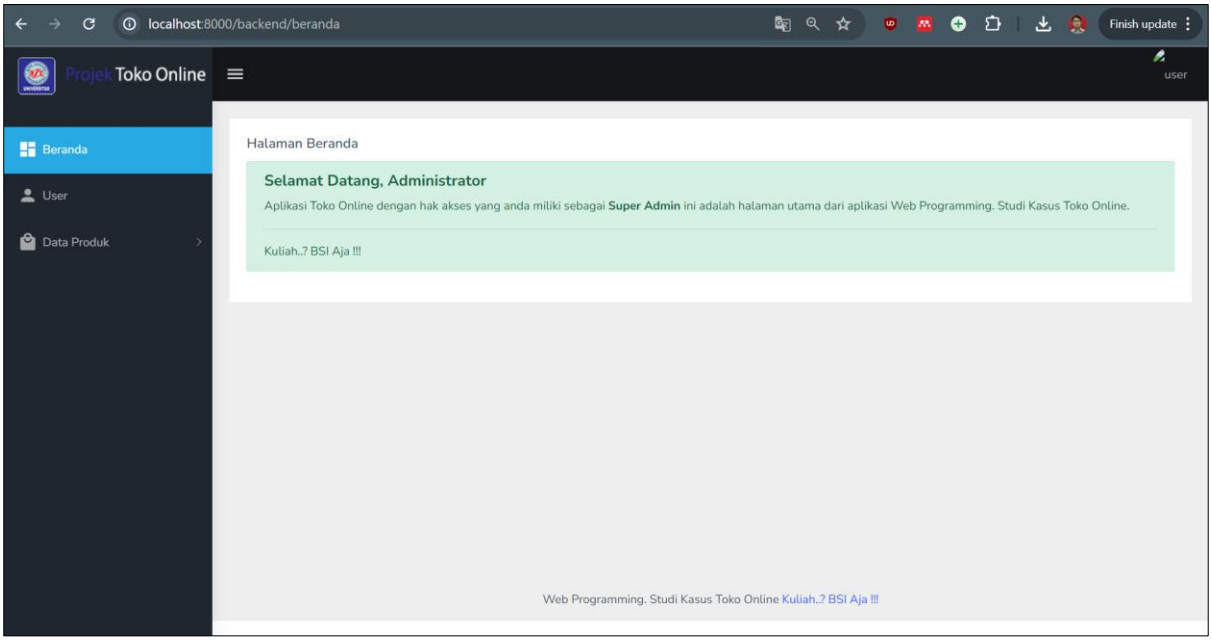


Gambar VII. 3
Indeks User menggunakan DataTabel

7.2. Menerapkan Template pada Halaman Beranda

Demikian juga pada **index**() BerandaController kita gunakan ***Elements Additional Content-alert-success*** seperti pada gambar Gambar VI.9 dengan direktori resources\views\backend\v_beranda**index.blade.php** berikut *script* lengkap & berikut perubahan seperti pada gambar VII.4

```
@extends('backend.v_layouts.app')
@section('content')
<!-- contentAwal -->

<div class="row">
    <div class="col-12">
        <div class="card">
            <div class="card-body border-top">
                <h5 class="card-title"> {{$judul}}</h5>
                <div class="alert alert-success" role="alert">
                    <h4 class="alert-heading"> Selamat Datang, {{ Auth::user()->nama
}}</h4>
                    Aplikasi Toko Online dengan hak akses yang anda miliki sebagai
                    <b>
                        @if (Auth::user()->role ==1)
                        Super Admin
                        @elseif(Auth::user()->role ==0)
                        Admin
                        @endif
                    </b>
                    ini adalah halaman utama dari aplikasi Web Programming. Studi Kasus
Toko Online.
                    <hr>
                    <p class="mb-0">Kuliah..? BSI Aja !!!</p>
                </div>
            </div>
        </div>
    </div>
</div>

<!-- contentAkhir -->
@endsection
```



Gambar VII. 4
Konten Beranda dengan Elements Additional Content- Alert-Success

7.3. Membuat Form Create Tanpa Template

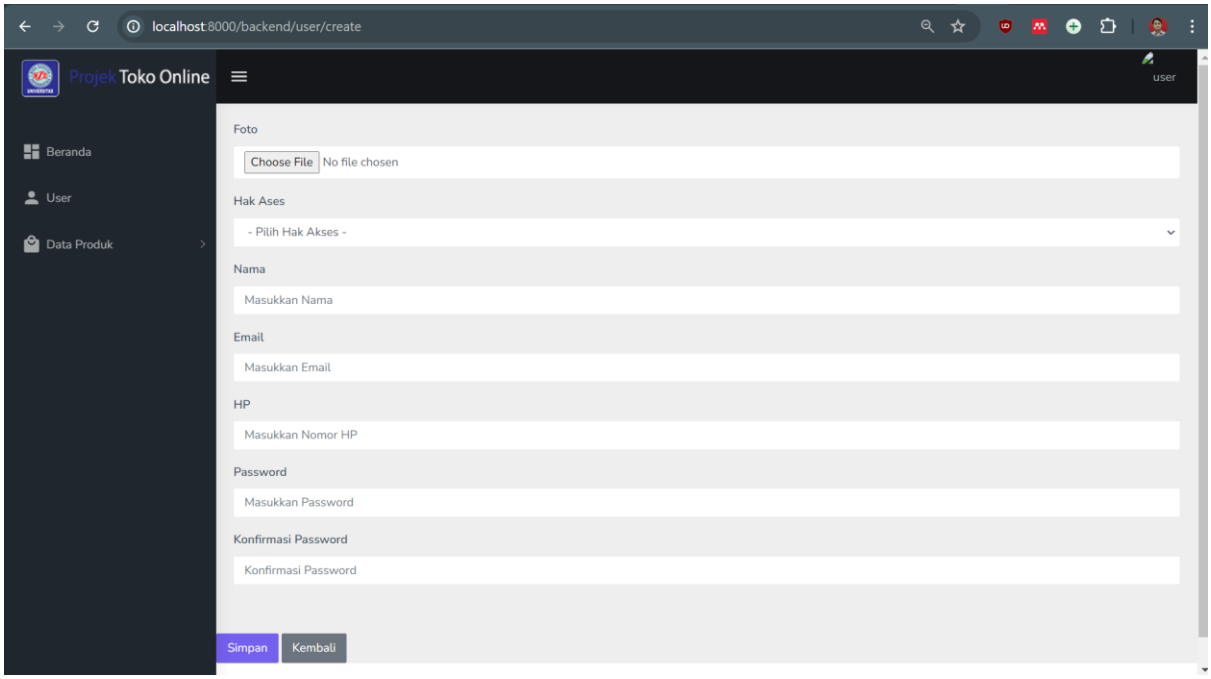
1. Menerapkan ***Template Form***, masih dengan **UserController** dengan *function create()*

```
    public function create()
    {
        return view('backend.v_user.create', [
            'judul' => 'Tambah User',
        ]);
    }
```

2. sehingga kita tambahkan **create.blade.php** pada direktori resources\views\backend\v_user berikut *script* tanpa *Template Form* dari **create.blade.php**

```
@extends('backend.v_layouts.app')
@section('content')
<!-- contentAwal -->

<form action="{{ route('backend.user.store') }}" method="post" enctype="multipart/form-
data">
    @csrf

    <label>Foto</label>
    <img class="foto-preview">
    <input type="file" name="foto" class="form-control @error('foto') is-invalid @enderror"
onchange="previewFoto()">
    @error('foto')
    <div class="invalid-feedback alert-danger">{{ $message }}</div>
    @enderror
    <p></p>

    <label>Hak Ases</label>
    <select name="role" class="form-control @error('role') is-invalid @enderror">
        <option value="" {{ old('role') == '' ? 'selected' : '' }}> - Pilih Hak Akses
            -
        </option>
        <option value="1" {{ old('role') == '1' ? 'selected' : '' }}> Super Admin</option>
        <option value="0" {{ old('role') == '0' ? 'selected' : '' }}> Admin
        </option>
```

```
    </select>
    @error('role')
    <span class="invalid-feedback alert-danger" role="alert">
        {{ $message }}
    </span>
    @enderror
    <p></p>

    <label>Nama</label>
    <input type="text" name="nama" value="{{ old('nama') }}" class="form-control
@error('nama') is-invalid @enderror" placeholder="Masukkan Nama">
    @error('nama')
    <span class="invalid-feedback alert-danger" role="alert">
        {{ $message }}
    </span>
    @enderror
    <p></p>

    <label>Email</label>
    <input type="text" name="email" value="{{ old('email') }}" class="form-control
@error('email') is-invalid @enderror" placeholder="Masukkan Email">
    @error('email')
    <span class="invalid-feedback alert-danger" role="alert">
        {{ $message }}
    </span>
    @enderror
    <p></p>

    <label>HP</label>
    <input type="text" onkeypress="return hanyaAngka(event)" name="hp" value="{{ old('hp')
}}" class="form-control @error('hp') is-invalid @enderror" placeholder="Masukkan Nomor HP">
    @error('hp')
    <span class="invalid-feedback alert-danger" role="alert">
        {{ $message }}
    </span>
    @enderror
    <p></p>

    <label>Password</label>
    <input type="password" name="password" class="form-control @error('password') is-
invalid @enderror" placeholder="Masukkan Password">
    @error('password')
    <span class="invalid-feedback alert-danger" role="alert">
        {{ $message }}
    </span>
    @enderror
    <p></p>

    <label>Konfirmasi Password</label>
    <input type="password" name="password_confirmation" class="form-control"
placeholder="Konfirmasi Password">
    </div>
    <p></p>

    <br>
    <button type="submit" class="btn btn-primary">Simpan</button>
    <a href="{{ route('backend.user.index') }}">
        <button type="button" class="btn btn-secondary">Kembali</button>
    </a>
</form>


<!-- contentAkhir -->
@endsection
```



Gambar VII. 5
Form Create User

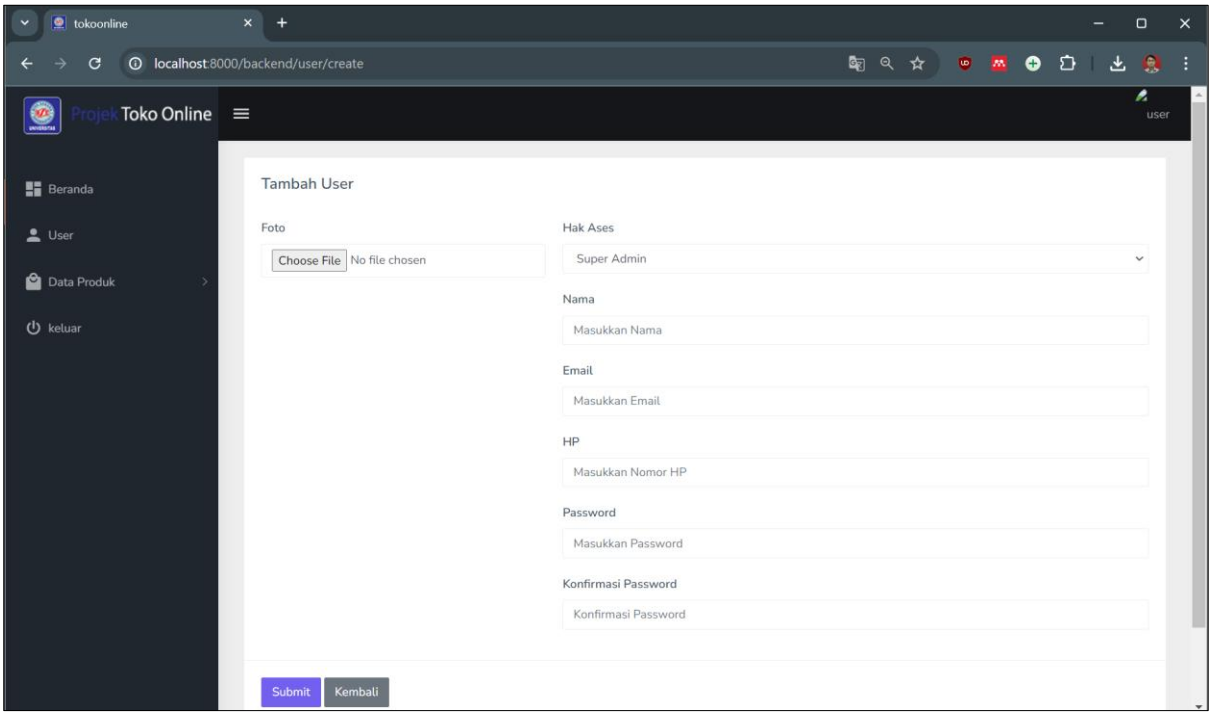
7.4. Menerapkan Form Create Dengan Template

Pada direktori resources\views\backend\v_user**create.blade.php** kita ganti kali ini kita gunakan **Template Form** dengan menggunakan **form-basic.html** seperti pada gambar VI.8, berikut *script* lengkap **create.blade.php** setelah menggunakan **Template Form**

```
@Extends('Backend.V_Layouts.App')
@section('content')
<!-- contentAwal -->

<div class="container-fluid">
    <div class="row">
        <div class="col-12">
            <div class="card">
                <form class="form-horizontal" action="{{ route('backend.user.store') }}"
method="post" enctype="multipart/form-data">
                    @csrf

                    <div class="card-body">
                        <h4 class="card-title"> {{$judul}} </h4>
                        <div class="row">
                            <div class="col-md-4">
                                <div class="form-group">
                                    <label>Foto</label>
                                    <img class="foto-preview">
                                    <input type="file" name="foto" class="form-control
@error('foto') is-invalid @enderror" onchange="previewFoto()">
                                    @error('foto')
                                    <div class="invalid-feedback alert-danger">{{ $message
}}</div>
                                    @enderror
                                </div>
                            </div>
                            <div class="col-md-8">
                                <div class="form-group">
                                    <label>Hak Ases</label>
                                    <select name="role" class="form-control @error('role')
is-invalid @enderror">
```

```
                                        <option value="" {{ old('role') == '' ? 'selected'
: '' }}> - Pilih Hak Akses
                                            -
                                        </option>
                                        <option value="1" {{ old('role') == '1' ?
'selected' : '' }}> Super Admin</option>
                                        <option value="0" {{ old('role') == '0' ?
'selected' : '' }}> Admin
                                        </option>
                                    </select>
                                    @error('role')
                                    <span class="invalid-feedback alert-danger"
role="alert">
                                        {{ $message }}
                                    </span>
                                    @enderror
                                </div>
                                <div class="form-group">
                                    <label>Nama</label>
                                    <input type="text" name="nama" value="{{ old('nama')
}}" class="form-control @error('nama') is-invalid @enderror" placeholder="Masukkan Nama">
                                    @error('nama')
                                    <span class="invalid-feedback alert-danger"
role="alert">
                                        {{ $message }}
                                    </span>
                                    @enderror
                                </div>
                                <div class="form-group">
                                    <label>Email</label>
                                    <input type="text" name="email" value="{{ old('email')
}}" class="form-control @error('email') is-invalid @enderror" placeholder="Masukkan Email">
                                    @error('email')
                                    <span class="invalid-feedback alert-danger"
role="alert">
                                        {{ $message }}
                                    </span>
                                    @enderror
                                </div>

                                <div class="form-group">
                                    <label>HP</label>
                                    <input type="text" onkeypress="return
hanyaAngka(event)" name="hp" value="{{ old('hp') }}" class="form-control @error('hp') is-
invalid @enderror" placeholder="Masukkan Nomor HP">
                                    @error('hp')
                                    <span class="invalid-feedback alert-danger"
role="alert">
                                        {{ $message }}
                                    </span>
                                    @enderror
                                </div>

                                <div class="form-group">
                                    <label>Password</label>
                                    <input type="password" name="password" class="form-
control @error('password') is-invalid @enderror" placeholder="Masukkan Password">
                                    @error('password')
                                    <span class="invalid-feedback alert-danger"
role="alert">
                                        {{ $message }}
                                    </span>
                                    @enderror
                                </div>

                                <div class="form-group">
```

```
                                <label>Konfirmasi Password</label>
                                <input type="password" name="password_confirmation"
class="form-control" placeholder="Konfirmasi Password">
                            </div>
                        </div>
                    </div>
                </div>
                <div class="border-top">
                    <div class="card-body">
                        <button type="button" class="btn btn-primary">Simpan</button>
                        <a href="{{ route('backend.user.index') }}">
                            <button type="button" class="btn btn-
secondary">Kembali</button>
                        </a>
                    </div>
                </div>
            </form>
        </div>
    </div>
</div>

<!-- contentAkhir -->
@endsection
```



Gambar VII. 6
Form Template

7.5. Membuat ImageHelper

1. Sebelum kita melanjutkan ke *function store()* pada **UserController**, form user mengirim Image maka diperlukan **ImageHelper. ImageHelper** digunakan untuk memanipulasi gambar, termasuk mengubah ukuran, memotong, memutar, dan mengubah format gambar
2. Membuat folder **Helper** dalam folder **app**, pada terminal

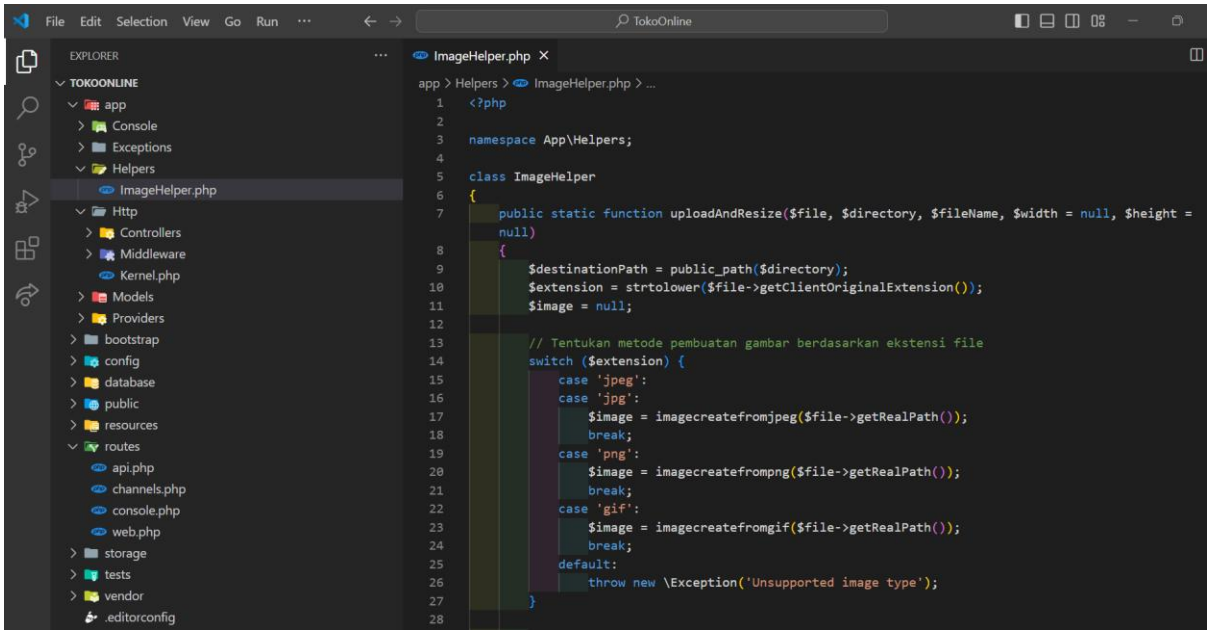
```
mkdir app/Helpers
```

3. Membuat file **ImageHelper.php** pada direktori app\Herpers, pada terminal

```
touch app/Helpers/ImageHelper.php
```

4. Berikut *script* lengkap ImageHelper.php seperti pada gambar VII.7

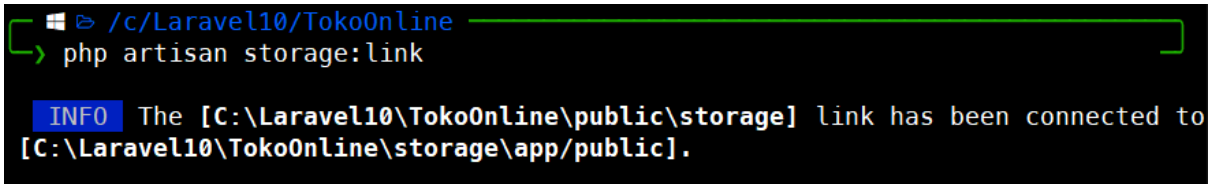
```php
<?php
namespace App\Helpers;

class ImageHelper
{
    public static function uploadAndResize($file, $directory, $fileName, $width = null,
$height = null)
    {
        $destinationPath = public_path($directory);
        $extension = strtolower($file->getClientOriginalExtension());
        $image = null;

        // Tentukan metode pembuatan gambar berdasarkan ekstensi file
        switch ($extension) {
            case 'jpeg':
            case 'jpg':
                $image = imagecreatefromjpeg($file->getRealPath());
                break;
            case 'png':
                $image = imagecreatefrompng($file->getRealPath());
                break;
            case 'gif':
                $image = imagecreatefromgif($file->getRealPath());
                break;
            default:
                throw new \Exception('Unsupported image type');
        }

        // Resize gambar jika lebar diset
        if ($width) {
            $oldWidth = imagesx($image);
            $oldHeight = imagesy($image);
            $aspectRatio = $oldWidth / $oldHeight;
            if (!$height) {
                $height = $width / $aspectRatio; // Hitung tinggi dengan mempertahankan
aspek rasio
            }
            $newImage = imagecreatetruecolor($width, $height);
            imagecopyresampled($newImage, $image, 0, 0, 0, 0, $width, $height, $oldWidth,
$oldHeight);
            imagedestroy($image);
            $image = $newImage;
        }

        // Simpan gambar dengan kualitas asli
        switch ($extension) {
            case 'jpeg':
            case 'jpg':
                imagejpeg($image, $destinationPath . '/' . $fileName);
                break;
            case 'png':
                imagepng($image, $destinationPath . '/' . $fileName);
                break;
            case 'gif':
                imagegif($image, $destinationPath . '/' . $fileName);
                break;
        }

        imagedestroy($image);
        return $fileName;
    }
}
```



Gambar VII. 7
Script ImageHelper

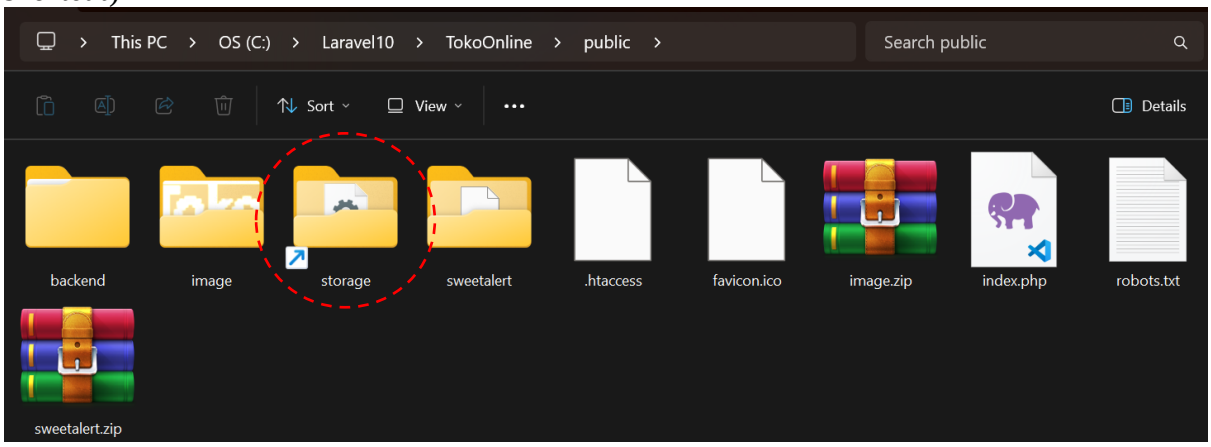
5. Perintah php artisan storage:link adalah perintah dalam Laravel yang digunakan untuk membuat symbolic link (symlink) dari direktori storage/app/public ke direktori public/storage. Ini memungkinkan file yang disimpan di direktori storage dapat diakses secara publik melalui URL. Jalankan perintah di terminal:

```
php artisan storage:link
```



Gambar VII. 8
Artisan Storage link

6. Jika berhasil maka pada folder *public* akan tertambah folder *storage* (folder berbentuk *shortcut*)



Gambar VII. 9
Storage

7. Setelah kita membuat **ImageHelper** maka kita panggil di **UserController** demikian kita tambahkan juga *Library* **Hash** untuk password pada *function store(). Library* **Hash** di Laravel digunakan untuk mengenkripsi (hashing) dan memverifikasi password dengan aman.

```php
<?php
namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\User;
use Illuminate\Support\Facades\Hash;
use App\Helpers\ImageHelper;

class UserController extends Controller
{
    /**
     * function lainnya
     */
    public function store(Request $request)
    {
        $validatedData = $request->validate([
            'nama' => 'required|max:255',
            'email' => 'required|max:255|email|unique:user',
            'role' => 'required',
            'hp' => 'required|min:10|max:13',
            'password' => 'required|min:4|confirmed',
            'foto' => 'image|mimes:jpeg,jpg,png,gif|file|max:1024',
        ], $messages = [
            'foto.image' => 'Format gambar gunakan file dengan ekstensi jpeg, jpg, png,
atau gif.',
            'foto.max' => 'Ukuran file gambar Maksimal adalah 1024 KB.'
        ]);
        $validatedData['status'] = 0;

        // menggunakan ImageHelper
        if ($request->file('foto')) {
            $file = $request->file('foto');
            $extension = $file->getClientOriginalExtension();
            $originalFileName = date('YmdHis') . '_' . uniqid() . '.' . $extension;
            $directory = 'storage/img-user/';
            // Simpan gambar dengan ukuran yang ditentukan
            ImageHelper::uploadAndResize($file, $directory, $originalFileName, 385, 400);
// null (jika tinggi otomatis)
            // Simpan nama file asli di database
            $validatedData['foto'] = $originalFileName;
        }

        // password kombinasi
        $password = $request->input('password');
        $pattern = '/^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[\W_]).+$/';
        // huruf kecil ([a-z]), huruf besar ([A-Z]), dan angka (\d) (?=.*[\W_]) simbol
karakter (non-alphanumeric)
        if (preg_match($pattern, $password)) {
            $validatedData['password'] = Hash::make($validatedData['password']);
            User::create($validatedData, $messages);
            return redirect()->route('backend.user.index')->with('success', 'Data berhasil
tersimpan');
        } else {
            return redirect()->back()->withErrors(['password' => 'Password harus terdiri
dari kombinasi huruf besar, huruf kecil, angka, dan simbol karakter.']);
        }
    }

    /**
     * function lainnya
     */
}
```

pada baris *script* ini kita dapat mengatur ukuran tinggi misalnya lebar 385px dengan tinggi 400px, namun kita juga dapat mengatur dengan lebar 385px dengan tinggi otomatis

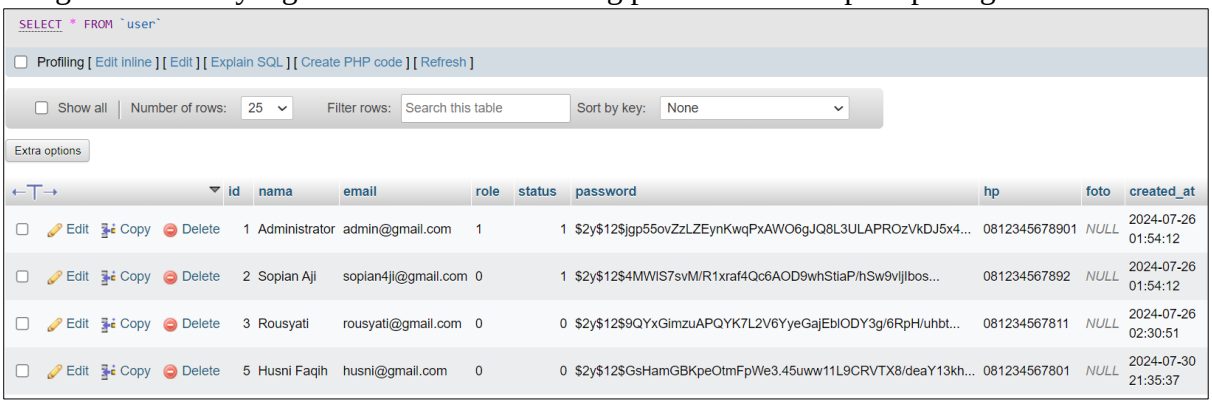
```
// Simpan gambar dengan ukuran yang ditentukan
            ImageHelper::uploadAndResize($file, $directory, $originalFileName, 385, 400);
// null (jika tinggi otomatis)
```

Sedangkan untuk // password kombinasi pada saat input data pada form password terdiri dari kombinasi huruf kecil, huruf besar, angka dan simbol karakter misalnya **P@55word**

8. Kemudian jangan lupa untuk membuat folder **img-user** pada terminal, sehingga kita memiliki folder **img-user** pada public/storage/ atau storage/app/public/

```
mkdir storage/app/public/img-user
```

* Tambahkan file gambar img-default.jpg dari https://bit.ly/LaravelWebPro2 di image.zip ke direktori public/storage/img-user atau storage/app/public/img-user. File ini digunakan untuk mengisi field foto yang bernilai null atau kosong pada database seperti pada gambar

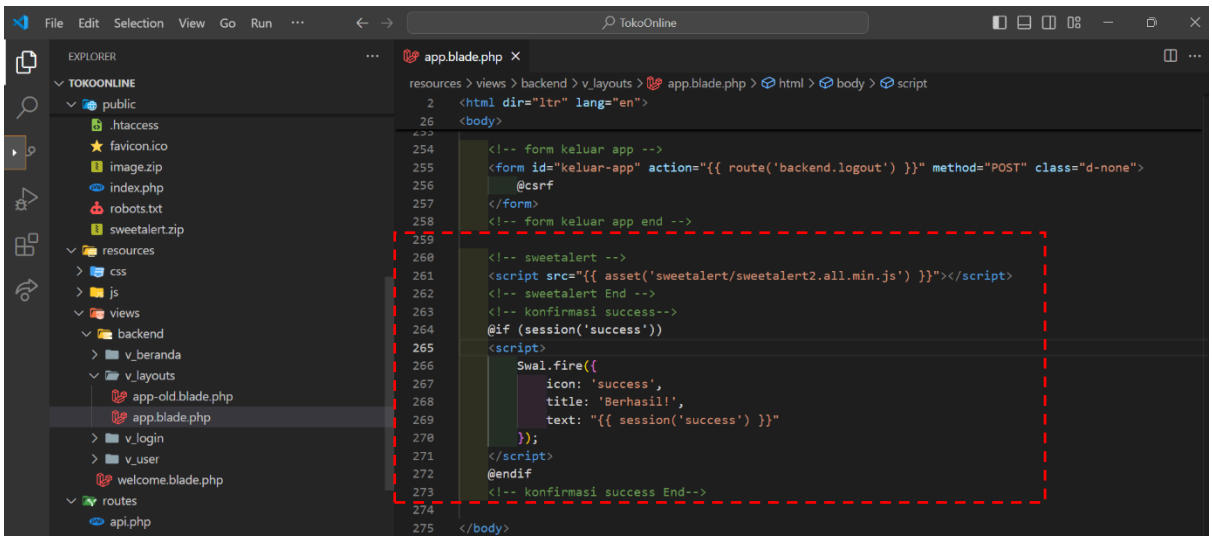


Gambar VII. 10
Field Foto Null

7.6. SweetAlert

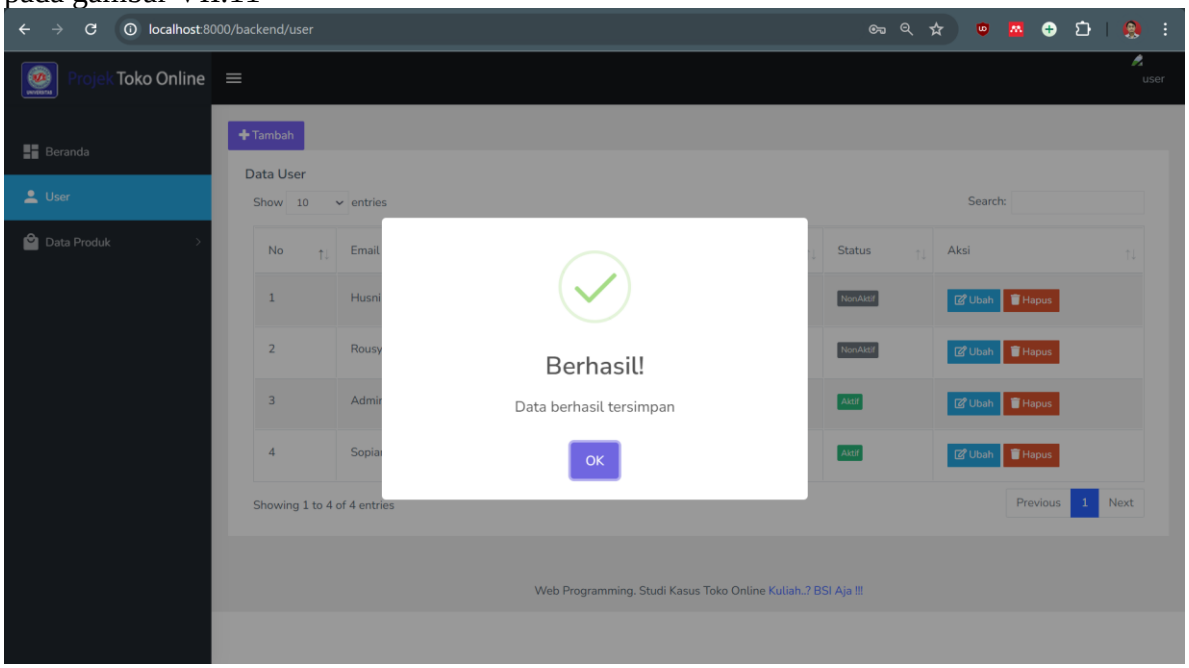
1. Silakan buat data user baru dengan mengklik tombol **Tambah**. Jika berhasil tersimpan, maka akan redirect ke halaman utama data user, yakni route('backend.user.index'). Namun, saat berhasil tersimpan, pesan **with('success')** tidak tampil (konfirmasi pesan data berhasil tersimpan). Hal ini terjadi karena kita belum memanggil **SweetAlert**. Untuk memanggil **SweetAlert**, terlebih dulu kita download dari https://bit.ly/LaravelWebPro2 dan ekstrak SweetAlert pada folder public seperti pada gambar VII.8 Pada **app.blade.php** di direktori resources\views\backend\v_layouts, tambahkan script untuk memanggil **SweetAlert**, kemudian tambahkan konfirmasi **success** seperti pada gambar VII.10

```
    <!-- sweetalert -->
    <script src="{{ asset('sweetalert/sweetalert2.all.min.js') }}"></script>
    <!-- sweetalert End -->
    <!-- konfirmasi success-->
    @if (session('success'))
    <script>
        Swal.fire({
            icon: 'success',
            title: 'Berhasil!',
            text: "{{ session('success') }}"
        });
    </script>
    @endif
    <!-- konfirmasi success End-->
```



Gambar VII. 11
Memanggil SweetAlert & Session('success')

2. Dengan demikian pada saat kita create data user maka konfirmasi *success* akan tampil seperti pada gambar VII.11



Gambar VII. 12
Konfirmasi Success

3. Kemudian pada Aksi Hapus. Pada UserContoller dengan *function destroy()*

```
    public function destroy(string $id)
    {
        $user = user::findOrFail($id);
        if ($user->foto) {
            $oldImagePath = public_path('storage/img-user/') . $user->foto;
            if (file_exists($oldImagePath)) {
                unlink($oldImagePath);
            }
        }
        $user->delete();
        return redirect()->route('backend.user.index')->with('success', 'Data berhasil
dihapus');
```

```
    }
```

Pada *script destroy*, selain menghapus data berdasarkan $id, *script* ini juga menghapus gambar jika $user memiliki gambar. Dimana gambar yang akan dihapus pada direktori storage/img-user/.

4. Lakukakan perubahan pada resources\views\backend\v_user\index.blade.php dimana baris *script* aksi hapus:

```
<form method="POST" action="{{ route('backend.user.destroy', $row->id) }}" style="display:
inline-block;">
    @method('delete')
    @csrf
    <button type="submit" class="btn btn-danger btn-sm" title='Hapus Data'>
        <i class="fas fa-trash"></i> Hapus</button>
</form>
```

Perubahan pada **Button Class** yakni **show_confirm** & **data-konf-delete,** kemudian kita akan deklarasikan pada *javascript* yakni pada konfirmasi delete.

```
<form method="POST" action="{{ route('backend.user.destroy', $row->id) }}" style="display:
inline-block;">
    @method('delete')
    @csrf
    <button type="submit" class="btn btn-danger btn-sm show_confirm" data-konf-delete="{{
$row->nama }}" title='Hapus Data'>
        <i class="fas fa-trash"></i> Hapus</button>
</form>
```

Berikut *script* lengkap resources\views\backend\v_user**index.blade.php**

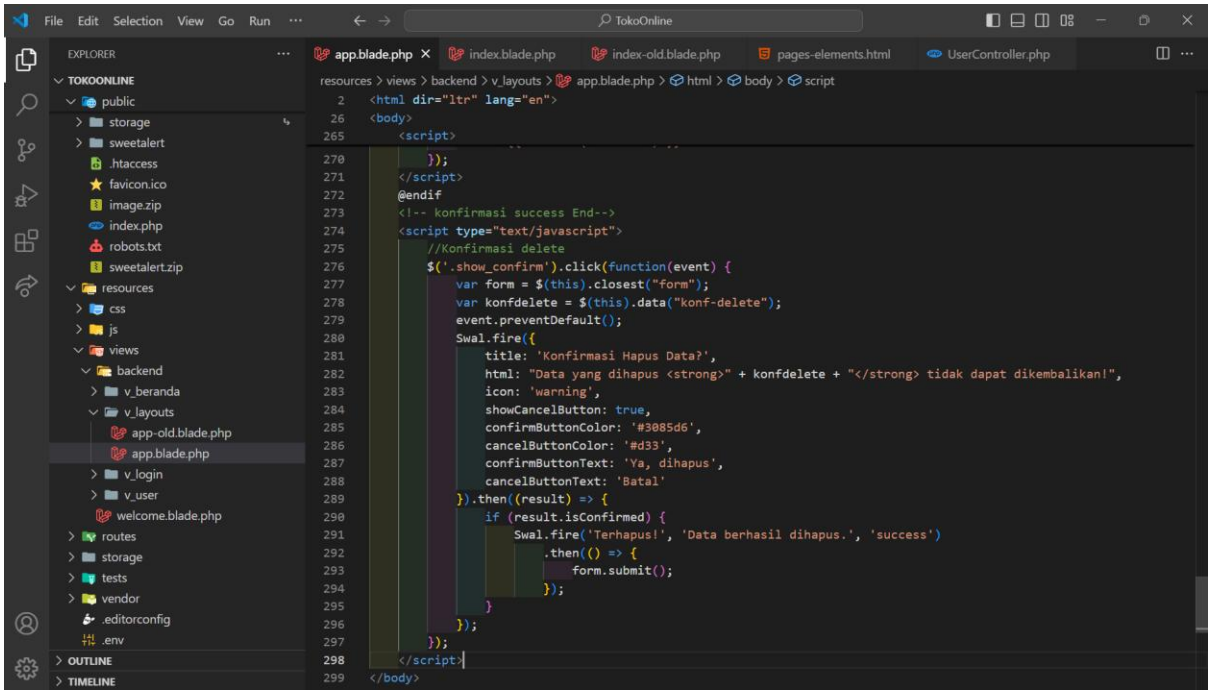
```
@extends('backend.v_layouts.app')
@section('content')
<!-- contentAwal -->

<div class="row">

    <div class="col-12">
        <a href="{{ route('backend.user.create') }}">
            <button type="button" class="btn btn-primary"><i class="fas fa-plus"></i>
Tambah</button>
        </a>
        <div class="card">
            <div class="card-body">
                <h5 class="card-title"> {{$judul}} </h5>
                <div class="table-responsive">
                    <table id="zero_config" class="table table-striped table-bordered">
                        <thead>
                            <tr>
                                <th>No</th>
                                <th>Email</th>
                                <th>Nama</th>
                                <th>Role</th>
                                <th>Status</th>
                                <th>Aksi</th>
                            </tr>
                        </thead>

                        <tbody>
                            @foreach ($index as $row)
                            <tr>
                                <td> {{ $loop->iteration }} </td>
                                <td> {{$row->nama}} </td>
                                <td> {{$row->email}} </td>
                                <td>
                                    @if ($row->role == 1)
```

```
                                    <span class="badge badge-success"></i>
                                        Super Admin</span>
                                    @elseif($row->role == 0)
                                    <span class="badge badge-primary"></i>
                                        Admin</span>
                                    @endif
                                </td>
                                <td>
                                    @if ($row->status ==1)
                                    <span class="badge badge-success"></i>
                                        Aktif</span>
                                    @elseif($row->status ==0)
                                    <span class="badge badge-secondary"></i>
                                        NonAktif</span>
                                    @endif
                                </td>
                                <td>
                                    <a href="{{ route('backend.user.edit', $row->id) }}"
title="Ubah Data">
                                        <button type="button" class="btn btn-cyan btn-
sm"><i class="far fa-edit"></i> Ubah</button>
                                    </a>

                                    <form method="POST" action="{{
route('backend.user.destroy', $row->id) }}" style="display: inline-block;">
                                        @method('delete')
                                        @csrf
                                        <button type="submit" class="btn btn-danger btn-sm
show_confirm" data-konf-delete="{{ $row->nama }}" title='Hapus Data'>
                                            <i class="fas fa-trash"></i> Hapus</button>
                                    </form>
                                </td>
                            </tr>
                            @endforeach
                        </tbody>
                    </table>
                </div>

            </div>
        </div>
    </div>
</div>

<!-- contentAkhir -->
@endsection
```

5. Untuk menangkap **show_confirm & data-konf-delete** maka pada resources\views\backend\v_layouts**app.blade.php** kita tambahkan *javascript* seperti pada gambar VII.12

```
<script type="text/javascript">
        //Konfirmasi delete
        $('.show_confirm').click(function(event) {
            var form = $(this).closest("form");
            var konfdelete = $(this).data("konf-delete");
            event.preventDefault();
            Swal.fire({
                title: 'Konfirmasi Hapus Data?',
                html: "Data yang dihapus <strong>" + konfdelete + "</strong> tidak dapat
dikembalikan!",
                icon: 'warning',
                showCancelButton: true,
                confirmButtonColor: '#3085d6',
                cancelButtonColor: '#d33',
                confirmButtonText: 'Ya, dihapus',
```

```
                cancelButtonText: 'Batal'
            }).then((result) => {
                if (result.isConfirmed) {
                    Swal.fire('Terhapus!', 'Data berhasil dihapus.', 'success')
                        .then(() => {
                            form.submit();
                        });
                }
            });
        });
</script>
```



Gambar VII. 13
javascrip Konfirmasi Delete

6. Kali ini, jika aksi Hapus diklik, akan tampil konfirmasi seperti pada gambar VII.13. Jika **Ya** dipilih, maka data akan dihapus & muncul konfimasi berikutny seperti pada gambar VII.14, namun jika **Batal** dipilih, maka akan kembali ke halaman **index()** User. Sehingga *script* lengkap dari resources\views\backend\v_layouts**app.blade.php** sebagai berikut:

```
<!DOCTYPE html>
<html dir="ltr" lang="en">

<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <!-- Tell the browser to be responsive to screen width -->
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="">
    <meta name="author" content="">
    <!-- Favicon icon -->
    <link rel="icon" type="image/png" sizes="16x16" href="{{
asset('image/icon_univ_bsi.png') }}">
    <title>tokoonline</title>
    <!-- Custom CSS -->
    <link rel="stylesheet" type="text/css" href="{{ asset('backend/extra-
libs/multicheck/multicheck.css') }}">
    <link href="{{ asset('backend/libs/datatables.net-bs4/css/dataTables.bootstrap4.css')
}}" rel="stylesheet">
    <link href="{{ asset('backend/dist/css/style.min.css') }}" rel="stylesheet">
```

```
    <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
    <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
    <script src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>
<![endif]-->
</head>

<body>
    <!-- ============================================================== -->
    <!-- Preloader - style you can find in spinners.css -->
    <!-- ============================================================== -->
    <div class="preloader">
        <div class="lds-ripple">
            <div class="lds-pos"></div>
            <div class="lds-pos"></div>
        </div>
    </div>
    <!-- ============================================================== -->
    <!-- Main wrapper - style you can find in pages.scss -->
    <!-- ============================================================== -->
    <div id="main-wrapper">
        <!-- ============================================================== -->
        <!-- Topbar header - style you can find in pages.scss -->
        <!-- ============================================================== -->
        <header class="topbar" data-navbarbg="skin5">
            <nav class="navbar top-navbar navbar-expand-md navbar-dark">
                <div class="navbar-header" data-logobg="skin5">
                    <!-- This is for the sidebar toggle which is visible on mobile only -->
                    <a class="nav-toggler waves-effect waves-light d-block d-md-none"
href="javascript:void(0)"><i class="ti-menu ti-close"></i></a>
                    <!-- ============================================================== -->
                    <!-- Logo -->
                    <!-- ============================================================== -->
                    <a class="navbar-brand" href="index.html">
                        <!-- Logo icon -->
                        <b class="logo-icon p-l-10">
                            <!--You can put here icon as well // <i class="wi wi-
sunset"></i> //-->
                            <!-- Dark Logo icon -->
                            <img src="{{ asset('image/icon_univ_bsi.png') }}"
alt="homepage" class="light-logo" />

                        </b>
                        <!--End Logo icon -->
                        <!-- Logo text -->
                        <span class="logo-text">
                            <!-- dark Logo text -->
                            <img src="{{ asset('image/logo_text.png') }}" alt="homepage"
class="light-logo" />

                        </span>
                        <!-- Logo icon -->
                        <!-- <b class="logo-icon"> -->
                        <!--You can put here icon as well // <i class="wi wi-sunset"></i>
//-->
                        <!-- Dark Logo icon -->
                        <!-- <img src="assets/images/logo-text.png" alt="homepage"
class="light-logo" /> -->

                        <!-- </b> -->
                        <!--End Logo icon -->
                    </a>
                    <!-- ============================================================== -->
                    <!-- End Logo -->
                    <!-- ============================================================== -->
```

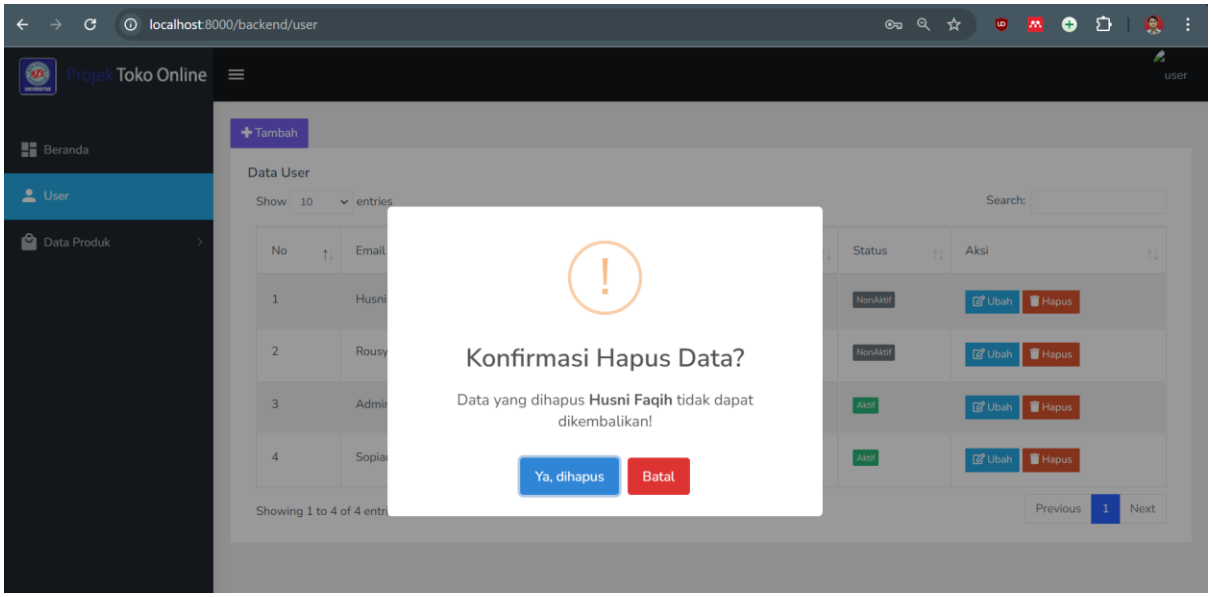
```
                    <!-- ============================================================== -->
                    <!-- Toggle which is visible on mobile only -->
                    <!-- ============================================================== -->
                    <a class="topbartoggler d-block d-md-none waves-effect waves-light"
href="javascript:void(0)" data-toggle="collapse" data-target="#navbarSupportedContent"
aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation"><i class="ti-more"></i></a>
                </div>
                <!-- ============================================================== -->
                <!-- End Logo -->
                <!-- ============================================================== -->
                <div class="navbar-collapse collapse" id="navbarSupportedContent" data-
navbarbg="skin5">
                    <!-- ============================================================== -->
                    <!-- toggle and nav items -->
                    <!-- ============================================================== -->
                    <ul class="navbar-nav float-left mr-auto">
                        <li class="nav-item d-none d-md-block"><a class="nav-link
sidebartoggler waves-effect waves-light" href="javascript:void(0)" data-sidebartype="mini-
sidebar"><i class="mdi mdi-menu font-24"></i></a></li>
                        <!-- ============================================================
-->
                        <!-- create new -->
                        <!-- ============================================================
-->

                        <!-- ============================================================
-->
                        <!-- Search -->
                        <!-- ============================================================
-->

                    </ul>
                    <!-- ============================================================== -->
                    <!-- Right side toggle and nav items -->
                    <!-- ============================================================== -->
                    <ul class="navbar-nav float-right">
                        <!-- ============================================================
-->
                        <!-- Comment -->
                        <!-- ============================================================
-->

                        <!-- ============================================================
-->
                        <!-- End Comment -->
                        <!-- ============================================================
-->
                        <!-- ============================================================
-->
                        <!-- Messages -->
                        <!-- ============================================================
-->

                        <!-- ============================================================
-->
                        <!-- End Messages -->
                        <!-- ============================================================
-->

                        <!-- ============================================================
-->
                        <!-- User profile and search -->
                        <!-- ============================================================
-->
                        <li class="nav-item dropdown">
```

```
                            <a class="nav-link dropdown-toggle text-muted waves-effect
waves-dark pro-pic" href="" data-toggle="dropdown" aria-haspopup="true" aria-
expanded="false"><img src="assets/images/users/1.jpg" alt="user" class="rounded-circle"
width="31"></a>
                            <div class="dropdown-menu dropdown-menu-right user-dd
animated">
                                <a class="dropdown-item" href="javascript:void(0)"><i
class="ti-user m-r-5 m-l-5"></i> Profil Saya</a>
                                <a class="dropdown-item" href=""
onclick="event.preventDefault(); document.getElementById('keluar-app').submit();"><i
class="fa fa-power-off m-r-5 m-l-5"></i> Keluar</a>
                                <div class="dropdown-divider"></div>

                            </div>
                        </li>
                        <!-- ============================================================
-->
                        <!-- User profile and search -->
                        <!-- ============================================================
-->
                    </ul>
                </div>
            </nav>
        </header>
        <!-- ============================================================== -->
        <!-- End Topbar header -->
        <!-- ============================================================== -->
        <!-- ============================================================== -->
        <!-- Left Sidebar - style you can find in sidebar.scss  -->
        <!-- ============================================================== -->
        <aside class="left-sidebar" data-sidebarbg="skin5">
            <!-- Sidebar scroll-->
            <div class="scroll-sidebar">
                <!-- Sidebar navigation-->
                <nav class="sidebar-nav">
                    <ul id="sidebarnav" class="p-t-30">
                        <li class="sidebar-item"> <a class="sidebar-link waves-effect
waves-dark sidebar-link" href="{{ route('backend.beranda') }}" aria-expanded="false"><i
class="mdi mdi-view-dashboard"></i><span class="hide-menu">Beranda</span></a>
                        </li>
                        <li class="sidebar-item"> <a class="sidebar-link waves-effect
waves-dark sidebar-link" href="{{ route('backend.user.index') }}" aria-expanded="false"><i
class="mdi mdi-account"></i><span class="hide-menu">User</span></a>
                        </li>
                        <li class="sidebar-item"> <a class="sidebar-link has-arrow waves-
effect waves-dark" href="javascript:void(0)" aria-expanded="false"><i class="mdi mdi-
shopping"></i><span class="hide-menu">Data Produk </span></a>
                            <ul aria-expanded="false" class="collapse  first-level">
                                <li class="sidebar-item"><a href="icon-material.html"
class="sidebar-link"><i class="mdi mdi-chevron-right"></i><span class="hide-menu"> Kategori
</span></a>
                                </li>
                                <li class="sidebar-item"><a href="icon-fontawesome.html"
class="sidebar-link"><i class="mdi mdi-chevron-right"></i><span class="hide-menu"> Produk
</span></a>
                                </li>
                            </ul>
                        </li>

                    </ul>
                </nav>
                <!-- End Sidebar navigation -->
            </div>
            <!-- End Sidebar scroll-->
        </aside>
        <!-- ============================================================== -->
```

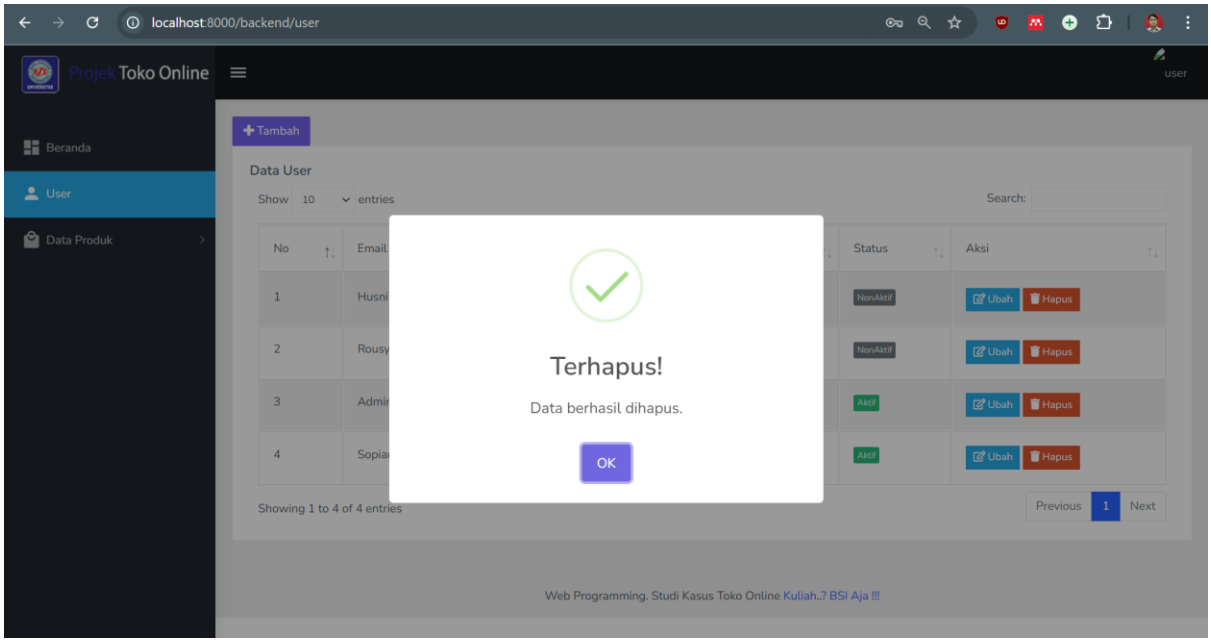
```
        <!-- End Left Sidebar - style you can find in sidebar.scss  -->
        <!-- ============================================================== -->
        <!-- ============================================================== -->
        <!-- Page wrapper  -->
        <!-- ============================================================== -->
        <div class="page-wrapper">
            <!-- ============================================================== -->
            <!-- Bread crumb and right sidebar toggle -->
            <!-- ============================================================== -->

            <!-- ============================================================== -->
            <!-- End Bread crumb and right sidebar toggle -->
            <!-- ============================================================== -->
            <!-- ============================================================== -->
            <!-- Container fluid  -->
            <!-- ============================================================== -->
            <div class="container-fluid">
                <!-- ============================================================== -->
                <!-- Start Page Content -->
                <!-- ============================================================== -->

                <!-- @yieldAwal -->
                @yield('content')
                <!-- @yieldAkhir-->

                <!-- ============================================================== -->
                <!-- End PAge Content -->
                <!-- ============================================================== -->
                <!-- ============================================================== -->
                <!-- Right sidebar -->
                <!-- ============================================================== -->
                <!-- .right-sidebar -->
                <!-- ============================================================== -->
                <!-- End Right sidebar -->
                <!-- ============================================================== -->
            </div>
            <!-- ============================================================== -->
            <!-- End Container fluid  -->
            <!-- ============================================================== -->
            <!-- ============================================================== -->
            <!-- footer -->
            <!-- ============================================================== -->
            <footer class="footer text-center">
                Web Programming. Studi Kasus Toko Online <a
href="https://bsi.ac.id/">Kuliah..? BSI Aja !!!</a>
            </footer>
            <!-- ============================================================== -->
            <!-- End footer -->
            <!-- ============================================================== -->
        </div>
        <!-- ============================================================== -->
        <!-- End Page wrapper  -->
        <!-- ============================================================== -->
    </div>
    <!-- ============================================================== -->
    <!-- End Wrapper -->
    <!-- ============================================================== -->
    <!-- ============================================================== -->
    <!-- All Jquery -->
    <!-- ============================================================== -->
    <script src="{{ asset('backend/libs/jquery/dist/jquery.min.js') }}"></script>
    <!-- Bootstrap tether Core JavaScript -->
    <script src="{{ asset('backend/libs/popper.js/dist/umd/popper.min.js') }}"></script>
    <script src="{{ asset('backend/libs/bootstrap/dist/js/bootstrap.min.js') }}"></script>
    <!-- slimscrollbar scrollbar JavaScript -->
```

```
    <script src="{{ asset('backend/libs/perfect-scrollbar/dist/perfect-
scrollbar.jquery.min.js') }}"></script>
    <script src="{{ asset('backend/extra-libs/sparkline/sparkline.js') }}"></script>
    <!--Wave Effects -->
    <script src="{{ asset('backend/dist/js/waves.js') }}"></script>
    <!--Menu sidebar -->
    <script src="{{ asset('backend/dist/js/sidebarmenu.js') }}"></script>
    <!--Custom JavaScript -->
    <script src="{{ asset('backend/dist/js/custom.min.js') }}"></script>
    <!-- this page js -->
    <script src="{{ asset('backend/extra-libs/multicheck/datatable-checkbox-init.js')
}}"></script>
    <script src="{{ asset('backend/extra-libs/multicheck/jquery.multicheck.js')
}}"></script>
    <script src="{{ asset('backend/extra-libs/DataTables/datatables.min.js') }}"></script>
    <script>
        /***************************************
         *       Basic Table                   *
         ***************************************/
        $('#zero_config').DataTable();
    </script>

    <!-- form keluar app -->
    <form id="keluar-app" action="{{ route('backend.logout') }}" method="POST" class="d-
none">
        @csrf
    </form>
    <!-- form keluar app end -->

    <!-- sweetalert -->
    <script src="{{ asset('sweetalert/sweetalert2.all.min.js') }}"></script>
    <!-- sweetalert End -->
    <!-- konfirmasi success-->
    @if (session('success'))
    <script>
        Swal.fire({
            icon: 'success',
            title: 'Berhasil!',
            text: "{{ session('success') }}"
        });
    </script>
    @endif
    <!-- konfirmasi success End-->
    <script type="text/javascript">
        //Konfirmasi delete
        $('.show_confirm').click(function(event) {
            var form = $(this).closest("form");
            var konfdelete = $(this).data("konf-delete");
            event.preventDefault();
            Swal.fire({
                title: 'Konfirmasi Hapus Data?',
                html: "Data yang dihapus <strong>" + konfdelete + "</strong> tidak dapat
dikembalikan!",
                icon: 'warning',
                showCancelButton: true,
                confirmButtonColor: '#3085d6',
                cancelButtonColor: '#d33',
                confirmButtonText: 'Ya, dihapus',
                cancelButtonText: 'Batal'
            }).then((result) => {
                if (result.isConfirmed) {
                    Swal.fire('Terhapus!', 'Data berhasil dihapus.', 'success')
                        .then(() => {
                            form.submit();
                        });
                }
```

```
            });
        });
    </script>
</body>

</html>
```



Gambar VII. 14
Konfirmasi Delete



Gambar VII. 15
Konfirmasi Data Berhasil di Hapus

7.7. Menerapkan Form Edit Dengan Template

1. Berikutnya adalah aksi pada Ubah, dimana ubah menggunakan *function edit*() pada *controller* UserController

```
    public function edit(string $id)
    {
        $user = User::findOrFail($id);
        return view('backend.v_user.edit', [
```

```
            'judul' => 'Ubah User',
            'edit' => $user
        ]);
    }
```

2. kemudian pada *function update()* yakni masih di controller UserController

```
    public function update(Request $request, string $id)
    {
        //ddd($request);
        $user = User::findOrFail($id);
        $rules = [
            'nama' => 'required|max:255',
            'role' => 'required',
            'status' => 'required',
            'hp' => 'required|min:10|max:13',
            'foto' => 'image|mimes:jpeg,jpg,png,gif|file|max:1024',
        ];
        $messages = [
            'foto.image' => 'Format gambar gunakan file dengan ekstensi jpeg, jpg, png,
atau gif.',
            'foto.max' => 'Ukuran file gambar Maksimal adalah 1024 KB.'
        ];

        if ($request->email != $user->email) {
            $rules['email'] = 'required|max:255|email|unique:user';
        }
        $validatedData = $request->validate($rules, $messages);

        // menggunakan ImageHelper
        if ($request->file('foto')) {
            //hapus gambar lama
            if ($user->foto) {
                $oldImagePath = public_path('storage/img-user/') . $user->foto;
                if (file_exists($oldImagePath)) {
                    unlink($oldImagePath);
                }
            }
            $file = $request->file('foto');
            $extension = $file->getClientOriginalExtension();
            $originalFileName = date('YmdHis') . '_' . uniqid() . '.' . $extension;
            $directory = 'storage/img-user/';
            // Simpan gambar dengan ukuran yang ditentukan
            ImageHelper::uploadAndResize($file, $directory, $originalFileName, 385, 400);
// null (jika tinggi otomatis)
            // Simpan nama file asli di database
            $validatedData['foto'] = $originalFileName;
        }

        $user->update($validatedData);
        return redirect()->route('backend.user.index')->with('success', 'Data berhasil
diperbaharui');
    }
```

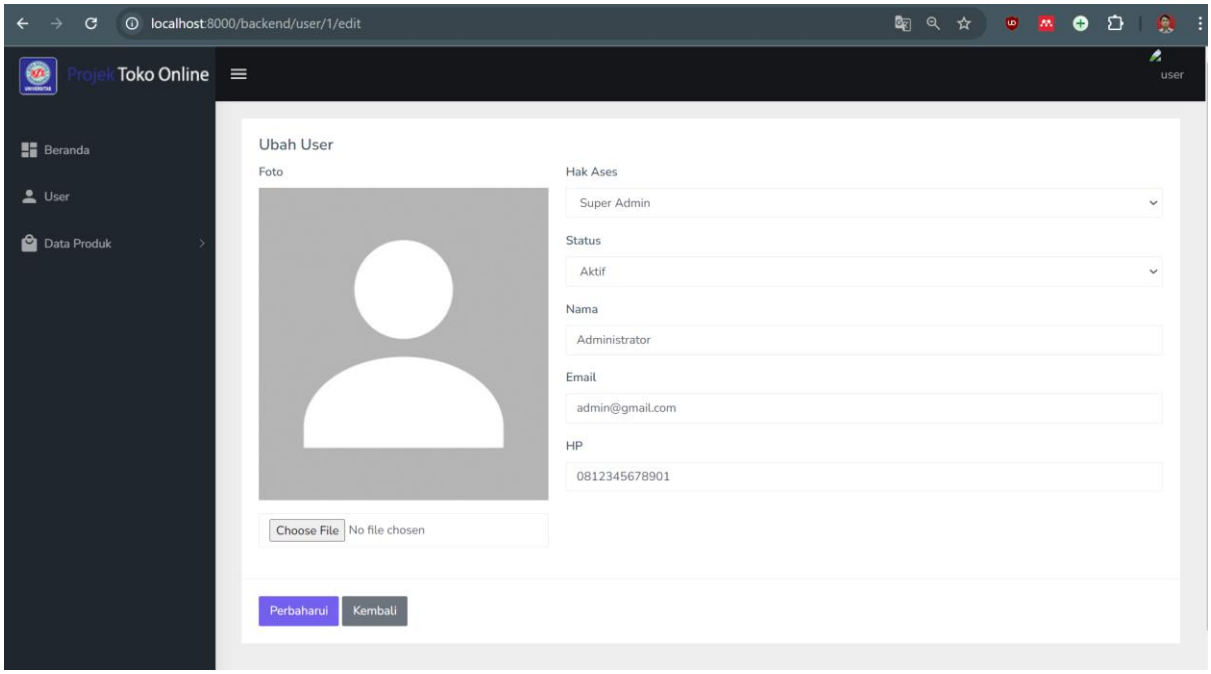
3. Tambahkan file **edit.blade.php** pada direktori resources\views\backend\v_user. Atau dengan cara menduplikat **create.blade.php** dan menggantinya dengan nama **edit.blade.php**. Berikut ini adalah *script* lengkap dari **edit.blade.php**. Pastikan kita sudah menambahkan file gambar **img-default.jpg** yang bisa diunduh dari **https://bit.ly/LaravelWebPro2** di **image.zip** ke dalam direktori public/storage/img-user. Jika sudah, tampilannya akan terlihat seperti pada gambar VII.15. Ketika Kita mengubah salah satu data pada form atau mengganti foto, kemudian mengklik tombol perbarui, akan muncul konfirmasi bahwa data berhasil diperbarui seperti yang ditampilkan pada gambar VII.16.

```
@extends('backend.v_layouts.app')
```

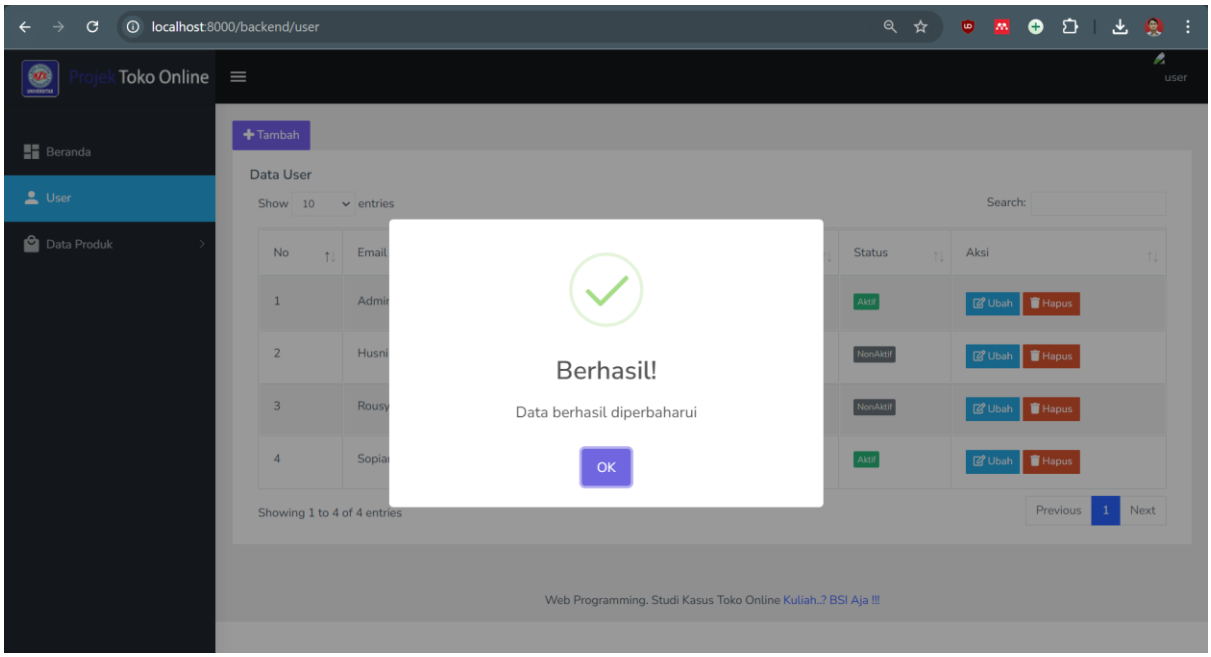
```
@section('content')
<!-- contentAwal -->

<div class="container-fluid">
    <div class="row">
        <div class="col-12">
            <div class="card">
                <form action="{{ route('backend.user.update', $edit->id) }}" method="post"
enctype="multipart/form-data">
                    @method('put')
                    @csrf

                    <div class="card-body">
                        <h4 class="card-title"> {{$judul}} </h4>
                        <div class="row">
                            <div class="col-md-4">
                                <div class="form-group">
                                    <label>Foto</label>
                                    {{-- view image --}}
                                    @if ($edit->foto)
                                    <img src="{{ asset('storage/img-user/' . $edit->foto)
}}" class="foto-preview" width="100%">
                                    <p></p>
                                    @else
                                    <img src="{{ asset('storage/img-user/img-default.jpg')
}}" class="foto-preview" width="100%">
                                    <p></p>
                                    @endif
                                    {{-- file foto --}}
                                    <input type="file" name="foto" class="form-control
@error('foto') is-invalid @enderror" onchange="previewFoto()">
                                    @error('foto')
                                    <div class="invalid-feedback alert-danger">{{ $message
}}</div>
                                    @enderror
                                </div>
                            </div>
                            <div class="col-md-8">
                                <div class="form-group">
                                    <label>Hak Ases</label>
                                    <select name="role" class="form-control @error('role')
is-invalid @enderror">
                                        <option value="" {{ old('role', $edit->role) == ''
? 'selected' : '' }}> -
                                            Pilih Hak Akses -</option>
                                        <option value="1" {{ old('role', $edit->role) ==
'1' ? 'selected' : '' }}>
                                            Super Admin</option>
                                        <option value="0" {{ old('role', $edit->role) ==
'0' ? 'selected' : '' }}>
                                            Admin</option>
                                    </select>
                                    @error('role')
                                    <span class="invalid-feedback alert-danger"
role="alert">
                                        {{ $message }}
                                    </span>
                                    @enderror
                                </div>
                                <div class="form-group">
                                    <label>Status</label>
                                    <select name="status" class="form-control
@error('status') is-invalid @enderror">
                                        <option value="" {{ old('status', $edit->status) ==
'' ? 'selected' : '' }}> -
                                            Pilih Hak Akses -</option>
```

```
                                    <option value="1" {{ old('status', $edit->status)
== '1' ? 'selected' : '' }}>
                                        Aktif</option>
                                    <option value="0" {{ old('status', $edit->status)
== '0' ? 'selected' : '' }}>
                                        NonAktif</option>
                                </select>
                                @error('status')
                                <span class="invalid-feedback alert-danger"
role="alert">
                                    {{ $message }}
                                </span>
                                @enderror
                            </div>

                            <div class="form-group">
                                <label>Nama</label>
                                <input type="text" name="nama" value="{{ old('nama',
$edit->nama) }}" class="form-control @error('nama') is-invalid @enderror"
placeholder="Masukkan Nama">
                                @error('nama')
                                <span class="invalid-feedback alert-danger"
role="alert">
                                    {{ $message }}
                                </span>
                                @enderror
                            </div>


                            <div class="form-group">
                                <label>Email</label>
                                <input type="text" name="email" value="{{ old('email',
$edit->email) }}" class="form-control @error('email') is-invalid @enderror"
placeholder="Masukkan Email">
                                @error('email')
                                <span class="invalid-feedback alert-danger"
role="alert">
                                    {{ $message }}
                                </span>
                                @enderror
                            </div>

                            <div class="form-group">
                                <label>HP</label>
                                <input type="text" onkeypress="return
hanyaAngka(event)" name="hp" value="{{ old('hp', $edit->hp) }}" class="form-control
@error('hp') is-invalid @enderror" placeholder="Masukkan Nomor HP">
                                @error('hp')
                                <span class="invalid-feedback alert-danger"
role="alert">
                                    {{ $message }}
                                </span>
                                @enderror
                            </div>
                        </div>

                    </div>
                </div>
                <div class="border-top">
                    <div class="card-body">
                        <button type="submit" class="btn btn-
primary">Perbaharui</button>
                        <a href="{{ route('backend.user.index') }}">
                            <button type="button" class="btn btn-
secondary">Kembali</button>
```

```
                        </a>
                    </div>
                </div>
            </form>
        </div>
    </div>
</div>

<!-- contentAkhir -->
@endsection
```



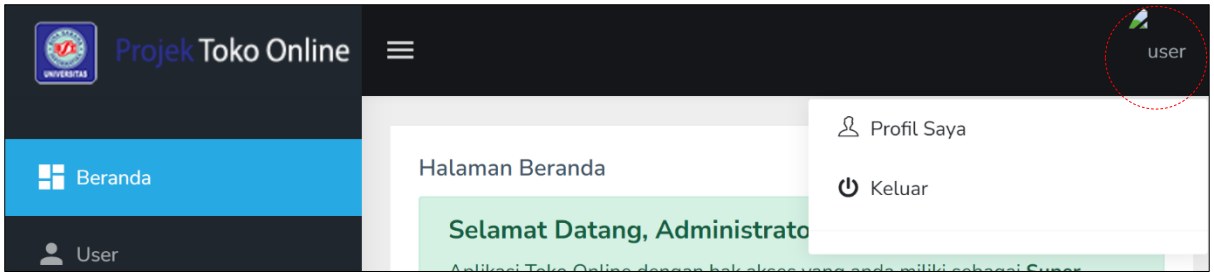
Gambar VII. 16
Foto User Administrator Dengan Avatar Default



Gambar VII. 17
Konfirmasi Data Berhasil Diperbaharui

Jika kita perhatikan pada avatar user, foto tidak tampil seperti pada gambar VII.16. Kali ini, kita akan memperbaikinya agar avatar menampilkan foto sesuai dengan foto avatar user yang login, jika berhasil seperti pada gambar VII.17. Untuk memanggil session dalam bentuk teks sebelumnya sudah dibahas pada pembahasan halaman utama. Namun, kali ini session yang dipanggil bukan hanya berbentuk teks yang digunakan untuk aksi (url) **Profil Saya**, di mana aksi tersebut akan mengarah ke profil akun user yang login, tetapi juga untuk menampilkan file atau foto. Berikut adalah perubahan yang perlu dilakukan pada resources\views\backend\v_layouts\app.blade.php dengan kata kunci pencarian berdasarkan dokumentasi yaitu "**User profile and search**".

```
                        <!-- ============================================================== -->
                        <!-- User profile and search -->
                        <!-- ============================================================== -->
                        <li class="nav-item dropdown">
                            <a class="nav-link dropdown-toggle text-muted waves-effect
waves-dark pro-pic" href="" data-toggle="dropdown" aria-haspopup="true" aria-
expanded="false">
                                <img src="assets/images/users/1.jpg" alt="user"
class="rounded-circle" width="31">
                            </a>
                            <div class="dropdown-menu dropdown-menu-right user-dd
animated">
                                <a class="dropdown-item" href="javascript:void(0)"><i
class="ti-user m-r-5 m-l-5"></i> Profil Saya</a>
                                <a class="dropdown-item" href=""
onclick="event.preventDefault(); document.getElementById('keluar-app').submit();"><i
class="fa fa-power-off m-r-5 m-l-5"></i> Keluar</a>
                                <div class="dropdown-divider"></div>

                            </div>
                        </li>
                        <!-- ============================================================== -->
                        <!-- User profile and search -->
                        <!-- ============================================================== -->
```

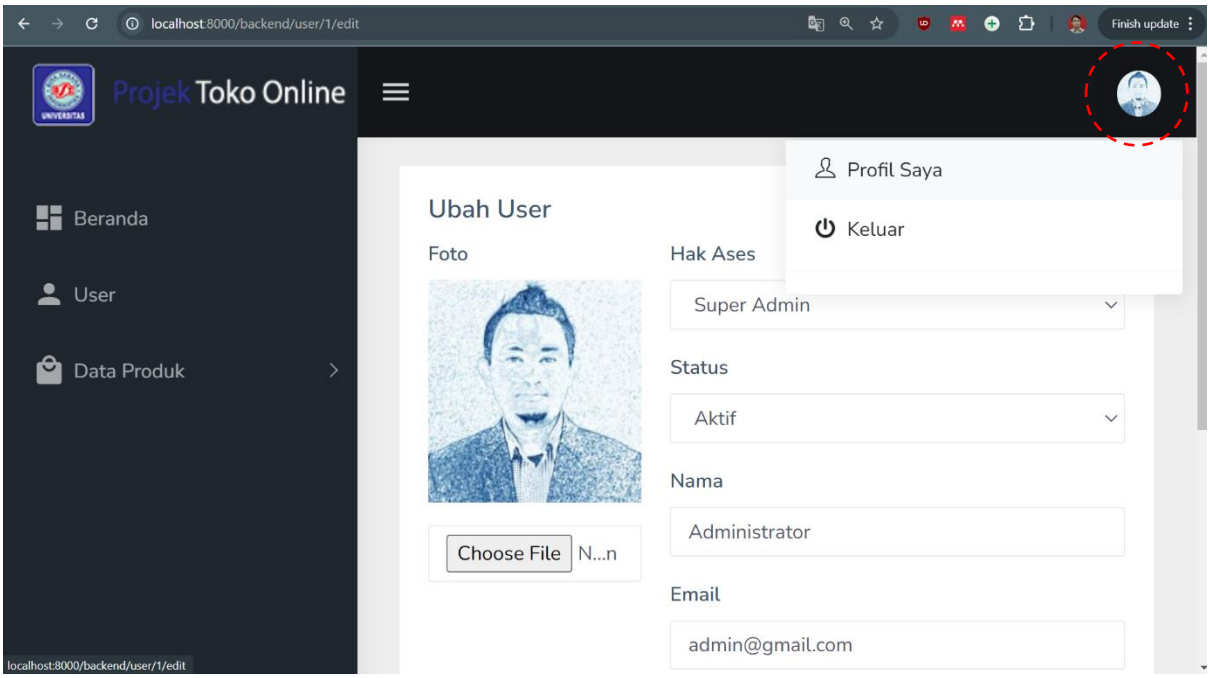


Gambar VII. 18
Foto Avatar Tidak Tampil

Berikut perubahan *script* lengkapnya

```
                        <!-- ============================================================== -->
                        <!-- User profile and search -->
                        <!-- ============================================================== -->
                        <li class="nav-item dropdown">
                            <a class="nav-link dropdown-toggle text-muted waves-effect
waves-dark pro-pic" href="" data-toggle="dropdown" aria-haspopup="true" aria-
expanded="false">
                                @if (Auth::user()->foto)
                                <img src="{{ asset('storage/img-user/' . Auth::user()-
>foto) }}" alt="user" class="rounded-circle" width="31">
                                @else
```

```
                                <img src="{{ asset('storage/img-user/img-default.jpg') }}"
alt="user" class="rounded-circle" width="31">
                                @endif
                            </a>
                            <div class="dropdown-menu dropdown-menu-right user-dd
animated">
                                <a class="dropdown-item" href="{{
route('backend.user.edit', Auth::user()->id) }}"><i class="ti-user m-r-5 m-l-5"></i> Profil
Saya</a>
                                <a class="dropdown-item" href=""
onclick="event.preventDefault(); document.getElementById('keluar-app').submit();"><i
class="fa fa-power-off m-r-5 m-l-5"></i> Keluar</a>
                                <div class="dropdown-divider"></div>

                            </div>
                        </li>
                        <!-- ============================================================== -->
                        <!-- User profile and search -->
                        <!-- ============================================================== -->
```



Gambar VII. 19
Foto Avatar Tampil

Berikut *script* lengkap dari resources\views\backend\v_layouts**app.blade.php**

```
<!DOCTYPE html>
<html dir="ltr" lang="en">

<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <!-- Tell the browser to be responsive to screen width -->
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="">
    <meta name="author" content="">
    <!-- Favicon icon -->
    <link rel="icon" type="image/png" sizes="16x16" href="{{
asset('image/icon_univ_bsi.png') }}">
    <title>tokoonline</title>
```

```
    <!-- Custom CSS -->
    <link rel="stylesheet" type="text/css" href="{{ asset('backend/extra-
libs/multicheck/multicheck.css') }}">
    <link href="{{ asset('backend/libs/datatables.net-bs4/css/dataTables.bootstrap4.css')
}}" rel="stylesheet">
    <link href="{{ asset('backend/dist/css/style.min.css') }}" rel="stylesheet">
    <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
    <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
    <script src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>
<![endif]-->
</head>

<body>
    <!-- ============================================================== -->
    <!-- Preloader - style you can find in spinners.css -->
    <!-- ============================================================== -->
    <div class="preloader">
        <div class="lds-ripple">
            <div class="lds-pos"></div>
            <div class="lds-pos"></div>
        </div>
    </div>
    <!-- ============================================================== -->
    <!-- Main wrapper - style you can find in pages.scss -->
    <!-- ============================================================== -->
    <div id="main-wrapper">
        <!-- ============================================================== -->
        <!-- Topbar header - style you can find in pages.scss -->
        <!-- ============================================================== -->
        <header class="topbar" data-navbarbg="skin5">
            <nav class="navbar top-navbar navbar-expand-md navbar-dark">
                <div class="navbar-header" data-logobg="skin5">
                    <!-- This is for the sidebar toggle which is visible on mobile only -->
                    <a class="nav-toggler waves-effect waves-light d-block d-md-none"
href="javascript:void(0)"><i class="ti-menu ti-close"></i></a>
                    <!-- ============================================================== -->
                    <!-- Logo -->
                    <!-- ============================================================== -->
                    <a class="navbar-brand" href="index.html">
                        <!-- Logo icon -->
                        <b class="logo-icon p-l-10">
                            <!--You can put here icon as well // <i class="wi wi-
sunset"></i> //-->
                            <!-- Dark Logo icon -->
                            <img src="{{ asset('image/icon_univ_bsi.png') }}"
alt="homepage" class="light-logo" />

                        </b>
                        <!--End Logo icon -->
                        <!-- Logo text -->
                        <span class="logo-text">
                            <!-- dark Logo text -->
                            <img src="{{ asset('image/logo_text.png') }}" alt="homepage"
class="light-logo" />

                        </span>
                        <!-- Logo icon -->
                        <!-- <b class="logo-icon"> -->
                        <!--You can put here icon as well // <i class="wi wi-sunset"></i>
//-->
                        <!-- Dark Logo icon -->
                        <!-- <img src="assets/images/logo-text.png" alt="homepage"
class="light-logo" /> -->
```

```
                        <!-- </b> -->
                        <!--End Logo icon -->
                    </a>
                    <!-- ============================================================== -->
                    <!-- End Logo -->
                    <!-- ============================================================== -->
                    <!-- ============================================================== -->
                    <!-- Toggle which is visible on mobile only -->
                    <!-- ============================================================== -->
                    <a class="topbartoggler d-block d-md-none waves-effect waves-light"
href="javascript:void(0)" data-toggle="collapse" data-target="#navbarSupportedContent"
aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation"><i class="ti-more"></i></a>
                </div>
                <!-- ============================================================== -->
                <!-- End Logo -->
                <!-- ============================================================== -->
                <div class="navbar-collapse collapse" id="navbarSupportedContent" data-
navbarbg="skin5">
                    <!-- ============================================================== -->
                    <!-- toggle and nav items -->
                    <!-- ============================================================== -->
                    <ul class="navbar-nav float-left mr-auto">
                        <li class="nav-item d-none d-md-block"><a class="nav-link
sidebartoggler waves-effect waves-light" href="javascript:void(0)" data-sidebartype="mini-
sidebar"><i class="mdi mdi-menu font-24"></i></a></li>
                        <!-- ============================================================
-->
                        <!-- create new -->
                        <!-- ============================================================
-->

                        <!-- ============================================================
-->
                        <!-- Search -->
                        <!-- ============================================================
-->

                    </ul>
                    <!-- ============================================================== -->
                    <!-- Right side toggle and nav items -->
                    <!-- ============================================================== -->
                    <ul class="navbar-nav float-right">
                        <!-- ============================================================
-->
                        <!-- Comment -->
                        <!-- ============================================================
-->

                        <!-- ============================================================
-->
                        <!-- End Comment -->
                        <!-- ============================================================
-->
                        <!-- ============================================================
-->
                        <!-- Messages -->
                        <!-- ============================================================
-->

                        <!-- ============================================================
-->
                        <!-- End Messages -->
                        <!-- ============================================================
-->
```

```
                        <!-- ============================================================
-->
                        <!-- User profile and search -->
                        <!-- ============================================================
-->
                        <li class="nav-item dropdown">
                            <a class="nav-link dropdown-toggle text-muted waves-effect
waves-dark pro-pic" href="" data-toggle="dropdown" aria-haspopup="true" aria-
expanded="false">
                                @if (Auth::user()->foto)
                                <img src="{{ asset('storage/img-user/' . Auth::user()-
>foto) }}" alt="user" class="rounded-circle" width="31">
                                @else
                                <img src="{{ asset('storage/img-user/img-default.jpg') }}"
alt="user" class="rounded-circle" width="31">
                                @endif
                            </a>
                            <div class="dropdown-menu dropdown-menu-right user-dd
animated">
                                <a class="dropdown-item" href="{{
route('backend.user.edit', Auth::user()->id) }}"><i class="ti-user m-r-5 m-l-5"></i> Profil
Saya</a>
                                <a class="dropdown-item" href=""
onclick="event.preventDefault(); document.getElementById('keluar-app').submit();"><i
class="fa fa-power-off m-r-5 m-l-5"></i> Keluar</a>
                                <div class="dropdown-divider"></div>

                            </div>
                        </li>
                        <!-- ============================================================
-->
                        <!-- User profile and search -->
                        <!-- ============================================================
-->
                    </ul>
                </div>
            </nav>
        </header>
        <!-- ============================================================== -->
        <!-- End Topbar header -->
        <!-- ============================================================== -->
        <!-- ============================================================== -->
        <!-- Left Sidebar - style you can find in sidebar.scss  -->
        <!-- ============================================================== -->
        <aside class="left-sidebar" data-sidebarbg="skin5">
            <!-- Sidebar scroll-->
            <div class="scroll-sidebar">
                <!-- Sidebar navigation-->
                <nav class="sidebar-nav">
                    <ul id="sidebarnav" class="p-t-30">
                        <li class="sidebar-item"> <a class="sidebar-link waves-effect
waves-dark sidebar-link" href="{{ route('backend.beranda') }}" aria-expanded="false"><i
class="mdi mdi-view-dashboard"></i><span class="hide-menu">Beranda</span></a>
                        </li>
                        <li class="sidebar-item"> <a class="sidebar-link waves-effect
waves-dark sidebar-link" href="{{ route('backend.user.index') }}" aria-expanded="false"><i
class="mdi mdi-account"></i><span class="hide-menu">User</span></a>
                        </li>
                        <li class="sidebar-item"> <a class="sidebar-link has-arrow waves-
effect waves-dark" href="javascript:void(0)" aria-expanded="false"><i class="mdi mdi-
shopping"></i><span class="hide-menu">Data Produk </span></a>
                            <ul aria-expanded="false" class="collapse  first-level">
                                <li class="sidebar-item"><a href="icon-material.html"
class="sidebar-link"><i class="mdi mdi-chevron-right"></i><span class="hide-menu"> Kategori
</span></a>
                                </li>
```

```
                                <li class="sidebar-item"><a href="icon-fontawesome.html"
class="sidebar-link"><i class="mdi mdi-chevron-right"></i><span class="hide-menu"> Produk
</span></a>
                                </li>
                            </ul>
                        </li>

                    </ul>
                </nav>
                <!-- End Sidebar navigation -->
            </div>
            <!-- End Sidebar scroll-->
        </aside>
        <!-- ============================================================== -->
        <!-- End Left Sidebar - style you can find in sidebar.scss  -->
        <!-- ============================================================== -->
        <!-- ============================================================== -->
        <!-- Page wrapper  -->
        <!-- ============================================================== -->
        <div class="page-wrapper">
            <!-- ============================================================== -->
            <!-- Bread crumb and right sidebar toggle -->
            <!-- ============================================================== -->

            <!-- ============================================================== -->
            <!-- End Bread crumb and right sidebar toggle -->
            <!-- ============================================================== -->
            <!-- ============================================================== -->
            <!-- Container fluid  -->
            <!-- ============================================================== -->
            <div class="container-fluid">
                <!-- ============================================================== -->
                <!-- Start Page Content -->
                <!-- ============================================================== -->

                <!-- @yieldAwal -->
                @yield('content')
                <!-- @yieldAkhir-->

                <!-- ============================================================== -->
                <!-- End PAge Content -->
                <!-- ============================================================== -->
                <!-- ============================================================== -->
                <!-- Right sidebar -->
                <!-- ============================================================== -->
                <!-- .right-sidebar -->
                <!-- ============================================================== -->
                <!-- End Right sidebar -->
                <!-- ============================================================== -->
            </div>
            <!-- ============================================================== -->
            <!-- End Container fluid  -->
            <!-- ============================================================== -->
            <!-- ============================================================== -->
            <!-- footer -->
            <!-- ============================================================== -->
            <footer class="footer text-center">
                Web Programming. Studi Kasus Toko Online <a
href="https://bsi.ac.id/">Kuliah..? BSI Aja !!!</a>
            </footer>
            <!-- ============================================================== -->
            <!-- End footer -->
            <!-- ============================================================== -->
        </div>
        <!-- ============================================================== -->
        <!-- End Page wrapper  -->
```

```
        <!-- ============================================================== -->
    </div>
    <!-- ============================================================== -->
    <!-- End Wrapper -->
    <!-- ============================================================== -->
    <!-- ============================================================== -->
    <!-- All Jquery -->
    <!-- ============================================================== -->
    <script src="{{ asset('backend/libs/jquery/dist/jquery.min.js') }}"></script>
    <!-- Bootstrap tether Core JavaScript -->
    <script src="{{ asset('backend/libs/popper.js/dist/umd/popper.min.js') }}"></script>
    <script src="{{ asset('backend/libs/bootstrap/dist/js/bootstrap.min.js') }}"></script>
    <!-- slimscrollbar scrollbar JavaScript -->
    <script src="{{ asset('backend/libs/perfect-scrollbar/dist/perfect-
scrollbar.jquery.min.js') }}"></script>
    <script src="{{ asset('backend/extra-libs/sparkline/sparkline.js') }}"></script>
    <!--Wave Effects -->
    <script src="{{ asset('backend/dist/js/waves.js') }}"></script>
    <!--Menu sidebar -->
    <script src="{{ asset('backend/dist/js/sidebarmenu.js') }}"></script>
    <!--Custom JavaScript -->
    <script src="{{ asset('backend/dist/js/custom.min.js') }}"></script>
    <!-- this page js -->
    <script src="{{ asset('backend/extra-libs/multicheck/datatable-checkbox-init.js')
}}"></script>
    <script src="{{ asset('backend/extra-libs/multicheck/jquery.multicheck.js')
}}"></script>
    <script src="{{ asset('backend/extra-libs/DataTables/datatables.min.js') }}"></script>
    <script>
        /***************************************
         *       Basic Table                   *
         ***************************************/
        $('#zero_config').DataTable();
    </script>

    <!-- form keluar app -->
    <form id="keluar-app" action="{{ route('backend.logout') }}" method="POST" class="d-
none">
        @csrf
    </form>
    <!-- form keluar app end -->

    <!-- sweetalert -->
    <script src="{{ asset('sweetalert/sweetalert2.all.min.js') }}"></script>
    <!-- sweetalert End -->
    <!-- konfirmasi success-->
    @if (session('success'))
    <script>
        Swal.fire({
            icon: 'success',
            title: 'Berhasil!',
            text: "{{ session('success') }}"
        });
    </script>
    @endif
    <!-- konfirmasi success End-->
    <script type="text/javascript">
        //Konfirmasi delete
        $('.show_confirm').click(function(event) {
            var form = $(this).closest("form");
            var konfdelete = $(this).data("konf-delete");
            event.preventDefault();
            Swal.fire({
                title: 'Konfirmasi Hapus Data?',
                html: "Data yang dihapus <strong>" + konfdelete + "</strong> tidak dapat
dikembalikan!",
```

```
                icon: 'warning',
                showCancelButton: true,
                confirmButtonColor: '#3085d6',
                cancelButtonColor: '#d33',
                confirmButtonText: 'Ya, dihapus',
                cancelButtonText: 'Batal'
            }).then((result) => {
                if (result.isConfirmed) {
                    Swal.fire('Terhapus!', 'Data berhasil dihapus.', 'success')
                        .then(() => {
                            form.submit();
                        });
                }
            });
        });
    </script>
</body>

</html>
```

Latihan Mandiri 7:

Update Portofolion sertifikasi kompetensi, Impentasikan Unit Kompetensi Software Development pada skema Menulis Kode Dengan Prinsip Sesuai Guidelines dan Best Practice, Melakukan Debugging, Membuat Dokumen Kode Program, Menggunakan Library atau Komponen Pre-Existing dan Menerapkan Akses Basis Data

Minggu Ke-9
Manajemen Data Master

Manajemen data master adalah proses pengelolaan data penting yang menjadi acuan utama dalam sebuah sistem, data tersebut akurat, konsisten, dan dapat diandalkan. Dalam studi kasus toko online, manajemen data master dapat mencakup pengelolaan berbagai jenis data seperti:

1. Kategori Produk: Informasi tentang berbagai kategori produk seperti elektronik, pakaian, peralatan rumah tangga, dll.
2. Merek Produk: Informasi tentang merek-merek yang tersedia dalam toko online.
3. Menu: Struktur navigasi atau menu yang digunakan untuk mengelompokkan produk atau layanan.
4. Tipe Produk: Spesifikasi atau jenis produk seperti ukuran, warna, bahan, dll.

Data-data ini digunakan sebagai acuan utama dan dihubungkan (join) dengan data produk dalam sistem untuk memastikan bahwa setiap produk memiliki informasi yang lengkap dan konsisten. Misalnya, setiap produk akan dihubungkan dengan kategori dan merek yang sesuai, sehingga memudahkan dalam pengelolaan dan pencarian produk.

Dengan manajemen data master yang baik, sebuah toko online dapat memastikan bahwa informasi produk yang ditampilkan kepada pelanggan selalu akurat dan up-to-date, serta mendukung berbagai fungsi bisnis seperti pemasaran, penjualan, dan analisis data.

9.1. Mempersiapkan Tabel Kategori

1. Masing dengan *Project* **TokoOnline** dengan database **db_tokoonline**, pada pembahasan pertemuan sebelumnya kita sudah membuat data master manajemen **User**, kali ini kita akan membuat data master **Kategori**, Pertama kita buat migration table kategori

```
php artisan make:migration create_kategori_table
```

```
 /c/Laravel10/TokoOnline
❯ php artisan make:migration create_kategori_table

  INFO  Migration [C:\Laravel10\TokoOnline\database\migrations/2024_08_05_021547_create_kategori_table.php] created successfully.
```

Gambar IX. 1
Migration Tabel Kategori

2. Kemudian, sesuaikan migration tabel kategori dengan Tabel IX.1. Berikut adalah *Blueprint* dari tabel **kategori**. Jalankan migration seperti yang ditunjukkan pada Gambar IX.1.

```
Schema::create('kategori', function (Blueprint $table) {
            $table->id();
            $table->string('nama_kategori');
        });
```

Tabel IX. 1
Tabel Kategori

Fild	Tipe Data	Keterangan
id	Bigint	Primary Key
Nama_kategori	Varchar(255)	

```
/c/Laravel10/TokoOnline                                              ✘ INT
❯ php artisan migrate:fresh --seed

  Dropping all tables ..................................... 316ms DONE

  INFO  Preparing database.

  Creating migration table ................................ 58ms DONE

  INFO  Running migrations.

  2014_10_12_000000_create_user_table ..................... 66ms DONE
  2014_10_12_100000_create_password_reset_tokens_table ..... 7ms DONE
  2019_08_19_000000_create_failed_jobs_table ............... 47ms DONE
  2019_12_14_000001_create_personal_access_tokens_table ..... 44ms DONE
  2024_08_05_021547_create_kategori_table ................... 11ms DONE


  INFO  Seeding database.
```

Gambar IX. 2
Migrate Pada Kategori

9.2. Model Kategori

1. Buat ***model*** dengan nama **Kategori**

```
php artisan make:model Kategori
```

```
/c/Laravel10/TokoOnline
❯ php artisan make:model Kategori

  INFO  Model [C:\Laravel10\TokoOnline\app\Models\Kategori.php] created success
fully.
```

Gambar IX. 3
Model Pada Katagori

2. *Script* lengkap pada ***model*** Kategori sebagai berikut:

```php
<?php

namespace App\Models;

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

class Kategori extends Model
{
    public $timestamps = false;
    protected $table = "kategori";
    // protected $fillable = [nama_kategori];
    protected $guarded = ['id'];
}
```

9.4. Controller Kategori

Buat ***Controller*** dengan nama **KatagoriController** menggunakan **resource**

```
php artisan make:controller KategoriController --resource
```

```
/c/Laravel10/TokoOnline                                          ✘ INT
❯ php artisan make:controller KategoriController --resource

  INFO  Controller [C:\Laravel10\TokoOnline\app\Http\Controllers\KategoriContro
ller.php] created successfully.
```

Gambar IX. 4
Controller Pada KatagoriController

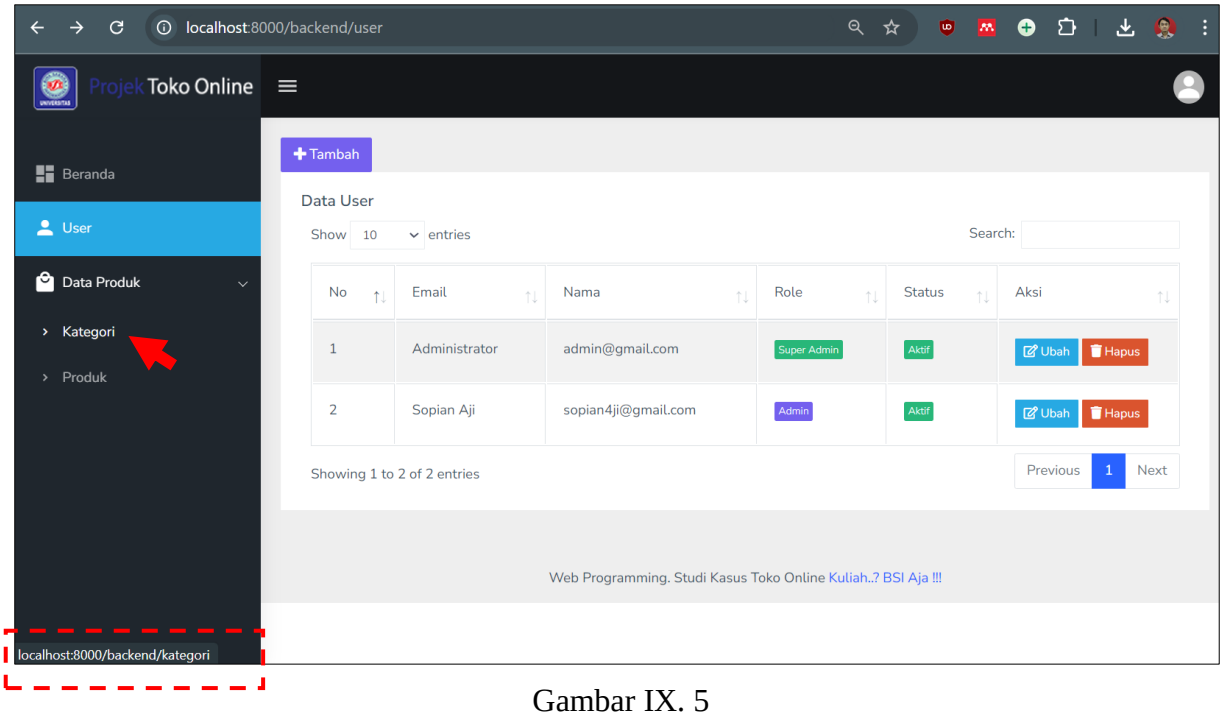
9.5. Konfigurasi Route Pada Kategori

1. Kemudian pada routes**web.php** kita tambahkan *script* sebagai berikut

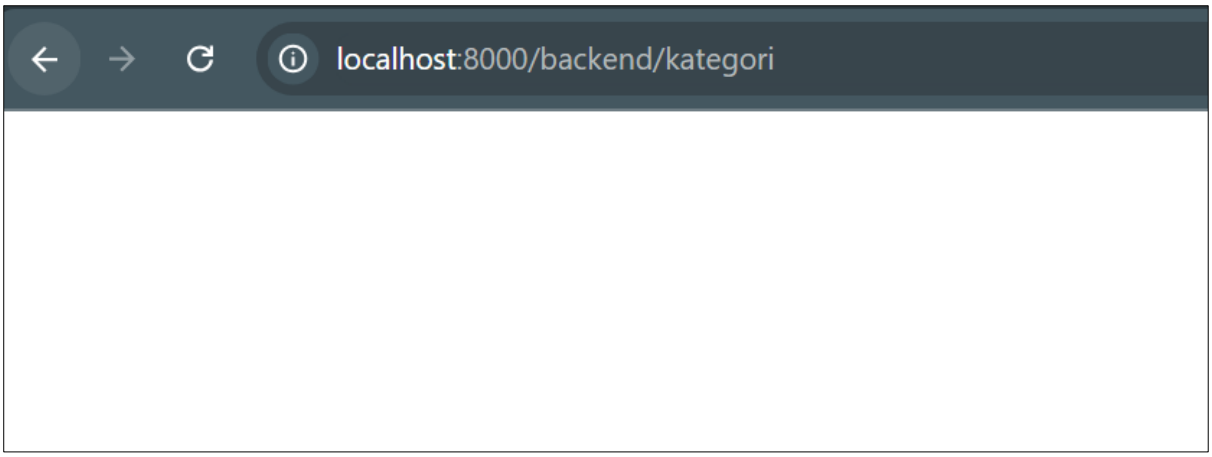
```php
<?php

use App\Http\Controllers\KategoriController;

Route::resource('backend/kategori', KategoriController::class, ['as' => 'backend'])-
>middleware('auth');
```

2. Pada sidebar, kita dapat menambahkan link pada tombol **Kategori** di direktori resources/views/backend/v_layouts/app.blade.php. Jika berhasil, saat tombol Kategori disentuh, link akan menuju http://localhost:8000/backend/kategori seperti pada Gambar IX.5. Namun, jika tombol Kategori diklik, URL akan menuju http://localhost:8000/backend/kategori tetapi halaman kategori akan tampil kosong (putih) seperti pada Gambar IX.6.

```php
 <li class="sidebar-item"><a href="{{ route('backend.kategori.index') }}"
class="sidebar-link"><i class="mdi mdi-chevron-right"></i><span class="hide-
menu"> Kategori </span></a>
</li>
```



Gambar IX. 5
Route Pada Kategori



Gambar IX. 6
URL Kategori

9.5. Halaman Index Kategori

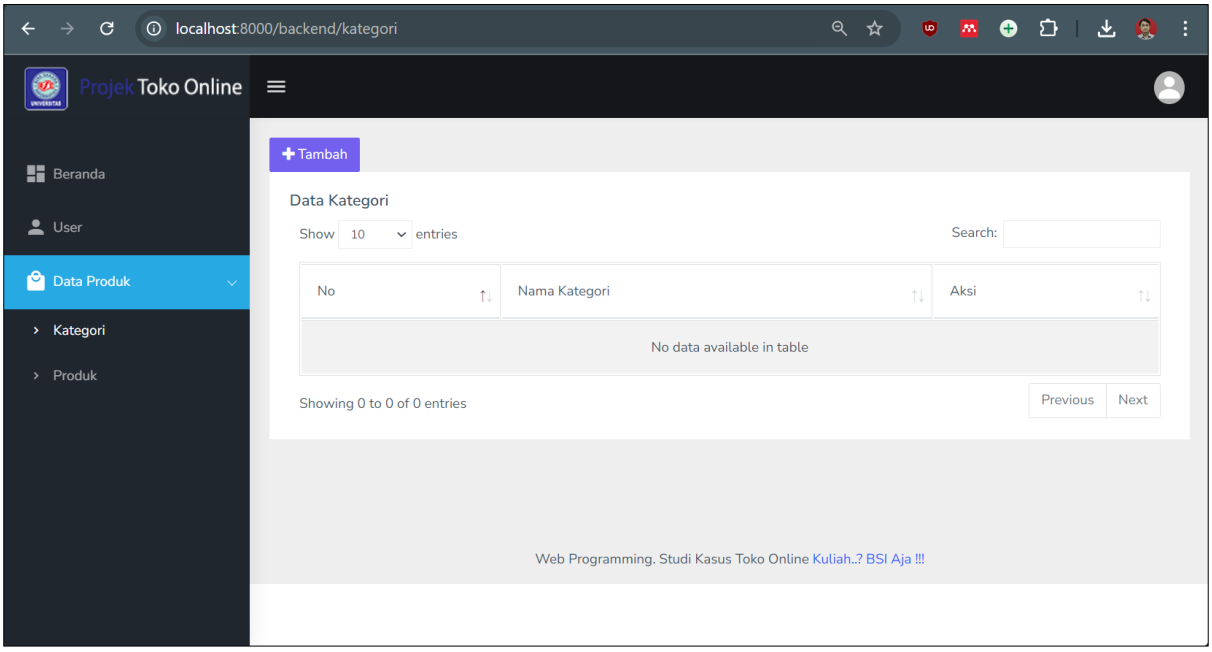
1. Tambahkan *script* pada **KategoriController** dalam *function* **index()** untuk menampilkan data berdasarkan **nama_kategori** secara ascending (asc)

```
use App\Models\Kategori;

class KategoriController extends Controller
{
    public function index()
    {
        $kategori = Kategori::orderBy('nama_kategori', 'asc')->get();
        return view('backend.v_kategori.index', [
            'judul' => 'Kategori',
            'index' => $kategori
        ]);
    }

    <!-- function lainnya-->
}
```

2. Pada view **index.blade.php** di direktori resources\backend\v_kategori, gunakan *script* berikut, jika berhasil maka akan tampil seperti pada gambar IX.7

```
@extends('backend.v_layouts.app')
@section('content')
<!-- contentAwal -->

<div class="row">

    <div class="col-12">
        <a href="{{ route('backend.kategori.create') }}">
            <button type="button" class="btn btn-primary"><i class="fas fa-plus"></i>
Tambah</button>
        </a>
        <div class="card">
            <div class="card-body">
                <h5 class="card-title"> {{$judul}} </h5>
                <div class="table-responsive">
                    <table id="zero_config" class="table table-striped table-bordered">
                        <thead>
                            <tr>
                                <th>No</th>
                                <th>Nama Kategori</th>
                                <th>Aksi</th>
                            </tr>
                        </thead>
```

```
                    <tbody>
                        @foreach ($index as $row)
                        <tr>
                            <td> {{ $loop->iteration }} </td>
                            <td> {{$row->nama_kategori}} </td>
                            <td>
                                <a href="{{ route('backend.kategori.edit', $row->id)
}}" title="Ubah Data">
                                    <button type="button" class="btn btn-cyan btn-
sm"><i class="far fa-edit"></i> Ubah</button>
                                </a>

                                <form method="POST" action="{{
route('backend.kategori.destroy', $row->id) }}" style="display: inline-block;">
                                    @method('delete')
                                    @csrf
                                    <button type="submit" class="btn btn-danger btn-sm
show_confirm" data-konf-delete="{{ $row->nama_kategori}}" title='Hapus Data'>
                                        <i class="fas fa-trash"></i> Hapus</button>
                                </form>
                            </td>
                        </tr>
                        @endforeach
                    </tbody>
                </table>
            </div>

        </div>
    </div>
</div>

<!-- contentAkhir -->
@endsection
```



Gambar IX. 7
Halaman Index Pada Kategori

9.6. Halaman Create Kategori

1. Tambahkan *script* pada **KategoriController** dalam *function* **create()** dan *function* **store()** sebagai berikut

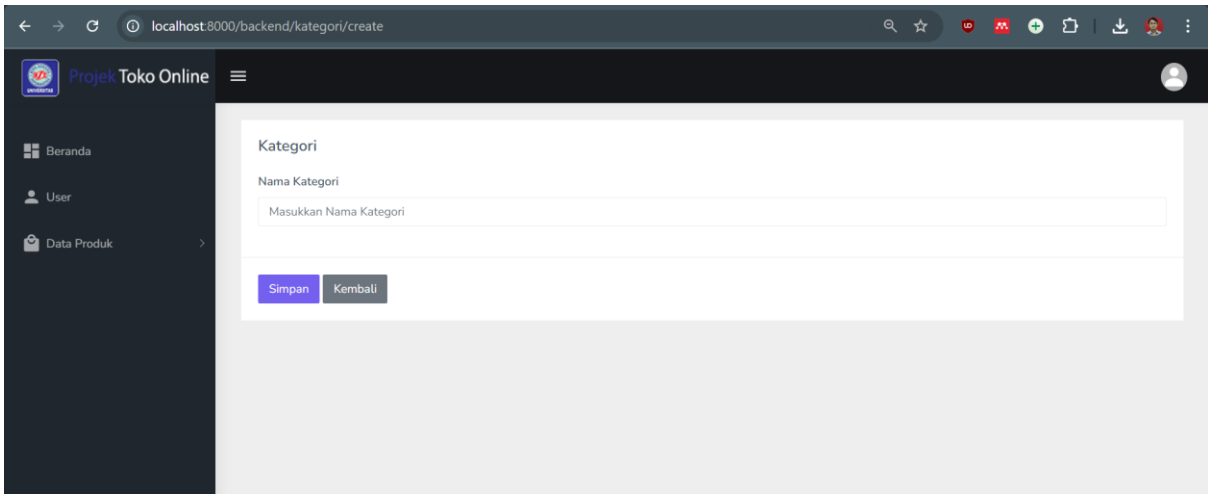
```
use App\Models\Kategori;

class KategoriController extends Controller
{
    <!-- function lainnya-->

    public function create()
    {
        return view('backend.v_kategori.create', [
            'judul' => 'Kategori',
        ]);
    }

    public function store(Request $request)
    {
        // dd($request);
        $validatedData = $request->validate([
            'nama_kategori' => 'required|max:255|unique:kategori ',
        ]);
        Kategori::create($validatedData);
        return redirect()->route('backend.kategori.index')->with('success', 'Data berhasil
tersimpan');
    }

   <!-- function lainnya-->
}
```

unique:kategori berfungsi untuk menyatakan bahwa nilai **nama_kategori** harus **unik** dalam tabel **kategori**. Artinya, tidak boleh ada dua atau lebih baris dalam tabel **kategori** yang memiliki nilai yang sama untuk kolom **nama_kategori**

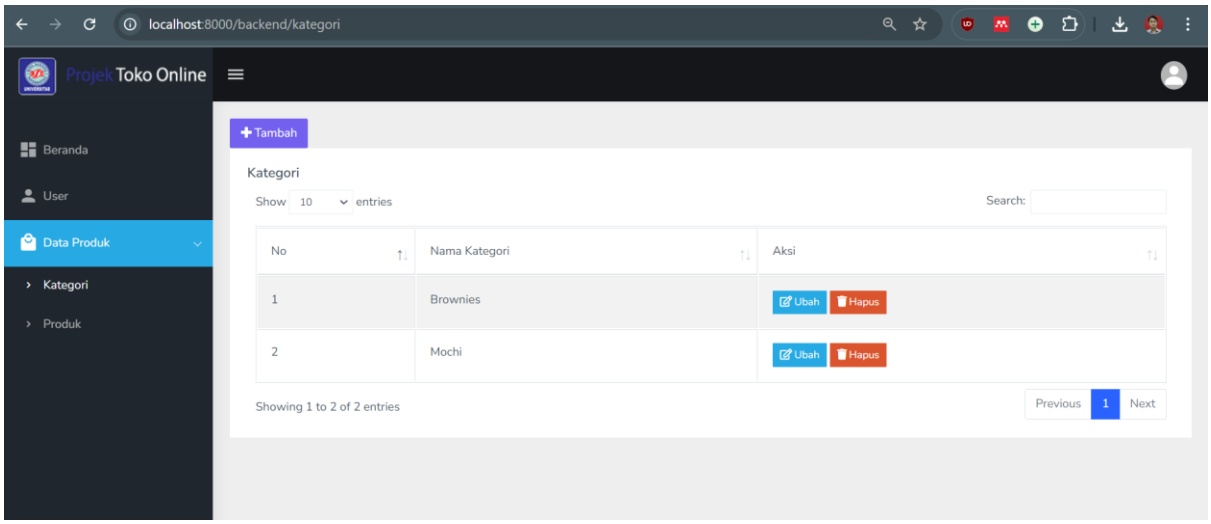
2. Pada view **create.blade.php** di direktori resources\backend\v_kategori, gunakan *script* berikut, untuk form kategori seperti pada gambar IX.8. Klik tombol **simpan** untuk menyimpan data dan jika berhasil tersimpan akan diarahkan ke halaman **index()** Kategori seperti yang ditunjukkan pada Gambar IX.9

```
@extends('backend.v_layouts.app')
@section('content')
<!-- contentAwal -->

<div class="container-fluid">
    <div class="row">
        <div class="col-12">
            <div class="card">
                <form class="form-horizontal" action="{{ route('backend.kategori.store')
}}" method="post">
                    @csrf

                    <div class="card-body">
                        <h4 class="card-title"> {{$judul}} </h4>

                        <div class="form-group">
                            <label>Nama Kategori</label>
                            <input type="text" name="nama_kategori" value="{{
old('nama_kategori') }}" class="form-control @error('nama_kategori') is-invalid @enderror"
placeholder="Masukkan Nama Kategori">
                            @error('nama_kategori')
                            <span class="invalid-feedback alert-danger" role="alert">
                                {{ $message }}
```

```
                            </span>
                            @enderror
                        </div>

                    </div>
                    <div class="border-top">
                        <div class="card-body">
                            <button type="submit" class="btn btn-primary">Simpan</button>
                            <a href="{{ route('backend.kategori.index') }}">
                                <button type="button" class="btn btn-
secondary">Kembali</button>
                            </a>
                        </div>
                    </div>
                </form>
            </div>
        </div>
    </div>
</div>

<!-- contentAkhir -->
@endsection
```



Gambar IX. 8
Form Create Kategori

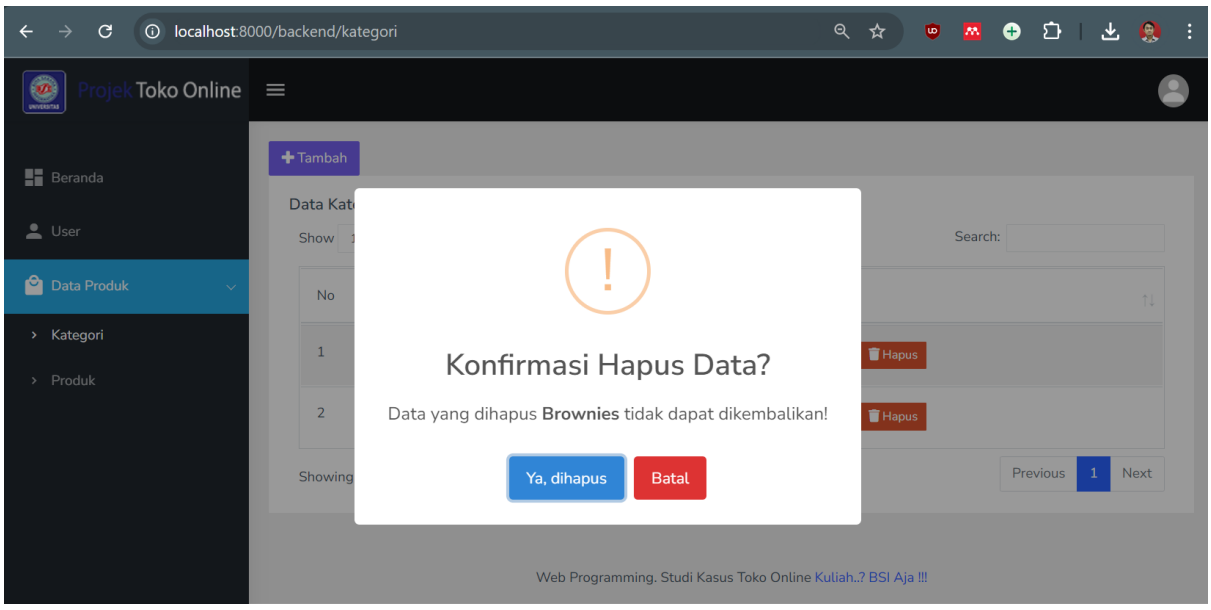


Gambar IX. 9
Halaman Index Dengan Menampilkan Data Kategori

9.5. Halaman Destroy Kategori

Untuk aksi Hapus pada direktori resources\backend\v_kategori\index.blade.php sudah tersedia, kita hanya tinggal menambahkan *script* pada **KategoriController** dengan *function* **destroy()** sebagai berikut & jika diklik tombol **Hapus** maka akan muncul konfirmasi seperti pada gambar IX.10

```
use App\Models\Kategori;

class KategoriController extends Controller
{
    <!-- function lainnya-->

    public function destroy(string $id)
    {
        $kategori = kategori::findOrFail($id);
        $kategori ->delete();
        return redirect()->route('backend.kategori.index')->with('success', 'Data berhasil
dihapus');
    }

    <!-- function lainnya-->
}
```



Gambar IX. 10
Konfimasi Hapus Data Pada Kategori

9.6. Halaman Edit Kategori

1. Berikutnya, kita akan melakukan perubahan data. Untuk mengubah data, kita tambahkan *script* pada *function* **edit()** dan **update()** di **KategoriController**.

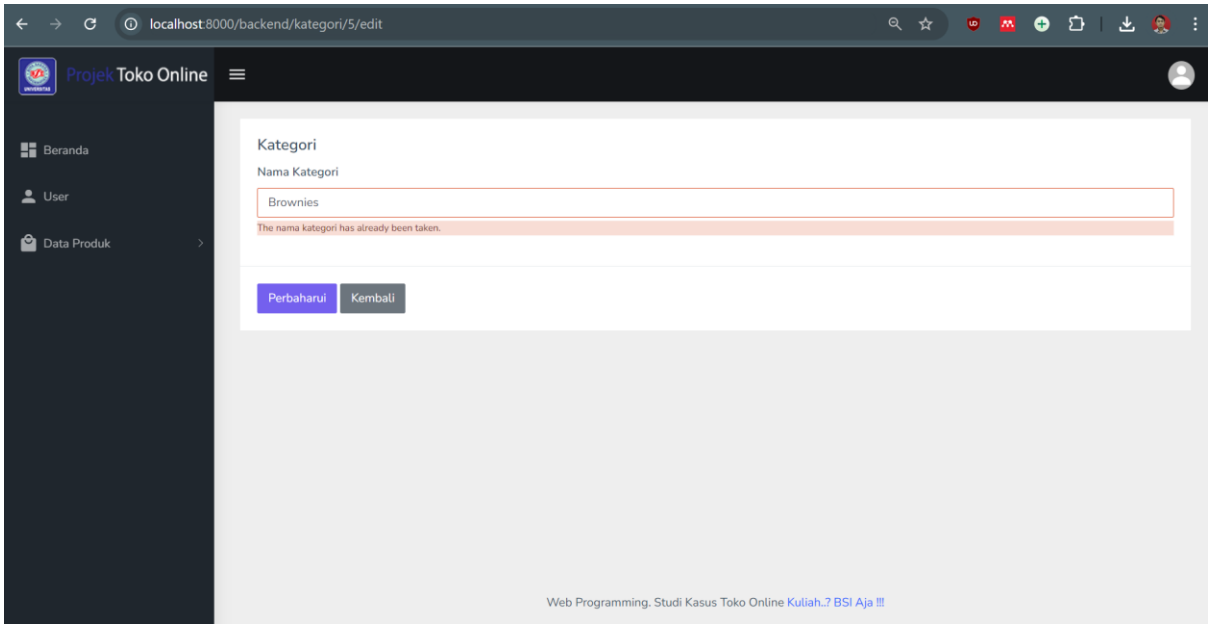
```
use App\Models\Kategori;

class KategoriController extends Controller
{
     <!-- function lainnya-->
     public function edit(string $id)
    {
        $kategori = Kategori::find($id);
        return view('backend.v_kategori.edit', [
            'judul' => 'Kategori',
            'edit' => $kategori
        ]);
```

```
    }

    /**
     * Update the specified resource in storage.
     */
    public function update(Request $request, string $id)
    {
        $rules = [
            'nama_kategori' => 'required|max:255|unique:kategori,nama_kategori,' . $id,
        ];
        $validatedData = $request->validate($rules);
        Kategori::where('id', $id)->update($validatedData);
        return redirect()->route('backend.kategori.index')->with('success', 'Data berhasil
diperbaharui');
    }

    <!-- function lainnya-->
}
```

2. Untuk view **edit.blade.php** di direktori resources/backend/v_kategori, gunakan *script* berikut. Misalnya, kita akan mengubah data seperti yang terlihat pada Gambar IX.9, yang terdiri dari dua record: Brownies dan Mochi. Jika kita mengubah Mochi menjadi Brownies, akan muncul pesan peringatan seperti pada Gambar IX.11. Hal ini terjadi karena Brownies sudah tersedia. Sebaiknya, ubahlah Mochi menjadi nama lain, misalnya Mochi Kacang atau Mochi Coklat.

```
@extends('backend.v_layouts.app')
@section('content')
<!-- contentAwal -->

<div class="container-fluid">
    <div class="row">
        <div class="col-12">
            <div class="card">
                <form action="{{ route('backend.kategori.update', $edit->id) }}"
method="post">
                    @method('put')
                    @csrf

                    <div class="card-body">
                        <h4 class="card-title"> {{$judul}} </h4>

                        <div class="form-group">
                            <label>Nama Kategori</label>
                            <input type="text" name="nama_kategori" value="{{
old('nama_kategori', $edit->nama_kategori) }}" class="form-control @error('nama_kategori')
is-invalid @enderror" placeholder="Masukkan Nama Kategori">
                            @error('nama_kategori')
                            <span class="invalid-feedback alert-danger" role="alert">
                                {{ $message }}
                            </span>
                            @enderror
                        </div>

                    </div>
                    <div class="border-top">
                        <div class="card-body">
                            <button type="submit" class="btn btn-
primary">Perbaharui</button>
                            <a href="{{ route('backend.kategori.index') }}">
                                <button type="button" class="btn btn-
secondary">Kembali</button>
                            </a>
                        </div>
```

```
                    </div>
                </form>
            </div>
        </div>
    </div>
</div>

<!-- contentAkhir -->
@endsection
```



Gambar IX. 11
Form Edit Kategori

Berikut *script* lengkap pada controller di **KategoriController** sebagai berikut:

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Kategori;

class KategoriController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        $kategori = Kategori::orderBy('nama_kategori', 'asc')->get();
        return view('backend.v_kategori.index', [
            'judul' => 'Kategori',
            'index' => $kategori
        ]);
    }

    /**
     * Show the form for creating a new resource.
     */
    public function create()
    {
        return view('backend.v_kategori.create', [
            'judul' => 'Kategori',
```

```
        ]);
    }

    /**
     * Store a newly created resource in storage.
     */
    public function store(Request $request)
    {
        // dd($request);
        $validatedData = $request->validate([
            'nama_kategori' => 'required|max:255|unique:kategori',
        ]);
        Kategori::create($validatedData);
        return redirect()->route('backend.kategori.index')->with('success', 'Data berhasil
tersimpan');
    }

    /**
     * Display the specified resource.
     */
    public function show(string $id)
    {
        //
    }

    /**
     * Show the form for editing the specified resource.
     */
    public function edit(string $id)
    {
        $kategori = Kategori::find($id);
        return view('backend.v_kategori.edit', [
            'judul' => 'Kategori',
            'edit' => $kategori
        ]);
    }

    /**
     * Update the specified resource in storage.
     */
    public function update(Request $request, string $id)
    {
        $rules = [
            'nama_kategori' => 'required|max:255|unique:kategori,nama_kategori,' . $id,
        ];
        $validatedData = $request->validate($rules);
        Kategori::where('id', $id)->update($validatedData);
        return redirect()->route('backend.kategori.index')->with('success', 'Data berhasil
diperbaharui');
    }

    /**
     * Remove the specified resource from storage.
     */
    public function destroy(string $id)
    {
        $user = kategori::findOrFail($id);
        $user->delete();
        return redirect()->route('backend.kategori.index')->with('success', 'Data berhasil
dihapus');
    }
}
```

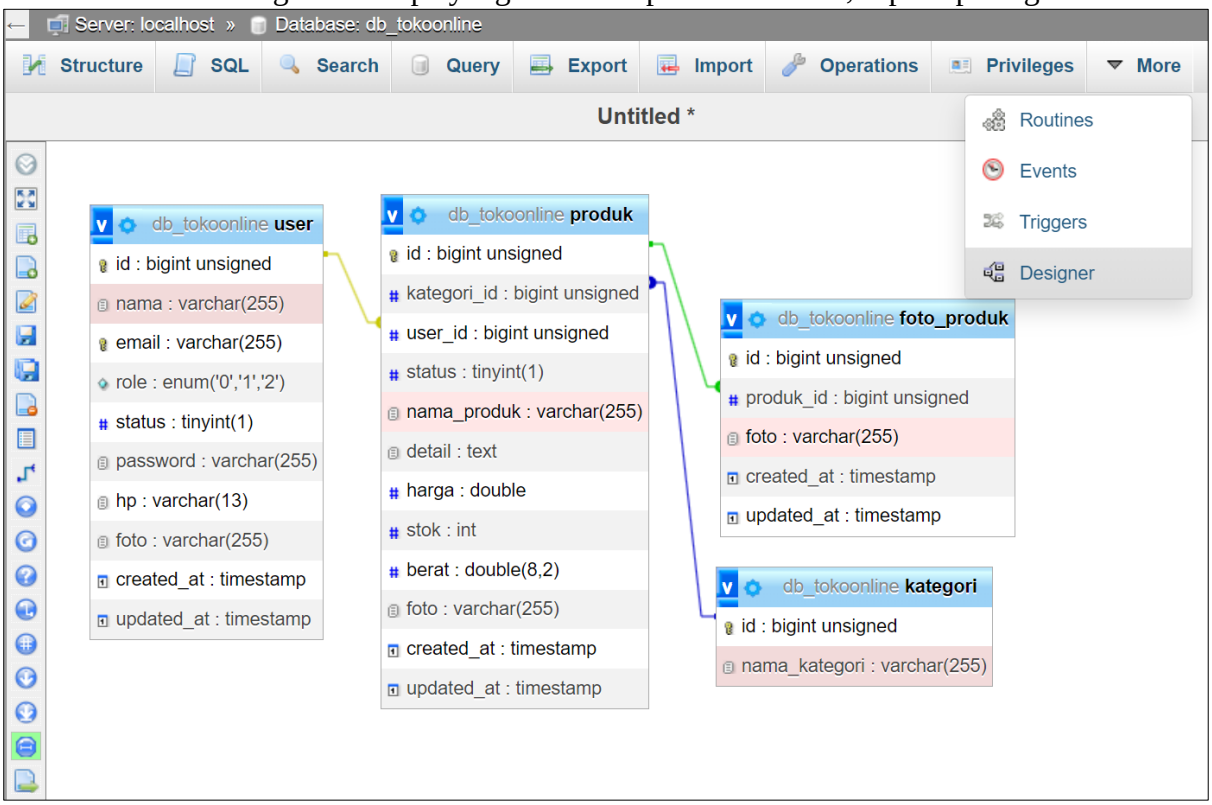
Berikut *script* lengkap routes**web.php** sampai dengan controller pada **KategoriController**

```php
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\BerandaController;
use App\Http\Controllers\LoginController;
use App\Http\Controllers\UserController;
use App\Http\Controllers\KategoriController;

Route::get('/', function () {
    // return view('welcome');
    return redirect()->route('backend.login');
});
Route::get('backend/beranda', [BerandaController::class, 'berandaBackend'])-
>name('backend.beranda')->middleware('auth');

Route::get('backend/login', [LoginController::class, 'loginBackend'])-
>name('backend.login');
Route::post('backend/login', [LoginController::class, 'authenticateBackend'])-
>name('backend.login');
Route::post('backend/logout', [LoginController::class, 'logoutBackend'])-
>name('backend.logout');

// Route::resource('backend/user', UserController::class)->middleware('auth');
Route::resource('backend/user', UserController::class, ['as' => 'backend'])-
>middleware('auth');
Route::resource('backend/kategori', KategoriController::class, ['as' => 'backend'])-
>middleware('auth');
```

Latihan Mandiri 9:
Portofolion sertifikasi kompetensi, Impentasikan Unit Kompetensi Software Development pada **Mengimplementasikan pemrograman berorientasi objek**.

Minggu Ke-10
Data Join Tabel Part 1

Data Join Tabel di Laravel 10 adalah teknik yang digunakan untuk menggabungkan data dari dua atau lebih tabel dalam basis data berdasarkan kolom yang terkait antara tabel-tabel tersebut. Berikut adalah rancangan LRS (Logical Record Structure) untuk studi kasus toko online pada Modul Web Programming II. Dalam rancangan ini, terjadi **join** antara tabel **produk** dengan tabel **kategori** untuk menentukan kategori dari produk, dan tabel **produk** juga join dengan tabel **user** untuk mengetahui siapa yang membuat produk tersebut, seperti pada gambar X.1



Gambar X. 1
LRS Toko Online Pada Web Programming II

10.1. Validasi Menggunakan Bahasa Indonesia

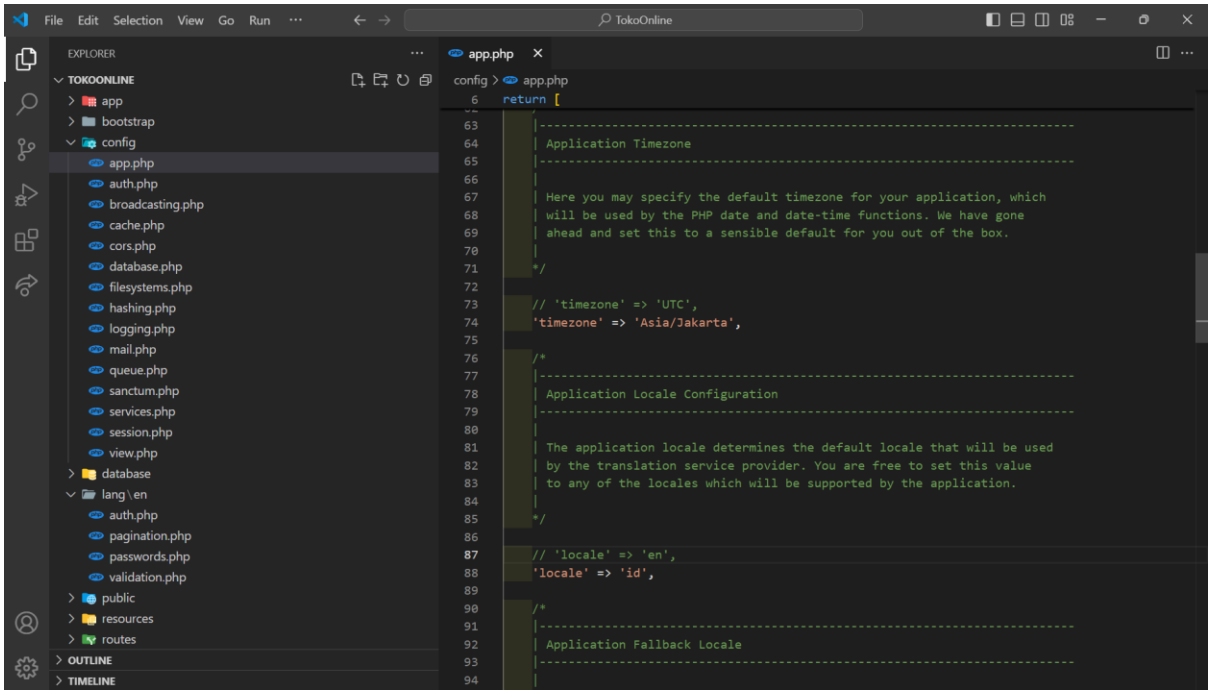
1. Sebelum kita masuk kepembahasan manajeman data master produk, kita konfigurasi validasi dengan menggunakan bahasa Indonesia, pada terminal kita ketikkan

```
php artisan lang:publish
```

```
/c/Laravel10/TokoOnline
php artisan lang:publish

  INFO  Language files published successfully.
```

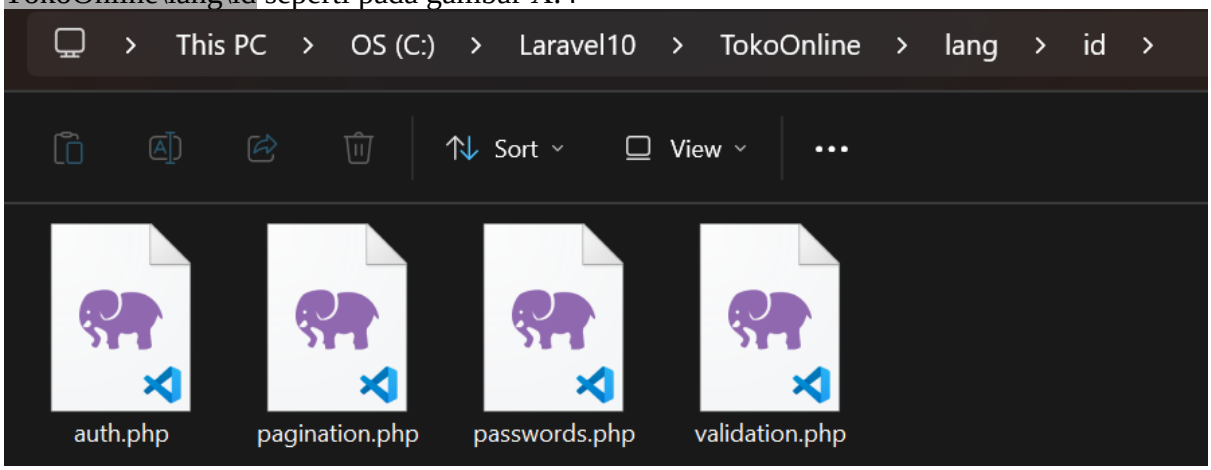
Gambar X. 2
Artisan Lang

2. Pada file config/**app.php**, ubah baris script 'locale' => 'en' menjadi 'locale' => 'id'. Kemudian, ubah waktu menjadi waktu Asia dengan mengubah baris script 'timezone' => 'UTC' menjadi 'timezone' => 'Asia/Jakarta', seperti yang ditunjukkan pada gambar X.3



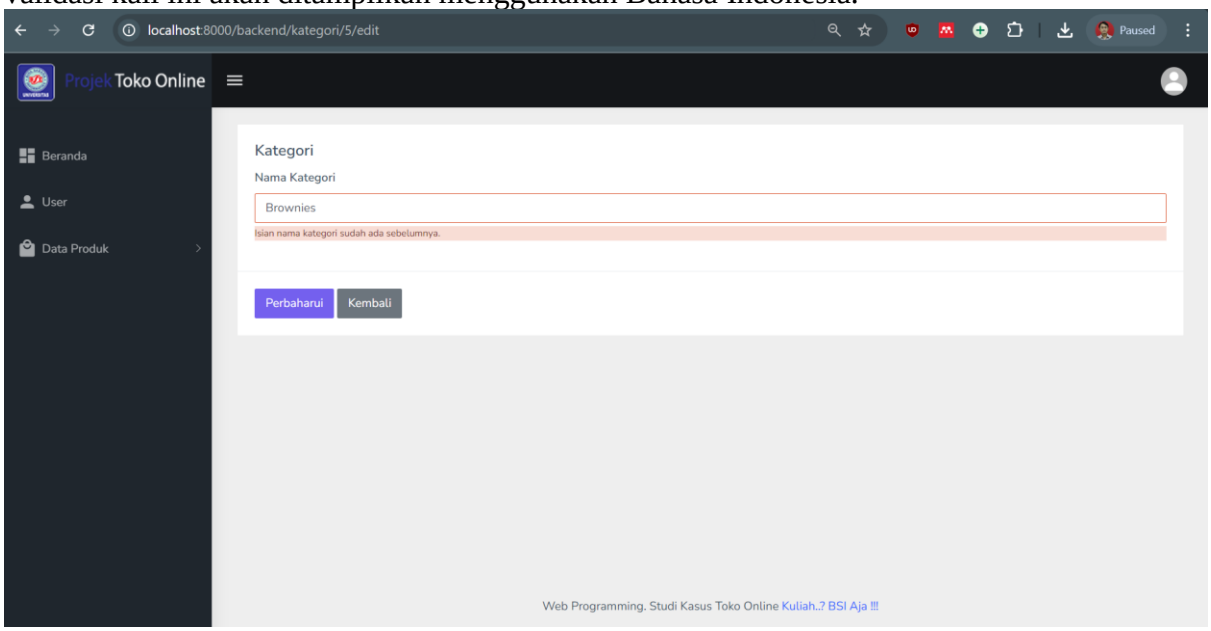
Gambar X. 3
konfigurasi Bahasa & Waktu Indonesia

3. Dapatkan file **id.zip** pada https://bit.ly/LaravelWebPro2 kemudian extrak ke direktori TokoOnline\lang\id seperti pada gambar X.4



Gambar X. 4
Folder Bahasa id (Indonesia)

4. Refresh form, misalnya pada halaman edit, dan ubah data dengan nilai yang sama. Maka validasi kali ini akan ditampilkan menggunakan Bahasa Indonesia.



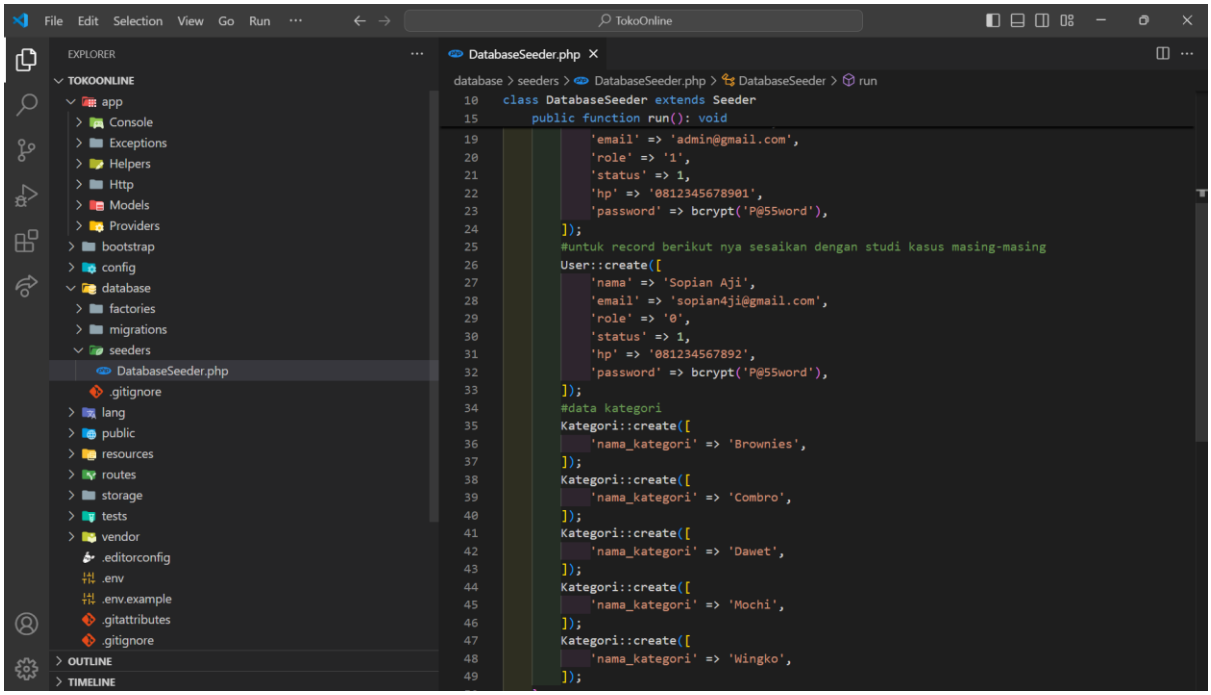
Gambar X. 5
Validasi Menggunakan Bahasa Indonesia

10.2. Seeder Kategori

Untuk membantu dalam mengisi record data master, seperti kategori, yang merupakan data tetap, kita bisa menggunakan seeder. Dengan demikian, saat kita mengubah relasi tabel, record tidak akan hilang dan tidak perlu diinput ulang melalui form. Buka file **DatabaseSeeder.php** pada direktori database/seeders seperti pada gambar IX.12 & berikut *script* lengkap *DatabaseSeeder.php*

```php
<?php

namespace Database\Seeders;

// use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;
use App\Models\User;
use App\Models\Kategori;

class DatabaseSeeder extends Seeder
{
    /**
     * Seed the application's database.
     */
    public function run(): void
    {
        User::create([
            'nama' => 'Administrator',
            'email' => 'admin@gmail.com',
            'role' => '1',
            'status' => 1,
            'hp' => '0812345678901',
            'password' => bcrypt('P@55word'),
        ]);
        #untuk record berikut nya sesaikan dengan studi kasus masing-masing
        User::create([
            'nama' => 'Sopian Aji',
            'email' => 'sopian4ji@gmail.com',
            'role' => '0',
            'status' => 1,
```

```
            'hp' => '081234567892',
            'password' => bcrypt('P@55word'),
        ]);
        #data kategori
        Kategori::create([
            'nama_kategori' => 'Brownies',
        ]);
        Kategori::create([
            'nama_kategori' => 'Combro',
        ]);
        Kategori::create([
            'nama_kategori' => 'Dawet',
        ]);
        Kategori::create([
            'nama_kategori' => 'Mochi',
        ]);
        Kategori::create([
            'nama_kategori' => 'Wingko',
        ]);
    }
}
```



Gambar X. 6
Seeder Kategori

Kemudian jalankan seeder dan cek data Kategori untuk memastikan data yang kita input pada **DatabaseSeeder** telah sesuai.

```
php artisan migrate:fresh --seed
```

10.3. Mempersiapkan Tabel Produk

1. Buat migration tabel **Produk**, ketikan pada terminal sebagai berikut seperti pada gambar X.7

```
php artisan make:migration create_produk_table
```

```
/c/Laravel10/TokoOnline
php artisan make:migration create_produk_table

  INFO  Migration [C:\Laravel10\TokoOnline\database\migrations/2024_08_06_113742_create_produk_table.php] created successfully.
```

Gambar X. 7
Migration Pada Produk

2. Kemudian, sesuaikan migration tabel kategori dengan Tabel X.1. Berikut adalah Blueprint dari tabel produk. Jalankan migration seperti yang ditunjukkan pada Gambar X.6.

```
Schema::create('produk', function (Blueprint $table) {
            $table->id();
            $table->unsignedBigInteger('kategori_id');
            $table->unsignedBigInteger('user_id');
            $table->boolean('status');
            $table->string('nama_produk');
            $table->text('detail');
            $table->double('harga');
            $table->integer('stok');
            $table->float('berat');
            $table->string('foto'); // Thumbnail image
            $table->timestamps();
            $table->foreign('kategori_id')->references('id')->on('kategori');
            $table->foreign('user_id')->references('id')->on('user');
        });
```

Tabel X. 1
Tabel Produk

Fild	Tipe Data	Keterangan
id	Bigint	Primary Key
kategori_id	Bigint	
user_id	Bigint	
status	Tinyint(1)	
Nama_produk	Varchar(255)	
detail	text	
harga	double	
stok	int	
berat	double	
foto	Varchar(255)	
created_at	Timestamp	
updated_at	Timestamp	

```
 /c/Laravel10/TokoOnline
❯ php artisan make:migration create_produk_table

  INFO  Migration [C:\Laravel10\TokoOnline\database\migrations/2024_08_06_113742_create_produk_table.php] created successfully.

 /c/Laravel10/TokoOnline
❯ php artisan migrate:fresh --seed

  Dropping all tables ..................................................... 227ms DONE

  INFO  Preparing database.

  Creating migration table ................................................ 67ms DONE

  INFO  Running migrations.

  2014_10_12_000000_create_user_table ..................................... 91ms DONE
  2014_10_12_100000_create_password_reset_tokens_table ..................... 9ms DONE
  2019_08_19_000000_create_failed_jobs_table ............................... 46ms DONE
  2019_12_14_000001_create_personal_access_tokens_table ................... 39ms DONE
  2024_08_05_021547_create_kategori_table ................................. 10ms DONE
  2024_08_06_113742_create_produk_table ................................... 100ms DONE


  INFO  Seeding database.
```

Gambar X. 8
Migration Table Pada Perubahan Tabel Produk

10.4. Model Produk

1. Membuat model dengan nama **Produk**

```
php artisan make:model Produk
```

```
 /c/Laravel10/TokoOnline
❯ php artisan make:model Produk

  INFO  Model [C:\Laravel10\TokoOnline\app\Models\Produk.php] created successfully.
```

Gambar X. 9
Model Pada Produk

2. *Script* lengkap pada ***model*** Produk sebagai berikut:

```php
<?php

namespace App\Models;

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

class Produk extends Model
{
    public $timestamps = true;
    protected $table = "produk";
    protected $guarded = ['id'];

    public function kategori()
    {
        return $this->belongsTo(Kategori::class);
```

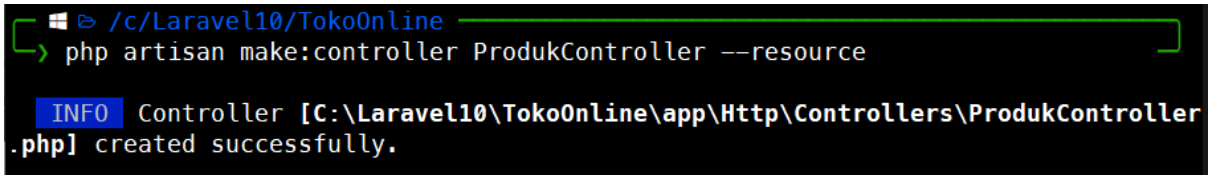
```
    }

    public function user()
    {
        return $this->belongsTo(User::class);
    }
}
```

Pada *script* produk di Laravel, **belongsTo** adalah metode Eloquent yang digunakan untuk mendefinisikan relasi satu-ke-banyak terbalik (inverse one-to-many) antara model. Ini digunakan ketika model saat ini berhubungan dengan satu entitas dari model lain. Dalam hal ini, model **Produk** memiliki relasi dengan model **Kategori** dan **User**, dimana setiap produk hanya memiliki satu kategori dan satu user yang membuatnya.

10.5. Controller Produk

Buat ***Controller*** dengan nama **ProdukController** menggunakan **resource**

```
php artisan make:controller ProdukController --resource
```



Gambar X. 10
Controller Pada ProdukController

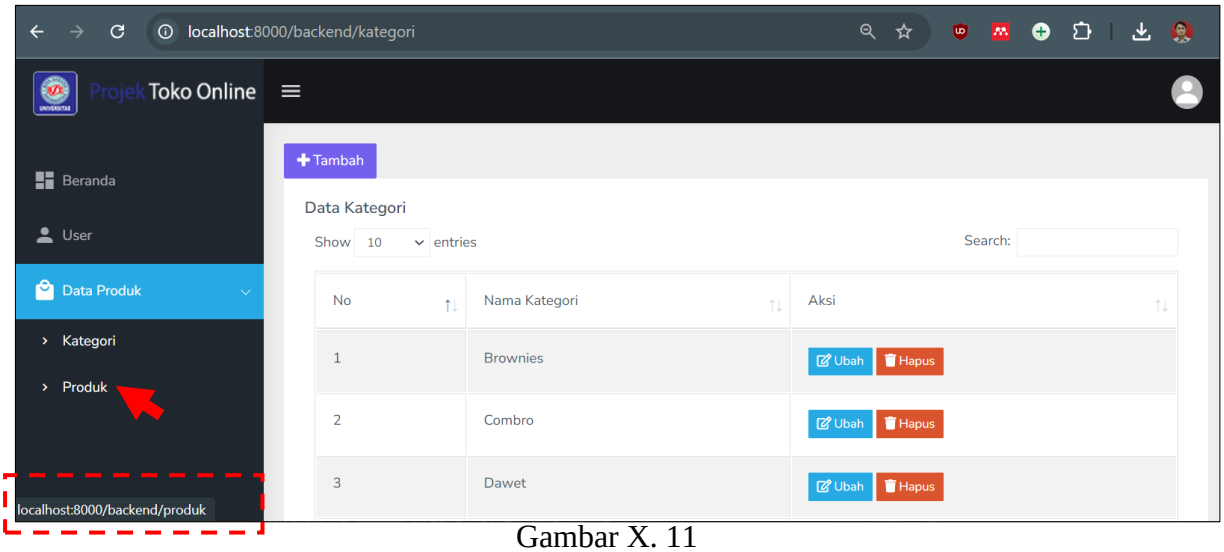
10.6. Konfigurasi Route Pada Produk

1. Kemudian pada routes**web.php** kita tambahkan *script* sebagai berikut

```
<?php

use App\Http\Controllers\ProdukController;

Route::resource('backend/produk', ProdukController::class, ['as' => 'backend'])->middleware('auth');
```

2. Pada sidebar, kita dapat menambahkan link pada tombol **Produk** di direktori resources/views/backend/v_layouts/app.blade.php. Jika berhasil, saat tombol Kategori disentuh, link akan menuju http://localhost:8000/backend/produk seperti pada Gambar IX.5.

```
 <li class="sidebar-item"><a href="{{ route('backend.produk.index') }}" class="sidebar-link"><i class="mdi mdi-chevron-right"></i><span class="hide-menu"> Produk </span></a></li>
```



Gambar X. 11
Route Pada Produk

10.7. Halaman Index Produk

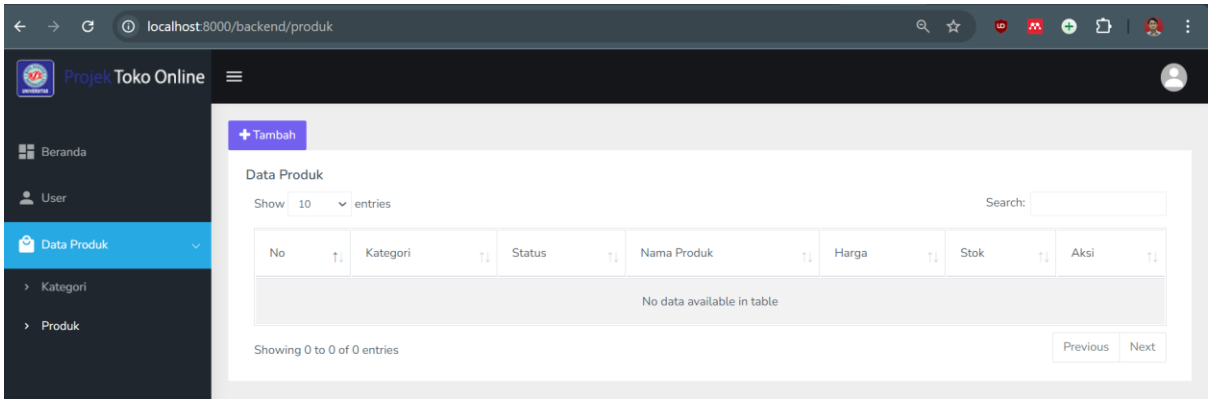
Tambahkan *script* pada **ProdukController** dalam *function* **index()** untuk menampilkan data berdasarkan **updated_at** secara descending (desc). Dengan demikian, data yang terbaru akan berada di posisi atas.

```
<?php

use App\Models\Produk;

class ProdukController extends Controller
{
    public function index()
    {
        $produk = Produk::orderBy('updated_at', 'desc')->get();
        return view('backend.v_produk.index', [
            'judul' => 'Data Produk',
            'index' => $produk
        ]);
    }
    <!-- function lainnya-->
}
```

2. Pada view tambahkan folder v_produk dengan file **index.blade.php** di direktori resources\backend\, gunakan *script* berikut, jika berhasil maka akan tampil seperti pada gambar X.12

```
@extends('backend.v_layouts.app')
@section('content')
<!-- contentAwal -->

<div class="row">

    <div class="col-12">
        <a href="{{ route('backend.produk.create') }}">
            <button type="button" class="btn btn-primary"><i class="fas fa-plus"></i>
Tambah</button>
        </a>
        <div class="card">
            <div class="card-body">
                <h5 class="card-title"> {{$judul}} </h5>
                <div class="table-responsive">
                    <table id="zero_config" class="table table-striped table-bordered">
                        <thead>
```

```
                            <tr>
                                <th>No</th>
                                <th>Kategori</th>
                                <th>Status</th>
                                <th>Nama Produk</th>
                                <th>Harga</th>
                                <th>Stok</th>
                                <th>Aksi</th>
                            </tr>
                        </thead>

                        <tbody>
                            @foreach ($index as $row)
                            <tr>
                                <td> {{ $loop->iteration }}</td>
                                <td> {{ $row->kategori->nama_kategori }} </td>
                                <td>
                                    @if ($row->status ==1)
                                    <span class="badge badge-success"></i>
                                        Publis</span>
                                    @elseif($row->status ==0)
                                    <span class="badge badge-secondary"></i>
                                        Blok</span>
                                    @endif
                                </td>
                                <td> {{ $row->nama_produk }} </td>
                                <td> Rp. {{ number_format($row->harga, 0, ',', '.') }}
</td>
                                <td> {{ $row->stok }} </td>
                                <td>
                                    <a href="{{ route('backend.produk.edit', $row->id) }}"
title="Ubah Data">
                                        <button type="button" class="btn btn-cyan btn-
sm"><i class="far fa-edit"></i> Ubah</button>
                                    </a>

                                    <form method="POST" action="{{
route('backend.produk.destroy', $row->id) }}" style="display: inline-block;">
                                        @method('delete')
                                        @csrf
                                        <button type="submit" class="btn btn-danger btn-sm
show_confirm" data-konf-delete="{{ $row->nama }}" title='Hapus Data'>
                                            <i class="fas fa-trash"></i> Hapus</button>
                                    </form>
                                </td>
                            </tr>
                            @endforeach
                        </tbody>
                    </table>
                </div>

            </div>
        </div>
    </div>
</div>

<!-- contentAkhir -->
@endsection
```



Gambar X. 12
Halaman Index Produk

10.8. Halaman Create Produk

1. Tambahkan *script* pada **ProdukController** dalam *function* **create()** dan *function* **store()** sebagai berikut:

```
use App\Models\Produk;
use App\Models\Kategori;

    public function create()
    {
        $kategori = Kategori::orderBy('nama_kategori', 'asc')->get();
        return view('backend.v_produk.create', [
            'judul' => 'Tambah Produk',
            'kategori' => $kategori
        ]);
    }
```

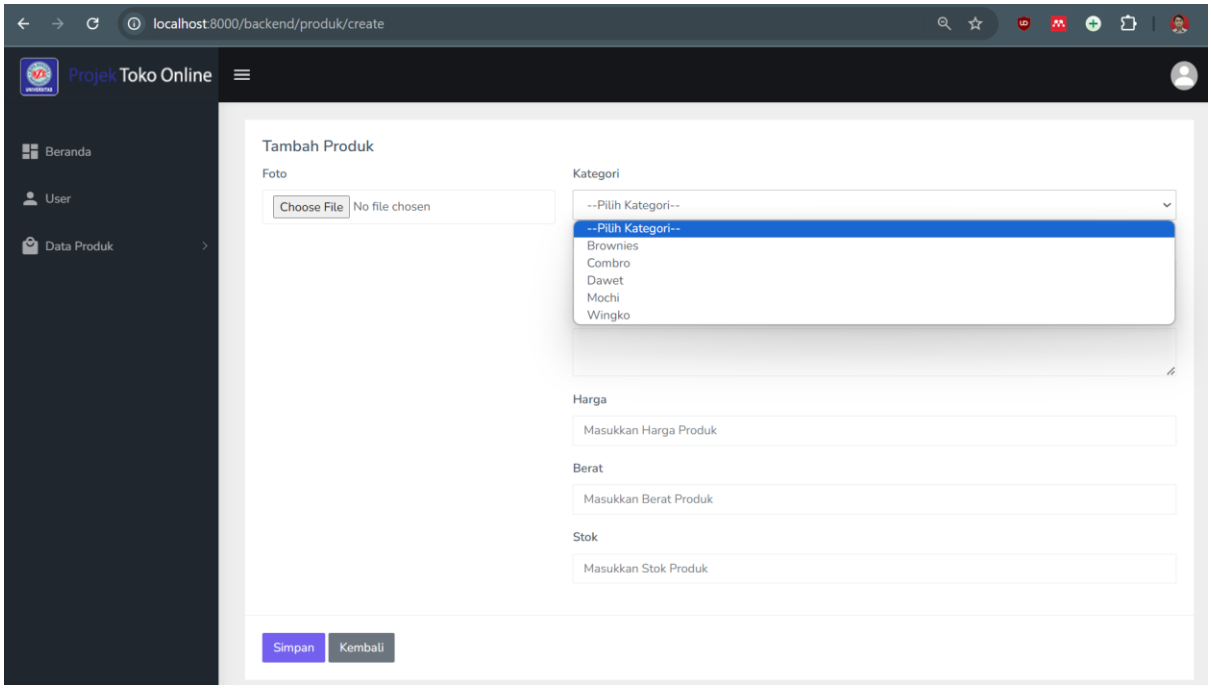
Pemanggilan **Kategori** dilakukan dalam metode create karena pada halaman halaman **create** produk, kita perlu menampilkan daftar kategori yang sudah ada untuk penggunan dimana memilih kategori yang sesuai untuk produk yang akan ditambahkan. Sehingga jika perhasil halaman **create** produk seperti pada gambar X.13

2. Tambahkan file **create.blade.php** di direktori resources\backend\v_produk, gunakan script berikut:

```
@extends('backend.v_layouts.app')
@section('content')
<!-- contentAwal -->

<div class="container-fluid">
    <div class="row">
        <div class="col-12">
            <div class="card">
                <form class="form-horizontal" action="{{ route('backend.produk.store') }}"
method="post" enctype="multipart/form-data">
                    @csrf

                    <div class="card-body">
                        <h4 class="card-title"> {{$judul}} </h4>
                        <div class="row">

                            <div class="col-md-4">
                                <div class="form-group">
                                    <label>Foto</label>
                                    <input type="file" name="foto" class="form-control
@error('foto') is-invalid @enderror">
                                    @error('foto')
```

```
                                    <div class="invalid-feedback alert-danger">{{ $message
}}</div>
                                    @enderror
                                </div>

                            </div>

                            <div class="col-md-8">
                                <div class="form-group">
                                    <label>Kategori</label>
                                    <select class="form-control @error('kategori') is-
invalid @enderror" name="kategori_id">
                                        <option value="" selected>--Pilih Kategori--
</option>
                                        @foreach ($kategori as $k)
                                        <option value="{{ $k->id }}"> {{ $k->nama_kategori
}} </option>
                                        @endforeach
                                    </select>
                                    @error('kategori_id')
                                    <span class="invalid-feedback alert-danger"
role="alert">
                                        {{ $message }}
                                    </span>
                                    @enderror
                                </div>

                                <div class="form-group">
                                    <label>Nama Produk</label>
                                    <input type="text" name="nama_produk" value="{{
old('nama_produk') }}" class="form-control @error('nama_produk') is-invalid @enderror"
placeholder="Masukkan Nama Prod">
                                    @error('nama_produk')
                                    <span class="invalid-feedback alert-danger"
role="alert">
                                        {{ $message }}
                                    </span>
                                    @enderror
                                </div>


                                <div class="form-group">
                                    <label>Detail</label><br>
                                    <textarea name="detail" class="form-control
@error('detail') is-invalid @enderror">{{ old('detail') }}</textarea>
                                    @error('detail')
                                    <span class="invalid-feedback alert-danger"
role="alert">
                                        {{ $message }}
                                    </span>
                                    @enderror
                                </div>

                                <div class="form-group">
                                    <label>Harga</label>
                                    <input type="text" onkeypress="return
hanyaAngka(event)" name="harga" value="{{ old('harga') }}" class="form-control
@error('harga') is-invalid @enderror" placeholder="Masukkan Harga Produk">
                                    @error('harga')
                                    <span class="invalid-feedback alert-danger"
role="alert">
                                        {{ $message }}
                                    </span>
                                    @enderror
                                </div>
```

```
                                <div class="form-group">
                                    <label>Berat</label>
                                    <input type="text" onkeypress="return
hanyaAngka(event)" name="berat" value="{{ old('berat') }}" class="form-control
@error('berat') is-invalid @enderror" placeholder="Masukkan Berat Produk">
                                    @error('berat')
                                    <span class="invalid-feedback alert-danger"
role="alert">
                                        {{ $message }}
                                    </span>
                                    @enderror
                                </div>

                                <div class="form-group">
                                    <label>Stok</label>
                                    <input type="text" onkeypress="return
hanyaAngka(event)" name="stok" value="{{ old('stok') }}" class="form-control @error('stok')
is-invalid @enderror" placeholder="Masukkan Stok Produk">
                                    @error('stok')
                                    <span class="invalid-feedback alert-danger"
role="alert">
                                        {{ $message }}
                                    </span>
                                    @enderror
                                </div>
                            </div>
                        </div>
                    </div>
                    <div class="border-top">
                        <div class="card-body">
                            <button type="submit" class="btn btn-primary">Simpan</button>
                            <a href="{{ route('backend.produk.index') }}">
                                <button type="button" class="btn btn-
secondary">Kembali</button>
                            </a>
                        </div>
                    </div>
                </form>
            </div>
        </div>
    </div>
</div>

<!-- contentAkhir -->
@endsection
```



Gambar X. 13
Halaman Create Produk

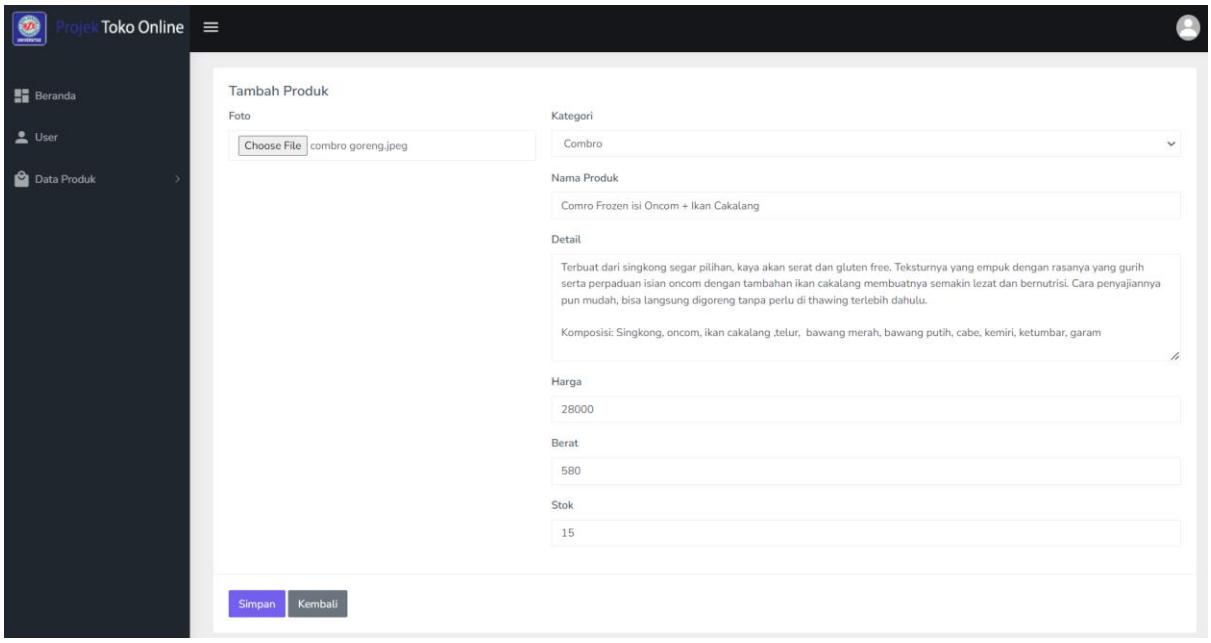
3. Tambahan *script* pada *function* store di **ProdukController** sebagai berikut

```
    public function store(Request $request)
    {
        $validatedData = $request->validate([
            'kategori_id' => 'required',
            'nama_produk' => 'required|max:255|unique:produk',
            'detail' => 'required',
            'harga' => 'required',
            'berat' => 'required',
            'stok' => 'required',
            'foto' => 'required|image|mimes:jpeg,jpg,png,gif|file|max:1024',
        ], $messages = [
            'foto.image' => 'Format gambar gunakan file dengan ekstensi jpeg, jpg, png,
atau gif.',
            'foto.max' => 'Ukuran file gambar Maksimal adalah 1024 KB.'
        ]);
        $validatedData['status'] = 0;

        if ($request->file('foto')) {
            $file = $request->file('foto');
            $extension = $file->getClientOriginalExtension();
            $originalFileName = date('YmdHis') . '_' . uniqid() . '.' . $extension;
            $directory = 'storage/img-produk/';

            // Simpan gambar asli
            $fileName = ImageHelper::uploadAndResize($file, $directory, $originalFileName);
            $validatedData['foto'] = $fileName;

            // create thumbnail 1 (lg)
            $thumbnailLg = 'thumb_lg_' . $originalFileName;
            ImageHelper::uploadAndResize($file, $directory, $thumbnailLg, 800, null);

            // create thumbnail 2 (md)
            $thumbnailMd = 'thumb_md_' . $originalFileName;
            ImageHelper::uploadAndResize($file, $directory, $thumbnailMd, 500, 519);

            // create thumbnail 3 (sm)
```

```
            $thumbnailSm = 'thumb_sm_' . $originalFileName;
            ImageHelper::uploadAndResize($file, $directory, $thumbnailSm, 100, 110);

            // Simpan nama file asli di database
            $validatedData['foto'] = $originalFileName;
        }

        Produk::create($validatedData, $messages);
        return redirect()->route('backend.produk.index')->with('success', 'Data berhasil
tersimpan');
    }
```

4. Tambah folder **img-produk** pada terminal, sehingga kita memiliki tambahan folder **img-produk** dan **img-user** pada public/storage/ atau storage/app/public/, kemudian baru lakukkan tambah data pada form **create** produk, misalnya seperti pada gambar X.14

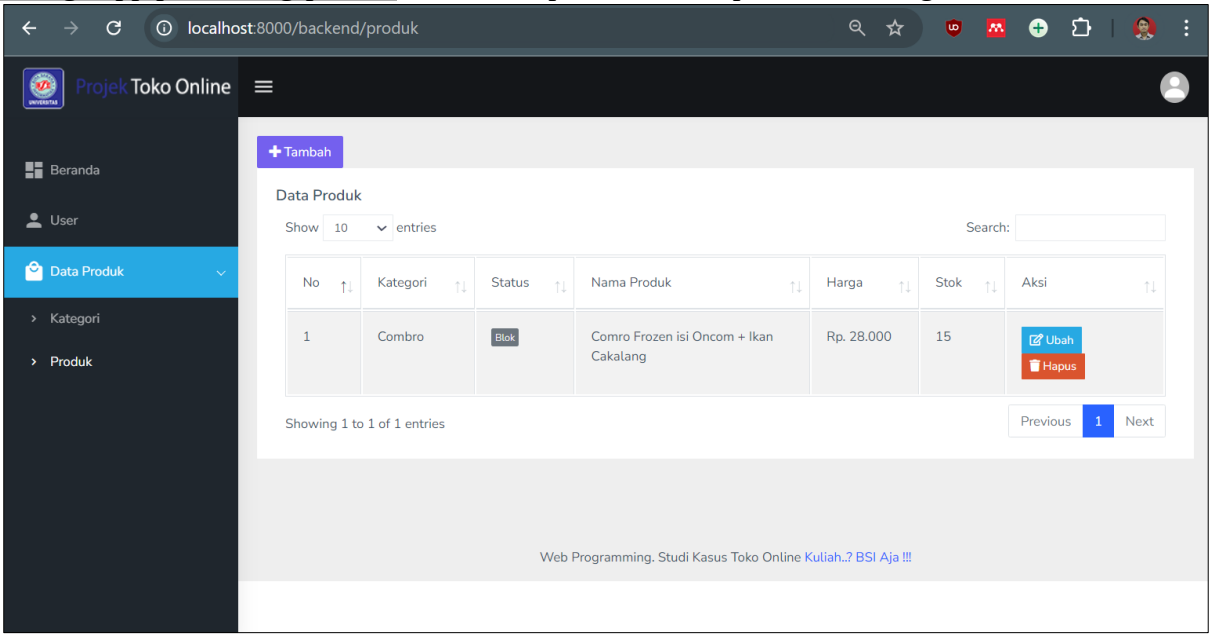
```
mkdir storage/app/public/img-produk
```



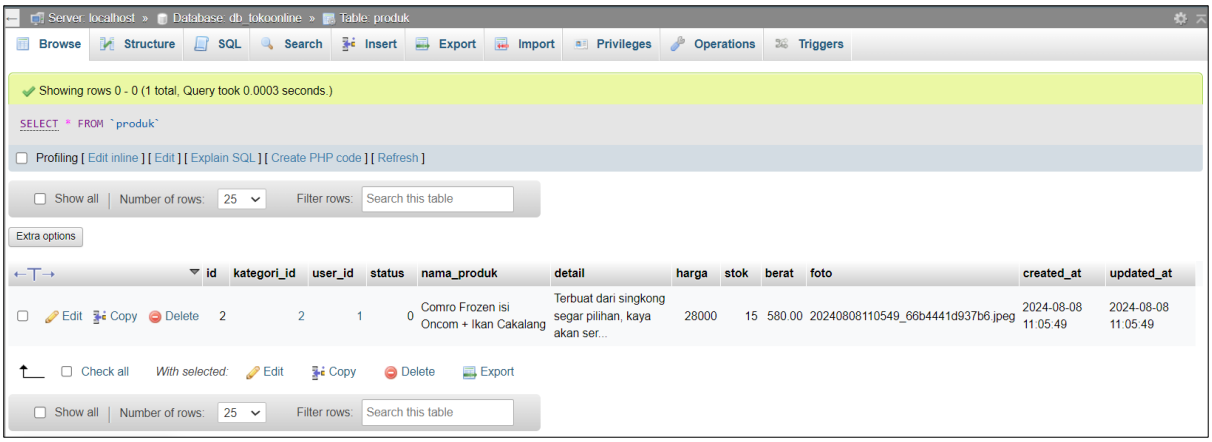
Gambar X. 14
Tambah Data Produk

5. Jika data berhasil tersimpan pada halaman **index** produk tampil seperti pada gambar X.15. Kemudian kita cek pada **phpmyadmin** record sudah masuk, terutama pada **foto** seperti pada gambar X.16 & kita pastikan foto berhasil dibuatkan thumbnail dengan ukuran yang sudah ditentukan pada controller, cek foto/gambar pada direktori public/storage/img-produk atau

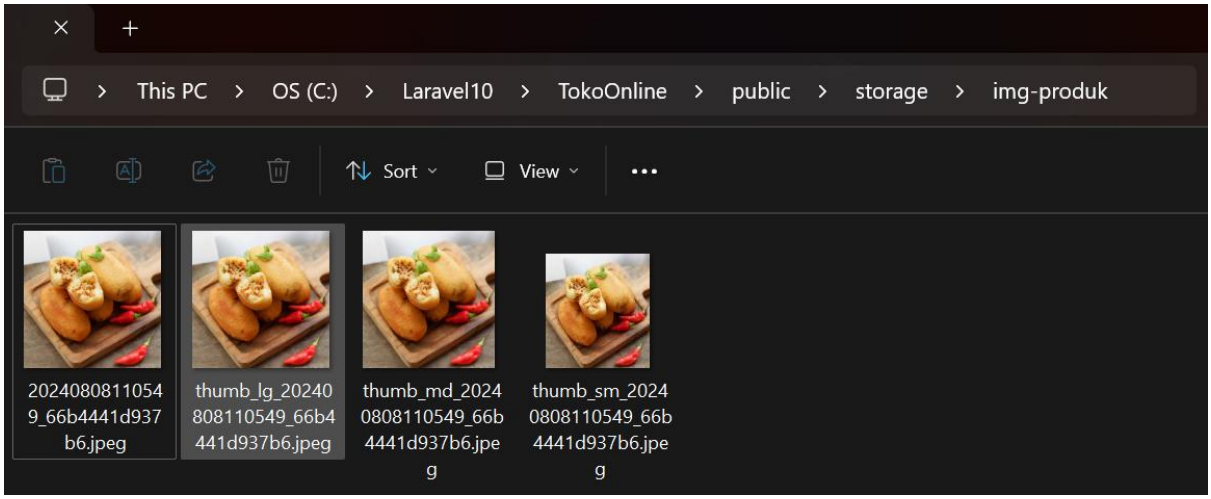
storage/app/public/img-produk seperti pada gambar X.17



Gambar X. 15
Halaman Index dengan Data Produk



Gambar X. 16
Foto Berhasil Tersimpan



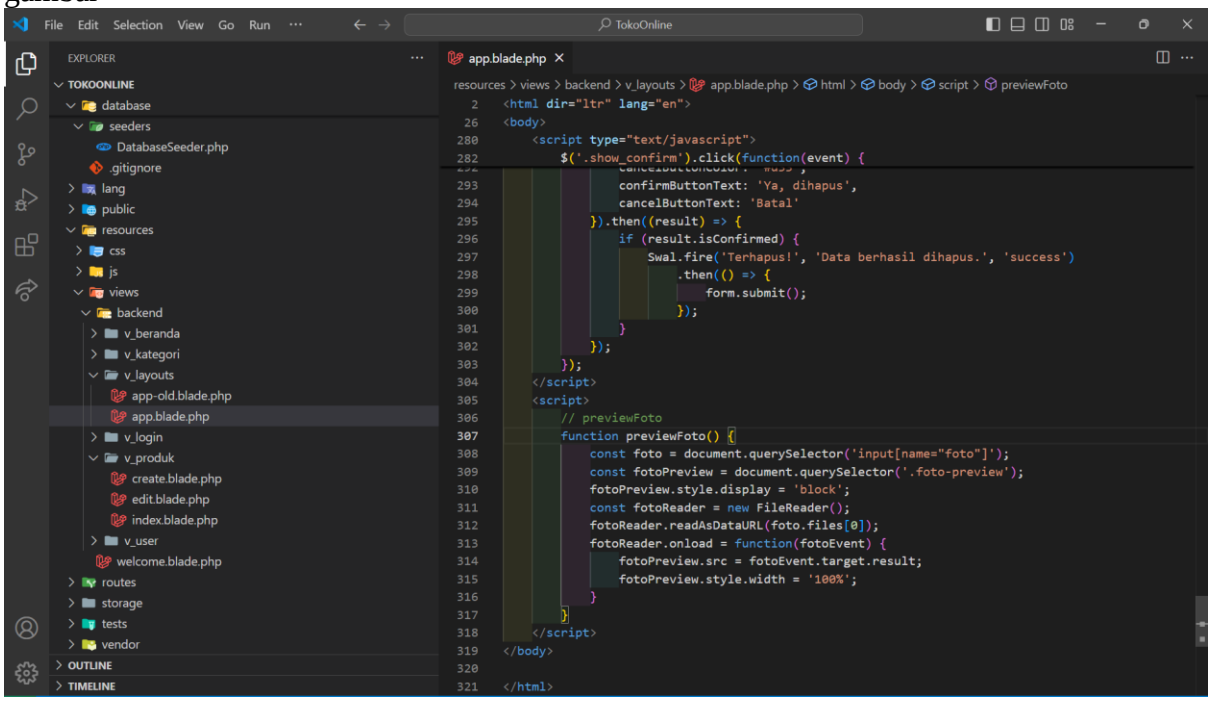
Gambar X. 17
Foto Berhasil Menggunakan Thumbnail

Latihan Mandiri 10:
Portofolion sertifikasi kompetensi, Impentasikan Unit Kompetensi Software Development pada **Menggunakan Spesifikasi Program**.

Minggu Ke-11
Data Join Tabel Part 2

11.1. Menampilkan Preview Gambar

1. Untuk memulai menggunakan Preview Gambar kita tambahkan JavaScript pada **app.blade.php** pada direktori resources\views\backend\v_layouts\app.blade.php seperti pada gambar



Gambar XI. 1
JavaScript Preview Gambar

2. Sehingga pada **create.blade.php** di dalam **ProdukController**, terdapat perubahan di mana input type="file" diarahkan ke JavaScript untuk preview gambar.

Sebelumnya

```
<div class="form-group">
    <label>Foto</label>
    <input type="file" name="foto" class="form-control @error('foto') is-invalid
@enderror">
    @error('foto')
    <div class="invalid-feedback alert-danger">{{ $message }}</div>
    @enderror
</div>
```

Menjadi

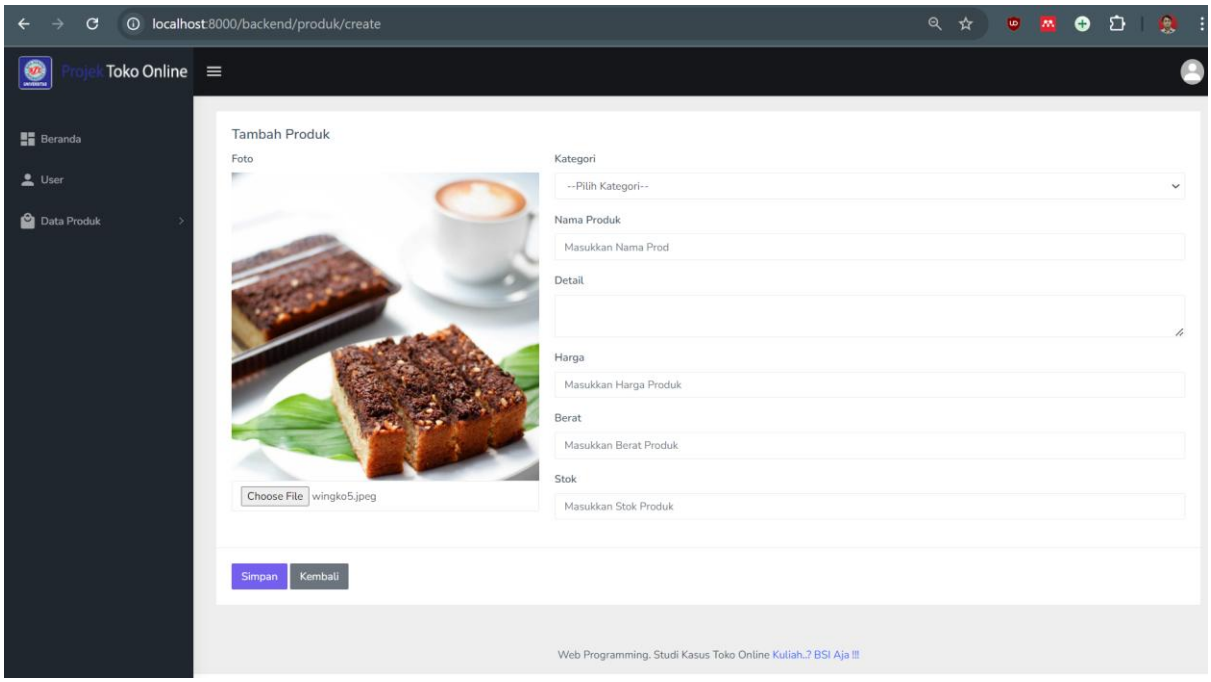
```
<div class="form-group">
    <label>Foto</label>
    <img class="foto-preview">
    <input type="file" name="foto" class="form-control @error('foto') is-invalid @enderror"
onchange="previewFoto()">
    @error('foto')
    <div class="invalid-feedback alert-danger">{{ $message }}</div>
    @enderror
</div>
```

Berikut *script* lengkap **create.blade.php** pada **ProdukController**, dan jika berhasil akan tampil seperti pada gambar XI.2

```
@extends('backend.v_layouts.app')
@section('content')
<!-- contentAwal -->

<div class="container-fluid">
    <div class="row">
        <div class="col-12">
            <div class="card">
                <form class="form-horizontal" action="{{ route('backend.produk.store') }}"
method="post" enctype="multipart/form-data">
                    @csrf

                    <div class="card-body">
                        <h4 class="card-title"> {{$judul}} </h4>
                        <div class="row">

                            <div class="col-md-4">
                                <div class="form-group">
                                    <label>Foto</label>
                                    <img class="foto-preview">
                                    <input type="file" name="foto" class="form-control
@error('foto') is-invalid @enderror" onchange="previewFoto()">
                                    @error('foto')
                                    <div class="invalid-feedback alert-danger">{{ $message
}}</div>
                                    @enderror
                                </div>
                            </div>

                            <div class="col-md-8">
                                <div class="form-group">
                                    <label>Kategori</label>
                                    <select class="form-control @error('kategori') is-
invalid @enderror" name="kategori_id">
                                        <option value="" selected>--Pilih Kategori--
</option>
                                        @foreach ($kategori as $k)
                                        <option value="{{ $k->id }}"> {{ $k->nama_kategori
}} </option>
                                        @endforeach
                                    </select>
                                    @error('kategori_id')
                                    <span class="invalid-feedback alert-danger"
role="alert">
                                        {{ $message }}
                                    </span>
                                    @enderror
                                </div>

                                <div class="form-group">
                                    <label>Nama Produk</label>
                                    <input type="text" name="nama_produk" value="{{
old('nama_produk') }}" class="form-control @error('nama_produk') is-invalid @enderror"
placeholder="Masukkan Nama Prod">
                                    @error('nama_produk')
                                    <span class="invalid-feedback alert-danger"
role="alert">
                                        {{ $message }}
                                    </span>
                                    @enderror
                                </div>
```

```
                                <div class="form-group">
                                    <label>Detail</label><br>
                                    <textarea name="detail" class="form-control
@error('detail') is-invalid @enderror">{{ old('detail') }}</textarea>
                                    @error('detail')
                                    <span class="invalid-feedback alert-danger"
role="alert">
                                        {{ $message }}
                                    </span>
                                    @enderror
                                </div>

                                <div class="form-group">
                                    <label>Harga</label>
                                    <input type="text" onkeypress="return
hanyaAngka(event)" name="harga" value="{{ old('harga') }}" class="form-control
@error('harga') is-invalid @enderror" placeholder="Masukkan Harga Produk">
                                    @error('harga')
                                    <span class="invalid-feedback alert-danger"
role="alert">
                                        {{ $message }}
                                    </span>
                                    @enderror
                                </div>

                                <div class="form-group">
                                    <label>Berat</label>
                                    <input type="text" onkeypress="return
hanyaAngka(event)" name="berat" value="{{ old('berat') }}" class="form-control
@error('berat') is-invalid @enderror" placeholder="Masukkan Berat Produk">
                                    @error('berat')
                                    <span class="invalid-feedback alert-danger"
role="alert">
                                        {{ $message }}
                                    </span>
                                    @enderror
                                </div>

                                <div class="form-group">
                                    <label>Stok</label>
                                    <input type="text" onkeypress="return
hanyaAngka(event)" name="stok" value="{{ old('stok') }}" class="form-control @error('stok')
is-invalid @enderror" placeholder="Masukkan Stok Produk">
                                    @error('stok')
                                    <span class="invalid-feedback alert-danger"
role="alert">
                                        {{ $message }}
                                    </span>
                                    @enderror
                                </div>
                            </div>
                        </div>
                    </div>
                    <div class="border-top">
                        <div class="card-body">
                            <button type="submit" class="btn btn-primary">Simpan</button>
                            <a href="{{ route('backend.produk.index') }}">
                                <button type="button" class="btn btn-
secondary">Kembali</button>
                            </a>
                        </div>
                    </div>
                </form>
            </div>
        </div>
    </div>
```

```
</div>

<!-- contentAkhir -->
@endsection
```



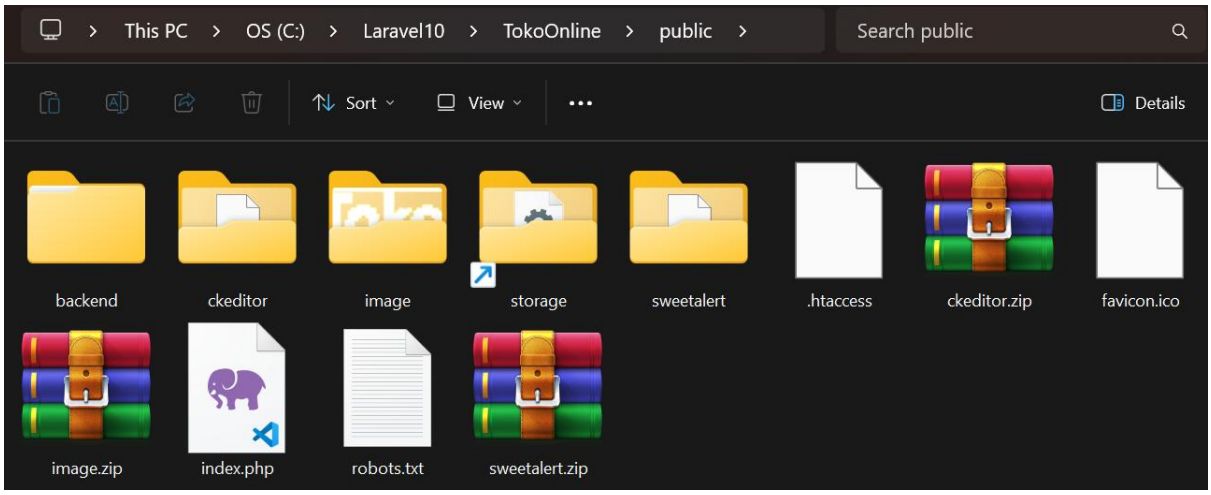
Gambar XI. 2
Preview Gambar Pada Create Produk

11.2. Menggunakan CKEditor

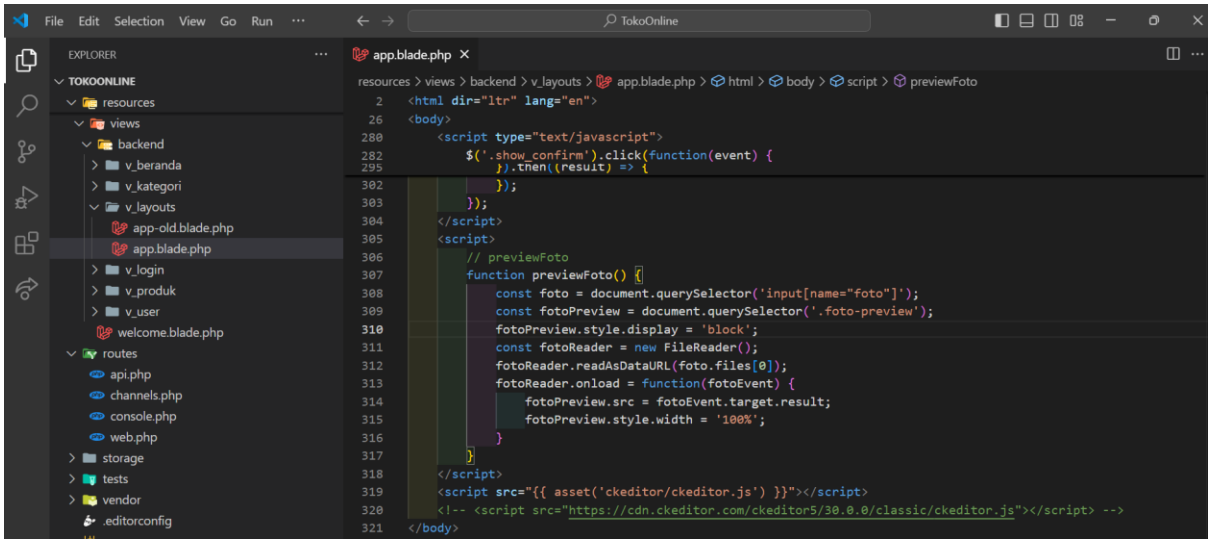
CKEditor adalah salah satu **WYSIWYG** (*What You See Is What You Get*) editor terpopuler yang digunakan untuk mengedit teks dalam aplikasi web. Dalam materi ini, akan dibahas mengenai pengenalan CKEditor, cara instalasi, dan implementasi dasar pada sebuah form di aplikasi web. CKEditor alat yang memungkinkan pengguna untuk menulis dan mengedit teks dalam konten rich text (teks dengan format HTML) dengan mudah melalui antarmuka grafis, tanpa perlu pengetahuan HTML yang mendalam. Ini banyak digunakan di berbagai aplikasi web untuk keperluan blogging, pembuatan konten, dan pengelolaan dokumen. CKEditor mendukung berbagai fitur seperti penambahan gambar, tabel, tautan, dan format teks, menjadikannya pilihan yang kuat untuk aplikasi web yang membutuhkan editor teks yang lengkap

1. siapkan CKeditor bisa langsung memanggil **ckeditor.js** yakni pada situs resmi https://cdn.ckeditor.com/ckeditor5/30.0.0/classic/ckeditor.js atau dengan cara download yang telah tersedia https://bit.ly/LaravelWebPro2 dengan nama file **ckeditor.zip** kemudian extrak **ckeditor** pada direktori TokoOnline\public seperti pada gambar XI.3
2. Kemudian panggil komponen **ckeditor** pada **app.blade.php** di direktori resources\views\backend\v_layouts\ seperti pada gambar XI.4

```
    <script src="{{ asset('ckeditor/ckeditor.js') }}"></script>
    <!-- <script
src="https://cdn.ckeditor.com/ckeditor5/30.0.0/classic/ckeditor.js"></script> -->
    <script>
        ClassicEditor
            .create(document.querySelector('#ckeditor'))
            .catch(error => {
                console.error(error);
            });
    </script>
```



Gambar XI. 3
CKeditor Pada Public



Gambar XI. 4
Memanggil ckeditor.js

3. Sekarang kita dapat memastikan apakah *ckeditor* sudah dapat berhasil kita gunakan, pada form yakni pada text, misalnya seperti pada text dengan label Detail, kita tambahkan ID= "ckeditor"

```
<textarea name="detail" class="form-control @error('detail') is-invalid @enderror">{{
old('detail') }}</textarea>
```

Menjadi

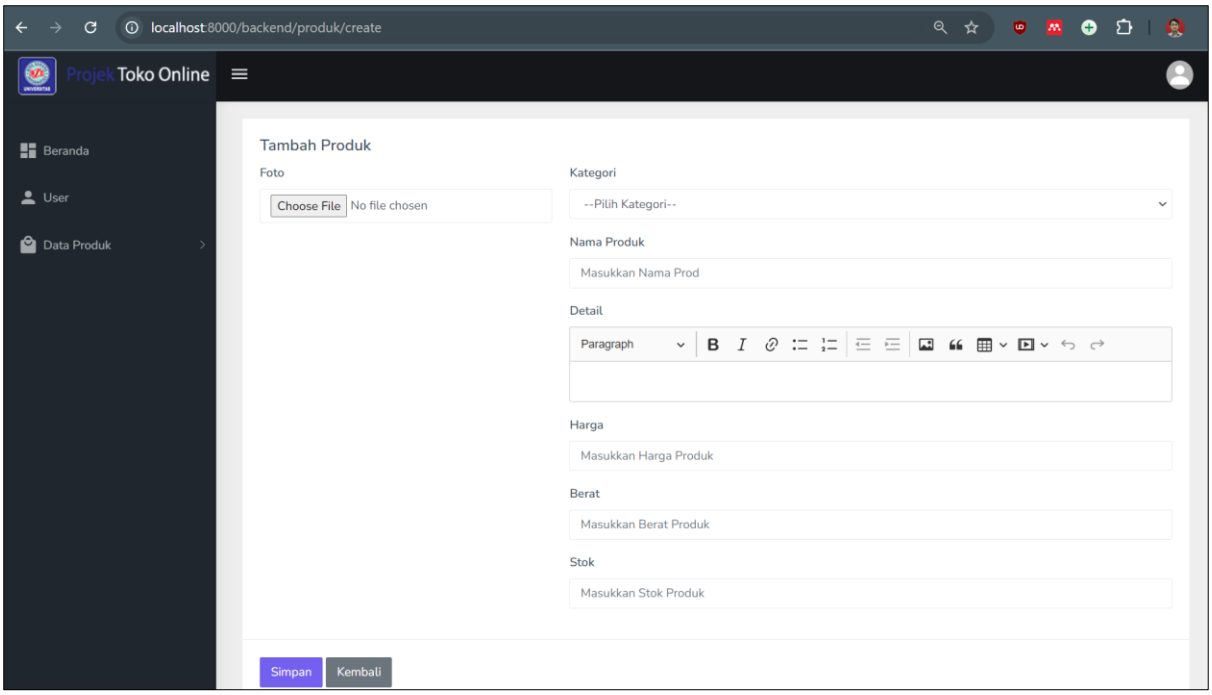
```
<textarea name="detail" class="form-control @error('detail') is-invalid @enderror"
id="ckeditor">{{ old('detail') }}</textarea>
```

Berikut *script* lengkap pada **create.blade.php** pada **ProdukController**, dan jika berhasil akan tampil seperti pada gambar XI.5

```
@extends('backend.v_layouts.app')
@section('content')
```

```
<!-- contentAwal -->

<div class="container-fluid">
    <div class="row">
        <div class="col-12">
            <div class="card">
                <form class="form-horizontal" action="{{ route('backend.produk.store') }}"
method="post" enctype="multipart/form-data">
                    @csrf
                    <div class="card-body">
                        <h4 class="card-title"> {{$judul}} </h4>
                        <div class="row">

                            <div class="col-md-4">
                                <div class="form-group">
                                    <label>Foto</label>
                                    <img class="foto-preview">
                                    <input type="file" name="foto" class="form-control
@error('foto') is-invalid @enderror" onchange="previewFoto()">
                                    @error('foto')
                                    <div class="invalid-feedback alert-danger">{{ $message
}}</div>
                                    @enderror
                                </div>
                            </div>

                            <div class="col-md-8">
                                <div class="form-group">
                                    <label>Kategori</label>
                                    <select class="form-control @error('kategori') is-
invalid @enderror" name="kategori_id">
                                        <option value="" selected>--Pilih Kategori--
</option>

                                        @foreach ($kategori as $k)
                                        <option value="{{ $k->id }}"> {{ $k->nama_kategori
}} </option>

                                        @endforeach
                                    </select>
                                    @error('kategori_id')
                                    <span class="invalid-feedback alert-danger"
role="alert">

                                        {{ $message }}
                                    </span>
                                    @enderror
                                </div>

                                <div class="form-group">
                                    <label>Nama Produk</label>
                                    <input type="text" name="nama_produk" value="{{
old('nama_produk') }}" class="form-control @error('nama_produk') is-invalid @enderror"
placeholder="Masukkan Nama Prod">
                                    @error('nama_produk')
                                    <span class="invalid-feedback alert-danger"
role="alert">

                                        {{ $message }}
                                    </span>
                                    @enderror
                                </div>

                                <div class="form-group">
                                    <label>Detail</label><br>
                                    <textarea name="detail" class="form-control
@error('detail') is-invalid @enderror" id="ckeditor">{{ old('detail') }}</textarea>
                                    @error('detail')
                                    <span class="invalid-feedback alert-danger"
role="alert">
```

```
                                    {{ $message }}
                                </span>
                                @enderror
                            </div>

                            <div class="form-group">
                                <label>Harga</label>
                                <input type="text" onkeypress="return
hanyaAngka(event)" name="harga" value="{{ old('harga') }}" class="form-control
@error('harga') is-invalid @enderror" placeholder="Masukkan Harga Produk">
                                @error('harga')
                                <span class="invalid-feedback alert-danger"
role="alert">
                                    {{ $message }}
                                </span>
                                @enderror
                            </div>

                            <div class="form-group">
                                <label>Berat</label>
                                <input type="text" onkeypress="return
hanyaAngka(event)" name="berat" value="{{ old('berat') }}" class="form-control
@error('berat') is-invalid @enderror" placeholder="Masukkan Berat Produk">
                                @error('berat')
                                <span class="invalid-feedback alert-danger"
role="alert">
                                    {{ $message }}
                                </span>
                                @enderror
                            </div>

                            <div class="form-group">
                                <label>Stok</label>
                                <input type="text" onkeypress="return
hanyaAngka(event)" name="stok" value="{{ old('stok') }}" class="form-control @error('stok')
is-invalid @enderror" placeholder="Masukkan Stok Produk">
                                @error('stok')
                                <span class="invalid-feedback alert-danger"
role="alert">
                                    {{ $message }}
                                </span>
                                @enderror
                            </div>
                        </div>

                    </div>
                </div>
                <div class="border-top">
                    <div class="card-body">
                        <button type="submit" class="btn btn-primary">Simpan</button>
                        <a href="{{ route('backend.produk.index') }}">
                            <button type="button" class="btn btn-
secondary">Kembali</button>
                        </a>
                    </div>
                </div>
            </form>
        </div>
    </div>
</div>

<!-- contentAkhir -->
@endsection
```



Gambar XI. 5
Label Detail Menggunakan CKeditor

11.3. Tabel Foto Produk

1. Membuat migration foto_produk, pada terminal sebagai berikut:

```
php artisan make:migration create_foto_produk_table
```

```
/c/Laravel10/TokoOnline took 3s
php artisan make:migration create_foto_produk_table

  INFO  Migration [C:\Laravel10\TokoOnline\database\migrations/2024_08_11_124430_create_foto_produk_table.php] cr
eated successfully.
```

Gambar XI. 6
Migration Foto Produk

2. Dengan Blueprint tabel sebagai berikut:

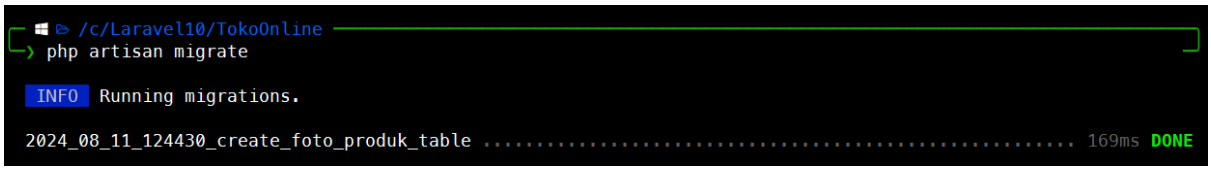
```
 Schema::create('foto_produk', function (Blueprint $table) {
            $table->id();
            $table->unsignedBigInteger('produk_id');
            $table->string('foto');
            $table->timestamps();
            $table->foreign('produk_id')->references('id')->on('produk')-
>onDelete('cascade');
        });
```

Tabel XI. 1
Tabel Foto_Produk

Fild	Tipe Data	Keterangan
id	Bigint	Primary Key
produk_id	Bigint	
foto	Varchar(255)	
created_at	Timestamp	
updated_at	Timestamp	

3. Jalankan migration

```
php artisan migrate
```

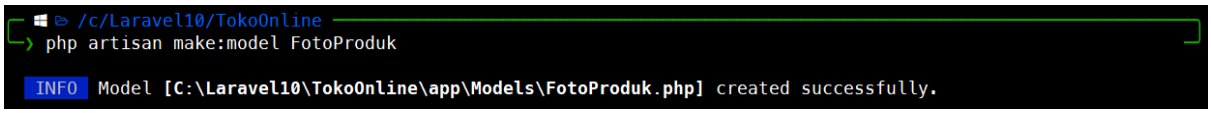


Gambar XI. 7
Migrations Pada FotoProduk

11.3. Model Pada Foto Produk

1. Pada terminal buat model dengan nama **FotoProduk** sebagai berikut:

```
php artisan make:model FotoProduk
```



Gambar XI. 8
Model Pada FotoProduk

2. Berikut *script* model pada **FotoProduk**

```php
<?php

namespace App\Models;

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

class FotoProduk extends Model
{
    public $timestamps = true;
    protected $table = 'foto_produk';
    protected $guarded = ['id'];
}
```

3. Sehingga pada model **Produk** kita tambahkan untuk memanggil **FotoProduk**, berikut script lengkap pada **Produk**

```php
<?php

namespace App\Models;

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

class Produk extends Model
{
    public $timestamps = true;
    protected $table = "produk";
    protected $guarded = ['id'];

    public function kategori()
    {
        return $this->belongsTo(Kategori::class);
    }

    public function user()
    {
```

```
        return $this->belongsTo(User::class);
    }

    public function fotoProduk()
    {
        return $this->hasMany(FotoProduk::class);
    }
}
```

11.4. Menambah Gambar Pada Produk

1. Tambahkan script Pada *function* **show()** di **ProdukController**, dimana kita akan memanggil view **show.blade.php** pada direktori resources/views/v_produk

```
    public function show(string $id)
    {
        $produk = Produk::with('gambar')->findOrFail($id);
        $kategori = Kategori::orderBy('nama_kategori', 'asc')->get();
        return view('backend.v_produk.show', [
            'judul' => 'Detail Produk',
            'show' => $produk,
            'kategori' => $kategori
        ]);
    }
```

2. Sehingga berikut *script* lengkap pada view **show.blade.php**

```
@extends('backend.v_layouts.app')
@section('content')
<!-- contentAwal -->

<div class="container-fluid">
    <div class="row">
        <div class="col-12">
            <div class="card">
                <div class="card-body">
                    <h4 class="card-title">{{ $judul }}</h4>
                    <div class="row">

                        <div class="col-md-6">
                            <div class="form-group">
                                <label>Kategori</label>
                                <select name="kategori_id" class="form-control
@error('kategori_id') is-invalid @enderror" disabled>
                                    <option value="" selected> - Pilih Kategori - </option>
                                    @foreach ($kategori as $row)
                                    <option value="{{ $row->id }}" {{ old('kategori_id',
$show->kategori_id) == $row->id ? 'selected' : '' }}>
                                        {{ $row->nama_kategori }}
                                    </option>
                                    @endforeach
                                </select>
                                @error('kategori_id')
                                <span class="invalid-feedback alert-danger" role="alert">
                                    {{ $message }}
                                </span>
                                @enderror
                            </div>

                            <div class="form-group">
                                <label>Nama Produk</label>
                                <input type="text" name="nama_produk" value="{{
old('nama_produk', $show->nama_produk) }}" class="form-control @error('nama_produk') is-
invalid @enderror" placeholder="Masukkan Nama Produk" disabled>
                                @error('nama_produk')
                                <span class="invalid-feedback alert-danger" role="alert">
                                    {{ $message }}
```

```
                                </span>
                                @enderror
                            </div>

                            <div class="form-group">
                                <label>Detail</label>
                                <textarea name="detail" class="form-control
@error('detail') is-invalid @enderror" id="ckeditor" disabled>{{ old('detail', $show-
>detail) }}</textarea>
                                @error('detail')
                                <span class="invalid-feedback alert-danger" role="alert">
                                    {{ $message }}
                                </span>
                                @enderror
                            </div>
                            <div class="form-group">
                                <label>Foto Utama</label> <br>
                                <img src="{{ asset('storage/img-produk/' . $show->foto) }}"
class="foto-preview" width="100%">
                            </div>
                        </div>

                        <div class="col-md-6">
                            <label>Foto Tambahan</label>
                            <div id="foto-container">
                                <div class="row">
                                    @foreach($show->gambar as $gambar)
                                    <div class="col-md-8">
                                        <img src="{{ asset('storage/img-produk/' . $gambar-
>foto) }}" width="100%">
                                    </div>
                                    <div class="col-md-4">
                                        <form action="{{
route('backend.foto_produk.destroy', $gambar->id) }}" method="post">
                                            @csrf
                                            @method('delete')
                                            <button type="submit" class="btn btn-
danger">Hapus</button>
                                        </form>
                                    </div>

                                    @endforeach
                                </div>
                                <br>
                            </div>
                            <button type="button" class="btn btn-primary add-foto mt-
2">Tambah Foto</button>
                        </div>

                    </div>
                </div>

                <div class="border-top">
                    <div class="card-body">
                        <a href="{{ route('backend.produk.index') }}">
                            <button type="button" class="btn btn-
secondary">Kembali</button>
                        </a>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>

<!-- contentAkhir -->
```

```
@endsection

<script>
    document.addEventListener('DOMContentLoaded', function() {
        const fotoContainer = document.getElementById('foto-container');
        const addFotoButton = document.querySelector('.add-foto');

        addFotoButton.addEventListener('click', function() {
            const fotoRow = document.createElement('div');
            fotoRow.classList.add('form-group', 'row');
            fotoRow.innerHTML = `
                <form action="{{ route('backend.foto_produk.store') }}" method="post"
enctype="multipart/form-data">
                    @csrf
                    <div class="col-md-12">
                        <input type="hidden" name="produk_id" value="{{ $show->id }}">
                        <input type="file" name="foto_produk[]" class="form-control
@error('foto_produk') is-invalid @enderror">
                        <button type="submit" class="btn btn-success">Simpan</button>
                    </div>
                </form>
            `;
            fotoContainer.appendChild(fotoRow);
        });
    });
</script>
```

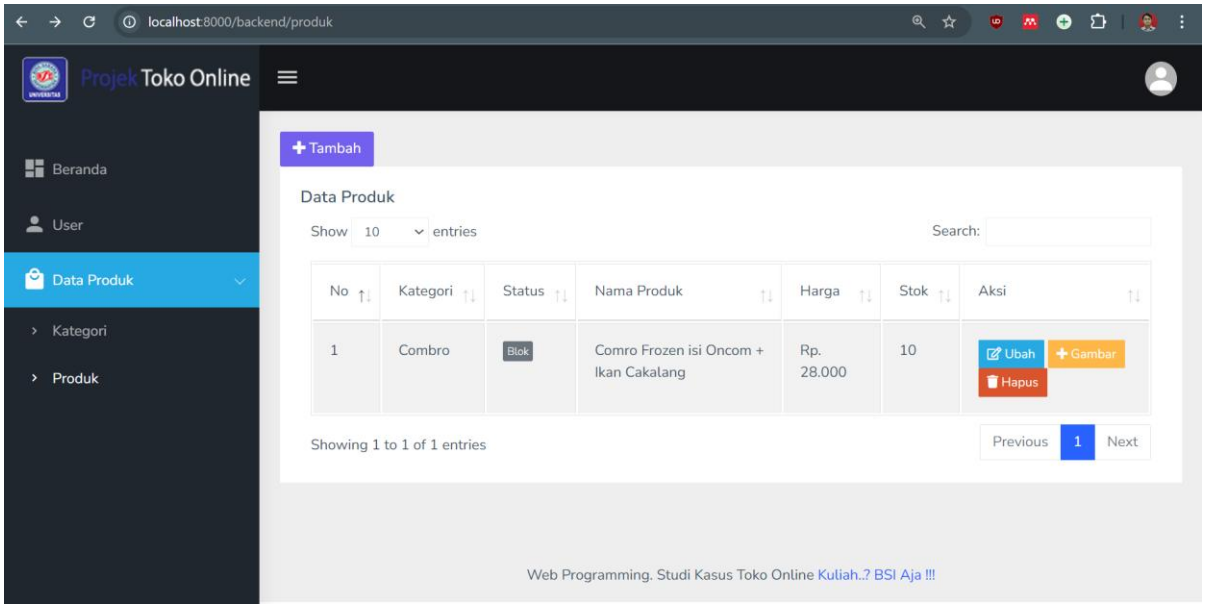
3. Agar kita dapat memanggil view **show()** aksi ini kita tambahkan pada view **index()** pada direktori resources/views/v_produk, dimana aksi **+Gambar** seperti pada gambar XI.9 & jika di klik aksi **+Gambar** maka seperti pada gambar XI.10

```
  <a href="{{ route('backend.produk.show', $row->id) }}" title="Ubah Data">
                                        <button type="button" class="btn btn-
warning btn-sm"><i class="fas fa-plus"></i> Gambar</button>
                                    </a>
```

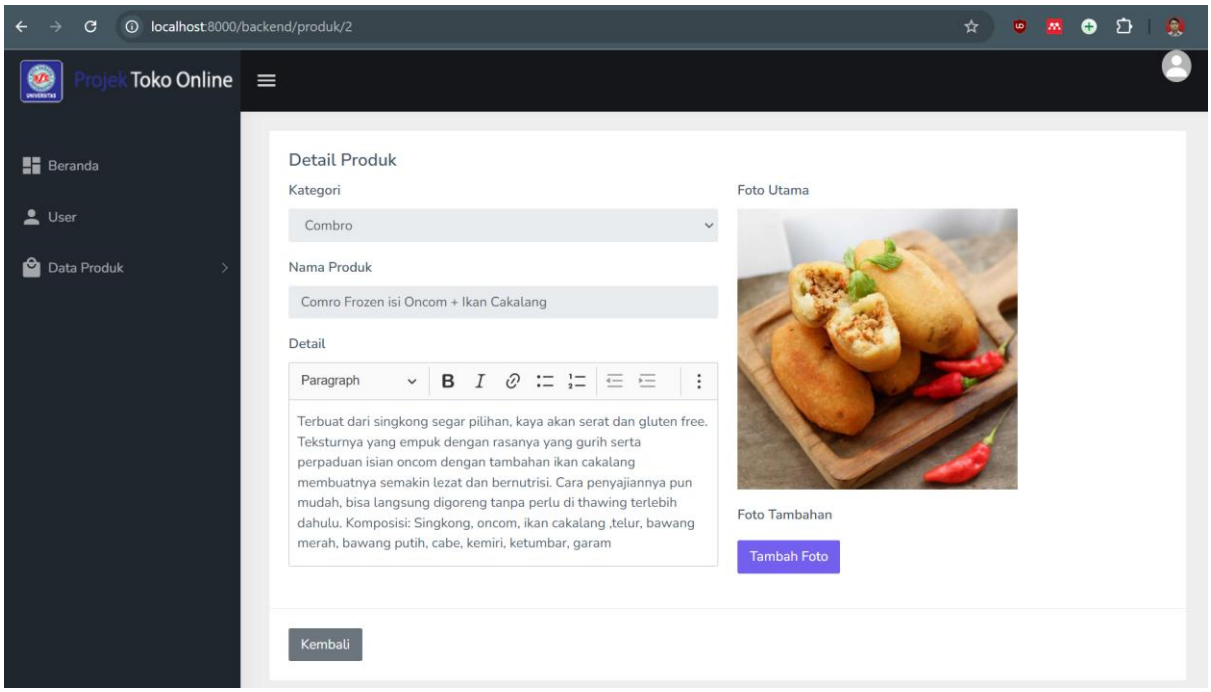
Berikut *script* lengkap **index.blade.php**

```
@extends('backend.v_layouts.app')
@section('content')
<!-- contentAwal -->

<div class="row">

    <div class="col-12">
        <a href="{{ route('backend.produk.create') }}">
            <button type="button" class="btn btn-primary"><i class="fas fa-plus"></i>
Tambah</button>
        </a>
        <div class="card">
            <div class="card-body">
                <h5 class="card-title"> {{$judul}} </h5>
                <div class="table-responsive">
                    <table id="zero_config" class="table table-striped table-bordered">
                        <thead>
                            <tr>
                                <th>No</th>
                                <th>Kategori</th>
                                <th>Status</th>
                                <th>Nama Produk</th>
                                <th>Harga</th>
                                <th>Stok</th>
                                <th>Aksi</th>
                            </tr>
```

```
                        </thead>

                        <tbody>
                            @foreach ($index as $row)
                            <tr>
                                <td> {{ $loop->iteration }}</td>
                                <td> {{ $row->kategori->nama_kategori }} </td>
                                <td>
                                    @if ($row->status ==1)
                                    <span class="badge badge-success"></i>
                                        Publis</span>
                                    @elseif($row->status ==0)
                                    <span class="badge badge-secondary"></i>
                                        Blok</span>
                                    @endif
                                </td>
                                <td> {{ $row->nama_produk }} </td>
                                <td> Rp. {{ number_format($row->harga, 0, ',', '.') }}
</td>
                                <td> {{ $row->stok }} </td>
                                <td>
                                    <a href="{{ route('backend.produk.edit', $row->id) }}"
title="Ubah Data">
                                        <button type="button" class="btn btn-cyan btn-
sm"><i class="far fa-edit"></i> Ubah</button>
                                    </a>
                                    <a href="{{ route('backend.produk.show', $row->id) }}"
title="Ubah Data">
                                        <button type="button" class="btn btn-warning btn-
sm"><i class="fas fa-plus"></i> Gambar</button>
                                    </a>

                                    <form method="POST" action="{{
route('backend.produk.destroy', $row->id) }}" style="display: inline-block;">
                                        @method('delete')
                                        @csrf
                                        <button type="submit" class="btn btn-danger btn-sm
show_confirm" data-konf-delete="{{ $row->nama }}" title='Hapus Data'>
                                            <i class="fas fa-trash"></i> Hapus</button>
                                    </form>
                                </td>
                            </tr>
                            @endforeach
                        </tbody>
                    </table>
                </div>

            </div>
        </div>
    </div>
</div>

<!-- contentAkhir -->
@endsection
```



Gambar XI. 9
Aksi Tambah Gambar Pada Halaman Produk



Gambar XI. 10
Halaman Detail Produk

4. Sebelum kita menambahkan Foto Tambahan, tambahkan script pada fungsi **storeFoto** dan **destroyFoto** di **ProdukController**. Dengan demikian, saat kita mengklik **Tambah Foto** -> Pilih gambar -> klik tombol **Simpan**, data akan tersimpan. Selain itu, terdapat aksi **Hapus** untuk menghapus data foto yang tidak diperlukan. Terlihat seperti pada gambar XI.11

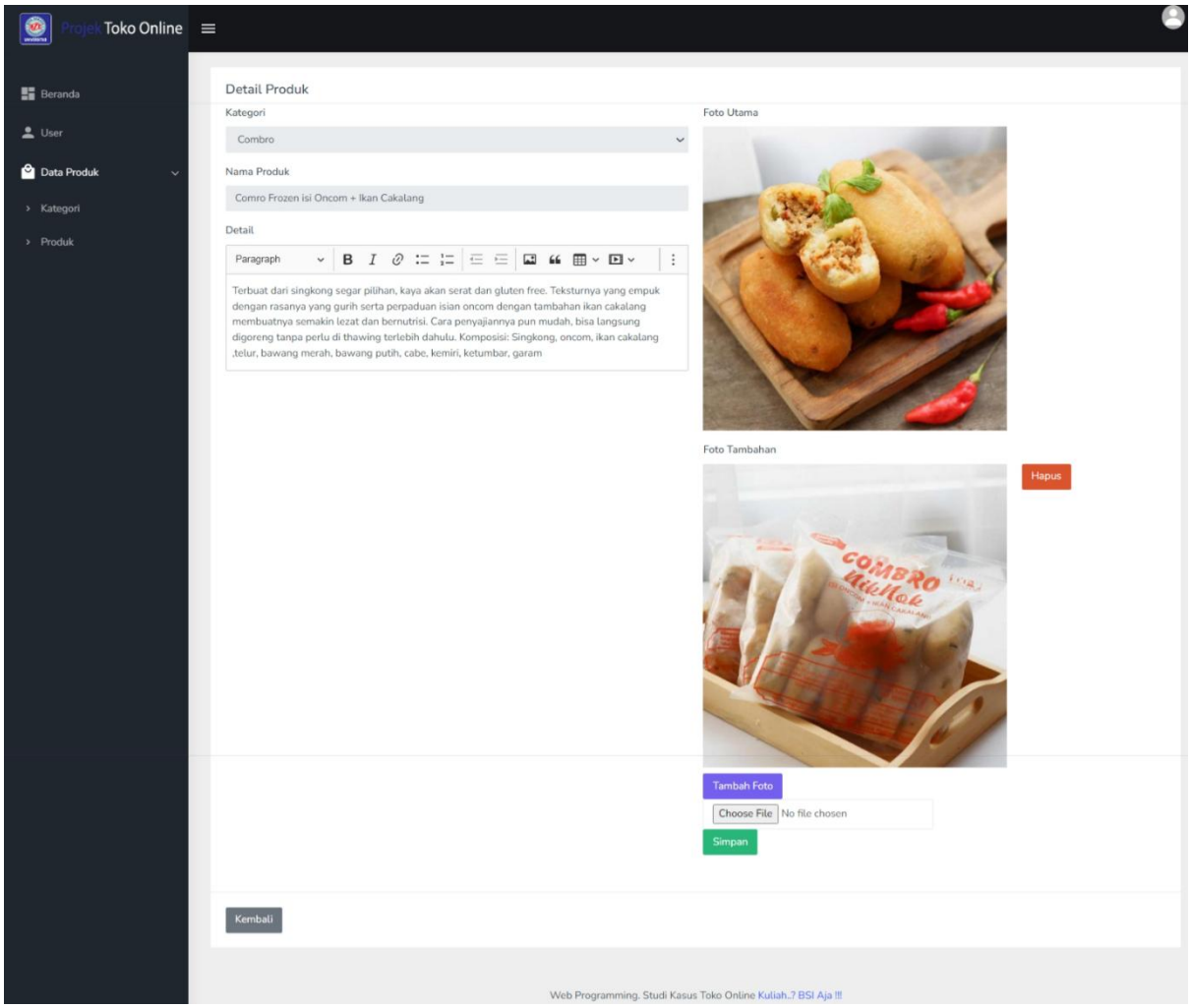
```
    // Method untuk menyimpan foto tambahan
    public function storeFoto(Request $request)
    {
        // Validasi input
        $request->validate([
            'produk_id' => 'required|exists:produk,id',
            'foto_produk.*' => 'image|mimes:jpeg,jpg,png,gif|file|max:1024',
        ]);
```

```
        if ($request->hasFile('foto_produk')) {
            foreach ($request->file('foto_produk') as $file) {
                // Buat nama file yang unik
                $extension = $file->getClientOriginalExtension();
                $filename = date('YmdHis') . '_' . uniqid() . '.' . $extension;
                $directory = 'storage/img-produk/';

                // Simpan dan resize gambar menggunakan ImageHelper
                ImageHelper::uploadAndResize($file, $directory, $filename, 800, null);
                // Simpan data ke database
                FotoProduk::create([
                    'produk_id' => $request->produk_id,
                    'foto' => $filename,
                ]);
            }
        }
        return redirect()->route('backend.produk.show', $request->produk_id)
            ->with('success', 'Foto berhasil ditambahkan.');
    }


    // Method untuk menghapus foto
    public function destroyFoto($id)
    {
        $foto = FotoProduk::findOrFail($id);
        $produkId = $foto->produk_id;

        // Hapus file gambar dari storage
        $imagePath = public_path('storage/img-produk/') . $foto->foto;
        if (file_exists($imagePath)) {
            unlink($imagePath);
        }
        // Hapus record dari database
        $foto->delete();

        return redirect()->route('backend.produk.show', $produkId)
            ->with('success', 'Foto berhasil dihapus.');
    }
```

5. Pada untuk menambahkan routes**web.php** dengan *script* lengkap sebagai berikut sehingga:

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\BerandaController;
use App\Http\Controllers\LoginController;
use App\Http\Controllers\UserController;
use App\Http\Controllers\KategoriController;
use App\Http\Controllers\ProdukController;

Route::get('/', function () {
    // return view('welcome');
    return redirect()->route('backend.login');
});
Route::get('backend/beranda', [BerandaController::class, 'berandaBackend'])-
>name('backend.beranda')->middleware('auth');

Route::get('backend/login', [LoginController::class, 'loginBackend'])-
>name('backend.login');
Route::post('backend/login', [LoginController::class, 'authenticateBackend'])-
>name('backend.login');
Route::post('backend/logout', [LoginController::class, 'logoutBackend'])-
>name('backend.logout');

// Route::resource('backend/user', UserController::class)->middleware('auth');
```

```
Route::resource('backend/user', UserController::class, ['as' => 'backend'])-
>middleware('auth');
Route::resource('backend/kategori', KategoriController::class, ['as' => 'backend'])-
>middleware('auth');
Route::resource('backend/produk', ProdukController::class, ['as' => 'backend'])-
>middleware('auth');
// Route untuk menambahkan foto
Route::post('foto-produk/store', [ProdukController::class, 'storeFoto'])-
>name('backend.foto_produk.store')->middleware('auth');
// Route untuk menghapus foto
Route::delete('foto-produk/{id}', [ProdukController::class, 'destroyFoto'])-
>name('backend.foto_produk.destroy')->middleware('auth');
```



Gambar XI. 11
Foto Tambahan Pada Produk

11.5. Ubah Data Pada Produk

1. Tambahkan function **edit** dan function **update** pada **ProdukController**, berikut script lengkapnya

```
/**
     * Show the form for editing the specified resource.
     */
    public function edit(string $id)
    {
        $produk = Produk::findOrFail($id);
        $kategori = Kategori::orderBy('nama_kategori', 'asc')->get();
        return view('backend.v_produk.edit', [
            'judul' => 'Ubah Produk',
```

```
            'edit' => $produk,
            'kategori' => $kategori
        ]);
    }

    /**
     * Update the specified resource in storage.
     */
    public function update(Request $request, string $id)
    {
        //ddd($request);
        $produk = Produk::findOrFail($id);
        $rules = [
            'nama_produk' => 'required|max:255|unique:produk,nama_produk,' . $id,
            'kategori_id' => 'required',
            'status' => 'required',
            'detail' => 'required',
            'harga' => 'required',
            'berat' => 'required',
            'stok' => 'required',
            'foto' => 'image|mimes:jpeg,jpg,png,gif|file|max:1024',
        ];
        $messages = [
            'foto.image' => 'Format gambar gunakan file dengan ekstensi jpeg, jpg, png,
atau gif.',
            'foto.max' => 'Ukuran file gambar Maksimal adalah 1024 KB.'
        ];
        $validatedData['user_id'] = auth()->id();
        $validatedData = $request->validate($rules, $messages);

        if ($request->file('foto')) {
            //hapus gambar lama
            if ($produk->foto) {
                $oldImagePath = public_path('storage/img-produk/') . $produk->foto;
                if (file_exists($oldImagePath)) {
                    unlink($oldImagePath);
                }
                $oldThumbnailLg = public_path('storage/img-produk/') . 'thumb_lg_' .
$produk->foto;
                if (file_exists($oldThumbnailLg)) {
                    unlink($oldThumbnailLg);
                }
                $oldThumbnailMd = public_path('storage/img-produk/') . 'thumb_md_' .
$produk->foto;
                if (file_exists($oldThumbnailMd)) {
                    unlink($oldThumbnailMd);
                }
                $oldThumbnailSm = public_path('storage/img-produk/') . 'thumb_sm_' .
$produk->foto;
                if (file_exists($oldThumbnailSm)) {
                    unlink($oldThumbnailSm);
                }
            }
            $file = $request->file('foto');
            $extension = $file->getClientOriginalExtension();
            $originalFileName = date('YmdHis') . '_' . uniqid() . '.' . $extension;
            $directory = 'storage/img-produk/';

            // Simpan gambar asli
            $fileName = ImageHelper::uploadAndResize($file, $directory, $originalFileName);
            $validatedData['foto'] = $fileName;

            // create thumbnail 1 (lg)
            $thumbnailLg = 'thumb_lg_' . $originalFileName;
            ImageHelper::uploadAndResize($file, $directory, $thumbnailLg, 800, null);
```

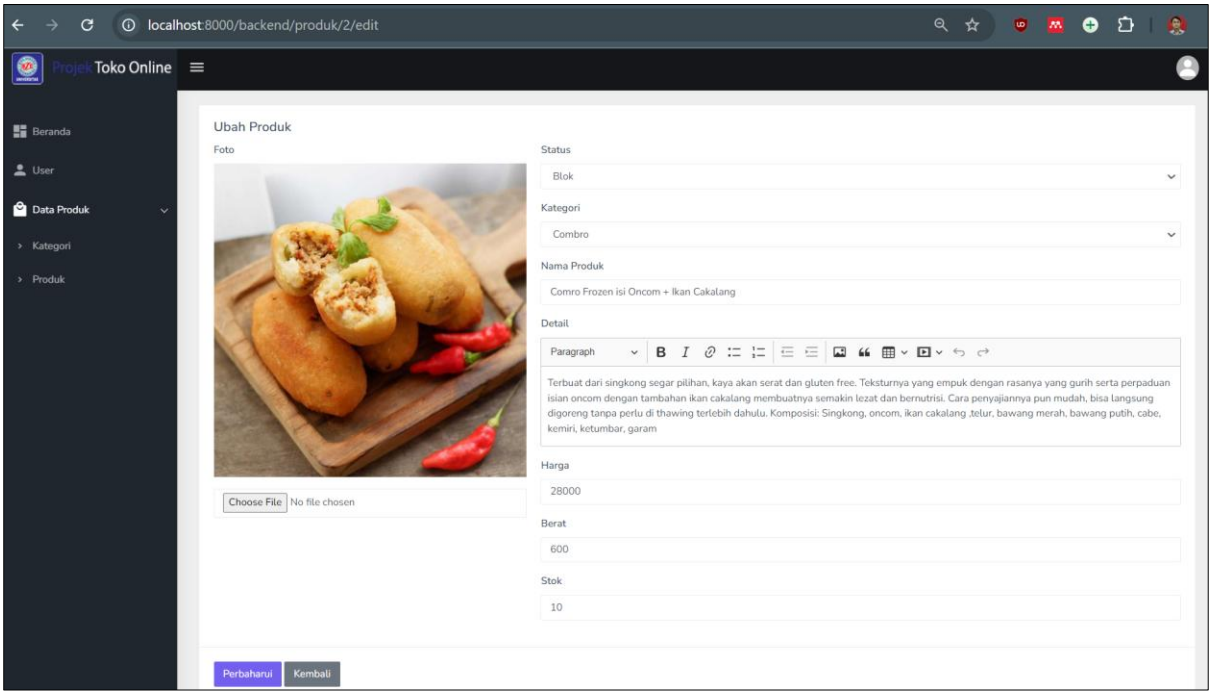
```
            // create thumbnail 2 (md)
            $thumbnailMd = 'thumb_md_' . $originalFileName;
            ImageHelper::uploadAndResize($file, $directory, $thumbnailMd, 500, 519);

            // create thumbnail 3 (sm)
            $thumbnailSm = 'thumb_sm_' . $originalFileName;
            ImageHelper::uploadAndResize($file, $directory, $thumbnailSm, 100, 110);

            // Simpan nama file asli di database
            $validatedData['foto'] = $originalFileName;
        }

        $produk->update($validatedData);
        return redirect()->route('backend.produk.index')->with('success', 'Data berhasil
diperbaharui');
    }
```

2. Tambahan **edit.blade.php** pada direktori resources/views/v_produk dan berikut script lengkap **edit.blade.php** seperti pada gambar XI.12

```
@extends('backend.v_layouts.app')
@section('content')
<!-- contentAwal -->

<div class="container-fluid">
    <div class="row">
        <div class="col-12">
            <div class="card">
                <form action="{{ route('backend.produk.update', $edit->id) }}"
method="post" enctype="multipart/form-data">
                    @method('put')
                    @csrf

                    <div class="card-body">
                        <h4 class="card-title"> {{$judul}} </h4>
                        <div class="row">

                            <div class="col-md-4">
                                <div class="form-group">
                                    <label>Foto</label>
                                    {{-- view image --}}
                                    @if ($edit->foto)
                                    <img src="{{ asset('storage/img-produk/' . $edit->foto)
}}" class="foto-preview" width="100%">
                                    <p></p>
                                    @else
                                    <img src="{{ asset('storage/img-produk/img-
default.jpg') }}" class="foto-preview" width="100%">
                                    <p></p>
                                    @endif
                                    {{-- file foto --}}
                                    <input type="file" name="foto" class="form-control
@error('foto') is-invalid @enderror" onchange="previewFoto()">
                                    @error('foto')
                                    <div class="invalid-feedback alert-danger">{{ $message
}}</div>
                                    @enderror
                                </div>
                            </div>

                            <div class="col-md-8">
                                <div class="form-group">
                                    <label>Status</label>
```

```
                                    <select name="status" class="form-control
@error('status') is-invalid @enderror">
                                        <option value="" {{ old('status', $edit->status) ==
'' ? 'selected' : '' }}> -
                                            Pilih Status -</option>
                                        <option value="1" {{ old('status', $edit->status)
== '1' ? 'selected' : '' }}>
                                            Public</option>
                                        <option value="0" {{ old('status', $edit->status)
== '0' ? 'selected' : '' }}>
                                            Blok</option>
                                    </select>
                                    @error('status')
                                    <span class="invalid-feedback alert-danger"
role="alert">
                                        {{ $message }}
                                    </span>
                                    @enderror
                                </div>

                                <div class="form-group">
                                    <label>Kategori</label>
                                    <select name="kategori_id" class="form-control
@error('kategori_id') is-invalid @enderror">
                                        <option value="" selected> - Pilih Katagori -
</option>
                                        @foreach ($kategori as $row)
                                        @if (old('kategori_id', $edit->kategori_id) ==
$row->id)
                                        <option value="{{ $row->id }}" selected> {{ $row-
>nama_kategori }} </option>
                                        @else
                                        <option value="{{ $row->id }}"> {{ $row->nama_type
}} </option>
                                        @endif
                                        @endforeach
                                    </select>
                                    @error('kategori_id')
                                    <span class="invalid-feedback alert-danger"
role="alert">
                                        {{ $message }}
                                    </span>
                                    @enderror
                                </div>

                                <div class="form-group">
                                    <label>Nama Produk</label>
                                    <input type="text" name="nama_produk" value="{{
old('nama_produk',$edit->nama_produk) }}" class="form-control @error('nama_produk') is-
invalid @enderror" placeholder="Masukkan Nama Prod">
                                    @error('nama_produk')
                                    <span class="invalid-feedback alert-danger"
role="alert">
                                        {{ $message }}
                                    </span>
                                    @enderror
                                </div>

                                <div class="form-group">
                                    <label>Detail</label><br>
                                    <textarea name="detail" class="form-control
@error('detail') is-invalid @enderror" id="ckeditor">{{ old('detail',$edit->detail)
}}</textarea>
                                    @error('detail')
                                    <span class="invalid-feedback alert-danger"
role="alert">
```

```
                                    {{ $message }}
                                </span>
                                @enderror
                            </div>

                            <div class="form-group">
                                <label>Harga</label>
                                <input type="text" onkeypress="return
hanyaAngka(event)" name="harga" value="{{ old('harga',$edit->harga) }}" class="form-control
@error('harga') is-invalid @enderror" placeholder="Masukkan Harga Produk">
                                @error('harga')
                                <span class="invalid-feedback alert-danger"
role="alert">
                                    {{ $message }}
                                </span>
                                @enderror
                            </div>

                            <div class="form-group">
                                <label>Berat</label>
                                <input type="text" onkeypress="return
hanyaAngka(event)" name="berat" value="{{ old('berat', $edit->berat) }}" class="form-
control @error('berat') is-invalid @enderror" placeholder="Masukkan Berat Produk">
                                @error('berat')
                                <span class="invalid-feedback alert-danger"
role="alert">
                                    {{ $message }}
                                </span>
                                @enderror
                            </div>

                            <div class="form-group">
                                <label>Stok</label>
                                <input type="text" onkeypress="return
hanyaAngka(event)" name="stok" value="{{ old('stok',$edit->stok) }}" class="form-control
@error('stok') is-invalid @enderror" placeholder="Masukkan Stok Produk">
                                @error('stok')
                                <span class="invalid-feedback alert-danger"
role="alert">
                                    {{ $message }}
                                </span>
                                @enderror
                            </div>
                        </div>

                    </div>
                </div>
                <div class="border-top">
                    <div class="card-body">
                        <button type="submit" class="btn btn-
primary">Perbaharui</button>
                        <a href="{{ route('backend.produk.index') }}">
                            <button type="button" class="btn btn-
secondary">Kembali</button>
                        </a>
                    </div>
                </div>
            </form>
        </div>
    </div>
</div>

<!-- contentAkhir -->
@endsection
```



Gambar XI. 12
Halaman Ubah Data Pada Produk

11.5. Hapus Data Pada Produk

Berikut logika hapus pada produk:

- Menghapus gambar utama dan semua thumbnail terkait (thumb_lg_, thumb_md_, dan thumb_sm_) pada tabel **produk**
- Demikian juga pada foto tambahan yang ada di tabel **foto_produk** juga dihapus, termasuk file fisiknya.

Dengan logika *script* dibawah ini, semua file gambar yang berkaitan dengan produk baik pada tabel produk dan foto_produk, yakni foto tambahan, gambar utama dan thumbnail, akan dihapus ketika produk dihapus dari database.

```
public function destroy($id)
    {
        $produk = Produk::findOrFail($id);
        $directory = public_path('storage/img-produk/');

        if ($produk->foto) {
            // Hapus gambar asli
            $oldImagePath = $directory . $produk->foto;
            if (file_exists($oldImagePath)) {
                unlink($oldImagePath);
            }

            // Hapus thumbnail lg
            $thumbnailLg = $directory . 'thumb_lg_' . $produk->foto;
            if (file_exists($thumbnailLg)) {
                unlink($thumbnailLg);
            }

            // Hapus thumbnail md
            $thumbnailMd = $directory . 'thumb_md_' . $produk->foto;
            if (file_exists($thumbnailMd)) {
                unlink($thumbnailMd);
            }
```

```
            // Hapus thumbnail sm
            $thumbnailSm = $directory . 'thumb_sm_' . $produk->foto;
            if (file_exists($thumbnailSm)) {
                unlink($thumbnailSm);
            }
        }
```

Berikut script lengkap pada controller **ProdukController**

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Produk;
use App\Models\Kategori;
use App\Models\FotoProduk;
use App\Helpers\ImageHelper;

class ProdukController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        $produk = Produk::orderBy('updated_at', 'desc')->get();
        return view('backend.v_produk.index', [
            'judul' => 'Data Produk',
            'index' => $produk
        ]);
    }

    /**
     * Show the form for creating a new resource.
     */
    public function create()
    {
        $kategori = Kategori::orderBy('nama_kategori', 'asc')->get();
        return view('backend.v_produk.create', [
            'judul' => 'Tambah Produk',
            'kategori' => $kategori
        ]);
    }

    /**
     * Store a newly created resource in storage.
     */
    public function store(Request $request)
    {
        $validatedData = $request->validate([
            'kategori_id' => 'required',
            'nama_produk' => 'required|max:255|unique:produk',
            'detail' => 'required',
            'harga' => 'required',
            'berat' => 'required',
            'stok' => 'required',
            'foto' => 'required|image|mimes:jpeg,jpg,png,gif|file|max:1024',
        ], $messages = [
            'foto.image' => 'Format gambar gunakan file dengan ekstensi jpeg, jpg, png,
atau gif.',
            'foto.max' => 'Ukuran file gambar Maksimal adalah 1024 KB.'
        ]);
        $validatedData['user_id'] = auth()->id();
        $validatedData['status'] = 0;
```

```
        if ($request->file('foto')) {
            $file = $request->file('foto');
            $extension = $file->getClientOriginalExtension();
            $originalFileName = date('YmdHis') . '_' . uniqid() . '.' . $extension;
            $directory = 'storage/img-produk/';

            // Simpan gambar asli
            $fileName = ImageHelper::uploadAndResize($file, $directory, $originalFileName);
            $validatedData['foto'] = $fileName;

            // create thumbnail 1 (lg)
            $thumbnailLg = 'thumb_lg_' . $originalFileName;
            ImageHelper::uploadAndResize($file, $directory, $thumbnailLg, 800, null);

            // create thumbnail 2 (md)
            $thumbnailMd = 'thumb_md_' . $originalFileName;
            ImageHelper::uploadAndResize($file, $directory, $thumbnailMd, 500, 519);

            // create thumbnail 3 (sm)
            $thumbnailSm = 'thumb_sm_' . $originalFileName;
            ImageHelper::uploadAndResize($file, $directory, $thumbnailSm, 100, 110);

            // Simpan nama file asli di database
            $validatedData['foto'] = $originalFileName;
        }

        Produk::create($validatedData, $messages);
        return redirect()->route('backend.produk.index')->with('success', 'Data berhasil
tersimpan');
    }

    /**
     * Display the specified resource.
     */
    public function show(string $id)
    {
        $produk = Produk::with('fotoProduk')->findOrFail($id);
        $kategori = Kategori::orderBy('nama_kategori', 'asc')->get();
        return view('backend.v_produk.show', [
            'judul' => 'Detail Produk',
            'show' => $produk,
            'kategori' => $kategori
        ]);
    }

    /**
     * Show the form for editing the specified resource.
     */
    public function edit(string $id)
    {
        $produk = Produk::findOrFail($id);
        $kategori = Kategori::orderBy('nama_kategori', 'asc')->get();
        return view('backend.v_produk.edit', [
            'judul' => 'Ubah Produk',
            'edit' => $produk,
            'kategori' => $kategori
        ]);
    }

    /**
     * Update the specified resource in storage.
     */
    public function update(Request $request, string $id)
    {
        //ddd($request);
        $produk = Produk::findOrFail($id);
```

```
        $rules = [
            'nama_produk' => 'required|max:255|unique:produk,nama_produk,' . $id,
            'kategori_id' => 'required',
            'status' => 'required',
            'detail' => 'required',
            'harga' => 'required',
            'berat' => 'required',
            'stok' => 'required',
            'foto' => 'image|mimes:jpeg,jpg,png,gif|file|max:1024',
        ];
        $messages = [
            'foto.image' => 'Format gambar gunakan file dengan ekstensi jpeg, jpg, png,
atau gif.',
            'foto.max' => 'Ukuran file gambar Maksimal adalah 1024 KB.'
        ];
        $validatedData['user_id'] = auth()->id();
        $validatedData = $request->validate($rules, $messages);

        if ($request->file('foto')) {
            //hapus gambar lama
            if ($produk->foto) {
                $oldImagePath = public_path('storage/img-produk/') . $produk->foto;
                if (file_exists($oldImagePath)) {
                    unlink($oldImagePath);
                }
                $oldThumbnailLg = public_path('storage/img-produk/') . 'thumb_lg_' .
$produk->foto;
                if (file_exists($oldThumbnailLg)) {
                    unlink($oldThumbnailLg);
                }
                $oldThumbnailMd = public_path('storage/img-produk/') . 'thumb_md_' .
$produk->foto;
                if (file_exists($oldThumbnailMd)) {
                    unlink($oldThumbnailMd);
                }
                $oldThumbnailSm = public_path('storage/img-produk/') . 'thumb_sm_' .
$produk->foto;
                if (file_exists($oldThumbnailSm)) {
                    unlink($oldThumbnailSm);
                }
            }
            $file = $request->file('foto');
            $extension = $file->getClientOriginalExtension();
            $originalFileName = date('YmdHis') . '_' . uniqid() . '.' . $extension;
            $directory = 'storage/img-produk/';

            // Simpan gambar asli
            $fileName = ImageHelper::uploadAndResize($file, $directory, $originalFileName);
            $validatedData['foto'] = $fileName;

            // create thumbnail 1 (lg)
            $thumbnailLg = 'thumb_lg_' . $originalFileName;
            ImageHelper::uploadAndResize($file, $directory, $thumbnailLg, 800, null);

            // create thumbnail 2 (md)
            $thumbnailMd = 'thumb_md_' . $originalFileName;
            ImageHelper::uploadAndResize($file, $directory, $thumbnailMd, 500, 519);

            // create thumbnail 3 (sm)
            $thumbnailSm = 'thumb_sm_' . $originalFileName;
            ImageHelper::uploadAndResize($file, $directory, $thumbnailSm, 100, 110);

            // Simpan nama file asli di database
            $validatedData['foto'] = $originalFileName;
        }
```

```
        $produk->update($validatedData);
        return redirect()->route('backend.produk.index')->with('success', 'Data berhasil
diperbaharui');
    }

    /**
     * Remove the specified resource from storage.
     */
    public function destroy($id)
    {
        $produk = Produk::findOrFail($id);
        $directory = public_path('storage/img-produk/');

        if ($produk->foto) {
            // Hapus gambar asli
            $oldImagePath = $directory . $produk->foto;
            if (file_exists($oldImagePath)) {
                unlink($oldImagePath);
            }

            // Hapus thumbnail lg
            $thumbnailLg = $directory . 'thumb_lg_' . $produk->foto;
            if (file_exists($thumbnailLg)) {
                unlink($thumbnailLg);
            }

            // Hapus thumbnail md
            $thumbnailMd = $directory . 'thumb_md_' . $produk->foto;
            if (file_exists($thumbnailMd)) {
                unlink($thumbnailMd);
            }

            // Hapus thumbnail sm
            $thumbnailSm = $directory . 'thumb_sm_' . $produk->foto;
            if (file_exists($thumbnailSm)) {
                unlink($thumbnailSm);
            }
        }

        // Hapus foto produk lainnya di tabel foto_produk
        $fotoProduks = FotoProduk::where('produk_id', $id)->get();
        foreach ($fotoProduks as $fotoProduk) {
            $fotoPath = $directory . $fotoProduk->foto;
            if (file_exists($fotoPath)) {
                unlink($fotoPath);
            }
            $fotoProduk->delete();
        }

        $produk->delete();

        return redirect()->route('backend.produk.index')->with('success', 'Data berhasil
dihapus');
    }


    // Method untuk menyimpan foto tambahan
    public function storeFoto(Request $request)
    {
        // Validasi input
        $request->validate([
            'produk_id' => 'required|exists:produk,id',
            'foto_produk.*' => 'image|mimes:jpeg,jpg,png,gif|file|max:1024',
        ]);
```

```
        if ($request->hasFile('foto_produk')) {
            foreach ($request->file('foto_produk') as $file) {
                // Buat nama file yang unik
                $extension = $file->getClientOriginalExtension();
                $filename = date('YmdHis') . '_' . uniqid() . '.' . $extension;
                $directory = 'storage/img-produk/';

                // Simpan dan resize gambar menggunakan ImageHelper
                ImageHelper::uploadAndResize($file, $directory, $filename, 800, null);
                // Simpan data ke database
                FotoProduk::create([
                    'produk_id' => $request->produk_id,
                    'foto' => $filename,
                ]);
            }
        }
        return redirect()->route('backend.produk.show', $request->produk_id)
            ->with('success', 'Foto berhasil ditambahkan.');
    }


    // Method untuk menghapus foto
    public function destroyFoto($id)
    {
        $foto = FotoProduk::findOrFail($id);
        $produkId = $foto->produk_id;

        // Hapus file gambar dari storage
        $imagePath = public_path('storage/img-produk/') . $foto->foto;
        if (file_exists($imagePath)) {
            unlink($imagePath);
        }
        // Hapus record dari database
        $foto->delete();

        return redirect()->route('backend.produk.show', $produkId)
            ->with('success', 'Foto berhasil dihapus.');
    }
}
```

Latihan Mandiri 11:
Update Portofolion sertifikasi kompetensi, Impentasikan Unit Kompetensi Software Development pada **Mengimplementasikan pemrograman berorientasi objek & Menggunakan Library atau Komponen Pre-Existing**.

Minggu Ke-12
Laporan Data Master

Pada pembuatan manajemen laporan data master, kita tidak membuat controller baru, melainkan hanya menambahkan fungsi seperti formProduk dan cetakProduk. Fungsi formProduk digunakan untuk menampilkan tampilan form filter laporan berdasarkan tanggal awal dan tanggal akhir. Sedangkan cetakProduk digunakan untuk menampilkan hasil filter dari form laporan tersebut.

12.1. Manajemen Laporan Data User

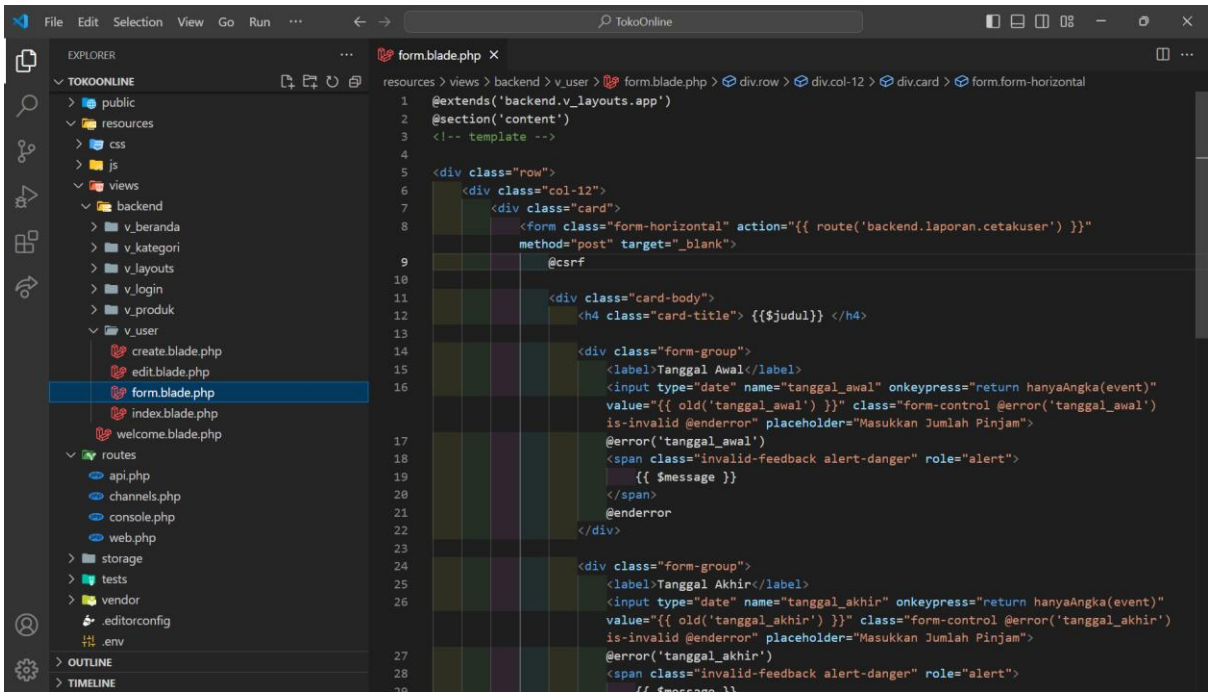
1. Pada contoller **UserController**, kita tambahkan *function* formUser dan cetakUser sebagai berikut

```
    public function formUser()
    {
        return view('backend.v_user.form', [
            'judul' => 'Laporan Data User',
        ]);
    }

    public function cetakUser(Request $request)
    {
        // Menambahkan aturan validasi
        $request->validate([
            'tanggal_awal' => 'required|date',
            'tanggal_akhir' => 'required|date|after_or_equal:tanggal_awal',
        ], [
            'tanggal_awal.required' => 'Tanggal Awal harus diisi.',
            'tanggal_akhir.required' => 'Tanggal Akhir harus diisi.',
            'tanggal_akhir.after_or_equal' => 'Tanggal Akhir harus lebih besar atau sama
dengan Tanggal Awal.',
        ]);

        $tanggalAwal = $request->input('tanggal_awal');
        $tanggalAkhir = $request->input('tanggal_akhir');

        $query =  User::whereBetween('created_at', [$tanggalAwal, $tanggalAkhir])
            ->orderBy('id', 'desc');

        $user = $query->get();
        return view('backend.v_user.cetak', [
            'judul' => 'Laporan User',
            'tanggalAwal' => $tanggalAwal,
            'tanggalAkhir' => $tanggalAkhir,
            'cetak' => $user
        ]);
    }
```

2. Sehingga pada view kita tambahkan file **form.blade.php** pada direktori resources/views/v_user seperti pada gambar

```
@extends('backend.v_layouts.app')
@section('content')
<!-- template -->

<div class="row">
    <div class="col-12">
        <div class="card">
            <form class="form-horizontal" action="{{ route('backend.laporan.cetakuser') }}"
method="post" target="_blank">
                @csrf
```

```
                <div class="card-body">
                    <h4 class="card-title"> {{$judul}} </h4>

                    <div class="form-group">
                        <label>Tanggal Awal</label>
                        <input type="date" name="tanggal_awal" onkeypress="return
hanyaAngka(event)" value="{{ old('tanggal_awal') }}" class="form-control
@error('tanggal_awal') is-invalid @enderror" placeholder="Masukkan Jumlah Pinjam">
                        @error('tanggal_awal')
                        <span class="invalid-feedback alert-danger" role="alert">
                            {{ $message }}
                        </span>
                        @enderror
                    </div>

                    <div class="form-group">
                        <label>Tanggal Akhir</label>
                        <input type="date" name="tanggal_akhir" onkeypress="return
hanyaAngka(event)" value="{{ old('tanggal_akhir') }}" class="form-control
@error('tanggal_akhir') is-invalid @enderror" placeholder="Masukkan Jumlah Pinjam">
                        @error('tanggal_akhir')
                        <span class="invalid-feedback alert-danger" role="alert">
                            {{ $message }}
                        </span>
                        @enderror
                    </div>

                    <br>
                    <button type="submit" class="btn btn-primary">Cetak</button>

            </form>
        </div>
    </div>
</div>

<!-- end template-->
@endsection
```



Gambar XII. 1
Form Pada Manajemen Laporan Data User

3. Dan pada view kita tambahkan file **cetak.blade.php** pada direktori resources/views/v_user

```
<style>
    table {
        border-collapse: collapse;
        width: 100%;
        border: 1px solid #ccc;
    }

    table tr td {
        padding: 6px;
        font-weight: normal;
        border: 1px solid #ccc;
    }

    table th {
        border: 1px solid #ccc;
    }
</style>
<table>
    <!-- <tr>
        <td align="center">
            <img src="{{ asset('images/header.png') }}" width="50%">
        </td>
    </tr> -->
    <tr>
        <td align="left">
            Perihal : {{ $judul }} <br>
            Tanggal Awal: {{ $tanggalAwal }} s/d Tanggal Akhir: {{ $tanggalAkhir }}
        </td>
    </tr>
</table>
<p></p>
<table>
    <thead>
        <tr>
            <th>No</th>
            <th>Email</th>
            <th>Nama</th>
            <th>Role</th>
            <th>Status</th>
        </tr>
    </thead>
    <tbody>
        @foreach ($cetak as $row)
        <tr>
            <td> {{ $loop->iteration }} </td>
            <td> {{$row->nama}} </td>
            <td> {{$row->email}} </td>
            <td>
                @if ($row->role == 1)
                Super Admin
                @elseif($row->role == 0)
                Admin
                @endif
            </td>
            <td>
                @if ($row->status ==1)
                Aktif
                @elseif($row->status ==0)
                NonAktif
                @endif
            </td>
        </tr>
        @endforeach
```

```
    </tbody>
</table>

<script>
    window.onload = function() {
        printStruk();
    }

    function printStruk() {
        window.print();
    }
</script>
```

4. Tambahkan *script* Pada routes**web.php**

```
Route::get('backend/laporan/formuser', [UserController::class, 'formUser'])-
>name('backend.laporan.formuser')->middleware('auth');
Route::post('backend/laporan/cetakuser', [UserController::class, 'cetakUser'])-
>name('backend.laporan.cetakuser')->middleware('auth');
```

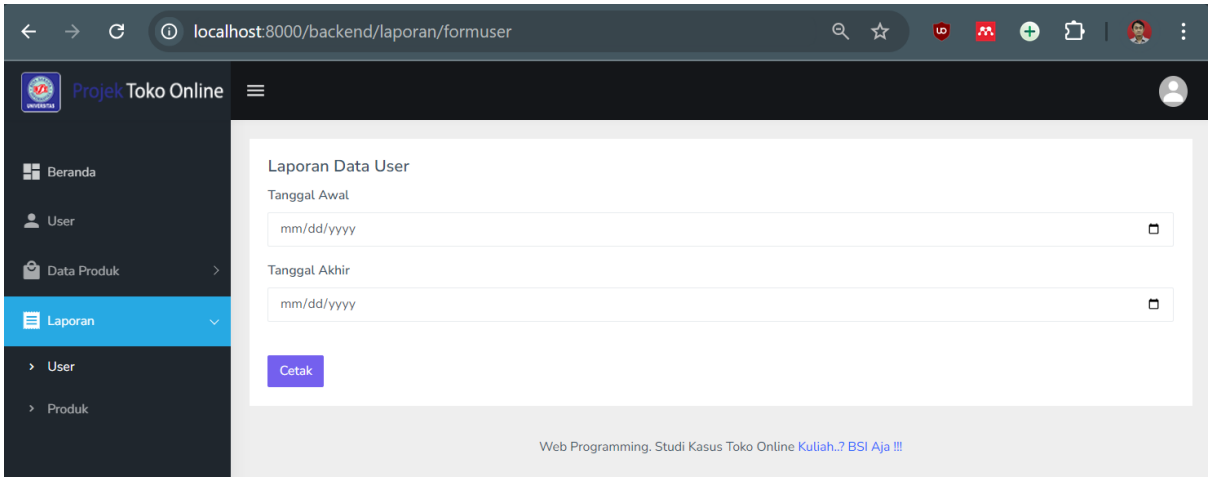
Dan berikut *script* lengkap dari routes**web.php**

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\BerandaController;
use App\Http\Controllers\LoginController;
use App\Http\Controllers\UserController;
use App\Http\Controllers\KategoriController;
use App\Http\Controllers\ProdukController;

Route::get('/', function () {
    // return view('welcome');
    return redirect()->route('backend.login');
});
Route::get('backend/beranda', [BerandaController::class, 'berandaBackend'])-
>name('backend.beranda')->middleware('auth');

Route::get('backend/login', [LoginController::class, 'loginBackend'])-
>name('backend.login');
Route::post('backend/login', [LoginController::class, 'authenticateBackend'])-
>name('backend.login');
Route::post('backend/logout', [LoginController::class, 'logoutBackend'])-
>name('backend.logout');

// Route untuk User
// Route::resource('backend/user', UserController::class)->middleware('auth');
Route::resource('backend/user', UserController::class, ['as' => 'backend'])-
>middleware('auth');
// Route untuk laporan user
Route::get('backend/laporan/formuser', [UserController::class, 'formUser'])-
>name('backend.laporan.formuser')->middleware('auth');
Route::post('backend/laporan/cetakuser', [UserController::class, 'cetakUser'])-
>name('backend.laporan.cetakuser')->middleware('auth');

// Route untuk Kategori
Route::resource('backend/kategori', KategoriController::class, ['as' => 'backend'])-
>middleware('auth');

// Route untuk Produk
Route::resource('backend/produk', ProdukController::class, ['as' => 'backend'])-
>middleware('auth');
// Route untuk menambahkan foto
Route::post('foto-produk/store', [ProdukController::class, 'storeFoto'])-
>name('backend.foto_produk.store')->middleware('auth');
// Route untuk menghapus foto
```

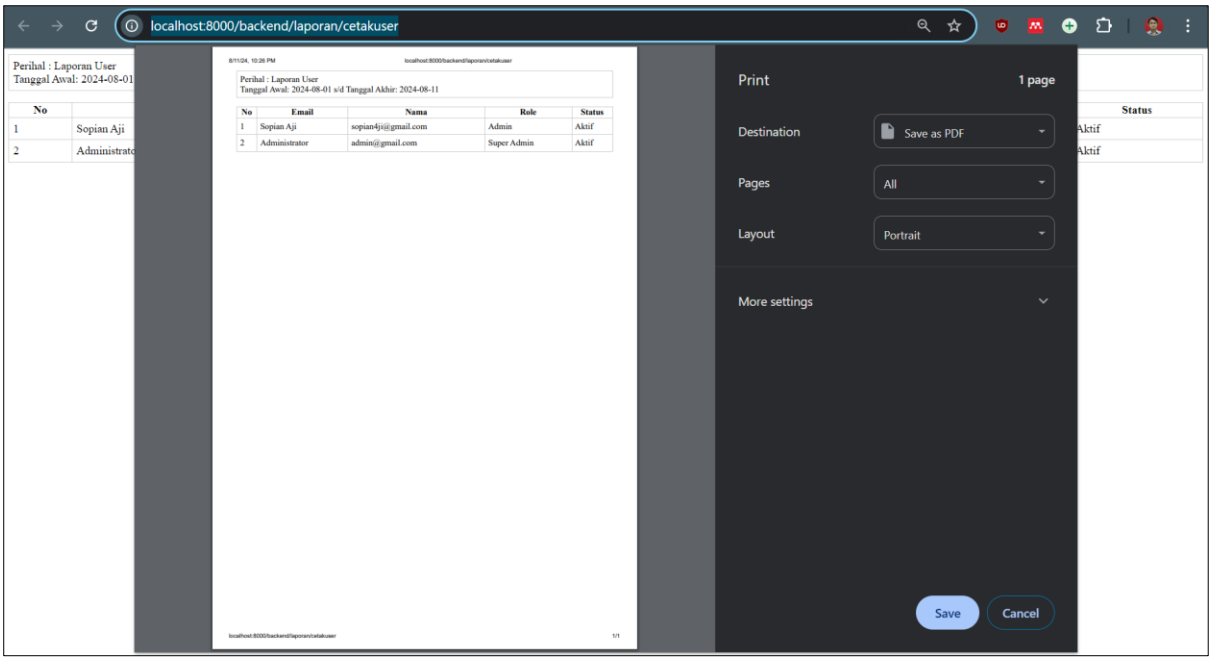
```
Route::delete('foto-produk/{id}', [ProdukController::class, 'destroyFoto'])-
>name('backend.foto_produk.destroy')->middleware('auth');
```

5. Tambahkan pada sidebar manajemen laporan data **User** yakni **app.blade.php** pada direktori resources\views\backend\ v_layouts\backend\. Jika berhasil, tampilan akan seperti pada Gambar XII.2 untuk melihat filter form laporan. Jika tombol cetak diklik, halaman browser baru akan terbuka dan menampilkan cetakan data yang diinginkan, seperti pada gambar XII.3

```
<li class="sidebar-item"> <a class="sidebar-link has-arrow waves-effect waves-dark"
href="javascript:void(0)" aria-expanded="false"><i class="mdi mdi-receipt"></i><span
class="hide-menu">Laporan </span></a>
                        <ul aria-expanded="false" class="collapse  first-level">
                            <li class="sidebar-item"><a href="{{
route('backend.laporan.formuser') }}" class="sidebar-link"><i class="mdi mdi-chevron-
right"></i><span class="hide-menu"> User </span></a></li>
                            <li class="sidebar-item"><a href="error-404.html"
class="sidebar-link"><i class="mdi mdi-chevron-right"></i><span class="hide-menu"> Produk
</span></a></li>

                        </ul>
                    </li>
```



Gambar XII. 2
Halaman Laporan Data User



Gambar XII. 3
Cetak Data User

Sehingga berikut *script* lengkap pada **UserController**:

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\User;
use Illuminate\Support\Facades\Hash;
use App\Helpers\ImageHelper;

class UserController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        $user = User::orderBy('updated_at', 'desc')->get();
        return view('backend.v_user.index', [
            'judul' => 'Data User',
            'index' => $user
        ]);
    }

    /**
     * Show the form for creating a new resource.
     */
    public function create()
    {
        return view('backend.v_user.create', [
            'judul' => 'Tambah User',
        ]);
    }

    /**
     * Store a newly created resource in storage.
     */
    public function store(Request $request)
```

```
    {
        $validatedData = $request->validate([
            'nama' => 'required|max:255',
            'email' => 'required|max:255|email|unique:user',
            'role' => 'required',
            'hp' => 'required|min:10|max:13',
            'password' => 'required|min:4|confirmed',
            'foto' => 'image|mimes:jpeg,jpg,png,gif|file|max:1024',
        ], $messages = [
            'foto.image' => 'Format gambar gunakan file dengan ekstensi jpeg, jpg, png,
atau gif.',
            'foto.max' => 'Ukuran file gambar Maksimal adalah 1024 KB.'
        ]);
        $validatedData['status'] = 0;

        // menggunakan ImageHelper
        if ($request->file('foto')) {
            $file = $request->file('foto');
            $extension = $file->getClientOriginalExtension();
            $originalFileName = date('YmdHis') . '_' . uniqid() . '.' . $extension;
            $directory = 'storage/img-user/';
            // Simpan gambar dengan ukuran yang ditentukan
            ImageHelper::uploadAndResize($file, $directory, $originalFileName, 385, 400);
// null (jika tinggi otomatis)
            // Simpan nama file asli di database
            $validatedData['foto'] = $originalFileName;
        }

        // password
        $password = $request->input('password');
        $pattern = '/^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[\W_]).+$/';
        // huruf kecil ([a-z]), huruf besar ([A-Z]), dan angka (\d) (?=.*[\W_]) simbol
karakter (non-alphanumeric)
        if (preg_match($pattern, $password)) {
            $validatedData['password'] = Hash::make($validatedData['password']);
            User::create($validatedData, $messages);
            return redirect()->route('backend.user.index')->with('success', 'Data berhasil
tersimpan');
        } else {
            return redirect()->back()->withErrors(['password' => 'Password harus terdiri
dari kombinasi huruf besar, huruf kecil, angka, dan simbol karakter.']);
        }
    }

    /**
     * Display the specified resource.
     */
    public function show(string $id)
    {
        //
    }

    /**
     * Show the form for editing the specified resource.
     */
    public function edit(string $id)
    {
        $user = User::findOrFail($id);
        return view('backend.v_user.edit', [
            'judul' => 'Ubah User',
            'edit' => $user
        ]);
    }

    /**
     * Update the specified resource in storage.
```

```
     */
    public function update(Request $request, string $id)
    {
        //ddd($request);
        $user = User::findOrFail($id);
        $rules = [
            'nama' => 'required|max:255',
            'role' => 'required',
            'status' => 'required',
            'hp' => 'required|min:10|max:13',
            'foto' => 'image|mimes:jpeg,jpg,png,gif|file|max:1024',
        ];
        $messages = [
            'foto.image' => 'Format gambar gunakan file dengan ekstensi jpeg, jpg, png,
atau gif.',
            'foto.max' => 'Ukuran file gambar Maksimal adalah 1024 KB.'
        ];

        if ($request->email != $user->email) {
            $rules['email'] = 'required|max:255|email|unique:user';
        }
        $validatedData = $request->validate($rules, $messages);

        // menggunakan ImageHelper
        if ($request->file('foto')) {
            //hapus gambar lama
            if ($user->foto) {
                $oldImagePath = public_path('storage/img-user/') . $user->foto;
                if (file_exists($oldImagePath)) {
                    unlink($oldImagePath);
                }
            }
            $file = $request->file('foto');
            $extension = $file->getClientOriginalExtension();
            $originalFileName = date('YmdHis') . '_' . uniqid() . '.' . $extension;
            $directory = 'storage/img-user/';
            // Simpan gambar dengan ukuran yang ditentukan
            ImageHelper::uploadAndResize($file, $directory, $originalFileName, 385, 400);
// null (jika tinggi otomatis)
            // Simpan nama file asli di database
            $validatedData['foto'] = $originalFileName;
        }

        $user->update($validatedData);
        return redirect()->route('backend.user.index')->with('success', 'Data berhasil
diperbaharui');
    }

    /**
     * Remove the specified resource from storage.
     */
    public function destroy(string $id)
    {
        $user = user::findOrFail($id);
        if ($user->foto) {
            $oldImagePath = public_path('storage/img-user/') . $user->foto;
            if (file_exists($oldImagePath)) {
                unlink($oldImagePath);
            }
        }
        $user->delete();
        return redirect()->route('backend.user.index')->with('success', 'Data berhasil
dihapus');
    }

    public function formUser()
```

```
    {
        return view('backend.v_user.form', [
            'judul' => 'Laporan Data User',
        ]);
    }

    public function cetakUser(Request $request)
    {
        // Menambahkan aturan validasi
        $request->validate([
            'tanggal_awal' => 'required|date',
            'tanggal_akhir' => 'required|date|after_or_equal:tanggal_awal',
        ], [
            'tanggal_awal.required' => 'Tanggal Awal harus diisi.',
            'tanggal_akhir.required' => 'Tanggal Akhir harus diisi.',
            'tanggal_akhir.after_or_equal' => 'Tanggal Akhir harus lebih besar atau sama
dengan Tanggal Awal.',
        ]);

        $tanggalAwal = $request->input('tanggal_awal');
        $tanggalAkhir = $request->input('tanggal_akhir');

        $query =  User::whereBetween('created_at', [$tanggalAwal, $tanggalAkhir])
            ->orderBy('id', 'desc');

        $user = $query->get();
        return view('backend.v_user.cetak', [
            'judul' => 'Laporan User',
            'tanggalAwal' => $tanggalAwal,
            'tanggalAkhir' => $tanggalAkhir,
            'cetak' => $user
        ]);
    }
}
```

12.2. laporan Manajemen Produk

1. Pada contoller **ProdukController**, kita tambahkan function **formProduk** dan **cetakProduk** sebagai berikut

```
    // Method untuk Form Laporan Produk
    public function formProduk()
    {
        return view('backend.v_produk.form', [
            'judul' => 'Laporan Data Produk',
        ]);
    }

    // Method untuk Cetak Laporan Produk
    public function cetakProduk(Request $request)
    {
        // Menambahkan aturan validasi
        $request->validate([
            'tanggal_awal' => 'required|date',
            'tanggal_akhir' => 'required|date|after_or_equal:tanggal_awal',
        ], [
            'tanggal_awal.required' => 'Tanggal Awal harus diisi.',
            'tanggal_akhir.required' => 'Tanggal Akhir harus diisi.',
            'tanggal_akhir.after_or_equal' => 'Tanggal Akhir harus lebih besar atau sama
dengan Tanggal Awal.',
        ]);

        $tanggalAwal = $request->input('tanggal_awal');
        $tanggalAkhir = $request->input('tanggal_akhir');
```

```
        $query =  Produk::whereBetween('updated_at', [$tanggalAwal, $tanggalAkhir])
            ->orderBy('id', 'desc');

        $produk = $query->get();
        return view('backend.v_produk.cetak', [
            'judul' => 'Laporan Produk',
            'tanggalAwal' => $tanggalAwal,
            'tanggalAkhir' => $tanggalAkhir,
            'cetak' => $produk
        ]);
    }
```

2. Sehingga pada view kita tambahkan file **form.blade.php** pada direktori resources/views/v_produk

```
@extends('backend.v_layouts.app')
@section('content')
<!-- template -->

<div class="row">
    <div class="col-12">
        <div class="card">
            <form class="form-horizontal" action="{{ route('backend.laporan.cetakproduk')
}}" method="post" target="_blank">
                @csrf

                <div class="card-body">
                    <h4 class="card-title"> {{$judul}} </h4>

                    <div class="form-group">
                        <label>Tanggal Awal</label>
                        <input type="date" name="tanggal_awal" onkeypress="return
hanyaAngka(event)" value="{{ old('tanggal_awal') }}" class="form-control
@error('tanggal_awal') is-invalid @enderror" placeholder="Masukkan Jumlah Pinjam">
                        @error('tanggal_awal')
                        <span class="invalid-feedback alert-danger" role="alert">
                            {{ $message }}
                        </span>
                        @enderror
                    </div>

                    <div class="form-group">
                        <label>Tanggal Akhir</label>
                        <input type="date" name="tanggal_akhir" onkeypress="return
hanyaAngka(event)" value="{{ old('tanggal_akhir') }}" class="form-control
@error('tanggal_akhir') is-invalid @enderror" placeholder="Masukkan Jumlah Pinjam">
                        @error('tanggal_akhir')
                        <span class="invalid-feedback alert-danger" role="alert">
                            {{ $message }}
                        </span>
                        @enderror
                    </div>

                    <br>
                    <button type="submit" class="btn btn-primary">Cetak</button>

            </form>
        </div>
    </div>
</div>


<!-- end template-->
@endsection
```

3. Dan pada view kita tambahkan file **cetak.blade.php** pada direktori resources/views/v_produk

```
<style>
    table {
        border-collapse: collapse;
        width: 100%;
        border: 1px solid #ccc;
    }

    table tr td {
        padding: 6px;
        font-weight: normal;
        border: 1px solid #ccc;
    }

    table th {
        border: 1px solid #ccc;
    }
</style>
<table>
    <!-- <tr>
        <td align="center">
            <img src="{{ asset('images/header.png') }}" width="50%">
        </td>
    </tr> -->
    <tr>
        <td align="left">
            Perihal : {{ $judul }} <br>
            Tanggal Awal: {{ $tanggalAwal }} s/d Tanggal Akhir: {{ $tanggalAkhir }}
        </td>
    </tr>
</table>
<p></p>
<table>
    <thead>
        <tr>
            <th>No</th>
            <th>Kategori</th>
            <th>Status</th>
            <th>Nama Produk</th>
            <th>Harga</th>
            <th>Stok</th>
        </tr>
    </thead>
    <tbody>
        @foreach ($cetak as $row)
        <tr>
            <td> {{ $loop->iteration }}</td>
            <td> {{ $row->kategori->nama_kategori }} </td>
            <td>
                @if ($row->status ==1)
                Publis
                @elseif($row->status ==0)
                Blok
                @endif
            </td>
            <td> {{ $row->nama_produk }} </td>
            <td> Rp. {{ number_format($row->harga, 0, ',', '.') }} </td>
            <td> {{ $row->stok }} </td>
        </tr>
        @endforeach

    </tbody>
</table>
```

```
<script>
    window.onload = function() {
        printStruk();
    }

    function printStruk() {
        window.print();
    }
</script>
```

4. Tambahkan *script* Pada routes\web.php

```
Route::get('backend/laporan/formproduk', [ProdukController::class, 'formProduk'])-
>name('backend.laporan.formproduk')->middleware('auth');
Route::post('backend/laporan/cetakproduk', [ProdukController::class, 'cetakProduk'])-
>name('backend.laporan.cetakproduk')->middleware('auth');
```

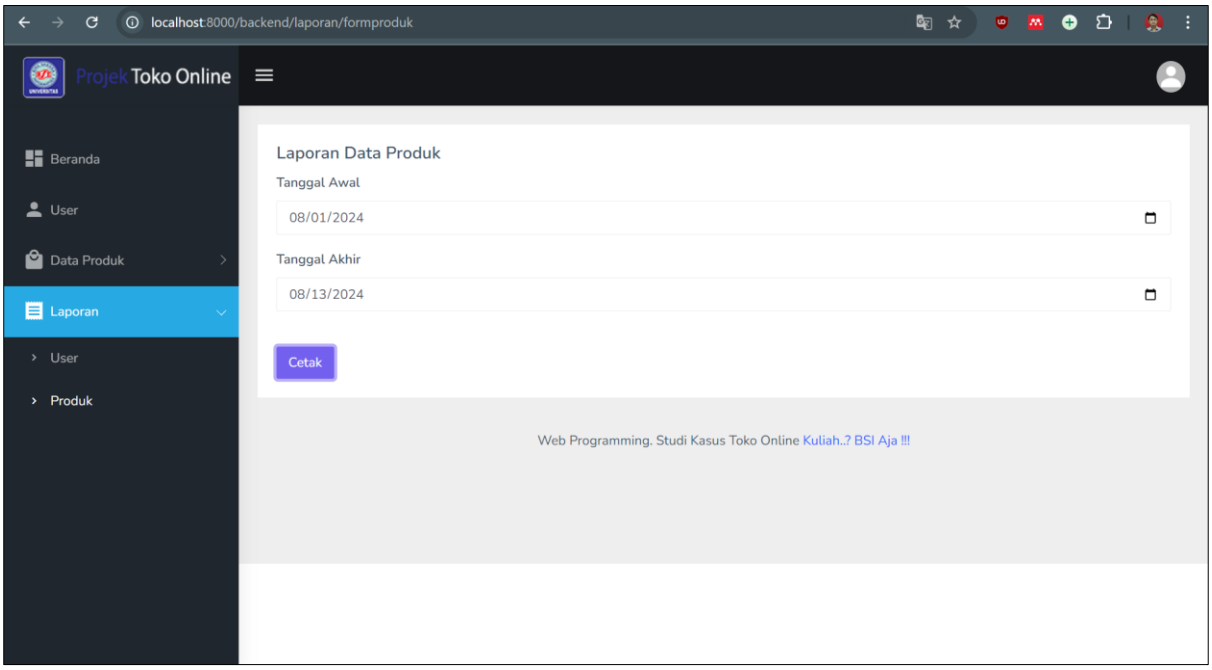
Dan berikut *script* lengkap dari routes\web.php

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\BerandaController;
use App\Http\Controllers\LoginController;
use App\Http\Controllers\UserController;
use App\Http\Controllers\KategoriController;
use App\Http\Controllers\ProdukController;

Route::get('/', function () {
    // return view('welcome');
    return redirect()->route('backend.login');
});
Route::get('backend/beranda', [BerandaController::class, 'berandaBackend'])-
>name('backend.beranda')->middleware('auth');

Route::get('backend/login', [LoginController::class, 'loginBackend'])-
>name('backend.login');
Route::post('backend/login', [LoginController::class, 'authenticateBackend'])-
>name('backend.login');
Route::post('backend/logout', [LoginController::class, 'logoutBackend'])-
>name('backend.logout');

// Route untuk User
// Route::resource('backend/user', UserController::class)->middleware('auth');
Route::resource('backend/user', UserController::class, ['as' => 'backend'])-
>middleware('auth');
// Route untuk laporan user
Route::get('backend/laporan/formuser', [UserController::class, 'formUser'])-
>name('backend.laporan.formuser')->middleware('auth');
Route::post('backend/laporan/cetakuser', [UserController::class, 'cetakUser'])-
>name('backend.laporan.cetakuser')->middleware('auth');

// Route untuk Kategori
Route::resource('backend/kategori', KategoriController::class, ['as' => 'backend'])-
>middleware('auth');

// Route untuk Produk
Route::resource('backend/produk', ProdukController::class, ['as' => 'backend'])-
>middleware('auth');
// Route untuk menambahkan foto
Route::post('foto-produk/store', [ProdukController::class, 'storeFoto'])-
>name('backend.foto_produk.store')->middleware('auth');
// Route untuk menghapus foto
Route::delete('foto-produk/{id}', [ProdukController::class, 'destroyFoto'])-
>name('backend.foto_produk.destroy')->middleware('auth');
// Route untuk laporan produk
```

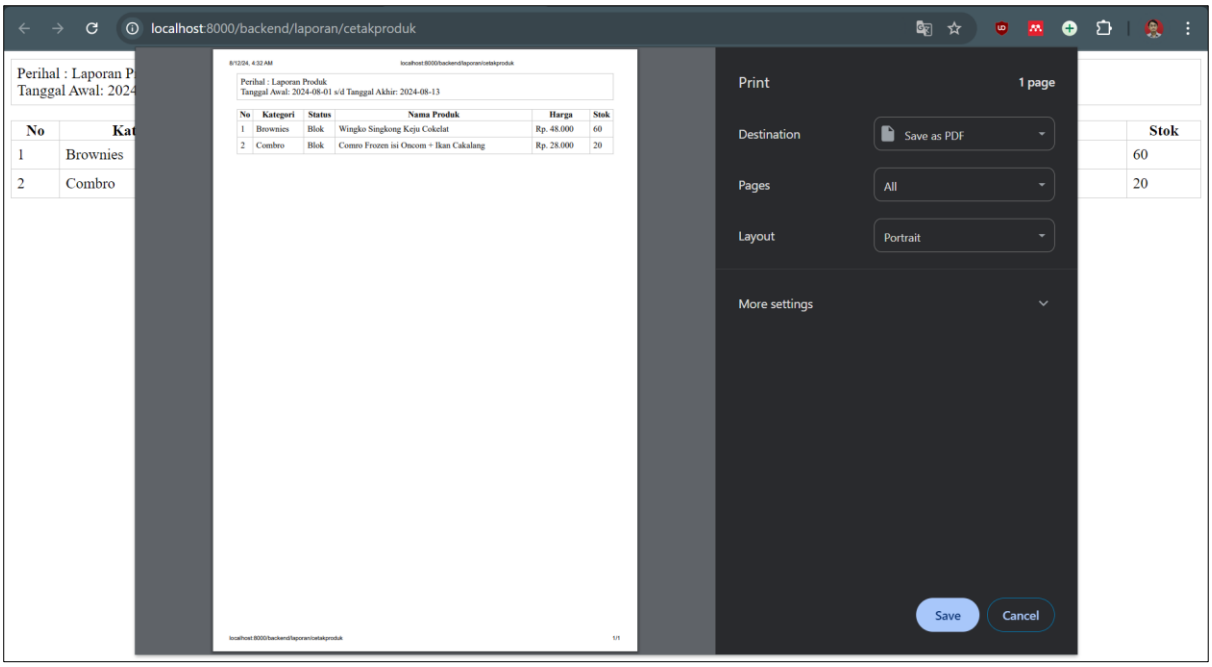
```
Route::get('backend/laporan/formproduk', [ProdukController::class, 'formProduk'])-
>name('backend.laporan.formproduk')->middleware('auth');
Route::post('backend/laporan/cetakproduk', [ProdukController::class, 'cetakProduk'])-
>name('backend.laporan.cetakproduk')->middleware('auth');
```

5. Tambahkan pada sidebar manajemen laporan **Produk** yakni **app.blade.php** pada direktori resources\views\backend\ v_layouts\backend\. Jika berhasil, tampilan akan seperti pada Gambar XII.4 untuk melihat filter form laporan. Jika tombol cetak diklik, halaman browser baru akan terbuka dan menampilkan cetakan data yang diinginkan, seperti pada gambar XII.5

```
<li class="sidebar-item"> <a class="sidebar-link has-arrow waves-effect waves-dark"
href="javascript:void(0)" aria-expanded="false"><i class="mdi mdi-receipt"></i><span
class="hide-menu">Laporan </span></a>
                            <ul aria-expanded="false" class="collapse  first-level">
                                <li class="sidebar-item"><a href="{{
route('backend.laporan.formuser') }}" class="sidebar-link"><i class="mdi mdi-chevron-
right"></i><span class="hide-menu"> User </span></a></li>
                                <li class="sidebar-item"><a href="{{
route('backend.laporan.formproduk') }}" class="sidebar-link"><i class="mdi mdi-chevron-
right"></i><span class="hide-menu"> Produk </span></a></li>

                            </ul>
                        </li>
```



Gambar XII. 4
Manajemen Laporan Data Produk



Gambar XII. 5
Cetak Laporan Data Produk

Sehingga berikut *script* lengkap pada **app.blade.php**:

```
<!DOCTYPE html>
<html dir="ltr" lang="en">

<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <!-- Tell the browser to be responsive to screen width -->
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="">
    <meta name="author" content="">
    <!-- Favicon icon -->
    <link rel="icon" type="image/png" sizes="16x16" href="{{
asset('image/icon_univ_bsi.png') }}">
    <title>tokoonline</title>
    <!-- Custom CSS -->
    <link rel="stylesheet" type="text/css" href="{{ asset('backend/extra-
libs/multicheck/multicheck.css') }}">
    <link href="{{ asset('backend/libs/datatables.net-bs4/css/dataTables.bootstrap4.css')
}}" rel="stylesheet">
    <link href="{{ asset('backend/dist/css/style.min.css') }}" rel="stylesheet">
    <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
    <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
    <script src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>
<![endif]-->
</head>

<body>
    <!-- ============================================================== -->
    <!-- Preloader - style you can find in spinners.css -->
    <!-- ============================================================== -->
    <div class="preloader">
        <div class="lds-ripple">
            <div class="lds-pos"></div>
            <div class="lds-pos"></div>
        </div>
```

```
    </div>
    <!-- ============================================================== -->
    <!-- Main wrapper - style you can find in pages.scss -->
    <!-- ============================================================== -->
    <div id="main-wrapper">
        <!-- ============================================================== -->
        <!-- Topbar header - style you can find in pages.scss -->
        <!-- ============================================================== -->
        <header class="topbar" data-navbarbg="skin5">
            <nav class="navbar top-navbar navbar-expand-md navbar-dark">
                <div class="navbar-header" data-logobg="skin5">
                    <!-- This is for the sidebar toggle which is visible on mobile only -->
                    <a class="nav-toggler waves-effect waves-light d-block d-md-none"
href="javascript:void(0)"><i class="ti-menu ti-close"></i></a>
                    <!-- ============================================================== -->
                    <!-- Logo -->
                    <!-- ============================================================== -->
                    <a class="navbar-brand" href="index.html">
                        <!-- Logo icon -->
                        <b class="logo-icon p-l-10">
                            <!--You can put here icon as well // <i class="wi wi-
sunset"></i> //-->
                            <!-- Dark Logo icon -->
                            <img src="{{ asset('image/icon_univ_bsi.png') }}"
alt="homepage" class="light-logo" />

                        </b>
                        <!--End Logo icon -->
                        <!-- Logo text -->
                        <span class="logo-text">
                            <!-- dark Logo text -->
                            <img src="{{ asset('image/logo_text.png') }}" alt="homepage"
class="light-logo" />

                        </span>
                        <!-- Logo icon -->
                        <!-- <b class="logo-icon"> -->
                        <!--You can put here icon as well // <i class="wi wi-sunset"></i>
//-->
                        <!-- Dark Logo icon -->
                        <!-- <img src="assets/images/logo-text.png" alt="homepage"
class="light-logo" /> -->

                        <!-- </b> -->
                        <!--End Logo icon -->
                    </a>
                    <!-- ============================================================== -->
                    <!-- End Logo -->
                    <!-- ============================================================== -->
                    <!-- ============================================================== -->
                    <!-- Toggle which is visible on mobile only -->
                    <!-- ============================================================== -->
                    <a class="topbartoggler d-block d-md-none waves-effect waves-light"
href="javascript:void(0)" data-toggle="collapse" data-target="#navbarSupportedContent"
aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation"><i class="ti-more"></i></a>
                </div>
                <!-- ============================================================== -->
                <!-- End Logo -->
                <!-- ============================================================== -->
                <div class="navbar-collapse collapse" id="navbarSupportedContent" data-
navbarbg="skin5">
                    <!-- ============================================================== -->
                    <!-- toggle and nav items -->
                    <!-- ============================================================== -->
                    <ul class="navbar-nav float-left mr-auto">
```

```
                        <li class="nav-item d-none d-md-block"><a class="nav-link
sidebartoggler waves-effect waves-light" href="javascript:void(0)" data-sidebartype="mini-
sidebar"><i class="mdi mdi-menu font-24"></i></a></li>
                        <!-- ============================================================
-->
                        <!-- create new -->
                        <!-- ============================================================
-->

                        <!-- ============================================================
-->
                        <!-- Search -->
                        <!-- ============================================================
-->

                    </ul>
                    <!-- ============================================================ -->
                    <!-- Right side toggle and nav items -->
                    <!-- ============================================================ -->
                    <ul class="navbar-nav float-right">
                        <!-- ============================================================
-->
                        <!-- Comment -->
                        <!-- ============================================================
-->

                        <!-- ============================================================
-->
                        <!-- End Comment -->
                        <!-- ============================================================
-->
                        <!-- ============================================================
-->
                        <!-- Messages -->
                        <!-- ============================================================
-->

                        <!-- ============================================================
-->
                        <!-- End Messages -->
                        <!-- ============================================================
-->

                        <!-- ============================================================
-->
                        <!-- User profile and search -->
                        <!-- ============================================================
-->
                        <li class="nav-item dropdown">
                            <a class="nav-link dropdown-toggle text-muted waves-effect
waves-dark pro-pic" href="" data-toggle="dropdown" aria-haspopup="true" aria-
expanded="false">
                                @if (Auth::user()->foto)
                                <img src="{{ asset('storage/img-user/' . Auth::user()-
>foto) }}" alt="user" class="rounded-circle" width="31">
                                @else
                                <img src="{{ asset('storage/img-user/img-default.jpg') }}"
alt="user" class="rounded-circle" width="31">
                                @endif
                            </a>
                            <div class="dropdown-menu dropdown-menu-right user-dd
animated">
                                <a class="dropdown-item" href="{{
route('backend.user.edit', Auth::user()->id) }}"><i class="ti-user m-r-5 m-l-5"></i> Profil
Saya</a>
```

```
                            <a class="dropdown-item" href=""
onclick="event.preventDefault(); document.getElementById('keluar-app').submit();"><i
class="fa fa-power-off m-r-5 m-l-5"></i> Keluar</a>
                            <div class="dropdown-divider"></div>

                        </div>
                    </li>
                    <!-- ============================================================
-->
                    <!-- User profile and search -->
                    <!-- ============================================================
-->
                </ul>
            </div>
        </nav>
    </header>
    <!-- ============================================================ -->
    <!-- End Topbar header -->
    <!-- ============================================================ -->
    <!-- ============================================================ -->
    <!-- Left Sidebar - style you can find in sidebar.scss  -->
    <!-- ============================================================ -->
    <aside class="left-sidebar" data-sidebarbg="skin5">
        <!-- Sidebar scroll-->
        <div class="scroll-sidebar">
            <!-- Sidebar navigation-->
            <nav class="sidebar-nav">
                <ul id="sidebarnav" class="p-t-30">
                    <li class="sidebar-item"> <a class="sidebar-link waves-effect
waves-dark sidebar-link" href="{{ route('backend.beranda') }}" aria-expanded="false"><i
class="mdi mdi-view-dashboard"></i><span class="hide-menu">Beranda</span></a>
                    </li>
                    <li class="sidebar-item"> <a class="sidebar-link waves-effect
waves-dark sidebar-link" href="{{ route('backend.user.index') }}" aria-expanded="false"><i
class="mdi mdi-account"></i><span class="hide-menu">User</span></a>
                    </li>
                    <li class="sidebar-item"> <a class="sidebar-link has-arrow waves-
effect waves-dark" href="javascript:void(0)" aria-expanded="false"><i class="mdi mdi-
shopping"></i><span class="hide-menu">Data Produk </span></a>
                        <ul aria-expanded="false" class="collapse  first-level">
                            <li class="sidebar-item"><a href="{{
route('backend.kategori.index') }}" class="sidebar-link"><i class="mdi mdi-chevron-
right"></i><span class="hide-menu"> Kategori </span></a>
                            </li>
                            <li class="sidebar-item"><a href="{{
route('backend.produk.index') }}" class="sidebar-link"><i class="mdi mdi-chevron-
right"></i><span class="hide-menu"> Produk </span></a>
                            </li>
                        </ul>
                    </li>
                    <li class="sidebar-item"> <a class="sidebar-link has-arrow waves-
effect waves-dark" href="javascript:void(0)" aria-expanded="false"><i class="mdi mdi-
receipt"></i><span class="hide-menu">Laporan </span></a>
                        <ul aria-expanded="false" class="collapse  first-level">
                            <li class="sidebar-item"><a href="{{
route('backend.laporan.formuser') }}" class="sidebar-link"><i class="mdi mdi-chevron-
right"></i><span class="hide-menu"> User </span></a></li>
                            <li class="sidebar-item"><a href="{{
route('backend.laporan.formproduk') }}" class="sidebar-link"><i class="mdi mdi-chevron-
right"></i><span class="hide-menu"> Produk </span></a></li>

                        </ul>
                    </li>

                </ul>
            </nav>
```

```
                <!-- End Sidebar navigation -->
            </div>
            <!-- End Sidebar scroll-->
        </aside>
        <!-- ============================================================== -->
        <!-- End Left Sidebar - style you can find in sidebar.scss  -->
        <!-- ============================================================== -->
        <!-- ============================================================== -->
        <!-- Page wrapper  -->
        <!-- ============================================================== -->
        <div class="page-wrapper">
            <!-- ============================================================== -->
            <!-- Bread crumb and right sidebar toggle -->
            <!-- ============================================================== -->

            <!-- ============================================================== -->
            <!-- End Bread crumb and right sidebar toggle -->
            <!-- ============================================================== -->
            <!-- ============================================================== -->
            <!-- Container fluid  -->
            <!-- ============================================================== -->
            <div class="container-fluid">
                <!-- ============================================================== -->
                <!-- Start Page Content -->
                <!-- ============================================================== -->

                <!-- @yieldAwal -->
                @yield('content')
                <!-- @yieldAkhir-->

                <!-- ============================================================== -->
                <!-- End PAge Content -->
                <!-- ============================================================== -->
                <!-- ============================================================== -->
                <!-- Right sidebar -->
                <!-- ============================================================== -->
                <!-- .right-sidebar -->
                <!-- ============================================================== -->
                <!-- End Right sidebar -->
                <!-- ============================================================== -->
            </div>
            <!-- ============================================================== -->
            <!-- End Container fluid  -->
            <!-- ============================================================== -->
            <!-- ============================================================== -->
            <!-- footer -->
            <!-- ============================================================== -->
            <footer class="footer text-center">
                Web Programming. Studi Kasus Toko Online <a
href="https://bsi.ac.id/">Kuliah..? BSI Aja !!!</a>
            </footer>
            <!-- ============================================================== -->
            <!-- End footer -->
            <!-- ============================================================== -->
        </div>
        <!-- ============================================================== -->
        <!-- End Page wrapper  -->
        <!-- ============================================================== -->
    </div>
    <!-- ============================================================== -->
    <!-- End Wrapper -->
    <!-- ============================================================== -->
    <!-- ============================================================== -->
    <!-- All Jquery -->
    <!-- ============================================================== -->
    <script src="{{ asset('backend/libs/jquery/dist/jquery.min.js') }}"></script>
```

```
    <!-- Bootstrap tether Core JavaScript -->
    <script src="{{ asset('backend/libs/popper.js/dist/umd/popper.min.js') }}"></script>
    <script src="{{ asset('backend/libs/bootstrap/dist/js/bootstrap.min.js') }}"></script>
    <!-- slimscrollbar scrollbar JavaScript -->
    <script src="{{ asset('backend/libs/perfect-scrollbar/dist/perfect-
scrollbar.jquery.min.js') }}"></script>
    <script src="{{ asset('backend/extra-libs/sparkline/sparkline.js') }}"></script>
    <!--Wave Effects -->
    <script src="{{ asset('backend/dist/js/waves.js') }}"></script>
    <!--Menu sidebar -->
    <script src="{{ asset('backend/dist/js/sidebarmenu.js') }}"></script>
    <!--Custom JavaScript -->
    <script src="{{ asset('backend/dist/js/custom.min.js') }}"></script>
    <!-- this page js -->
    <script src="{{ asset('backend/extra-libs/multicheck/datatable-checkbox-init.js')
}}"></script>
    <script src="{{ asset('backend/extra-libs/multicheck/jquery.multicheck.js')
}}"></script>
    <script src="{{ asset('backend/extra-libs/DataTables/datatables.min.js') }}"></script>
    <script>
        /****************************************
         *       Basic Table                    *
         ****************************************/
        $('#zero_config').DataTable();
    </script>

    <!-- form keluar app -->
    <form id="keluar-app" action="{{ route('backend.logout') }}" method="POST" class="d-
none">
        @csrf
    </form>
    <!-- form keluar app end -->

    <!-- sweetalert -->
    <script src="{{ asset('sweetalert/sweetalert2.all.min.js') }}"></script>
    <!-- sweetalert End -->
    <!-- konfirmasi success-->
    @if (session('success'))
    <script>
        Swal.fire({
            icon: 'success',
            title: 'Berhasil!',
            text: "{{ session('success') }}"
        });
    </script>
    @endif
    <!-- konfirmasi success End-->
    <script type="text/javascript">
        //Konfirmasi delete
        $('.show_confirm').click(function(event) {
            var form = $(this).closest("form");
            var konfdelete = $(this).data("konf-delete");
            event.preventDefault();
            Swal.fire({
                title: 'Konfirmasi Hapus Data?',
                html: "Data yang dihapus <strong>" + konfdelete + "</strong> tidak dapat
dikembalikan!",
                icon: 'warning',
                showCancelButton: true,
                confirmButtonColor: '#3085d6',
                cancelButtonColor: '#d33',
                confirmButtonText: 'Ya, dihapus',
                cancelButtonText: 'Batal'
            }).then((result) => {
                if (result.isConfirmed) {
                    Swal.fire('Terhapus!', 'Data berhasil dihapus.', 'success')
```

```
                        .then(() => {
                            form.submit();
                        });
                }
            });
        });
    </script>
    <script>
        // previewFoto
        function previewFoto() {
            const foto = document.querySelector('input[name="foto"]');
            const fotoPreview = document.querySelector('.foto-preview');
            fotoPreview.style.display = 'block';
            const fotoReader = new FileReader();
            fotoReader.readAsDataURL(foto.files[0]);
            fotoReader.onload = function(fotoEvent) {
                fotoPreview.src = fotoEvent.target.result;
                fotoPreview.style.width = '100%';
            }
        }
    </script>
    <script src="{{ asset('ckeditor/ckeditor.js') }}"></script>
    <!-- <script
src="https://cdn.ckeditor.com/ckeditor5/30.0.0/classic/ckeditor.js"></script> -->
    <script>
        ClassicEditor
            .create(document.querySelector('#ckeditor'))
            .catch(error => {
                console.error(error);
            });
    </script>
</body>

</html>
```

Sehingga berikut *script* lengkap pada **ProdukController**:

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Produk;
use App\Models\Kategori;
use App\Models\FotoProduk;
use App\Helpers\ImageHelper;

class ProdukController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        $produk = Produk::orderBy('updated_at', 'desc')->get();
        return view('backend.v_produk.index', [
            'judul' => 'Data Produk',
            'index' => $produk
        ]);
    }


    /**
     * Show the form for creating a new resource.
     */
    public function create()
```

```
    {
        $kategori = Kategori::orderBy('nama_kategori', 'asc')->get();
        return view('backend.v_produk.create', [
            'judul' => 'Tambah Produk',
            'kategori' => $kategori
        ]);
    }

    /**
     * Store a newly created resource in storage.
     */
    public function store(Request $request)
    {
        $validatedData = $request->validate([
            'kategori_id' => 'required',
            'nama_produk' => 'required|max:255|unique:produk',
            'detail' => 'required',
            'harga' => 'required',
            'berat' => 'required',
            'stok' => 'required',
            'foto' => 'required|image|mimes:jpeg,jpg,png,gif|file|max:1024',
        ], $messages = [
            'foto.image' => 'Format gambar gunakan file dengan ekstensi jpeg, jpg, png,
atau gif.',
            'foto.max' => 'Ukuran file gambar Maksimal adalah 1024 KB.'
        ]);
        $validatedData['user_id'] = auth()->id();
        $validatedData['status'] = 0;

        if ($request->file('foto')) {
            $file = $request->file('foto');
            $extension = $file->getClientOriginalExtension();
            $originalFileName = date('YmdHis') . '_' . uniqid() . '.' . $extension;
            $directory = 'storage/img-produk/';

            // Simpan gambar asli
            $fileName = ImageHelper::uploadAndResize($file, $directory, $originalFileName);
            $validatedData['foto'] = $fileName;

            // create thumbnail 1 (lg)
            $thumbnailLg = 'thumb_lg_' . $originalFileName;
            ImageHelper::uploadAndResize($file, $directory, $thumbnailLg, 800, null);

            // create thumbnail 2 (md)
            $thumbnailMd = 'thumb_md_' . $originalFileName;
            ImageHelper::uploadAndResize($file, $directory, $thumbnailMd, 500, 519);

            // create thumbnail 3 (sm)
            $thumbnailSm = 'thumb_sm_' . $originalFileName;
            ImageHelper::uploadAndResize($file, $directory, $thumbnailSm, 100, 110);

            // Simpan nama file asli di database
            $validatedData['foto'] = $originalFileName;
        }

        Produk::create($validatedData, $messages);
        return redirect()->route('backend.produk.index')->with('success', 'Data berhasil
tersimpan');
    }

    /**
     * Display the specified resource.
     */
    public function show(string $id)
    {
        $produk = Produk::with('fotoProduk')->findOrFail($id);
```

```
        $kategori = Kategori::orderBy('nama_kategori', 'asc')->get();
        return view('backend.v_produk.show', [
            'judul' => 'Detail Produk',
            'show' => $produk,
            'kategori' => $kategori
        ]);
    }

    /**
     * Show the form for editing the specified resource.
     */
    public function edit(string $id)
    {
        $produk = Produk::findOrFail($id);
        $kategori = Kategori::orderBy('nama_kategori', 'asc')->get();
        return view('backend.v_produk.edit', [
            'judul' => 'Ubah Produk',
            'edit' => $produk,
            'kategori' => $kategori
        ]);
    }

    /**
     * Update the specified resource in storage.
     */
    public function update(Request $request, string $id)
    {
        //ddd($request);
        $produk = Produk::findOrFail($id);
        $rules = [
            'nama_produk' => 'required|max:255|unique:produk,nama_produk,' . $id,
            'kategori_id' => 'required',
            'status' => 'required',
            'detail' => 'required',
            'harga' => 'required',
            'berat' => 'required',
            'stok' => 'required',
            'foto' => 'image|mimes:jpeg,jpg,png,gif|file|max:1024',
        ];
        $messages = [
            'foto.image' => 'Format gambar gunakan file dengan ekstensi jpeg, jpg, png,
atau gif.',
            'foto.max' => 'Ukuran file gambar Maksimal adalah 1024 KB.'
        ];
        $validatedData['user_id'] = auth()->id();
        $validatedData = $request->validate($rules, $messages);

        if ($request->file('foto')) {
            //hapus gambar lama
            if ($produk->foto) {
                $oldImagePath = public_path('storage/img-produk/') . $produk->foto;
                if (file_exists($oldImagePath)) {
                    unlink($oldImagePath);
                }
                $oldThumbnailLg = public_path('storage/img-produk/') . 'thumb_lg_' .
$produk->foto;
                if (file_exists($oldThumbnailLg)) {
                    unlink($oldThumbnailLg);
                }
                $oldThumbnailMd = public_path('storage/img-produk/') . 'thumb_md_' .
$produk->foto;
                if (file_exists($oldThumbnailMd)) {
                    unlink($oldThumbnailMd);
                }
                $oldThumbnailSm = public_path('storage/img-produk/') . 'thumb_sm_' .
$produk->foto;
```

```
                if (file_exists($oldThumbnailSm)) {
                    unlink($oldThumbnailSm);
                }
            }
            $file = $request->file('foto');
            $extension = $file->getClientOriginalExtension();
            $originalFileName = date('YmdHis') . '_' . uniqid() . '.' . $extension;
            $directory = 'storage/img-produk/';

            // Simpan gambar asli
            $fileName = ImageHelper::uploadAndResize($file, $directory, $originalFileName);
            $validatedData['foto'] = $fileName;

            // create thumbnail 1 (lg)
            $thumbnailLg = 'thumb_lg_' . $originalFileName;
            ImageHelper::uploadAndResize($file, $directory, $thumbnailLg, 800, null);

            // create thumbnail 2 (md)
            $thumbnailMd = 'thumb_md_' . $originalFileName;
            ImageHelper::uploadAndResize($file, $directory, $thumbnailMd, 500, 519);

            // create thumbnail 3 (sm)
            $thumbnailSm = 'thumb_sm_' . $originalFileName;
            ImageHelper::uploadAndResize($file, $directory, $thumbnailSm, 100, 110);

            // Simpan nama file asli di database
            $validatedData['foto'] = $originalFileName;
        }

        $produk->update($validatedData);
        return redirect()->route('backend.produk.index')->with('success', 'Data berhasil
diperbaharui');
    }

    /**
     * Remove the specified resource from storage.
     */
    public function destroy($id)
    {
        $produk = Produk::findOrFail($id);
        $directory = public_path('storage/img-produk/');

        if ($produk->foto) {
            // Hapus gambar asli
            $oldImagePath = $directory . $produk->foto;
            if (file_exists($oldImagePath)) {
                unlink($oldImagePath);
            }

            // Hapus thumbnail lg
            $thumbnailLg = $directory . 'thumb_lg_' . $produk->foto;
            if (file_exists($thumbnailLg)) {
                unlink($thumbnailLg);
            }

            // Hapus thumbnail md
            $thumbnailMd = $directory . 'thumb_md_' . $produk->foto;
            if (file_exists($thumbnailMd)) {
                unlink($thumbnailMd);
            }

            // Hapus thumbnail sm
            $thumbnailSm = $directory . 'thumb_sm_' . $produk->foto;
            if (file_exists($thumbnailSm)) {
                unlink($thumbnailSm);
            }
```

```
        }

        // Hapus foto produk lainnya di tabel foto_produk
        $fotoProduks = FotoProduk::where('produk_id', $id)->get();
        foreach ($fotoProduks as $fotoProduk) {
            $fotoPath = $directory . $fotoProduk->foto;
            if (file_exists($fotoPath)) {
                unlink($fotoPath);
            }
            $fotoProduk->delete();
        }

        $produk->delete();

        return redirect()->route('backend.produk.index')->with('success', 'Data berhasil
dihapus');
    }


    // Method untuk menyimpan foto tambahan
    public function storeFoto(Request $request)
    {
        // Validasi input
        $request->validate([
            'produk_id' => 'required|exists:produk,id',
            'foto_produk.*' => 'image|mimes:jpeg,jpg,png,gif|file|max:1024',
        ]);

        if ($request->hasFile('foto_produk')) {
            foreach ($request->file('foto_produk') as $file) {
                // Buat nama file yang unik
                $extension = $file->getClientOriginalExtension();
                $filename = date('YmdHis') . '_' . uniqid() . '.' . $extension;
                $directory = 'storage/img-produk/';

                // Simpan dan resize gambar menggunakan ImageHelper
                ImageHelper::uploadAndResize($file, $directory, $filename, 800, null);
                // Simpan data ke database
                FotoProduk::create([
                    'produk_id' => $request->produk_id,
                    'foto' => $filename,
                ]);
            }
        }
        return redirect()->route('backend.produk.show', $request->produk_id)
            ->with('success', 'Foto berhasil ditambahkan.');
    }


    // Method untuk menghapus foto
    public function destroyFoto($id)
    {
        $foto = FotoProduk::findOrFail($id);
        $produkId = $foto->produk_id;

        // Hapus file gambar dari storage
        $imagePath = public_path('storage/img-produk/') . $foto->foto;
        if (file_exists($imagePath)) {
            unlink($imagePath);
        }
        // Hapus record dari database
        $foto->delete();

        return redirect()->route('backend.produk.show', $produkId)
            ->with('success', 'Foto berhasil dihapus.');
```

```
    }

    // Method untuk Form Laporan Produk
    public function formProduk()
    {
        return view('backend.v_produk.form', [
            'judul' => 'Laporan Data Produk',
        ]);
    }

    // Method untuk Cetak Laporan Produk
    public function cetakProduk(Request $request)
    {
        // Menambahkan aturan validasi
        $request->validate([
            'tanggal_awal' => 'required|date',
            'tanggal_akhir' => 'required|date|after_or_equal:tanggal_awal',
        ], [
            'tanggal_awal.required' => 'Tanggal Awal harus diisi.',
            'tanggal_akhir.required' => 'Tanggal Akhir harus diisi.',
            'tanggal_akhir.after_or_equal' => 'Tanggal Akhir harus lebih besar atau sama
dengan Tanggal Awal.',
        ]);

        $tanggalAwal = $request->input('tanggal_awal');
        $tanggalAkhir = $request->input('tanggal_akhir');

        $query =  Produk::whereBetween('updated_at', [$tanggalAwal, $tanggalAkhir])
            ->orderBy('id', 'desc');

        $produk = $query->get();
        return view('backend.v_produk.cetak', [
            'judul' => 'Laporan Produk',
            'tanggalAwal' => $tanggalAwal,
            'tanggalAkhir' => $tanggalAkhir,
            'cetak' => $produk
        ]);
    }
}
```

Minggu Ke-13
Persentasi Tugas Kelompok

Presentasi Tugas Kelompok yakni mempresentasikan hasil dari project akhir yang dikerjakan secara berkelompok dengan Indikator Penilaian:

1. Ketepatan waktu pengumpulan tugas
2. Kelengkapan makalah project
3. Dapat menjelaskan, menampilkan dan membuktikan hasil project akhir secara berkelompok

Minggu Ke-14
Persentasi Tugas Kelompok

Presentasi Tugas Kelompok yakni mempresentasikan hasil dari project akhir yang dikerjakan secara berkelompok dengan Indikator Penilaian:

1. Ketepatan waktu pengumpulan tugas
2. Kelengkapan makalah project
3. Dapat menjelaskan, menampilkan dan membuktikan hasil project akhir secara berkelompok

Minggu Ke-15
Persentasi Tugas Kelompok

Presentasi Tugas Kelompok yakni mempresentasikan hasil dari project akhir yang dikerjakan secara berkelompok dengan Indikator Penilaian:

1. Ketepatan waktu pengumpulan tugas
2. Kelengkapan makalah project
3. Dapat menjelaskan, menampilkan dan membuktikan hasil project akhir secara berkelompok