**LAPORAN PEMBUATAN APLIKASI**

**TOKO BUKU CHEERFUL**

**TAHUN 2024**



**DISUSUN OLEH:**

**Nama : T. Amylia Safitri**

**NISN : 0074586930**

**Program Keahlian : Teknik Komputer dan Informatika**

**Kompetensi Keahlian : Rekayasa Perangkat Lunak**

**PEMERINTAH ACEH**

**DINAS PENDIDIKAN**

**SEKOLAH MENENGAH KEJURUAN (SMK) NEGERI 1 KARANG BARU**

**KABUPATEN ACEH TAMIANG**

**TAHUN 2024**

# LEMBAR PENGESAHAN

**LAPORAN PEMBUATAN APLIKASI**

**TOKO BUKU CHEERFUL**

**TAHUN 2024**

**Nama : T. Amylia Safitri**

**NISN : 0074586930**

**Program Keahlian : Teknik Komputer dan Informatika**

**Kompetensi Keahlian : Rekayasa Perangkat Lunak**

Telah disahkan tanggal, ……………………………….

Menyutujui:

|  |  |  |
| --- | --- | --- |
|  | | |
|  |  | Pembimbing Sekolah  **Ahmadi Muslim, MP**  NIP. 19860505 201403 1 003 |

# KATA PENGANTAR

Puji syukur saya ucapkan Kehadirat Allah SWT. karena dengan rahmat dan karunia-Nya penulis dapat menyelesaikan laporan praktek kerja industri (Prakerin) yang telah dilaksanakan di SMKN 1 Karang Baru Kabupaten Aceh Tamiang.

Sehubungan dengan terlaksanakannya Prakerin ini tidak terlepas dari bantuan dan dorongan dari semua pihak secara moril maupun materil, oleh karena itu penulis mengucapkan terima kasih sebesar-besarnya kepada:

1. Bapak Fahmi Putra, S.Pd selaku Kepala SMK Negeri 1 Karang Baru
2. Bapak Ahmadi Muslim, MP selaku Kepala Badan Kepegawaian Dan Pengembangan Sumber daya Manusia Kabupaten Aceh Tamiang
3. Bapak Safrizal, S.ST selaku Kepala Bidang Rekayasa Perangkat Lunak SMK Negeri 1 Karang Baru
4. Bapak Ahmadi Muslim, MP selaku Pembimbing di SMK Negeri 1 Karang Baru
5. Bapak Ahmadi Muslim, MP selaku pembimbing di SMKN 1 Karang Baru Aceh Tamiang
6. Bapak/Ibu Guru di SMK Negeri 1 Karang Baru
7. Para pegawai di SMKN 1 Karang Baru

Penulis menyadari bahwa laporan ini masih jauh dari kesempurnaan. Oleh karena itu penulis menghrapkan kritik dan saran yang sifatnya membangun bagi pembaca, penulis berharap semoga laporan ini bermanfaat.

Karang Baru, 20 November 2023

Penyusun,

**Ahmadi Muslim**

# DAFTAR ISI

[**LEMBAR PENGESAHAN i**](#_Toc164929218)

[**KATA PENGANTAR ii**](#_Toc164929219)

[**DAFTAR ISI iii**](#_Toc164929220)

[**BAB I PENDAHULUAN 1**](#_Toc164929221)

[A. Latar Belakang 1](#_Toc164929222)

[B. Tujuan dan Manfaat Aplikasi 1](#_Toc164929223)

[C. Cara Kerja Aplikasi 1](#_Toc164929224)

[D. Dokumen Pendukung Perancangan Aplikasi 1](#_Toc164929225)

[BA**B II PERANCANGAN BASIS DATA** 2](#_Toc164929226)

[A. Entitas Relationship Diagram 2](#_Toc164929228)

[B. Desain Tabel Sebelum Normalisasi 2](#_Toc164929229)

[C. Normalisasi 2](#_Toc164929230)

[D. Design Tabel Setelah Normalisasi 2](#_Toc164929231)

[E. Migration 3](#_Toc164929232)

[**BAB III TAMPILAN APLIKASI 8**](#_Toc164929233)

[A. Fitur Utama / Dashboard 8](#_Toc164929235)

[B. Fitur Petugas 8](#_Toc164929236)

[C. Fitur Login 8](#_Toc164929237)

[D. Fitur Produk 8](#_Toc164929238)

[E. Fitur Pelanggan 8](#_Toc164929239)

[F. Fitur Transaksi 8](#_Toc164929240)

[G. Fitur Detiltransaksi 8](#_Toc164929241)

[H. Fitur Laporan 8](#_Toc164929242)

# BAB I PENDAHULUAN

## Latar Belakang

Selama ini sistem penjualan di Toko Buku Cheerful masih dilakukan secara manual dan memakan waktu yang lama. Untuk itu, dengan pembuatan Aplikasi POS ini yang memadai fitur-fitur didalamnya, diharapkan berguna agar sistem penjualan di Toko Buku Cheerful tidak dilakukan secara manual lagi dan juga menghindari terjadinya kesalahan.

## Tujuan dan Manfaat Aplikasi

1. Untuk mempercepat dalam proses peproduk terhadap konsumen/pelanggan
2. Memudahkan dan lebih praktis dalam proses perhitungan jumlah pembayaran

## Cara Kerja Aplikasi

* Seorang petugas dapat menjual banyak produk
* Seorang pelanggan dapat membeli lebih dari satu produk yang dijual oleh petugas
* Seorang petugas dapat merekap banyak transaksi
* Transaksi memiliki detiltransaksi
* Stok produk akan berkurang sesuai jumlah produk yang dibeli
* Nomor telepon petugas boleh lebih dari satu
* Petugas, pelanggan, produk dan transaksi dapat di identifikasi dengan id yang berbeda (unik)

## Dokumen Pendukung Perancangan Aplikasi

Barang yang diperjualbelikan

Daftar harga barang

# BAB II PERANCANGAN BASIS DATA



## Entitas Relationship Diagram (ERD)



**Penjelasan ERD:**

* *Seorang petugas dapat menjual banyak produk*
* *Seorang pelanggan dapat membeli lebih dari satu produk yang dijual oleh petugas*
* *Seorang petugas dapat merekap banyak transaksi*

## Desain Tabel Aplikasi



## Migration

**Versi 1: Skema Relasi by Table Migration files**

Skema relasi by tabel adalah pembuatan tabel berbarengan dengan relasi tabel. Penyusunan skema relasi dari tabel child ke tabel parent dibuat menjadi satu file dengan pembuatan atribut/fields pada tabel. Skema ini dilakukan dengan membuat tabel induk/parent terlebih dahulu, baru diikuti dengan membuat tabel child.

* + 1. **Tabel User**

Untuk tabel user sebenarnya tidak wajib dibuat, karena secara default sudah disediakan oleh laravel. Akan tetapi jika harus dibuat, maka dapat dituliskan seperti berikut ini

1. Memahami desain tabel dari model ERT



* *Tabel user merupakan tabel induk yang akan berisi tentang daftar user yang akan login untuk mengelola aplikasi.*
* *Tabel ini nantinya akan direferensi oleh tabel transaksi sehingga kedudukannya adalah* ***parent****.*

1. Menuliskan prompt berikut pada terminal VS Code:

php artisan make:model User -m

1. Buka file create\_produks\_table.php yang terdapat dalam folder migration di folder Database.
2. Kemudian ubah seperti berikut:

public function up(): void

{

Schema::create('users', function (Blueprint $table) {

$table->id();

$table->string('password');

$table->string('email');

$table->string('name');

$table->timestamps();

});

}

* + 1. **Tabel Pelanggan**

1. Memahami desain tabel dari model ERT



* *Tabel pelanggan merupakan tabel yang akan berisi tentang daftar pelanggan yang akan menjadi pelanggan/stakeholder/customer pada proses penjualan.*
* *Tabel ini nantinya akan direferensi oleh tabel transaksi sehingga kedudukannya adalah* ***parent****.*

1. Menuliskan prompt berikut pada terminal VS Code:

php artisan make:model Pelanggan -m

1. Buka file create\_produks\_table.php yang terdapat dalam folder migration di folder Database.
2. Kemudian ubah seperti berikut:

public function up(): void

{

Schema::create('pelanggans', function (Blueprint $table) {

$table->id();

$table->string('nama');

$table->string('telepon')->nullable();

$table->text('alamat')->nullable();

$table->timestamps();

});

}

* + 1. **Tabel Produk**

1. Memahami desain tabel dari model ERT



* *Tabel produk merupakan tabel yang akan berisi tentang daftar menu produk yang akan dijual oleh user dan dibeli oleh pelanggan.*
* *Tabel ini nantinya akan direferensi oleh tabel detiltransaksi sekaligus mereferensi tabel suplier dna tabel teknisi sehingga kedudukannya adalah* ***parent*** *sekaligus* ***child****.*

1. Menuliskan prompt berikut pada terminal VS Code:

php artisan make:model Produk -m

1. Buka file create\_produks\_table.php yang terdapat dalam folder migration di folder Database.
2. Kemudian ubah seperti berikut:

public function up(): void

{

Schema::create('produks', function (Blueprint $table) {

$table->id();

$table->string('nama');

$table->string('deskripsi')->nullable();

$table->integer('stok');

$table->double('harga');

$table->timestamps();

});

}

1. Untuk function down() menjadi seperti berikut:

public function down()

    {

        Schema::dropIfExists('produks');

    }

* + 1. **Tabel Transaksi**

1. Memahami desain tabel dari model ERT



* *Tabel transaksi merupakan tabel yang akan berisi tentang transaksi produk yang dijual oleh user dan dibeli oleh pelanggan.*
* *Tabel ini nantinya akan direferensi oleh tabel detiltransaksi sekaligus mereferensi tabel suplier dan tabel teknisi sehingga kedudukannya adalah* ***parent*** *sekaligus* ***child****.*

1. Menuliskan prompt berikut pada terminal VS Code:

php artisan make:model Transaksi -m

1. Ubah file migration transaksi menjadi seperti berikut:

public function up(): void

{

Schema::create('transaksis', function (Blueprint $table) {

$table->id();

$table->string('invoice')->unique();

$table->bigInteger('pelanggan\_id')->unsigned();

$table->bigInteger('user\_id')->unsigned();

$table->double('total');

$table->timestamps();

});

Schema::table('transaksis',function(Blueprint $table){

$table->foreign('pelanggan\_id')->references('id')->on('pelanggans')

->onUpdate('cascade')->onDelete('cascade');

$table->foreign('user\_id')->references('id')->on('users')

->onUpdate('cascade')->onDelete('cascade');

});

}

1. Untuk function down() menjadi seperti berikut:

public function down(): void

    {

        Schema::table('transaksis', function(Blueprint $table) {

            $table->dropForeign('transaksis\_pelanggan\_id\_foreign');

        });

        Schema::table('transaksis', function(Blueprint $table) {

            $table->dropIndex('transaksis\_pelanggan\_id\_foreign');

        });

        Schema::table('transaksis', function(Blueprint $table) {

            $table->dropForeign('transaksis\_user\_id\_foreign');

        });

        Schema::table('transaksis', function(Blueprint $table) {

            $table->dropIndex('transaksis\_user\_id\_foreign');

        });

        Schema::dropIfExists('transaksis');

    }

* + 1. **Tabel Detiltransaksi**

1. Memahami desain tabel dari model ERT



* *Tabel detiltransaksi merupakan tabel yang akan berisi tentang transaksi produk yang dijual oleh user dan dibeli oleh pelanggan.*
* *Tabel ini nantinya akan mereferensi tabel transakssi dan tabel produk sehingga kedudukannya adalah* ***child****.*

1. Menuliskan prompt berikut pada terminal VS Code:

php artisan make:model Detiltransaksi -m

1. Ubah file migration detiltransaksi menjadi seperti berikut:

public function up(): void

{

Schema::create('detiltransaksis', function (Blueprint $table) {

$table->id();

$table->bigInteger('transaksis\_id')->unsigned();

$table->bigInteger('produk\_id')->unsigned();

$table->integer('qty');

$table->double('harga');

$table->timestamps();

});

Schema::table('detiltransaksis', function(Blueprint $table) {

$table->foreign('transaksis\_id')->references('id')->on('transaksis')

->onUpdate('cascade') ->onDelete('cascade');

$table->foreign('produk\_id')->references('id')->on('produks')

->onUpdate('cascade') ->onDelete('cascade');

});

}

1. Kemudian untuk function down() menjadi seperti berikut:

public function down(): void

{

Schema::table('detiltransaksis', function(Blueprint $table) {

$table->dropForeign('detiltransaksis\_transaksis\_id\_foreign');

});

Schema::table('detiltransaksis', function(Blueprint $table) {

$table->dropIndex('detiltransaksis\_transaksis\_id\_foreign');

});

Schema::table('detiltransaksis', function(Blueprint $table) {

$table->dropForeign('detiltransaksis\_produk\_id\_foreign');

});

Schema::table('detiltransaksis', function(Blueprint $table) {

$table->dropIndex('detiltransaksis\_produk\_id\_foreign');

});

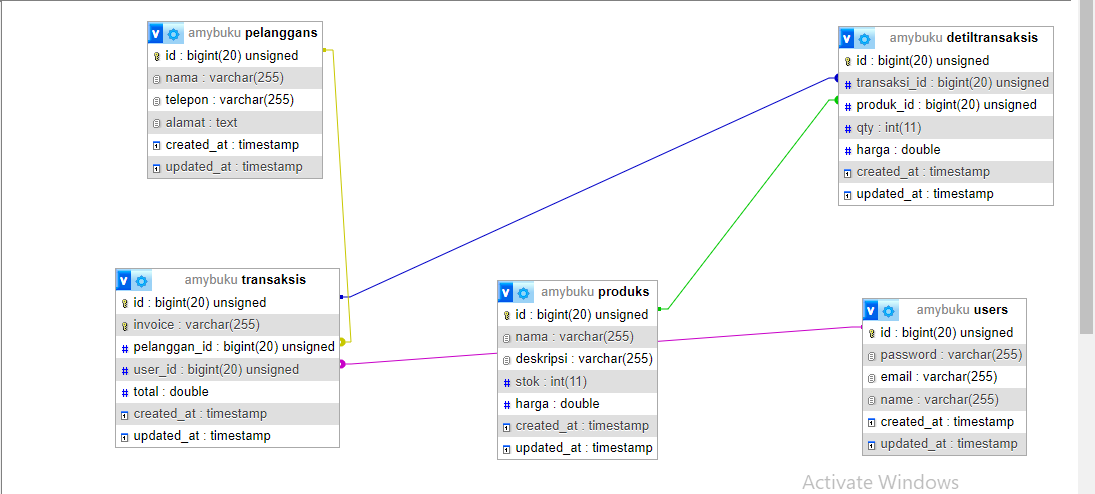
Schema::dropIfExists('detiltransaksis');

}

* + 1. **Migrasi/migrate ke database server**

php artisan migrate

* + 1. **Cek hasilnya di localhost/phpmyadmin**



**Versi 2: Skema Relasi out Table Migration Files**

Skema relasi out tabel adalah pembuatan database tanpa memperhatikan atau mengesampingkan dahulu skema relasi tabel-tabel pada database. Penyusunan skema relasi dari tabel child ke tabel parent dibuat di dalam satu file tersendiri melalui skema yang sudah disusun dalam rencana permodelan ERT setelah semua tabel yang dibutuhkan pada aplikasi selesai dikerjakan. Penerapan model ini dirasa lebih mudah untuk menyusun skema tabel pada databse karena tidak akan terjadi error akibat tabel anak/child mencari referensi ke tabel induk/parent.

* + 1. **Tabel User**

Untuk tabel user sebenarnya tidak wajib dibuat, karena secara default sudah disediakan oleh laravel. Akan tetapi jika harus dibuat, maka dapat dituliskan seperti berikut ini

1. Memahami desain tabel dari model ERT



* *Tabel user merupakan tabel induk yang akan berisi tentang daftar user yang akan login untuk mengelola aplikasi.*
* *Tabel ini nantinya akan direferensi oleh tabel transaksi sehingga kedudukannya adalah* ***parent****.*

1. Menuliskan prompt berikut pada terminal VS Code:

php artisan make:model User -m

1. Buka file create\_produks\_table.php yang terdapat dalam folder migration di folder Database.
2. Kemudian ubah seperti berikut:

public function up(): void

{

Schema::create('users', function (Blueprint $table) {

$table->id();

$table->string('password');

$table->string('email');

$table->string('name');

$table->timestamps();

});

}

* + 1. **Tabel Pelanggan**

1. Memahami desain tabel dari model ERT



* *Tabel pelanggan merupakan tabel yang akan berisi tentang daftar pelanggan yang akan menjadi pelanggan/stakeholder/customer pada proses penjualan.*
* *Tabel ini nantinya akan direferensi oleh tabel transaksi sehingga kedudukannya adalah* ***parent****.*

1. Menuliskan prompt berikut pada terminal VS Code:

php artisan make:model Pelanggan -m

1. Buka file create\_produks\_table.php yang terdapat dalam folder migration di folder Database.
2. Kemudian ubah seperti berikut:

public function up(): void

{

Schema::create('pelanggans', function (Blueprint $table) {

$table->id();

$table->string('nama');

$table->string('telepon')->nullable();

$table->text('alamat')->nullable();

$table->timestamps();

});

}

* + 1. **Tabel Produk**

1. Memahami desain tabel dari model ERT



* *Tabel produk merupakan tabel yang akan berisi tentang daftar menu produk yang akan dijual oleh user dan dibeli oleh pelanggan.*
* *Tabel ini nantinya akan direferensi oleh tabel detiltransaksi sekaligus mereferensi tabel suplier dna tabel teknisi sehingga kedudukannya adalah* ***parent*** *sekaligus* ***child****.*

1. Menuliskan prompt berikut pada terminal VS Code:

php artisan make:model Produk -m

1. Buka file create\_produks\_table.php yang terdapat dalam folder migration di folder Database.
2. Kemudian ubah seperti berikut:

public function up(): void

{

Schema::create('produks', function (Blueprint $table) {

$table->id();

$table->string('nama');

$table->string('deskripsi')->nullable();

$table->integer('stok');

$table->double('harga');

$table->timestamps();

});

}

* + 1. **Tabel Transaksi**

1. Memahami desain tabel dari model ERT



* *Tabel transaksi merupakan tabel yang akan berisi tentang transaksi produk yang dijual oleh user dan dibeli oleh pelanggan.*
* *Tabel ini nantinya akan direferensi oleh tabel detiltransaksi sekaligus mereferensi tabel suplier dan tabel teknisi sehingga kedudukannya adalah* ***parent*** *sekaligus* ***child****.*

1. Menuliskan prompt berikut pada terminal VS Code:

php artisan make:model Transaksi -m

1. Ubah file migration transaksi menjadi seperti berikut:

public function up(): void

{

Schema::create('transaksis', function (Blueprint $table) {

$table->id();

$table->string('invoice')->unique();

$table->bigInteger('pelanggan\_id')->unsigned();

$table->bigInteger('user\_id')->unsigned();

$table->double('total');

$table->timestamps();

});

Schema::table('transaksis',function(Blueprint $table){

$table->foreign('pelanggan\_id')->references('id')->on('pelanggans')

->onUpdate('cascade')->onDelete('cascade');

$table->foreign('user\_id')->references('id')->on('users')

->onUpdate('cascade')->onDelete('cascade');

});

}

* + 1. **Tabel Detiltransaksi**

1. Memahami desain tabel dari model ERT



* *Tabel detiltransaksi merupakan tabel yang akan berisi tentang transaksi produk yang dijual oleh user dan dibeli oleh pelanggan.*
* *Tabel ini nantinya akan mereferensi tabel transakssi dan tabel produk sehingga kedudukannya adalah* ***child****.*

1. Menuliskan prompt berikut pada terminal VS Code:

php artisan make:model Detiltransaksi -m

1. Ubah file migration detiltransaksi menjadi seperti berikut:

public function up(): void

{

Schema::create('detiltransaksis', function (Blueprint $table) {

$table->id();

$table->bigInteger('transaksis\_id')->unsigned();

$table->bigInteger('produk\_id')->unsigned();

$table->integer('qty');

$table->double('harga');

$table->timestamps();

});

Schema::table('detiltransaksis', function(Blueprint $table) {

$table->foreign('transaksis\_id')->references('id')->on('transaksis')

->onUpdate('cascade') ->onDelete('cascade');

$table->foreign('produk\_id')->references('id')->on('produks')

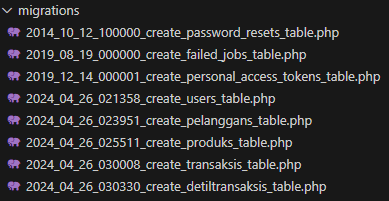
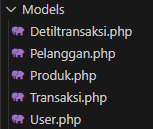
->onUpdate('cascade') ->onDelete('cascade');

});

}

* + 1. **Tabel Relasi pada Folder Migrasi**

Hasil tabel migrasi jika kita berhasil makan akan muncul pada folder model dan migrasi

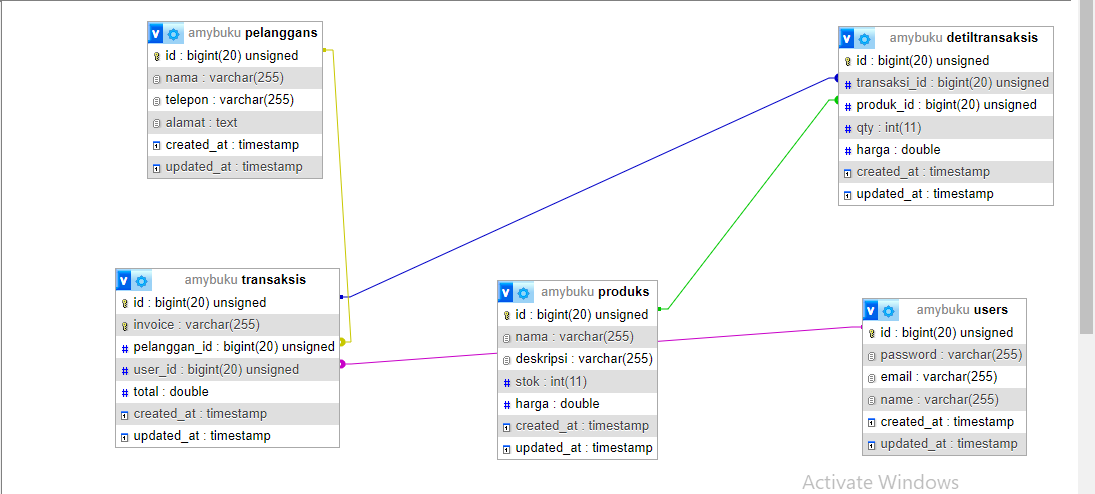
 

* + 1. **Tabel Relasi Migrasi**

1. Memahami desain tabel dari model ERT



* + 1. **Cek hasil migrasinya di localhost/phpmyadmin**



# BAB IV BACKUP APLIKASI



## Cara Install Git di Windows

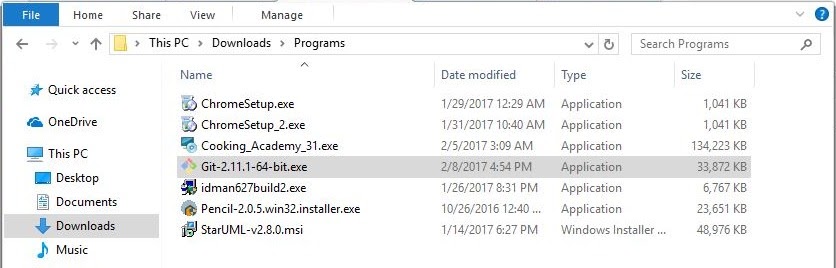
Instalasi Git di Windows memang tidak seperti di Linux yang ketik perintah langsung terinstal.

Kita harus men-download dulu, kemudian melakukan ritual next>next>finish.

Tapi dalam ritual tersebut, ada pilihan yang harus diperhatikan agar perintah git dapat dikenali di CMD.

Silahkan buka website resminya Git ( [git-scm.com](https://git-scm.com/)). Kemudian unduh Git sesuai dengan arsitektur komputer kita. Kalau menggunakan 64bit, unduh yang 64bit. Begitu juga kalau menggunakan 32bit.

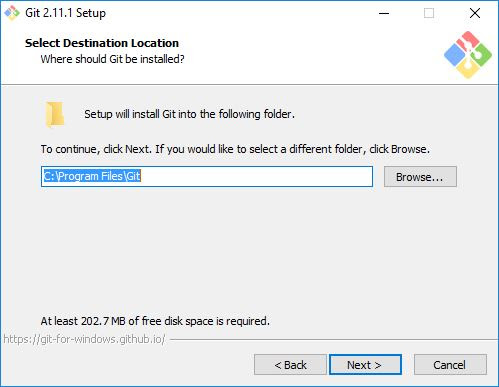
Baiklah, mari kita mulai ritual instalnya. Silahkan klik 2x file instaler Git yang sudah diunduh.



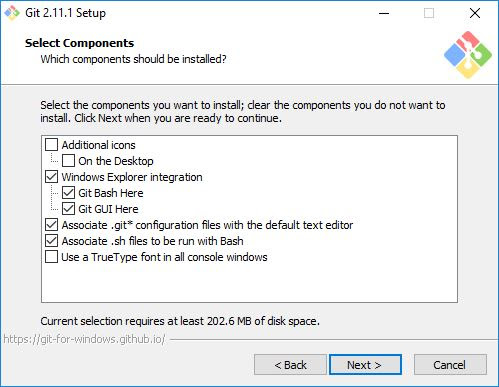
Maka akan muncul infomasi lisensi Git, klik Next > untuk melanjutkan.



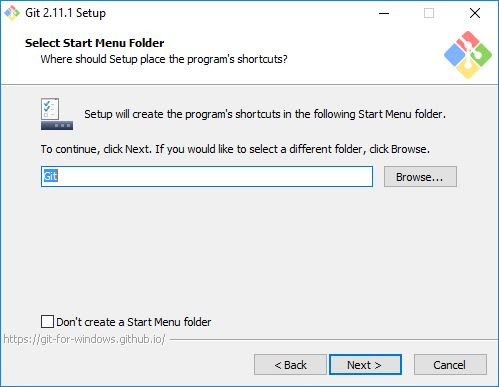
Selanjutnya menentukan lokasi instalasi. Biarkan saja apa adanya, kemudian klik Next >.



Selanjutnya pemilihan komoponen, biarkan saja seperti ini kemudian klik Next >.



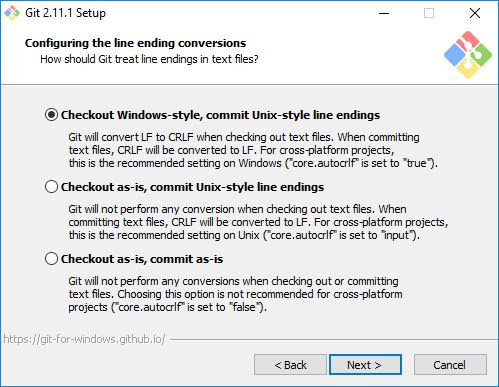
Selanjutnya pemlilihan direktori start menu, klik Next >.



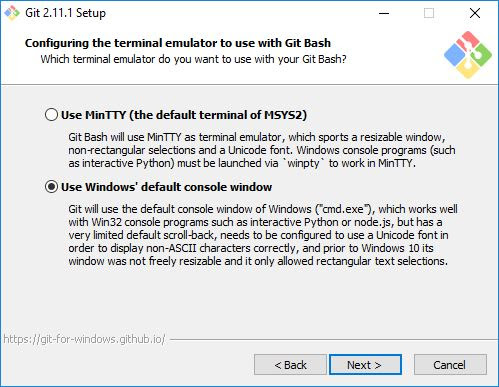
Selanjutnya pengaturan PATH Environment. Pilih yang tengah agar perintah git dapat di kenali di Command Prompt (CMD). Setelah itu klik Next >.



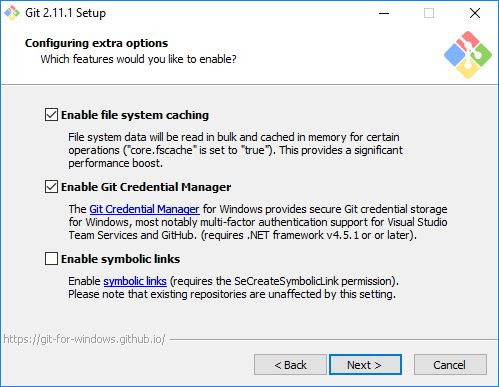
Selanjutnya konversi line ending. Biarkan saja seperti ini, kemudian klik Next >.



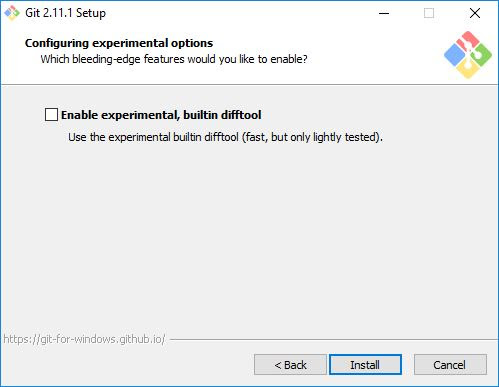
Selanjutnya pemilihan emulator terminal. Pilih saja yang bawah, kemudian klik Next >.



Selanjutnya pemilihan opsi ekstra. Klik saja Next >.



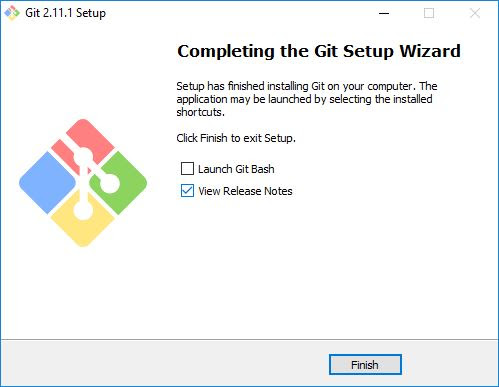
Selanjutnya pemilihan opsi ekspreimental, langsung saja klik Install untuk memaulai instalasi.



Tunggu beberapa saat, instalasi sedang dilakukan.



Setelah selesai, kita bisa langsung klik Finish.



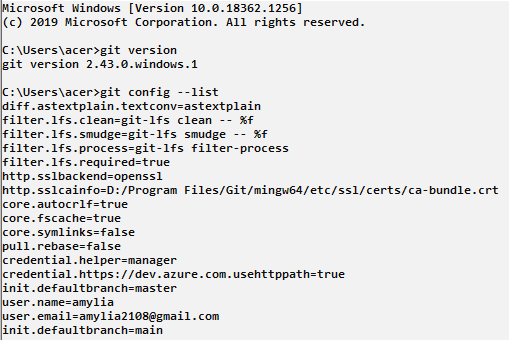
## Setting Dan Konfigurasi Github

Berikut adalah cara melakukan setingan konfigurasi git local desktop. Setingan ini akan menentukan bagaimana cara git terhubung ke github dan lain sebagainya.

1. **Melihat version dan default branch**

Git –v : untuk melihat versi github, jika belum terinstal maka akan muncul pesan error.

Git config –list : untuk melihat daftar konfigurasi setingan yang sudah dilakukan.



1. **Melakukan setting Configurasi Global**

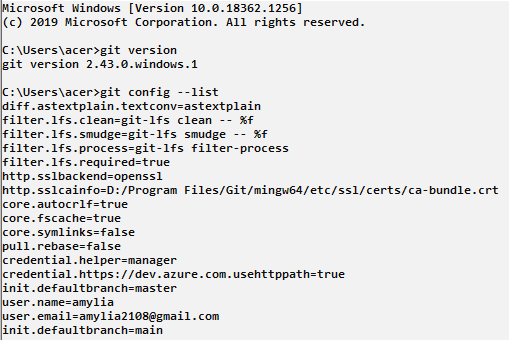
Lakukan setingan configurasi global jika belum ada config ke usesrname di github. Username dan email yang digunakan harus benar benar yang sudah terdaftar.

**Git config --global user.email “email\_user”**:

**Git config --global user.name “nama\_user”**:

**Git config --global user.email “amylia2108@gmail.com”**:

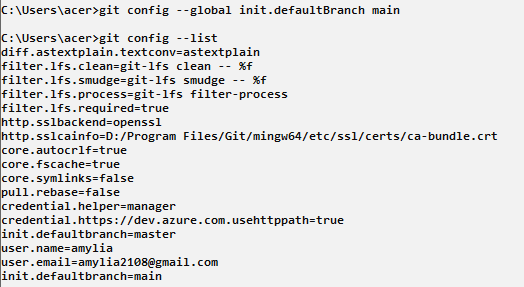
**Git config --global user.name “amyliaa”**:



1. **Melakukan setting Default Branch Main di Configurasi Global**

Lakukan setingan default branch main agar branch yang di update adalah branch main.

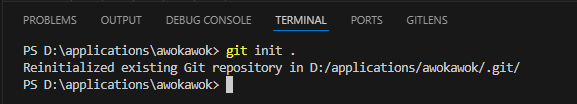
git config --global init.defaultBranch main



1. **Git Init**

Lakukan git init . untuk menginisialisasi project yang akan di update.

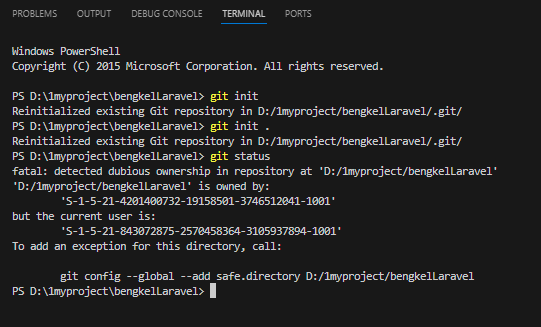
git init.



1. **Git status**

Lakukan git status . untuk melihat status project yang akan di update.

git status.

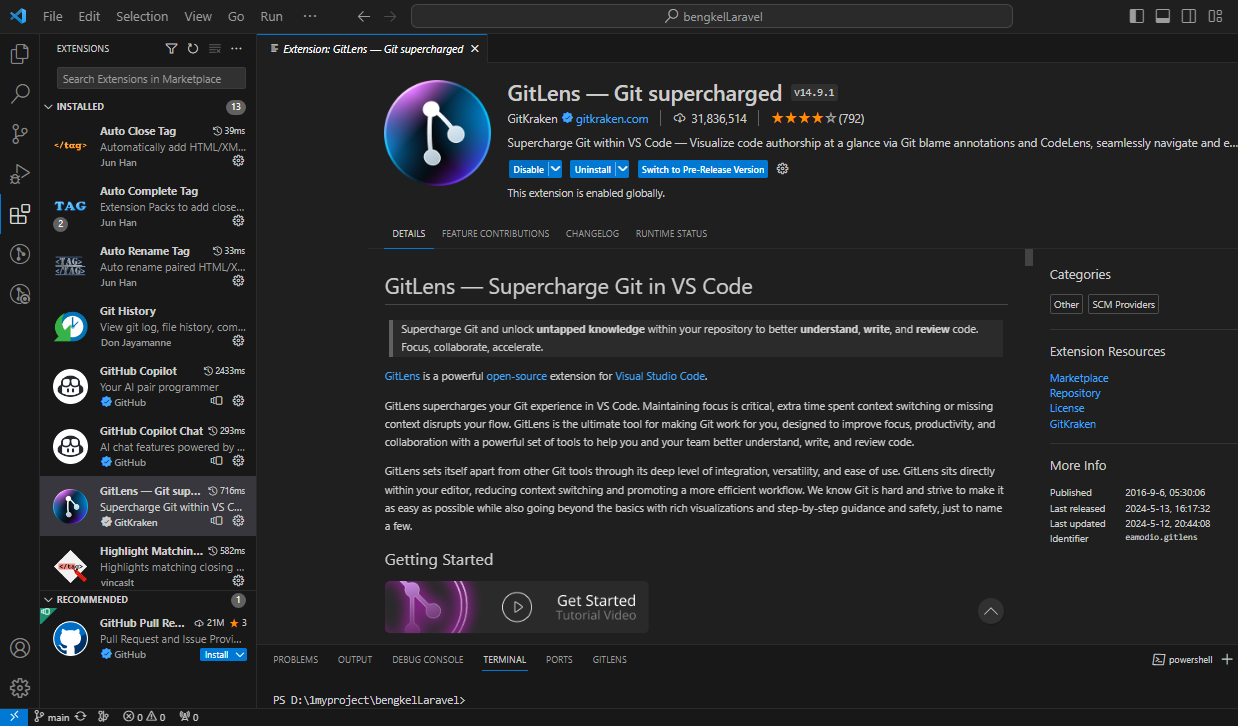


1. **Git Add**

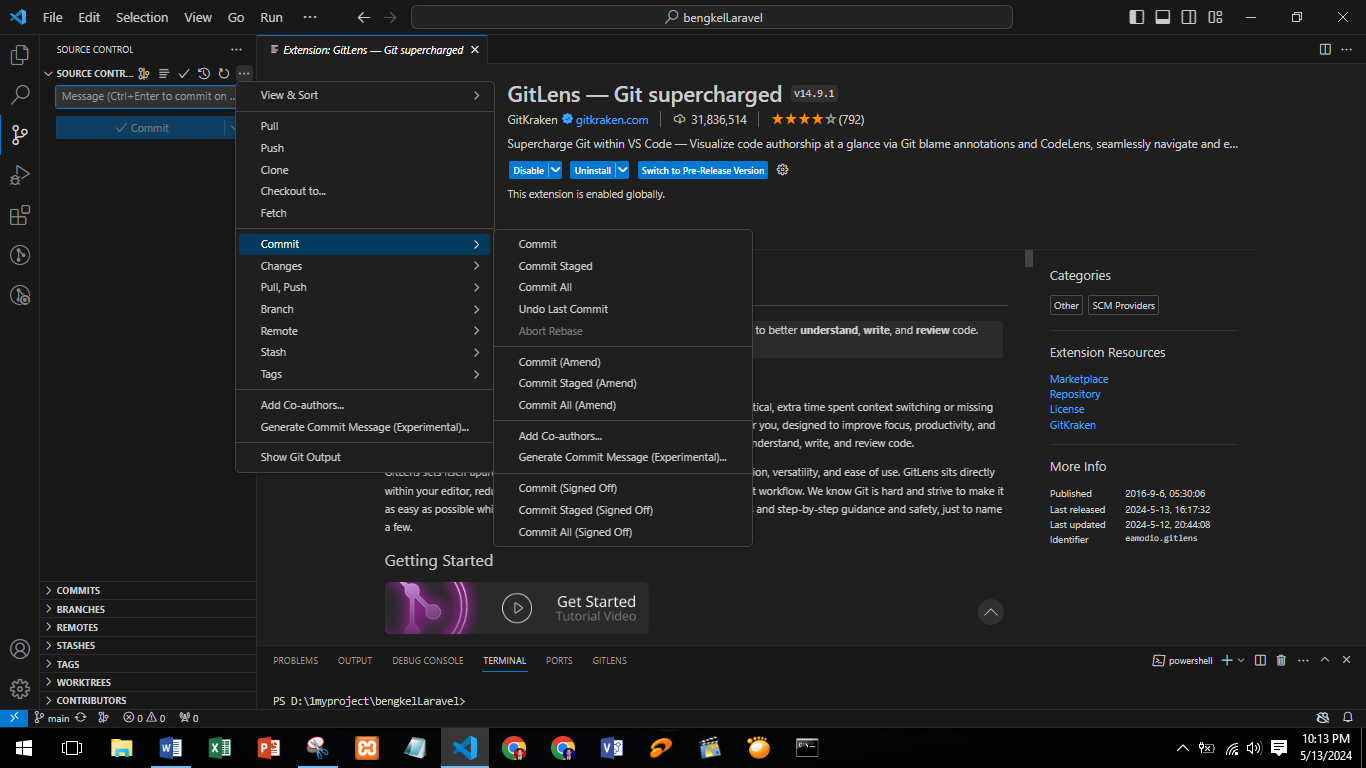
Melakukan git add . untuk melakukan penambahan project yang akan di update.

git add.

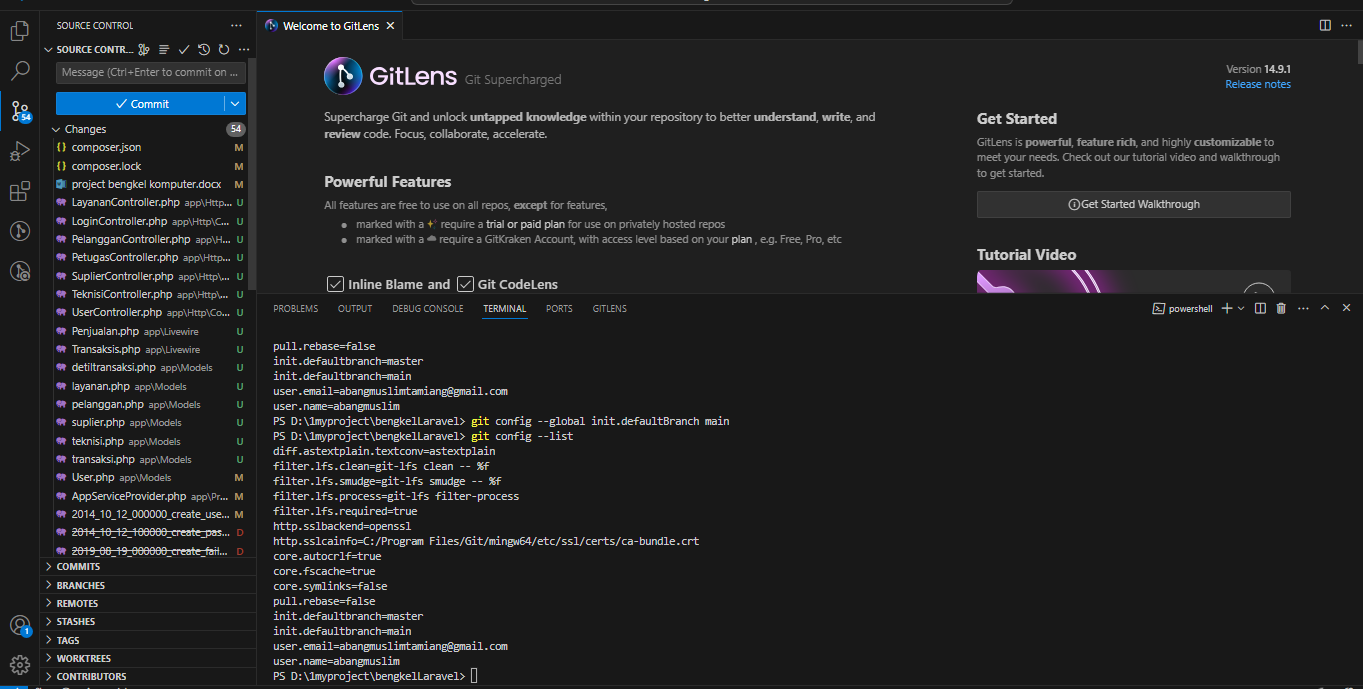
Namun, ada cara yang lebih efektif dan effisien adalah dengan menggunakan extensi dari vs code yaitu **Git Lens**.



Di Ekstensi Ini tidak perlu mengetikkan prompt git add untuk melihat jumlah perubahan yang masuk ke katalog commit. Melainkan hanya perlu mengklik di bagian git tab sidebar kiri vs code.



Selanjutnya hanya perlu mengklik perintah commit jika terlihat bagian yang akan di update ke github sudah sesuai.



Lalu tinggal tunggu proses git di vs code melakukan updat/singkronisasi ke main branch kita yang sudah terinisialisasi di github.

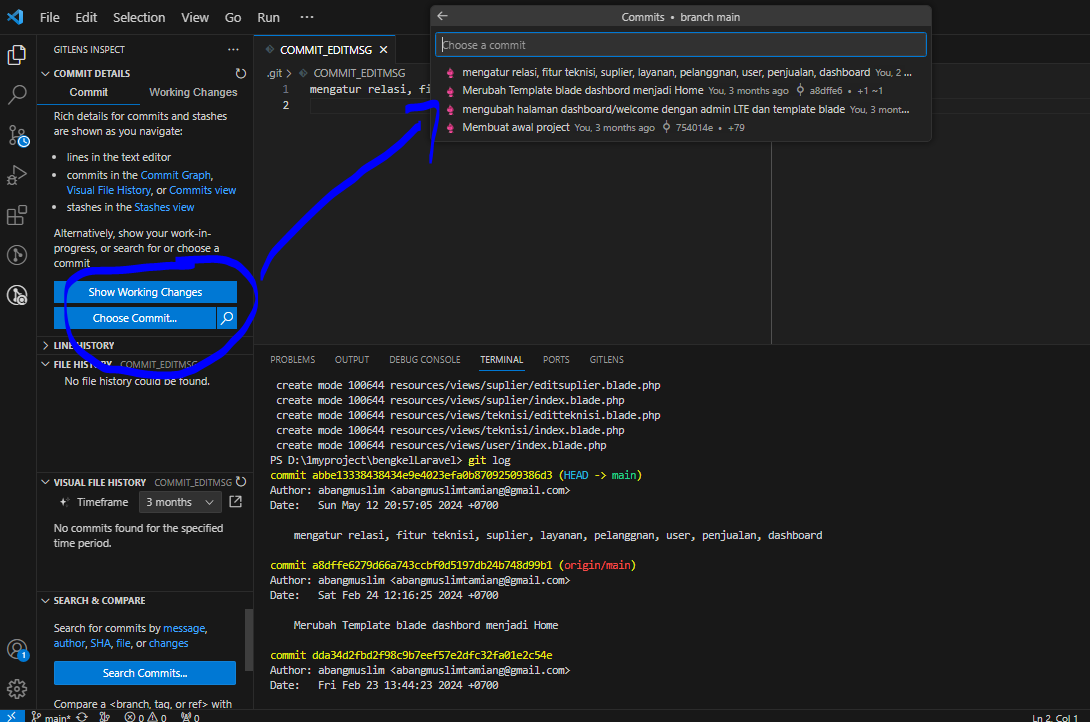


1. **Git Commit**

Melakukan git commit . untuk melakukan commit project yang akan di update.

git commit –m “ komentar atau keterangan mengenai update perubahan yang dilakukan di project ”

Agar lebih efektif dan effisien kita dapat menggunakan git lense. Dan Untuk melakukan commit Jangan lupa memberikan nama pada commit yang kita buat agar perubahan (check point) menjadi lebih dapat dimengerti semua orang, terutama kita sendiri.

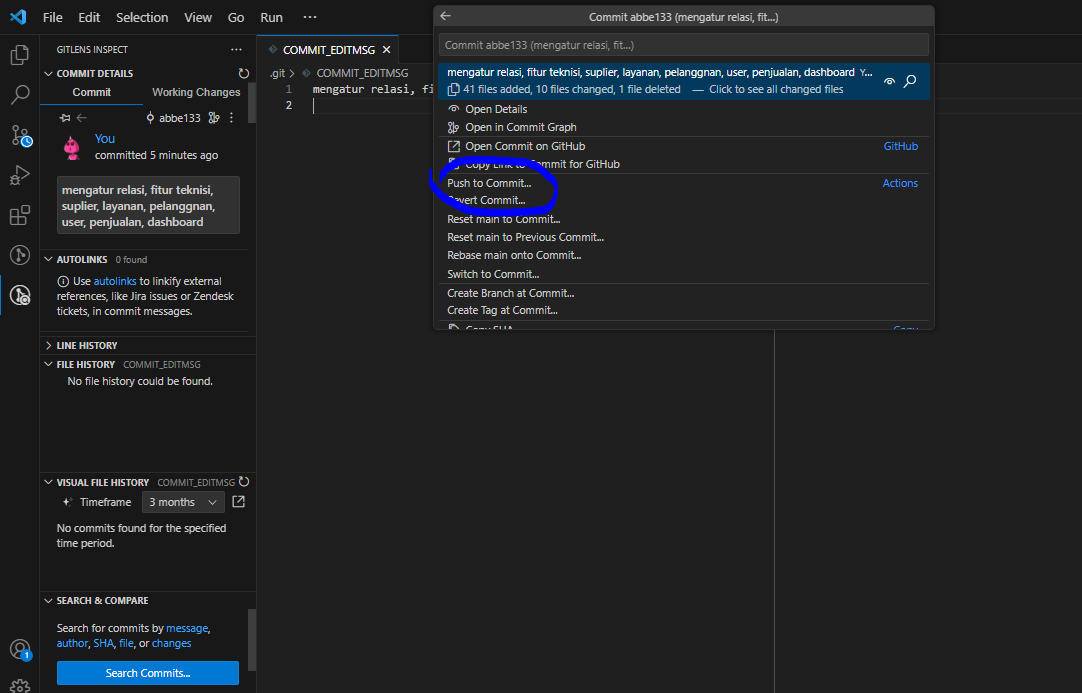


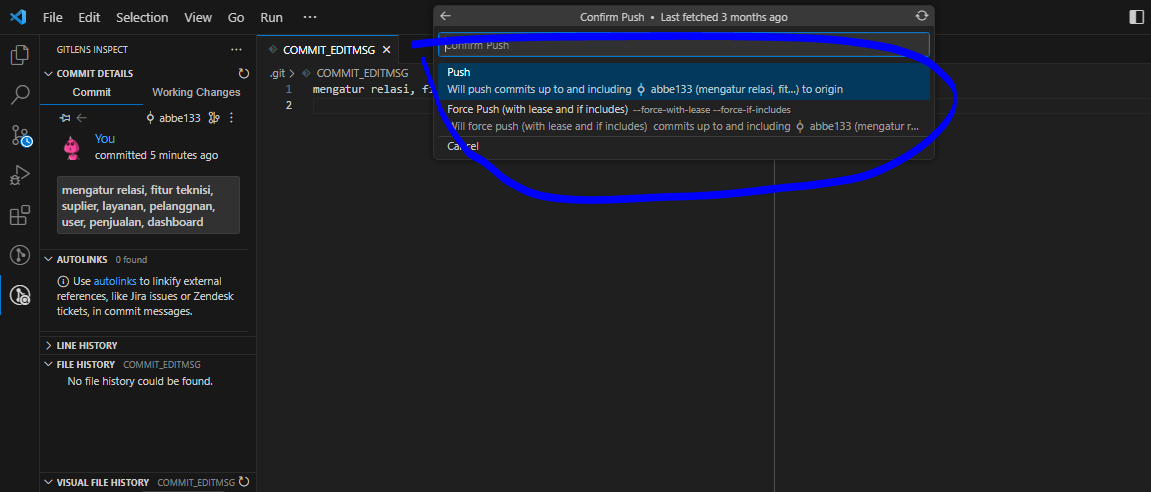
1. **Git push**

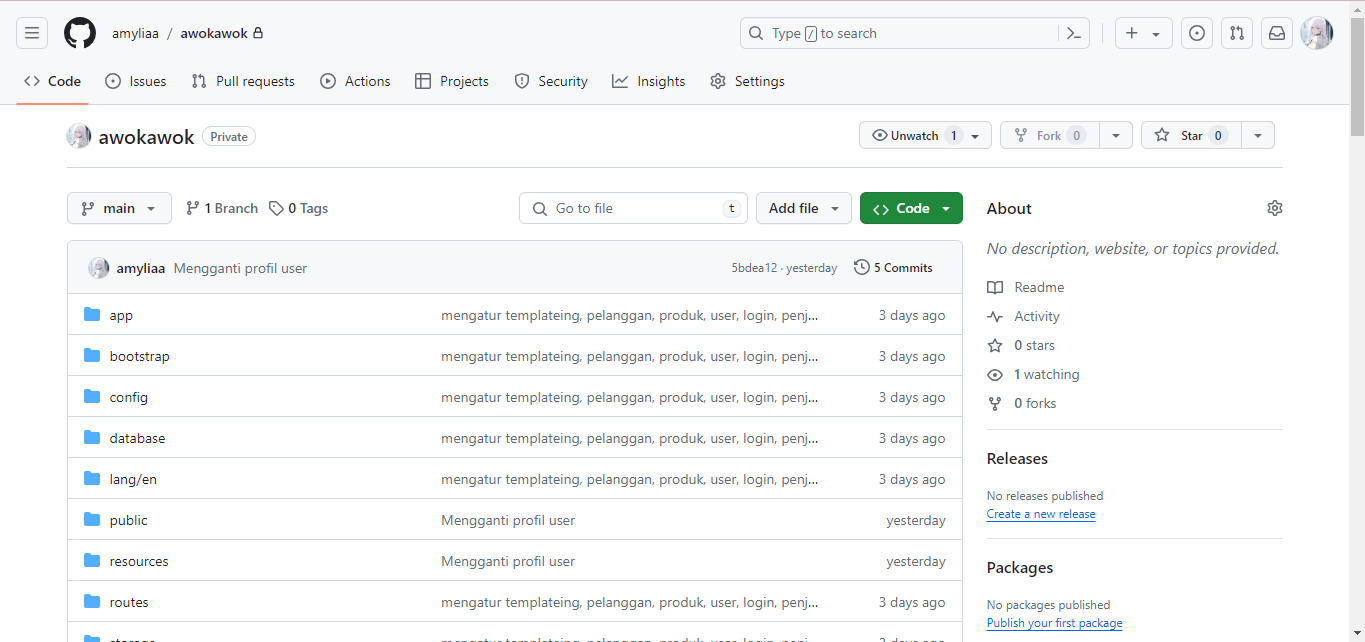
Melakukan git push . untuk melakukan update project kita ke github.

git push.

Agar lebih efektif dan effisien kita dapat menggunakan extensi git lens yang sudah terinstal di vs code.







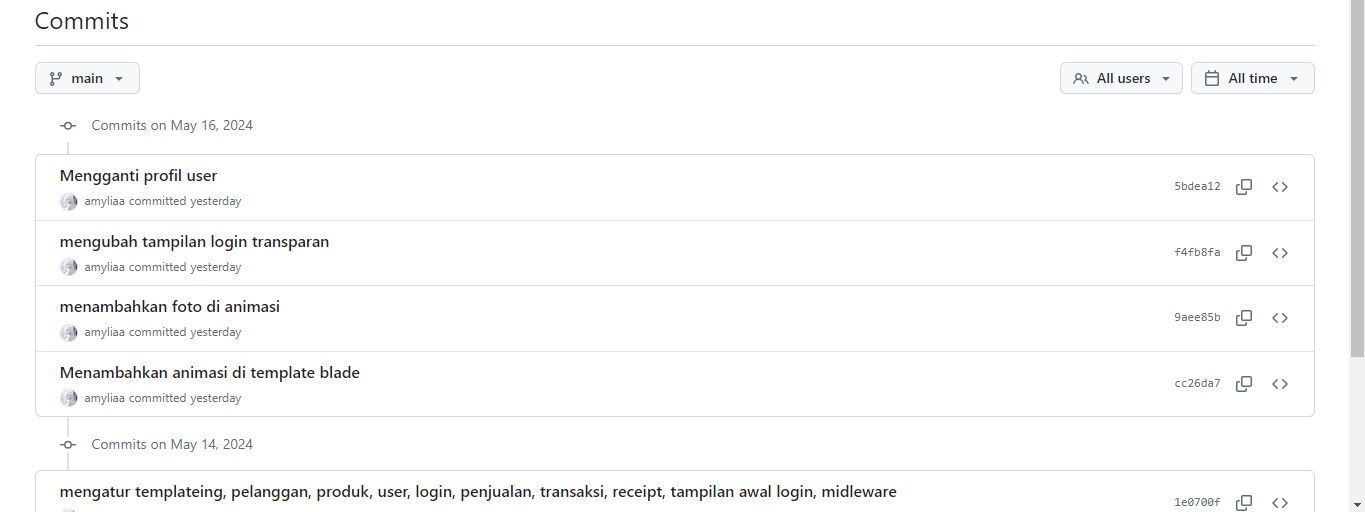
1. **Git log**

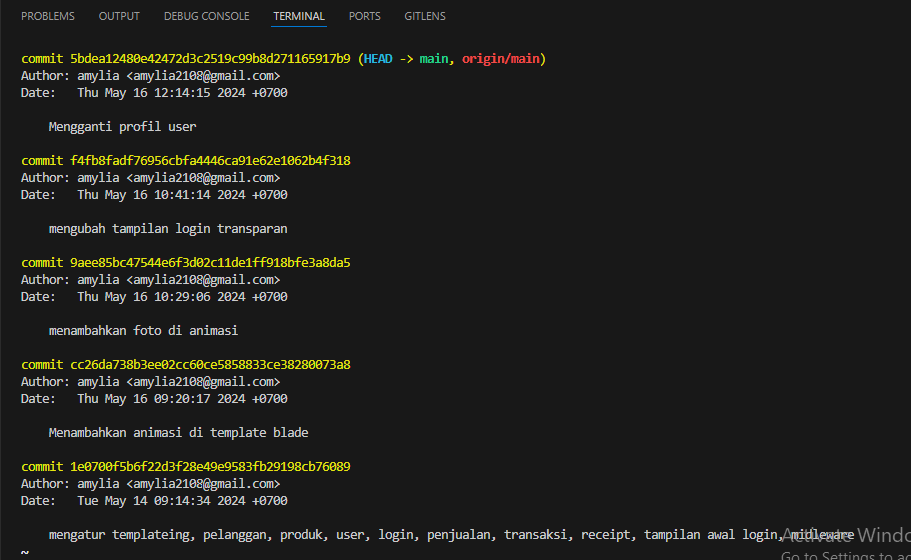
Melakukan git log . untuk melihat/check update file project kita ke github.

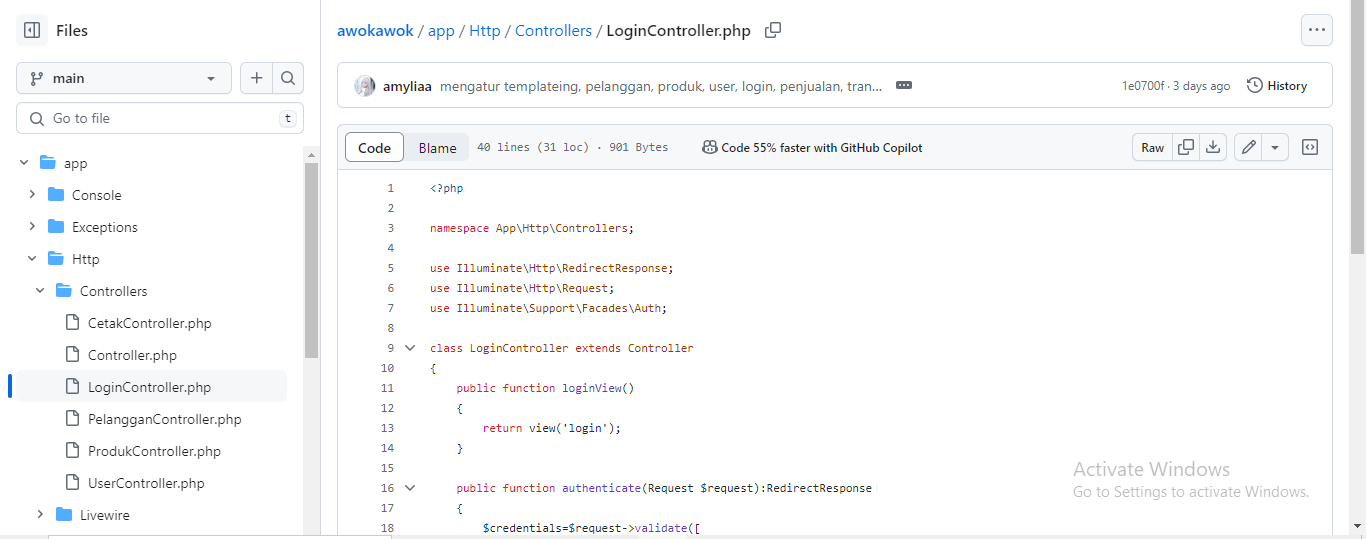
git log.

Agar lebih efektif dan effisien kita dapat menggunakan extensi git lens yang sudah terinstal di vs code.

Namun kita bisa juga langsung mengecek ke repository kita di github. Yaitu dengan mengklik bagian commit. Di menu commit kita dapat juga melihat riwayat dan jumlah commit yang pernah kita buat, lengkap dengan tanggal dan keterangan dari penamaan commit tersebut.







# BAB III TAMPILAN APLIKASI

## Alur Proses Pembuatan Fitur/Menu Pada Aplikasi

Berikut adalah urutan/tahapan yang dilakukan untuk pembuatan aplikasi. Urutan ini sangat mempengaruhi mana fitur yang lebih dahulu dibuat, sehingga proses pembuatan aplikasi dapat berjalan dengan baik.



1. Fitur/menu pertama yang dibuat adalah **Dashboard /Home**, fitur ini perlu dibuat sebagai halaman/pangkalan utama (halaman sentral) yang akan diakses setelah login atau menjadi tempat kembali dari mengakses sebuah menu. Disamping itu dapat pula menjadi halaman rekap dan laporan ringkasan/resume dari aplikasi yang sedang dikerjakan.
2. Fitur/menu selanjutnya yang dibuat adalah **Suplier** dan **Teknisi**. Hal ini karena suplier dan teknisi akan menjadi menu/tabel yang isi/data recordnya diakses dan *diselect* oleh menu produk, sehingga isi/data record dari kedua tabel induk/parent tersebut wajib ada untuk menjadi referensi tabel produk sebagai tabel child. Jika tabel teknisi dan suplier masih kosong, maka user tidak akan dapat mengisi data pada menu produk.
3. Fitur/menu selanjutnya yang dibuat adalah **Produk** dan **Pelanggan**. Data record dari menu produk dan pelanggan nantinya akan diakses oleh menu transaksi, sehingga menu transaksi nantinya tidak akan dapat berjalan normal jika menu produk dan pelanggan belum selesai.
4. Fitur/menu selanjutnya yang dibuat menu **User**. Menu user nantinya akan menjadi menu yang sangat krusial karena menjadi menu yang menangani data pengguna yang bertugas untuk menjalankan aplikasi, oleh karena itu data user pada menu ini akan dilengkapi dengan akun dan passwordnya. Disamping itu menu user nantinya akan dilengkapi dengan authentikasi melalui login untuk melindungi isi dari aplikasi sebagi kendali dan keamanan.
5. Fitur/menu selanjutnya yang dibuat adalah **Transaksi** dan **Detiltransaksi**. Kedua menu ini sebenarnya tergabung dalam 1 menu, yaitu penjualan. Namun dalam pengerjaan melibatkan dua buah tabel child yaitu tabel transaksi dan detiltransaksi.
6. Setelah semua fitur/menu selsai dibuat, maka perlu adanya sebuah **Receipt** atau nota cash bon untuk menjadi bukti/struck pembayaran untuk pelanggan dan penjual. Sebenarnya menu ini sudah include menjadi satu di dalam menu penjualan. Namun dalam pembuatannya menu ini dapat ditempatkan khusus agar setiap level user dapat melacak laporan dari transaksi penjualan yang sudah dijalankan melalui aplikasi.

## Fitur Utama / Dashboard

1. Copy 2 buah folder dari Admin LTE yaitu **plugins** dan **dist** ke dalam folder **public** yang terdapat di project yang telah kita kerjakan.
2. Sehingga di dalam folder public project kita terdapat 2 buah folder tambahan yaitu plugins dan dist.



1. Membuat folder baru dengan nama **layouts** di dalam folder **resources/views**.
2. Lalu buatlah file baru dengan nama **template.blade.php**.



1. Tuliskan script berikut pada file **template.blade.php**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>{{$title}}</title>

    <!-- Google Font: Source Sans Pro -->

    <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback">

    <!-- Font Awesome Icons -->

    <link rel="stylesheet" href="{{ asset('plugins/fontawesome-free/css/all.min.css') }}">

    @yield('tambahanCSS')

    <!-- IonIcons -->

    <link rel="stylesheet" href="https://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css">

    <!-- Theme style -->

    <link rel="stylesheet" href="{{ asset('dist/css/adminlte.min.css')}}">

</head>

<!-- Preloader -->

<div class="preloader flex-column justify-content-center align-items-center">

    <img class="animation\_\_shake" src="dist/img/pp.jpg" alt="bsss" height="150" width="150">

  </div>

<body class="hold-transition sidebar-mini">

    <div class="wrapper">

        <!-- Navbar -->

        <nav class="main-header navbar navbar-expand navbar-white navbar-light">

            <!-- Left navbar links -->

            <ul class="navbar-nav">

                <li class="nav-item">

                    <a class="nav-link" data-widget="pushmenu" href="#" role="button"><i class="fas fa-bars"></i></a>

                </li>

            </ul>

            <!-- Right navbar links -->

            <ul class="navbar-nav ml-auto">

                <li class="nav-item dropdown">

                    <a class="nav-link" data-toggle="dropdown" href="#">

                        <i class="far fa-user mr-2"></i>{{ Auth::user()->name }}

                        <span class="badge badge-warning navbar-badge"></span>

                    </a>

                    <div class="dropdown-menu dropdown-menu-lg dropdown-menu-right">

                        <span class="dropdown-item dropdown-header">User Menu</span>

                        <div class="dropdown-divider"></div>

                        <a href="#" class="dropdown-item">

                            <i class="fas fa-user mr-2"></i> {{ Auth::user()->name }}

                            <span class="float-right text-muted text-sm"></span>

                        </a>

                        <div class="dropdown-divider"></div>

                        <form action="logout" method="POST">

                            @csrf

                            <button type="submit" class="dropdown-item">

                                <i class="fas fa-sign-out-alt mr-2"></i>Logout

                            </button>

                        </form>

                    </div>

                </li>

                <li class="nav-item">

                    <a class="nav-link" data-widget="fullscreen" href="#" role="button">

                        <i class="fas fa-expand-arrows-alt"></i>

                    </a>

                </li>

            </ul>

        </nav>

        <!-- /.navbar -->

        <!-- Main Sidebar Container -->

        <aside class="main-sidebar sidebar-dark-primary elevation-4">

            <!-- Brand Logo -->

            <a href="index3.html" class="brand-link">

                <img src="{{ asset('dist/img/Lg.png')}}" alt="AdminLTE Logo" class="brand-image img-circle elevation-3" style="opacity: .8">

                <span class="brand-text font-weight-light">Toko Buku Cheerful</span>

            </a>

            <!-- Sidebar -->

            <div class="sidebar">

                <!-- Sidebar user panel (optional) -->

                <div class="user-panel mt-3 pb-3 mb-3 d-flex">

                    <div class="image">

                        <img src="{{ asset('dist/img/ssakura.jpg')}}" class="img-circle elevation-2" alt="User Image">

                    </div>

                    <div class="info">

                        <a href="#" class="d-block">{{ Auth::user()->name}}</a>

                    </div>

                </div>

                <!-- Sidebar Menu -->

                <nav class="mt-2">

                    <ul class="nav nav-pills nav-sidebar flex-column" data-widget="treeview" role="menu" data-accordion="false">

                        <!-- Add icons to the links using the .nav-icon class

               with font-awesome or any other icon font library -->

                        <li class="nav-item">

                            <a href="/" class="nav-link {{ ($title==='Dashboard')?'active':'' }}">

                                <i class="nav-icon fas fa-tachometer-alt"></i>

                                <p>

                                    Dashboard

                                </p>

                            </a>

                        </li>

                        <li class="nav-item">

                            <a href="#" class="nav-link {{($title==='Produk')?'active':''}}">

                                <i class="nav-icon fas fa-boxes"></i>

                                <p>

                                    Produk

                                    <i class="fas fa-angle-left right"></i>

                                </p>

                            </a>

                            <ul class="nav nav-treeview">

                                <li class="nav-item">

                                    <a href="{{route('produk.index')}}" class="nav-link">

                                        <i class="far fa-circle nav-icon"></i>

                                        <p>Input Produk</p>

                                    </a>

                                </li>

                                <!-- <li class="nav-item">

                                    <a href="#" class="nav-link">

                                        <i class="far fa-circle nav-icon"></i>

                                        <p>Laporan Produk</p>

                                    </a>

                                </li> -->

                            </ul>

                        </li>

                        <li class="nav-item">

                            <a href="{{ route('pelanggan.index')}}" class="nav-link {{ ($title==='Pelanggan')?'active':''}}">

                                <i class="nav-icon fas fa-users"></i>

                                <p>

                                    Pelanggan

                                </p>

                            </a>

                        </li>

                        <li class="nav-item">

                            <a href="{{ route('pengguna.index')}}" class="nav-link {{ ($title==='Users')?'active':''}}">

                                <i class="nav-icon fas fa-users"></i>

                                <p>

                                    Petugas

                                </p>

                            </a>

                        </li>

                        <li class="nav-item">

                            <a href="penjualan" target="\_blank" class="nav-link {{ ($title==='Penjualan') ? 'active':'' }}">

                                <i class="nav-icon fas fa-cash-register"></i>

                                <p>

                                    Penjualan

                                </p>

                            </a>

                        </li>

                    </ul>

                </nav>

                <!-- /.sidebar-menu -->

            </div>

            <!-- /.sidebar -->

        </aside>

        <!-- Content Wrapper. Contains page content -->

        <div class="content-wrapper">

            <!-- Content Header (Page header) -->

            <div class="content-header">

                <div class="container-fluid">

                    <div class="row mb-2">

                        <div class="col-sm-12">

                            <h1 class="m-0">@yield('judulh1')</h1>

                        </div><!-- /.col -->

                    </div><!-- /.row -->

                </div><!-- /.container-fluid -->

            </div>

            <!-- /.content-header -->

            <!-- Main content -->

            <div class="content">

                <div class="container-fluid">

                    <div class="row">

                        @yield('konten')

                    </div>

                    <!-- /.row -->

                </div>

                <!-- /.container-fluid -->

            </div>

            <!-- /.content -->

        </div>

        <!-- /.content-wrapper -->

        <!-- Control Sidebar -->

        <aside class="control-sidebar control-sidebar-dark">

            <!-- Control sidebar content goes here -->

        </aside>

        <!-- /.control-sidebar -->

    </div>

    <!-- ./wrapper -->

    <!-- REQUIRED SCRIPTS -->

    <!-- jQuery -->

    <script src="{{ asset('plugins/jquery/jquery.min.js')}}"></script>

    <!-- Bootstrap -->

    <script src="{{ asset('plugins/bootstrap/js/bootstrap.bundle.min.js')}}"></script>

    @yield('tambahanJS')

    <!-- AdminLTE -->

    <script src="{{ asset('dist/js/adminlte.js')}}"></script>

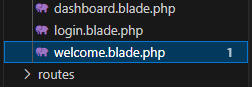
    <!-- OPTIONAL SCRIPTS -->

    <script src="{{ asset('plugins/chart.js/Chart.min.js')}}"></script>

    @yield('tambahScript')

Catatan : script di atas bisa didapatkan dari file index.php yang terdapat pada folder Admin LTE, selanjutnya dimodifikasi sehingga menghasilkan tampilan seperti di atas.

1. Selanjutnya copy script berikut ke file **welcome.blade.php** pada folder **resources/views**.



<!DOCTYPE html>

<html lang="{{ str\_replace('\_', '-', app()->getLocale()) }}">

    <head>

        <meta charset="utf-8">

        <meta name="viewport" content="width=device-width, initial-scale=1">

        <title>Laravel</title>

        <!-- Fonts -->

        <link href="https://fonts.bunny.net/css2?family=Nunito:wght@400;600;700&display=swap" rel="stylesheet">

        <!-- Styles -->

        <style>

            /\*! normalize.css v8.0.1 | MIT License | github.com/necolas/normalize.css \*/html{line-height:1.15;-webkit-text-size-adjust:100%}body{margin:0}a{background-color:transparent}[hidden]{display:none}html{font-family:system-ui,-apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Helvetica Neue,Arial,Noto Sans,sans-serif,Apple Color Emoji,Segoe UI Emoji,Segoe UI Symbol,Noto Color Emoji;line-height:1.5}\*,:after,:before{box-sizing:border-box;border:0 solid #e2e8f0}a{color:inherit;text-decoration:inherit}svg,video{display:block;vertical-align:middle}video{max-width:100%;height:auto}.bg-white{--tw-bg-opacity: 1;background-color:rgb(255 255 255 / var(--tw-bg-opacity))}.bg-gray-100{--tw-bg-opacity: 1;background-color:rgb(243 244 246 / var(--tw-bg-opacity))}.border-gray-200{--tw-border-opacity: 1;border-color:rgb(229 231 235 / var(--tw-border-opacity))}.border-t{border-top-width:1px}.flex{display:flex}.grid{display:grid}.hidden{display:none}.items-center{align-items:center}.justify-center{justify-content:center}.font-semibold{font-weight:600}.h-5{height:1.25rem}.h-8{height:2rem}.h-16{height:4rem}.text-sm{font-size:.875rem}.text-lg{font-size:1.125rem}.leading-7{line-height:1.75rem}.mx-auto{margin-left:auto;margin-right:auto}.ml-1{margin-left:.25rem}.mt-2{margin-top:.5rem}.mr-2{margin-right:.5rem}.ml-2{margin-left:.5rem}.mt-4{margin-top:1rem}.ml-4{margin-left:1rem}.mt-8{margin-top:2rem}.ml-12{margin-left:3rem}.-mt-px{margin-top:-1px}.max-w-6xl{max-width:72rem}.min-h-screen{min-height:100vh}.overflow-hidden{overflow:hidden}.p-6{padding:1.5rem}.py-4{padding-top:1rem;padding-bottom:1rem}.px-6{padding-left:1.5rem;padding-right:1.5rem}.pt-8{padding-top:2rem}.fixed{position:fixed}.relative{position:relative}.top-0{top:0}.right-0{right:0}.shadow{--tw-shadow: 0 1px 3px 0 rgb(0 0 0 / .1), 0 1px 2px -1px rgb(0 0 0 / .1);--tw-shadow-colored: 0 1px 3px 0 var(--tw-shadow-color), 0 1px 2px -1px var(--tw-shadow-color);box-shadow:var(--tw-ring-offset-shadow, 0 0 #0000),var(--tw-ring-shadow, 0 0 #0000),var(--tw-shadow)}.text-center{text-align:center}.text-gray-200{--tw-text-opacity: 1;color:rgb(229 231 235 / var(--tw-text-opacity))}.text-gray-300{--tw-text-opacity: 1;color:rgb(209 213 219 / var(--tw-text-opacity))}.text-gray-400{--tw-text-opacity: 1;color:rgb(156 163 175 / var(--tw-text-opacity))}.text-gray-500{--tw-text-opacity: 1;color:rgb(107 114 128 / var(--tw-text-opacity))}.text-gray-600{--tw-text-opacity: 1;color:rgb(75 85 99 / var(--tw-text-opacity))}.text-gray-700{--tw-text-opacity: 1;color:rgb(55 65 81 / var(--tw-text-opacity))}.text-gray-900{--tw-text-opacity: 1;color:rgb(17 24 39 / var(--tw-text-opacity))}.underline{text-decoration:underline}.antialiased{-webkit-font-smoothing:antialiased;-moz-osx-font-smoothing:grayscale}.w-5{width:1.25rem}.w-8{width:2rem}.w-auto{width:auto}.grid-cols-1{grid-template-columns:repeat(1,minmax(0,1fr))}@media (min-width:640px){.sm\:rounded-lg{border-radius:.5rem}.sm\:block{display:block}.sm\:items-center{align-items:center}.sm\:justify-start{justify-content:flex-start}.sm\:justify-between{justify-content:space-between}.sm\:h-20{height:5rem}.sm\:ml-0{margin-left:0}.sm\:px-6{padding-left:1.5rem;padding-right:1.5rem}.sm\:pt-0{padding-top:0}.sm\:text-left{text-align:left}.sm\:text-right{text-align:right}}@media (min-width:768px){.md\:border-t-0{border-top-width:0}.md\:border-l{border-left-width:1px}.md\:grid-cols-2{grid-template-columns:repeat(2,minmax(0,1fr))}}@media (min-width:1024px){.lg\:px-8{padding-left:2rem;padding-right:2rem}}@media (prefers-color-scheme:dark){.dark\:bg-gray-800{--tw-bg-opacity: 1;background-color:rgb(31 41 55 / var(--tw-bg-opacity))}.dark\:bg-gray-900{--tw-bg-opacity: 1;background-color:rgb(17 24 39 / var(--tw-bg-opacity))}.dark\:border-gray-700{--tw-border-opacity: 1;border-color:rgb(55 65 81 / var(--tw-border-opacity))}.dark\:text-white{--tw-text-opacity: 1;color:rgb(255 255 255 / var(--tw-text-opacity))}.dark\:text-gray-400{--tw-text-opacity: 1;color:rgb(156 163 175 / var(--tw-text-opacity))}.dark\:text-gray-500{--tw-text-opacity: 1;color:rgb(107 114 128 / var(--tw-text-opacity))}}

        </style>

        <style>

            body {

                font-family: 'Nunito', sans-serif;

            }

        </style>

    </head>

    <body class="antialiased">

        <div class="relative flex items-top justify-center min-h-screen bg-gray-100 dark:bg-gray-900 sm:items-center py-4 sm:pt-0">

            @if (Route::has('login'))

                <div class="hidden fixed top-0 right-0 px-6 py-4 sm:block">

                    @auth

                        <a href="{{ url('/home') }}" class="text-sm text-gray-700 dark:text-gray-500 underline">Home</a>

                    @else

                        <a href="{{ route('login') }}" class="text-sm text-gray-700 dark:text-gray-500 underline">Log in</a>

                        @if (Route::has('register'))

                            <a href="{{ route('register') }}" class="ml-4 text-sm text-gray-700 dark:text-gray-500 underline">Register</a>

                        @endif

                    @endauth

                </div>

            @endif

            <div class="max-w-6xl mx-auto sm:px-6 lg:px-8">

                <div class="flex justify-center pt-8 sm:justify-start sm:pt-0">

                    <svg viewBox="0 0 651 192" fill="none" xmlns="http://www.w3.org/2000/svg" class="h-16 w-auto text-gray-700 sm:h-20">

                        <g clip-path="url(#clip0)" fill="#EF3B2D">

                            <path d="M248.032 44.676h-16.466v100.23h47.394v-14.748h-30.928V44.676zM337.091 87.202c-2.101-3.341-5.083-5.965-8.949-7.875-3.865-1.909-7.756-2.864-11.669-2.864-5.062 0-9.69.931-13.89 2.792-4.201 1.861-7.804 4.417-10.811 7.661-3.007 3.246-5.347 6.993-7.016 11.239-1.672 4.249-2.506 8.713-2.506 13.389 0 4.774.834 9.26 2.506 13.459 1.669 4.202 4.009 7.925 7.016 11.169 3.007 3.246 6.609 5.799 10.811 7.66 4.199 1.861 8.828 2.792 13.89 2.792 3.913 0 7.804-.955 11.669-2.863 3.866-1.908 6.849-4.533 8.949-7.875v9.021h15.607V78.182h-15.607v9.02zm-1.431 32.503c-.955 2.578-2.291 4.821-4.009 6.73-1.719 1.91-3.795 3.437-6.229 4.582-2.435 1.146-5.133 1.718-8.091 1.718-2.96 0-5.633-.572-8.019-1.718-2.387-1.146-4.438-2.672-6.156-4.582-1.719-1.909-3.032-4.152-3.938-6.73-.909-2.577-1.36-5.298-1.36-8.161 0-2.864.451-5.585 1.36-8.162.905-2.577 2.219-4.819 3.938-6.729 1.718-1.908 3.77-3.437 6.156-4.582 2.386-1.146 5.059-1.718 8.019-1.718 2.958 0 5.656.572 8.091 1.718 2.434 1.146 4.51 2.674 6.229 4.582 1.718 1.91 3.054 4.152 4.009 6.729.953 2.577 1.432 5.298 1.432 8.162-.001 2.863-.479 5.584-1.432 8.161zM463.954 87.202c-2.101-3.341-5.083-5.965-8.949-7.875-3.865-1.909-7.756-2.864-11.669-2.864-5.062 0-9.69.931-13.89 2.792-4.201 1.861-7.804 4.417-10.811 7.661-3.007 3.246-5.347 6.993-7.016 11.239-1.672 4.249-2.506 8.713-2.506 13.389 0 4.774.834 9.26 2.506 13.459 1.669 4.202 4.009 7.925 7.016 11.169 3.007 3.246 6.609 5.799 10.811 7.66 4.199 1.861 8.828 2.792 13.89 2.792 3.913 0 7.804-.955 11.669-2.863 3.866-1.908 6.849-4.533 8.949-7.875v9.021h15.607V78.182h-15.607v9.02zm-1.432 32.503c-.955 2.578-2.291 4.821-4.009 6.73-1.719 1.91-3.795 3.437-6.229 4.582-2.435 1.146-5.133 1.718-8.091 1.718-2.96 0-5.633-.572-8.019-1.718-2.387-1.146-4.438-2.672-6.156-4.582-1.719-1.909-3.032-4.152-3.938-6.73-.909-2.577-1.36-5.298-1.36-8.161 0-2.864.451-5.585 1.36-8.162.905-2.577 2.219-4.819 3.938-6.729 1.718-1.908 3.77-3.437 6.156-4.582 2.386-1.146 5.059-1.718 8.019-1.718 2.958 0 5.656.572 8.091 1.718 2.434 1.146 4.51 2.674 6.229 4.582 1.718 1.91 3.054 4.152 4.009 6.729.953 2.577 1.432 5.298 1.432 8.162 0 2.863-.479 5.584-1.432 8.161zM650.772 44.676h-15.606v100.23h15.606V44.676zM365.013 144.906h15.607V93.538h26.776V78.182h-42.383v66.724zM542.133 78.182l-19.616 51.096-19.616-51.096h-15.808l25.617 66.724h19.614l25.617-66.724h-15.808zM591.98 76.466c-19.112 0-34.239 15.706-34.239 35.079 0 21.416 14.641 35.079 36.239 35.079 12.088 0 19.806-4.622 29.234-14.688l-10.544-8.158c-.006.008-7.958 10.449-19.832 10.449-13.802 0-19.612-11.127-19.612-16.884h51.777c2.72-22.043-11.772-40.877-33.023-40.877zm-18.713 29.28c.12-1.284 1.917-16.884 18.589-16.884 16.671 0 18.697 15.598 18.813 16.884h-37.402zM184.068 43.892c-.024-.088-.073-.165-.104-.25-.058-.157-.108-.316-.191-.46-.056-.097-.137-.176-.203-.265-.087-.117-.161-.242-.265-.345-.085-.086-.194-.148-.29-.223-.109-.085-.206-.182-.327-.252l-.002-.001-.002-.002-35.648-20.524a2.971 2.971 0 00-2.964 0l-35.647 20.522-.002.002-.002.001c-.121.07-.219.167-.327.252-.096.075-.205.138-.29.223-.103.103-.178.228-.265.345-.066.089-.147.169-.203.265-.083.144-.133.304-.191.46-.031.085-.08.162-.104.25-.067.249-.103.51-.103.776v38.979l-29.706 17.103V24.493a3 3 0 00-.103-.776c-.024-.088-.073-.165-.104-.25-.058-.157-.108-.316-.191-.46-.056-.097-.137-.176-.203-.265-.087-.117-.161-.242-.265-.345-.085-.086-.194-.148-.29-.223-.109-.085-.206-.182-.327-.252l-.002-.001-.002-.002L40.098 1.396a2.971 2.971 0 00-2.964 0L1.487 21.919l-.002.002-.002.001c-.121.07-.219.167-.327.252-.096.075-.205.138-.29.223-.103.103-.178.228-.265.345-.066.089-.147.169-.203.265-.083.144-.133.304-.191.46-.031.085-.08.162-.104.25-.067.249-.103.51-.103.776v122.09c0 1.063.568 2.044 1.489 2.575l71.293 41.045c.156.089.324.143.49.202.078.028.15.074.23.095a2.98 2.98 0 001.524 0c.069-.018.132-.059.2-.083.176-.061.354-.119.519-.214l71.293-41.045a2.971 2.971 0 001.489-2.575v-38.979l34.158-19.666a2.971 2.971 0 001.489-2.575V44.666a3.075 3.075 0 00-.106-.774zM74.255 143.167l-29.648-16.779 31.136-17.926.001-.001 34.164-19.669 29.674 17.084-21.772 12.428-43.555 24.863zm68.329-76.259v33.841l-12.475-7.182-17.231-9.92V49.806l12.475 7.182 17.231 9.92zm2.97-39.335l29.693 17.095-29.693 17.095-29.693-17.095 29.693-17.095zM54.06 114.089l-12.475 7.182V46.733l17.231-9.92 12.475-7.182v74.537l-17.231 9.921zM38.614 7.398l29.693 17.095-29.693 17.095L8.921 24.493 38.614 7.398zM5.938 29.632l12.475 7.182 17.231 9.92v79.676l.001.005-.001.006c0 .114.032.221.045.333.017.146.021.294.059.434l.002.007c.032.117.094.222.14.334.051.124.088.255.156.371a.036.036 0 00.004.009c.061.105.149.191.222.288.081.105.149.22.244.314l.008.01c.084.083.19.142.284.215.106.083.202.178.32.247l.013.005.011.008 34.139 19.321v34.175L5.939 144.867V29.632h-.001zm136.646 115.235l-65.352 37.625V148.31l48.399-27.628 16.953-9.677v33.862zm35.646-61.22l-29.706 17.102V66.908l17.231-9.92 12.475-7.182v33.841z"/>

                        </g>

                    </svg>

                </div>

                <div class="mt-8 bg-white dark:bg-gray-800 overflow-hidden shadow sm:rounded-lg">

                    <div class="grid grid-cols-1 md:grid-cols-2">

                        <div class="p-6">

                            <div class="flex items-center">

                                <svg xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24" stroke-width="1.5" stroke="currentColor" class="w-8 h-8 text-gray-500"><path stroke-linecap="round" stroke-linejoin="round" d="M12 6.042A8.967 8.967 0 006 3.75c-1.052 0-2.062.18-3 .512v14.25A8.987 8.987 0 016 18c2.305 0 4.408.867 6 2.292m0-14.25a8.966 8.966 0 016-2.292c1.052 0 2.062.18 3 .512v14.25A8.987 8.987 0 0018 18a8.967 8.967 0 00-6 2.292m0-14.25v14.25" /></svg>

                                <div class="ml-4 text-lg leading-7 font-semibold"><a href="https://laravel.com/docs" class="underline text-gray-900 dark:text-white">Documentation</a></div>

                            </div>

                            <div class="ml-12">

                                <div class="mt-2 text-gray-600 dark:text-gray-400 text-sm">

                                    Laravel has wonderful, thorough documentation covering every aspect of the framework. Whether you are new to the framework or have previous experience with Laravel, we recommend reading all of the documentation from beginning to end.

                                </div>

                            </div>

                        </div>

                        <div class="p-6 border-t border-gray-200 dark:border-gray-700 md:border-t-0 md:border-l">

                            <div class="flex items-center">

                                <svg xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24" stroke-width="1.5" stroke="currentColor" class="w-8 h-8 text-gray-500"><path stroke-linecap="round" d="M15.75 10.5l4.72-4.72a.75.75 0 011.28.53v11.38a.75.75 0 01-1.28.53l-4.72-4.72M4.5 18.75h9a2.25 2.25 0 002.25-2.25v-9a2.25 2.25 0 00-2.25-2.25h-9A2.25 2.25 0 002.25 7.5v9a2.25 2.25 0 002.25 2.25z" /></svg>

                                <div class="ml-4 text-lg leading-7 font-semibold"><a href="https://laracasts.com" class="underline text-gray-900 dark:text-white">Laracasts</a></div>

                            </div>

                            <div class="ml-12">

                                <div class="mt-2 text-gray-600 dark:text-gray-400 text-sm">

                                    Laracasts offers thousands of video tutorials on Laravel, PHP, and JavaScript development. Check them out, see for yourself, and massively level up your development skills in the process.

                                </div>

                            </div>

                        </div>

                        <div class="p-6 border-t border-gray-200 dark:border-gray-700">

                            <div class="flex items-center">

                                <svg xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24" stroke-width="1.5" stroke="currentColor" class="w-8 h-8 text-gray-500"><path stroke-linecap="round" stroke-linejoin="round" d="M7.5 8.25h9m-9 3H12m-9.75 1.51c0 1.6 1.123 2.994 2.707 3.227 1.129.166 2.27.293 3.423.379.35.026.67.21.865.501L12 21l2.755-4.133a1.14 1.14 0 01.865-.501 48.172 48.172 0 003.423-.379c1.584-.233 2.707-1.626 2.707-3.228V6.741c0-1.602-1.123-2.995-2.707-3.228A48.394 48.394 0 0012 3c-2.392 0-4.744.175-7.043.513C3.373 3.746 2.25 5.14 2.25 6.741v6.018z" /></svg>

                                <div class="ml-4 text-lg leading-7 font-semibold"><a href="https://laravel-news.com/" class="underline text-gray-900 dark:text-white">Laravel News</a></div>

                            </div>

                            <div class="ml-12">

                                <div class="mt-2 text-gray-600 dark:text-gray-400 text-sm">

                                    Laravel News is a community driven portal and newsletter aggregating all of the latest and most important news in the Laravel ecosystem, including new package releases and tutorials.

                                </div>

                            </div>

                        </div>

                        <div class="p-6 border-t border-gray-200 dark:border-gray-700 md:border-l">

                            <div class="flex items-center">

                                <svg xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24" stroke-width="1.5" stroke="currentColor" class="w-8 h-8 text-gray-500"><path stroke-linecap="round" stroke-linejoin="round" d="M6.115 5.19l.319 1.913A6 6 0 008.11 10.36L9.75 12l-.387.775c-.217.433-.132.956.21 1.298l1.348 1.348c.21.21.329.497.329.795v1.089c0 .426.24.815.622 1.006l.153.076c.433.217.956.132 1.298-.21l.723-.723a8.7 8.7 0 002.288-4.042 1.087 1.087 0 00-.358-1.099l-1.33-1.108c-.251-.21-.582-.299-.905-.245l-1.17.195a1.125 1.125 0 01-.98-.314l-.295-.295a1.125 1.125 0 010-1.591l.13-.132a1.125 1.125 0 011.3-.21l.603.302a.809.809 0 001.086-1.086L14.25 7.5l1.256-.837a4.5 4.5 0 001.528-1.732l.146-.292M6.115 5.19A9 9 0 1017.18 4.64M6.115 5.19A8.965 8.965 0 0112 3c1.929 0 3.716.607 5.18 1.64" /></svg>

                                <div class="ml-4 text-lg leading-7 font-semibold text-gray-900 dark:text-white">Vibrant Ecosystem</div>

                            </div>

                            <div class="ml-12">

                                <div class="mt-2 text-gray-600 dark:text-gray-400 text-sm">

                                    Laravel's robust library of first-party tools and libraries, such as <a href="https://forge.laravel.com" class="underline">Forge</a>, <a href="https://vapor.laravel.com" class="underline">Vapor</a>, <a href="https://nova.laravel.com" class="underline">Nova</a>, and <a href="https://envoyer.io" class="underline">Envoyer</a> help you take your projects to the next level. Pair them with powerful open source libraries like <a href="https://laravel.com/docs/billing" class="underline">Cashier</a>, <a href="https://laravel.com/docs/dusk" class="underline">Dusk</a>, <a href="https://laravel.com/docs/broadcasting" class="underline">Echo</a>, <a href="https://laravel.com/docs/horizon" class="underline">Horizon</a>, <a href="https://laravel.com/docs/sanctum" class="underline">Sanctum</a>, <a href="https://laravel.com/docs/telescope" class="underline">Telescope</a>, and more.

                                </div>

                            </div>

                        </div>

                    </div>

                </div>

                <div class="flex justify-center mt-4 sm:items-center sm:justify-between">

                    <div class="text-center text-sm text-gray-500 sm:text-left">

                        <div class="flex items-center">

                            <svg xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24" stroke-width="1.5" stroke="currentColor" class="-mt-px w-5 h-5 text-gray-400">

                                <path stroke-linecap="round" stroke-linejoin="round" d="M2.25 3h1.386c.51 0 .955.343 1.087.835l.383 1.437M7.5 14.25a3 3 0 00-3 3h15.75m-12.75-3h11.218c1.121-2.3 2.1-4.684 2.924-7.138a60.114 60.114 0 00-16.536-1.84M7.5 14.25L5.106 5.272M6 20.25a.75.75 0 11-1.5 0 .75.75 0 011.5 0zm12.75 0a.75.75 0 11-1.5 0 .75.75 0 011.5 0z" />

                            </svg>

                            <a href="https://laravel.bigcartel.com" class="ml-1 underline">

                                Shop

                            </a>

                            <svg xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24" stroke-width="1.5" stroke="currentColor" class="ml-4 -mt-px w-5 h-5 text-gray-400">

                                <path stroke-linecap="round" stroke-linejoin="round" d="M21 8.25c0-2.485-2.099-4.5-4.688-4.5-1.935 0-3.597 1.126-4.312 2.733-.715-1.607-2.377-2.733-4.313-2.733C5.1 3.75 3 5.765 3 8.25c0 7.22 9 12 9 12s9-4.78 9-12z" />

                            </svg>

                            <a href="https://github.com/sponsors/taylorotwell" class="ml-1 underline">

                                Sponsor

                            </a>

                        </div>

                    </div>

                    <div class="ml-4 text-center text-sm text-gray-500 sm:text-right sm:ml-0">

                        Laravel v{{ Illuminate\Foundation\Application::VERSION }} (PHP v{{ PHP\_VERSION }})

                    </div>

                </div>

            </div>

        </div>

    </body>

</html>

1. Selanjutnya tambahkan script berikut pada routes/web.php.

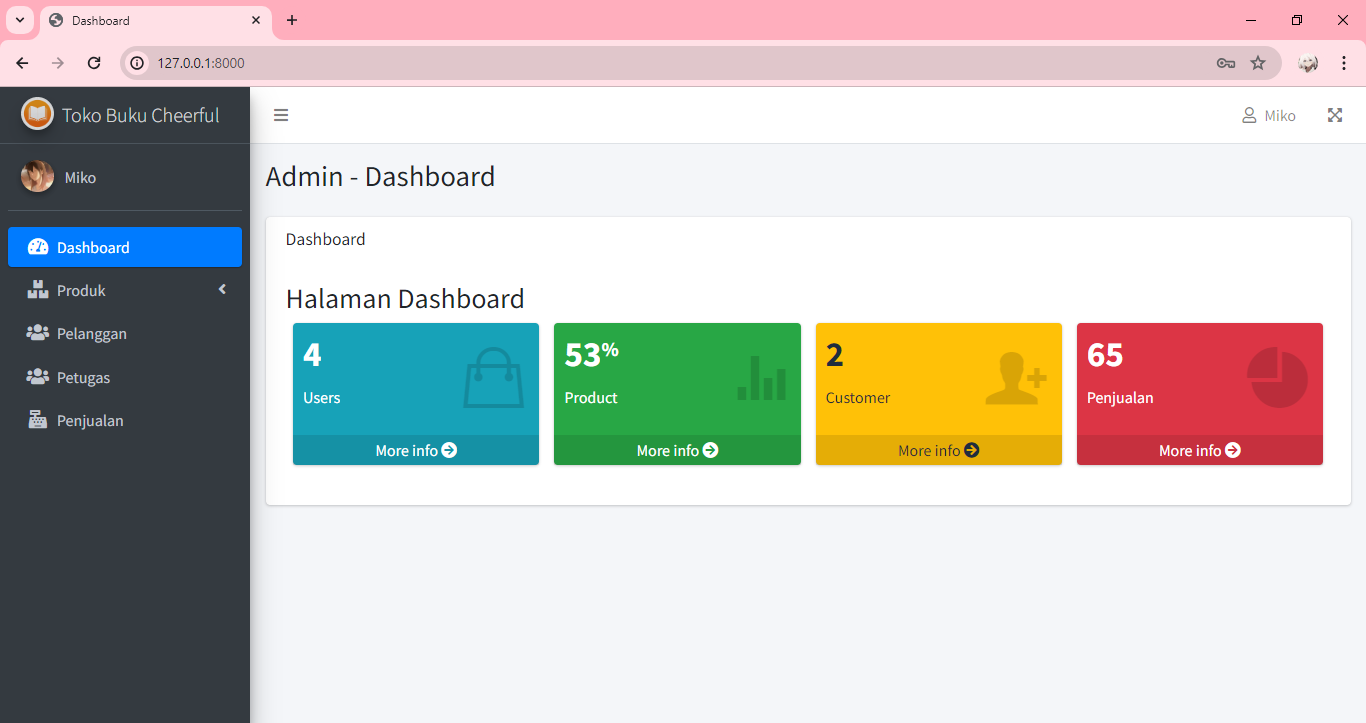
Route::get('/', function () {

    return view('dashboard', [

        "title" => "Dashboard"

    ]);

1. Jalankan server kemudian silahkan buka alamat <http://127.0.0.1:8000/> atau localhost:8000



## Fitur Produk

### **Membuat ProdukController dan Modifikasi Model Produk dan Model Category**

* Buatlah controller produk dengan cara menjalankan perintah php artisan berikut:

php artisan make:controller ProdukController

* Jika sudah berhasil cek kembali **ProdukController.php** pada **App\Http\Controller**

### 

### **Membuat Route Produk dan Method index()**

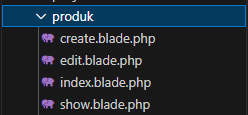
* Buka file **routes\web.php** lalu tambahkan Route berikut:

Route::resource('produk',ProdukController::class)->middleware('auth');

* Jangan lupa untuk meng-import class **ProdukController** ke dalam **routes/web.php**.

use App\Http\Controllers\ProdukController;

* Kemudian buat sebuah folder di dalam folder views dengan nama produk, lalu buat 4 file dengan nama **index.blade.php**, **create.blade.php**, **edit.blade.php**, dan **show.blade.php**.



Kalau diperhatikan pada bagian produk ini kita mulai menggunakan penamaan standar yang biasanya digunakan dalam project laravel. File **index**, akan menjadi tampilan utama dari produk, kemudian file **create** adalah tampilan untuk menambahkan produk, file **edit** untuk merubah/update produk, sedangkan file **show** untuk menampilkan detail produk.

* Buka file index.blade.php lalu tambahkan script berikut:

@extends('layouts.template')

@section('tambahanCSS')

<!-- DataTables -->

<link rel="stylesheet" href="plugins/datatables-bs4/css/dataTables.bootstrap4.min.css">

<link rel="stylesheet" href="plugins/datatables-responsive/css/responsive.bootstrap4.min.css">

<link rel="stylesheet" href="plugins/datatables-buttons/css/buttons.bootstrap4.min.css">

<!-- Toastr -->

<link rel="stylesheet" type="text/css" href="//cdnjs.cloudflare.com/ajax/libs/toastr.js/latest/css/toastr.css">

@endsection

@section('judulh1','Admin - Produk')

@section('konten')

<div class="col-md-12">

    <div class="card card-info">

        <div class="card-header">

            <h2 class="card-title">Data Pelanggan</h2>

            <a type="button" class="btn btn-success float-right" href="{{ route('produk.create') }}">

                <i class=" fas fa-plus"></i> Tambah Produk

            </a>

        </div>

        <!-- /.card-header -->

        <div class="card-body">

            <table id="example1" class="table table-bordered table-striped ">

                <thead>

                    <tr>

                        <th>No</th>

                        <th>Nama</th>

                        <th>Stok</th>

                        <th>Harga</th>

                        <th>Deskripsi</th>

                        <th>Aksi</th>

                    </tr>

                </thead>

                <tbody>

                    @foreach($data as $dt)

                    <tr>

                        <td>{{ $loop->iteration }}</td>

                        <td>{{ $dt->nama }}</td>

                        <td>{{ $dt->stok }}</td>

                        <td>{{ $dt->deskripsi }}</td>

                        <td>@money($dt->harga) </td>

                        <td>

                            <div class="btn-group">

                                <form action="{{ route('produk.destroy',$dt->id)}}" method="POST">

                                    @csrf

                                    @method('DELETE')

                                    <button type="submit" class="btn btn-danger">

                                        <i class=" fas fa-trash"></i>

                                    </button>

                                </form>

                                <a type="button" class="btn btn-warning" href="{{ route('produk.edit',$dt->id) }}">

                                    <i class=" fas fa-edit"></i>

                                </a>

                                <a type="button" class="btn btn-success" href="{{ route('produk.show',$dt->id) }}">

                                    <i class=" fas fa-eye"></i>

                                </a>

                            </div>

                        </td>

                    </tr>

                    @endforeach

                </tbody>

            </table>

        </div>

    </div>

</div>

@endsection

@section('tambahanJS')

<!-- DataTables  & Plugins -->

<script src="plugins/datatables/jquery.dataTables.min.js"></script>

<script src="plugins/datatables-bs4/js/dataTables.bootstrap4.min.js"></script>

<script src="plugins/datatables-responsive/js/dataTables.responsive.min.js"></script>

<script src="plugins/datatables-responsive/js/responsive.bootstrap4.min.js"></script>

<script src="plugins/datatables-buttons/js/dataTables.buttons.min.js"></script>

<script src="plugins/datatables-buttons/js/buttons.bootstrap4.min.js"></script>

<script src="plugins/jszip/jszip.min.js"></script>

<script src="plugins/pdfmake/pdfmake.min.js"></script>

<script src="plugins/pdfmake/vfs\_fonts.js"></script>

<script src="plugins/datatables-buttons/js/buttons.html5.min.js"></script>

<script src="plugins/datatables-buttons/js/buttons.print.min.js"></script>

<script src="plugins/datatables-buttons/js/buttons.colVis.min.js"></script>

<!-- Toastr -->

<script src="https://cdnjs.cloudflare.com/ajax/libs/toastr.js/latest/toastr.min.js"></script>

@endsection

@section('tambahScript')

<script>

$(function() {

    $("#example1").DataTable({

        "responsive": true,

        "lengthChange": true,

        "autoWidth": false,

        "responsive": true,

    }).buttons().container().appendTo('#example1\_wrapper .col-md-6:eq(0)');

});

@if($message = Session::get('success'))

toastr.success("{{ $message}}");

@elseif($message = Session::get('updated'))

toastr.warning("{{ $message}}");

@elseif($message = Session::get('deleted'))

toastr.error("{{ $message}}");

@endif

</script>

@endsection

* Buka file **template.blade.php** pada folder **views\layouts**. Lalu tambahkan script yang ditandai berikut:

<ul class="nav nav-treeview">

                                <li class="nav-item">

                                    <a href="{{route('produk.index')}}" class="nav-link">

                                        <i class="far fa-circle nav-icon"></i>

                                        <p>Input Produk</p>

                                    </a>

                                </li>

* Buka file **ProdukController.php**, lalu tambahkan Method index() di dalam class ProdukController.

public function index()

    {

        $produk=Produk::all();

        return view('produk.index',[

            "title"=>"Produk",

            "data"=>$produk

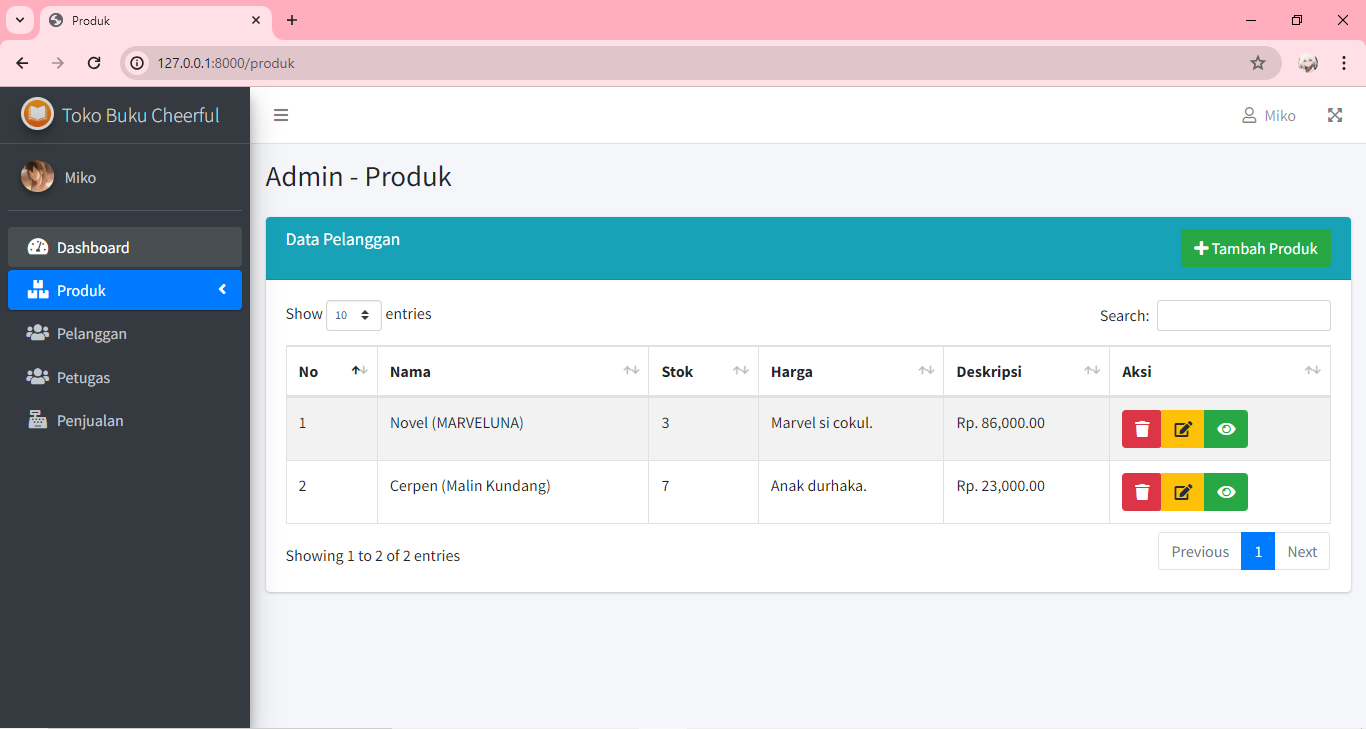
        ]);

    }

* Jangan lupa untuk import model Produk

use App\Models\Produk;

* Uji coba yang telah dikerjakan.



### **Membuat Method create() dan store()**

* Jika diperhatikan dari manajemen customer sebelumnya, method create() digunakan untuk menampilkan tampilan untuk menambah data sementara method store() digunakan untuk menyimpan data ke dalam database.
* Memperbaiki Controller Produk pada file ProdukController.php

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\Produk;

use Illuminate\Http\RedirectResponse;

use Illuminate\Support\Facades\Redirect;

use Illuminate\View\View;

class ProdukController extends Controller

{

    public function index()

    {

        $produk=Produk::all();

        return view('produk.index',[

            "title"=>"Produk",

            "data"=>$produk

        ]);

    }

    public function create():View

    {

        return view('produk.create')->with([

            "title"=> "Tambah Data Produk",

        ]);

    }

    public function store(Request $request):RedirectResponse

    {

        $request->validate([

            "nama"=>"required",

            "deskripsi"=>"nullable",

            "stok"=>"required",

            "harga"=>"required",

        ]);

        Produk::create($request->all());

        return redirect()->route('produk.index')->with('success','Data Produk Berhasil Ditambahkan');

    }

    public function edit(Produk $produk):View

    {

        return view('produk.edit',compact('produk'))->with([

            "title" => "Ubah Data Produk",

        ]);

    }

    public function update(Produk $produk, Request $request):RedirectResponse

    {

        $request->validate([

            "nama"=>"required",

            "deskripsi"=>"nullable",

            "stok"=>"required",

            "harga"=>"required",

        ]);

        $produk->update($request->all());

        return redirect()->route('produk.index')->with('Updated','Data Produk Berhasil Diubah');

    }

    public function show():View

    {

        $produk=Produk::all();

        return view('produk.show')->with([

            "title" => "Tampil Data Produk",

            "data"=>$produk

        ]);

    }

    public function destroy($id):RedirectResponse

    {

        Produk::where('id',$id)->delete();

        return redirect()->route('produk.index')->with('deleted','Data Produk Berhasil Dilihat');

    }

}

* Buka file create.blade.php pada folder views\produk. Tambahkan script berikut:

@extends('layouts.template')

@section('judulh1','Admin - Produk')

@section('konten')

<div class="col-md-6">

    @if ($errors->any())

    <div class="alert alert-danger">

        <strong>Whoops!</strong> There were some problems with your input.<br><br>

        <ul>

            @foreach ($errors->all() as $error)

            <li>{{ $error }}</li>

            @endforeach

        </ul>

    </div>

    @endif

    <div class="card card-success">

        <div class="card-header">

            <h3 class="card-title">Tambah Data Produk</h3>

        </div>

        <!-- /.card-header -->

        <!-- form start -->

        <form action="{{ route('produk.store') }}" method="POST">

            @csrf

            <div class=" card-body">

                <div class="form-group">

                    <label for="nama">Nama Produk</label>

                    <input type="text" class="form-control" id="nama" name="nama" placeholder="">

                </div>

                <div class="form-group">

                    <label for="stok">Stok</label>

                    <input type="number" class="form-control" id="stok" name="stok">

                </div>

                <div class="form-group">

                    <label for="harga">Harga</label>

                    <input type="number" class="form-control" id="harga" name="harga">

                </div>

                <div class="form-group">

                    <label for="deskripsi">Deskripsi</label>

                    <textarea id="deskripsi" name="deskripsi" class=" form-control" rows="4"></textarea>

                </div>

            </div>

            <!-- /.card-body -->

            <div class="card-footer">

                <button type="submit" class="btn btn-success float-right">Simpan</button>

            </div>

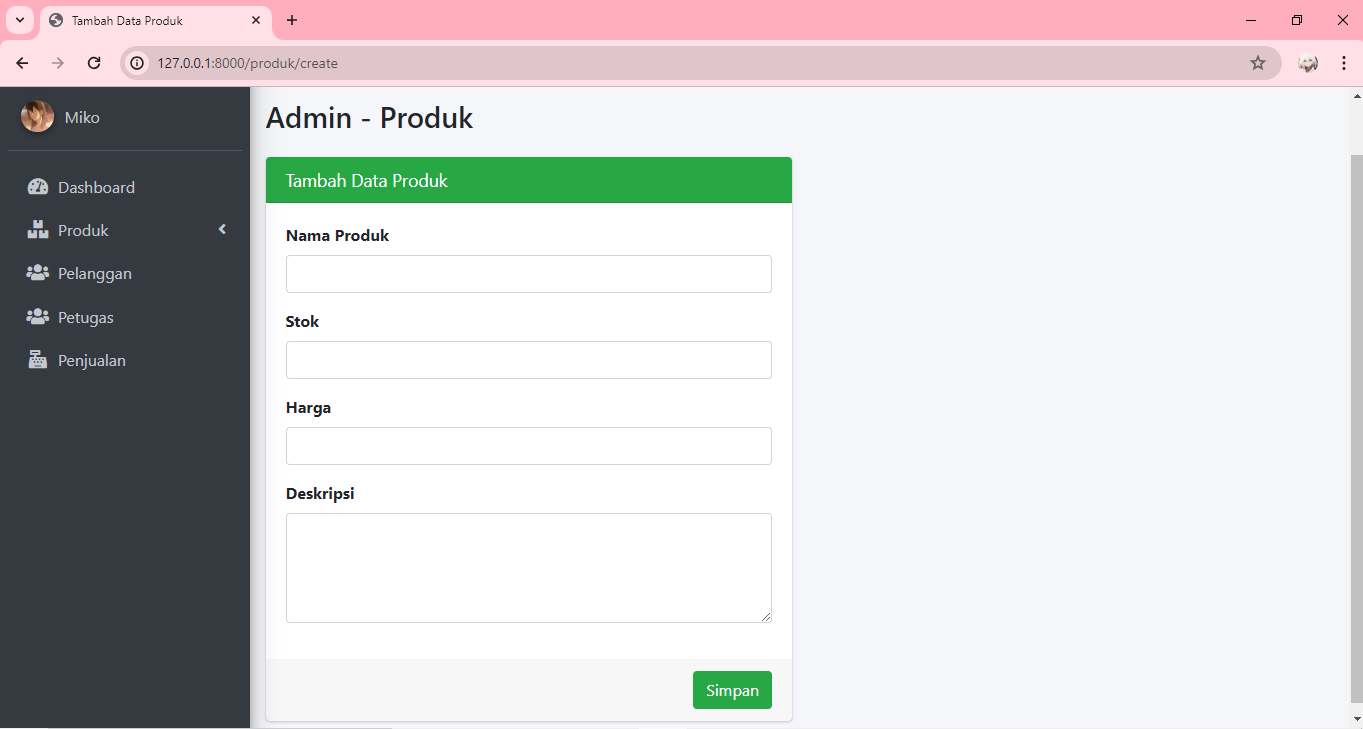
        </form>

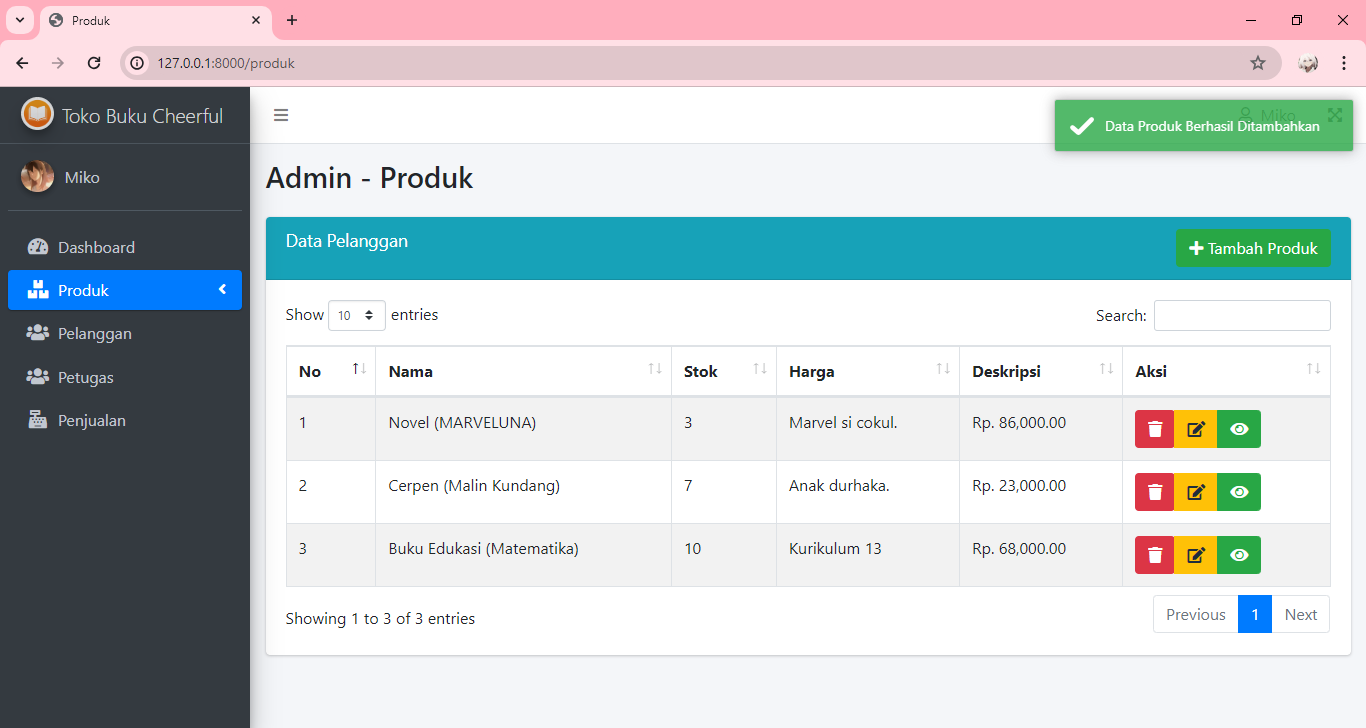
    </div>

</div>

@endsection

* Uji coba penambahan data

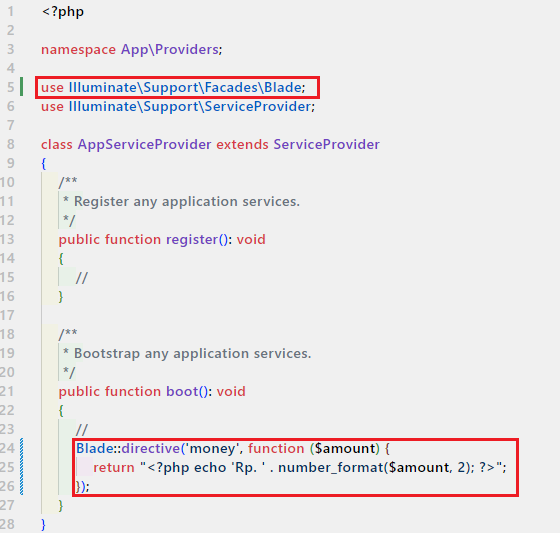


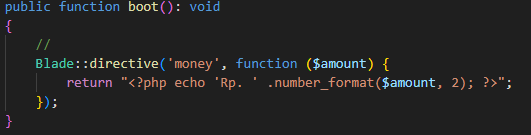


Apakah sudah berhasil ? Selanjutnya perhatikan kolom harga, seharusnya yang sudah dikerjakan tidak muncul format mata uang. Lalu bagaimana caranya agar muncul format uangnya. Untuk melakukan hal tersebut kita harus membuat fungsi baru.

### **Membuat Format Uang**

* Buka file AppServiceProvider.php pada folder app\Providers.
* Tambahkan script berikut:





Lalu tambahkan:

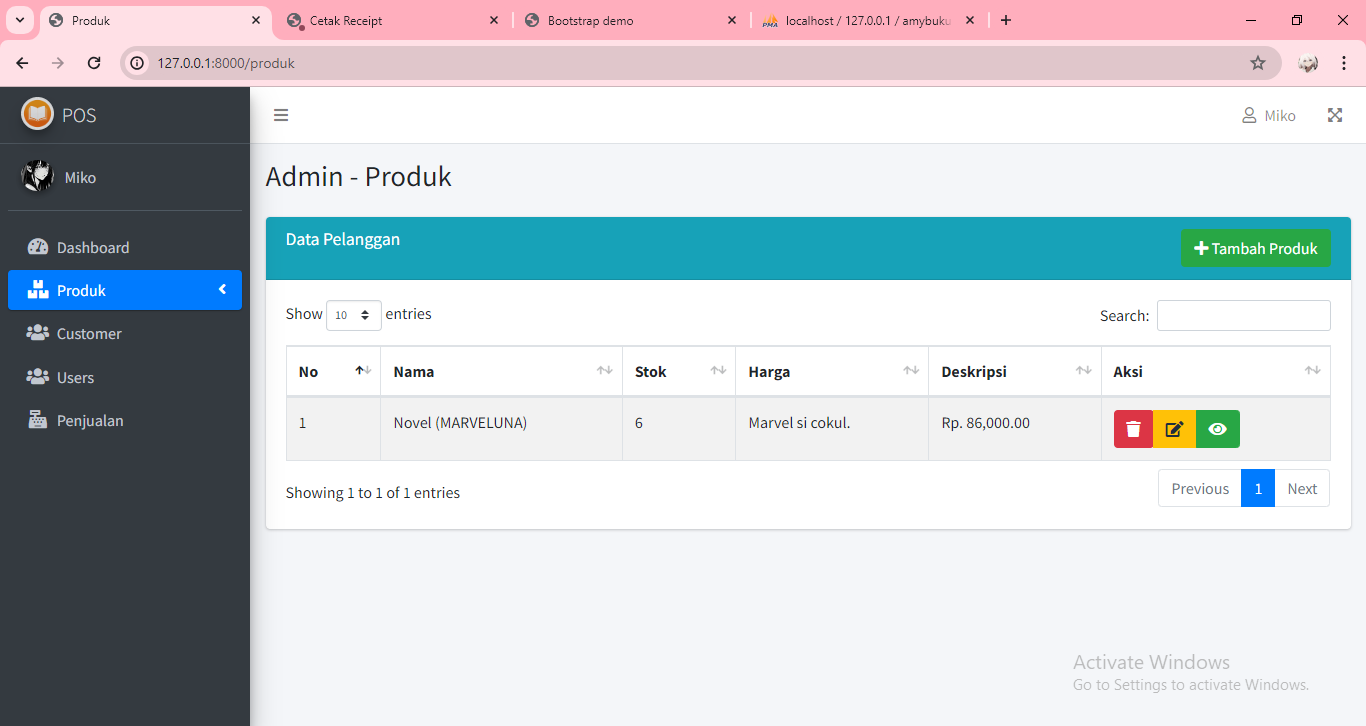
use Illuminate\Support\Facades\Blade;

* Rubah file index.blade.php yang berada di dalam folder views\produks



<td>@money($dt->harga) </td>

* Buka kembali aplikasi pos di browser, silahkan refresh. Apakah sudah muncul format uang?



### **Membuat Method edit() dan Method update()**

* Method edit digunakan untuk menampilkan tampilan edit produk sedangkan method update digunakan untuk merubah record pada tabel produks di database.
* Tambahkan method edit() pada class ProdukController.

public function edit(Produk $produk):View

    {

        return view('produk.edit',compact('produk'))->with([

            "title" => "Ubah Data Produk",

        ]);

    }

* Buka file edit.blade.php lalu tambahkan script berikut:

@extends('layouts.template')

@section('judulh1','Admin - Produk')

@section('konten')

<div class="col-md-6">

    @if ($errors->any())

    <div class="alert alert-danger">

        <strong>Whoops!</strong> There were some problems with your input.<br><br>

        <ul>

            @foreach ($errors->all() as $error)

            <li>{{ $error }}</li>

            @endforeach

        </ul>

    </div>

    @endif

    <div class="card card-warning">

        <div class="card-header">

            <h3 class="card-title">Ubah Data Produk</h3>

        </div>

        <!-- /.card-header -->

        <!-- form start -->

        <form action="{{ route('produk.update',$produk->id) }}" method="POST">

            @csrf

            @method('PUT')

            <div class=" card-body">

                <div class="form-group">

                    <label for="nama">Nama Produk</label>

                    <input type="text" class="form-control" id="nama" name="nama" placeholder=""

                        value="{{$produk->nama}}">

                </div>

                <div class="form-group">

                    <label for="stok">Stok</label>

                    <input type="number" class="form-control" id="stok" name="stok" value="{{$produk->stok}}">

                </div>

                <div class="form-group">

                    <label for="harga">Harga</label>

                    <input type="number" class="form-control" id="harga" name="harga" value="{{$produk->harga}}">

                </div>

                <div class=" form-group">

                    <label for="deskripsi">Deskripsi</label>

                    <textarea id="deskripsi" name="deskripsi" class=" form-control"

                        rows="4">{{ $produk->deskripsi }}</textarea>

                </div>

            </div>

            <!-- /.card-body -->

            <div class="card-footer">

                <button type="submit" class="btn btn-warning float-right">Ubah</button>

            </div>

        </form>

    </div>

</div>

@endsection

* Buka kembali ProdukController lalu tambahkan script berikut pada class ProdukController.

public function update(Produk $produk, Request $request):RedirectResponse

    {

        $request->validate([

            "nama"=>"required",

            "deskripsi"=>"nullable",

            "stok"=>"required",

            "harga"=>"required",

        ]);

        $produk->update($request->all());

        return redirect()->route('produk.index')->with('Updated','Data Produk Berhasil Diubah');

    }

* Selanjutnya uji coba proses update, apakah berjalan lancar atau tidak.

### **Membuat Method show()**

* Tambahkan script berikut pada class ProdukController.

public function show():View

    {

        $produk=Produk::all();

        return view('produk.show')->with([

            "title" => "Tampil Data Produk",

            "data"=>$produk

        ]);

    }

* Buka file show.blade.php pada folder views\produk, lalu tambahkan script berikut:

@extends('layouts.template')

@section('judulh1','Admin - Produk')

@section('konten')

<div class="col-md-6">

    @if ($errors->any())

    <div class="alert alert-danger">

        <strong>Whoops!</strong> There were some problems with your input.<br><br>

        <ul>

            @foreach ($errors->all() as $error)

            <li>{{ $error }}</li>

            @endforeach

        </ul>

    </div>

    @endif

    <div class="card card-primary">

        <div class="card-header">

            <h3 class="card-title">Data Produk</h3>

        </div>

        <!-- /.card-header -->

        <div class=" card-body">

            <table>

                <tr>

                    <th>Nama Produk</th>

                    <td>:</td>

                    <td>{{ $data[0]->nama }}</td>

                </tr>

                <tr>

                    <th>Stok</th>

                    <td>:</td>

                    <td>{{ $data[0]->stok }}</td>

                </tr>

                <tr>

                    <th>Harga</th>

                    <td>:</td>

                    <td>@money($data[0]->harga)</td>

                </tr>

                <tr>

                    <th>Deskripsi</th>

                    <td>:</td>

                    <td>{{ $data[0]->deskripsi}}</td>

                </tr>

            </table>

        </div>

        <!-- /.card-body -->

    </div>

</div>

@endsection

* Uji coba method show.

### **Membuat Method destroy()**

* Tambahkan script berikut pada class ProdukController:

public function destroy($id):RedirectResponse

    {

        Produk::where('id',$id)->delete();

        return redirect()->route('produk.index')->with('deleted','Data Produk Berhasil Dilihat');

    }

* Selanjutnya uji coba seluruh method yang telah dibuat.

## Fitur Pelanggan

### **Membuat Controller Pelanggan**

* Buat controller Pelanggan dengan menggunakan php artisan

php artisan make:controller PelangganController

* Tambahkan script berikut pada **PelangganController.php**

<?php

namespace App\Http\Controllers;

use App\Models\Pelanggan;

use Illuminate\Http\RedirectResponse as HttpRedirectResponse;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Redirect;

use Illuminate\View\View;

use Iluminate\Http\RedirectResponse;

use Symfony\Component\HttpFoundation\RedirectResponse as HttpFoundationRedirectResponse;

use function PHPUnit\Framework\returnValueMap;

class PelangganController extends Controller

{

    //

    public function index()

    {

        return view('pelanggan.tabel', [

        "title" => "Pelanggan",

        "data" => Pelanggan::all()

        ]);

    }

    public function create():View

    {

        return view('pelanggan.tambah')->with(["title" => "Tambah Data Pelanggan"]);

    }

    public function store(Request $request): HttpRedirectResponse

    {

            $request->validate([

            "nama"=>"required",

            "telepon"=>"required",

            "alamat"=>"nullable"

        ]);

        Pelanggan::create($request->all());

        return redirect()->route ('pelanggan.index')->with('success','Data Pelanggan Berhasil Ditambahkan');

    }

    public function edit(Pelanggan $pelanggan):View

    {

        return view('pelanggan.edit',compact('pelanggan'))->with(["title" => " Ubah Data Pelanggan"]);

    }

        public function update(Request $request, Pelanggan $pelanggan): HttpRedirectResponse

        {

                $request->validate([

                "nama"=>"required",

                "telepon"=>"required",

                "alamat"=>"nullable"

            ]);

            $pelanggan->update($request->all());

            return redirect()->route ('pelanggan.index')->with('Updated','Data Pelanggan Berhasil Diubah');

        }

        public function show(Pelanggan $pelanggan):View

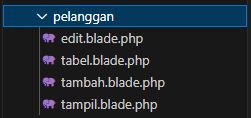
        {

            return view('pelanggan.tampil',compact('pelanggan'))->with(["title" => "Data Pelanggan"]);

        }

}

* Buat folder baru didalam views dengan nama pelanggan
* Buat file baru di dalam folder pelanggan dengan nama **index.blade.php**



* Buka file **views/layouts/template.blade.php**, lalu tambahkan script berikut pada menu pelanggan.

<li class="nav-item">

                            <a href="{{ route('pelanggan.index')}}" class="nav-link {{ ($title==='Pelanggan')?'active':''}}">

                                <i class="nav-icon fas fa-users"></i>

                                <p>

                                    Pelanggan

                                </p>

                            </a>

                        </li>

* Lalu tambahkan script berikut pada file **index.blade.php**:

@extends('layouts.template')

    @section('tambahanCSS')

    <!-- DataTables -->

    <link rel="stylesheet" href="plugins/datatables-bs4/css/dataTables.bootstrap4.min.css">

    <link rel="stylesheet" href="plugins/datatables-responsive/css/responsive.bootstrap4.min.css">

    <link rel="stylesheet" href="plugins/datatables-buttons/css/buttons.bootstrap4.min.css">

    <!-- Toastr -->

    <link rel="stylesheet" type="text/css" href="//cdnjs.cloudflare.com/ajax/libs/toastr.js/latest/css/toastr.css">

    @endsection

    @section('judulh1','Admin - Pelanggan')

    @section('konten')

    <div class="col-md-4">

    <div class="card card-success">

        <div class="card-header">

            <h3 class="card-title">Input Customer</h3>

        </div>

        <!-- /.card-header -->

        <!-- form start -->

        <form action="{{ route('pelanggan.store') }}" method="POST">

            @csrf

            <div class=" card-body">

                <div class="form-group">

                    <label for="nama">Nama</label>

                    <input type="text" class="form-control" id="nama" name="nama" placeholder=" Nama Lengkap" required value="{{ old('nama') }}">

                </div>

                <div class="form-group">

                    <label for="alamat">Alamat</label>

                    <input type="text" class="form-control" id="alamat" name="alamat" placeholder=" Alamat" required value="{{ old('alamat') }}">

                </div>

                <div class="form-group">

                    <label for="telepon">No Telepon</label>

                    <input type="text" class="form-control" id="telepon" name="telepon" placeholder=" No telepon" required value="{{ old('telepon') }}">

                </div>

            </div>

            <!-- /.card-body -->

            <div class="card-footer">

                <button type="submit" class="btn btn-success float-right">Simpan</button>

            </div>

        </form>

    </div>

</div>

    <div class="col-md-8">

        <div class="card card-info">

            <div class="card-header">

                <h2 class="card-title">Data Customer</h2>

                <!-- <a type="button" class="btn btn-success float-right" href="{{ route('pelanggan.create') }}">

                    <i class=" fas fa-plus"></i>Tambah Customer

                </a> -->

            </div>

            <!-- /.card-header -->

            <div class="card-body">

                <table id="example1" class="table table-bordered table-striped ">

                    <thead>

                        <tr>

                            <th>No</th>

                            <th>Nama</th>

                            <th>Alamat</th>

                            <th>No Telepon</th>

                            <th>Aksi</th>

                        </tr>

                    </thead>

                    <tbody>

                        @foreach($data as $dt)

                        <tr>

                            <td>{{ $loop->iteration }}</td>

                            <td>{{ $dt->nama }}</td>

                            <td>{{ $dt->alamat }}</td>

                            <td>{{ $dt->telepon }}</td>

                            <td>

                                <div class="btn-group">

                                    <a type="button" class="btn btn-warning" href="{{ route('pelanggan.edit',$dt->id) }}">

                                        <i class=" fas fa-edit"></i>

                                    </a>

                                    <a type="button" class="btn btn-success" href="{{ route('pelanggan.show',$dt->id) }}">

                                        <i class=" fas fa-eye"></i>

                                    </a>

                                </div>

                            </td>

                        </tr>

                        @endforeach

                    </tbody>

                </table>

            </div>

        </div>

    </div>

    @endsection

    @section('tambahanJS')

    <!-- DataTables  & Plugins -->

    <script src="plugins/datatables/jquery.dataTables.min.js"></script>

    <script src="plugins/datatables-bs4/js/dataTables.bootstrap4.min.js"></script>

    <script src="plugins/datatables-responsive/js/dataTables.responsive.min.js"></script>

    <script src="plugins/datatables-responsive/js/responsive.bootstrap4.min.js"></script>

    <script src="plugins/datatables-buttons/js/dataTables.buttons.min.js"></script>

    <script src="plugins/datatables-buttons/js/buttons.bootstrap4.min.js"></script>

    <script src="plugins/jszip/jszip.min.js"></script>

    <script src="plugins/pdfmake/pdfmake.min.js"></script>

    <script src="plugins/pdfmake/vfs\_fonts.js"></script>

    <script src="plugins/datatables-buttons/js/buttons.html5.min.js"></script>

    <script src="plugins/datatables-buttons/js/buttons.print.min.js"></script>

    <script src="plugins/datatables-buttons/js/buttons.colVis.min.js"></script>

    <!-- Toastr -->

    <script src="https://cdnjs.cloudflare.com/ajax/libs/toastr.js/latest/toastr.min.js"></script>

    @endsection

    @section('tambahScript')

    <script>

        $(function() {

            $("#example1").DataTable({

                "responsive": true,

                "lengthChange": true,

                "autoWidth": false,

                "responsive": true,

            }).buttons().container().appendTo('#example1\_wrapper .col-md-6:eq(0)');

        });

        @if($message = Session::get('success'))

        toastr.success("{{ $message}}");

        @endif

        @if($message = Session::get('updated'))

        toastr.warning("{{ $message}}");

        @endif

    </script>

    @endsection

</body>

</html>

### **Membuat Route**

* Tambahkan script berikut pada **route/web.php**

Route::resource('pelanggan',PelangganController::class)->except('destroy')->middleware('auth');

* Lalu jalankan perintah php artisan route:list untuk melihat daftar method dari pelanggan.

php artisan route:list

* Pada **model Pelanggan** tambahkan variabel $fillable

class Pelanggan extends Model

{

    use HasFactory;

    protected $fillable=['nama','telepon','alamat'];

    public function transaksi():HasMany

    {

        return $this->hasMany(Transaksi::class);

    }

}

* Buat file baru pada Views/Pelanggan dengan nama **editpelanggan.blade.php**, lalu tambahkan script berikut:

@extends('layouts.template')

@section('judulh1','Admin - Pelanggan')

@section('konten')

<div class="col-md-6">

    @if ($errors->any())

    <div class="alert alert-danger">

        <strong>Whoops!</strong> There were some problems with your input.<br><br>

        <ul>

            @foreach ($errors->all() as $error)

            <li>{{ $error }}</li>

            @endforeach

        </ul>

    </div>

    @endif

    <div class="card card-warning">

        <div class="card-header">

            <h3 class="card-title">Ubah Data Pelanggan</h3>

        </div>

        <!-- /.card-header -->

        <!-- form start -->

        <form action="{{ route('pelanggan.update',$pelanggan->id) }}" method="POST">

            @csrf

            @method('PUT')

            <div class=" card-body">

                <div class="form-group">

                    <label for="nama">Nama Pelanggan</label>

                    <input type="text" class="form-control" id="nama" name="nama" placeholder="" value="{{ $pelanggan->nama }}">

                </div>

                <div class="form-group">

                    <label for="telepon">No Telepon</label>

                    <input type="text" class="form-control" id="telepon" name="telepon" value="{{ $pelanggan->telepon }}">

                </div>

                <div class="form-group">

                    <label for="alamat">Alamat</label>

                    <textarea id="alamat" name="alamat" class=" form-control"

                        rows="4">{{ $pelanggan->alamat }}</textarea>

                </div>

            </div>

            <!-- /.card-body -->

            <div class="card-footer">

                <button type="submit" class="btn btn-warning float-right">Simpan</button>

            </div>

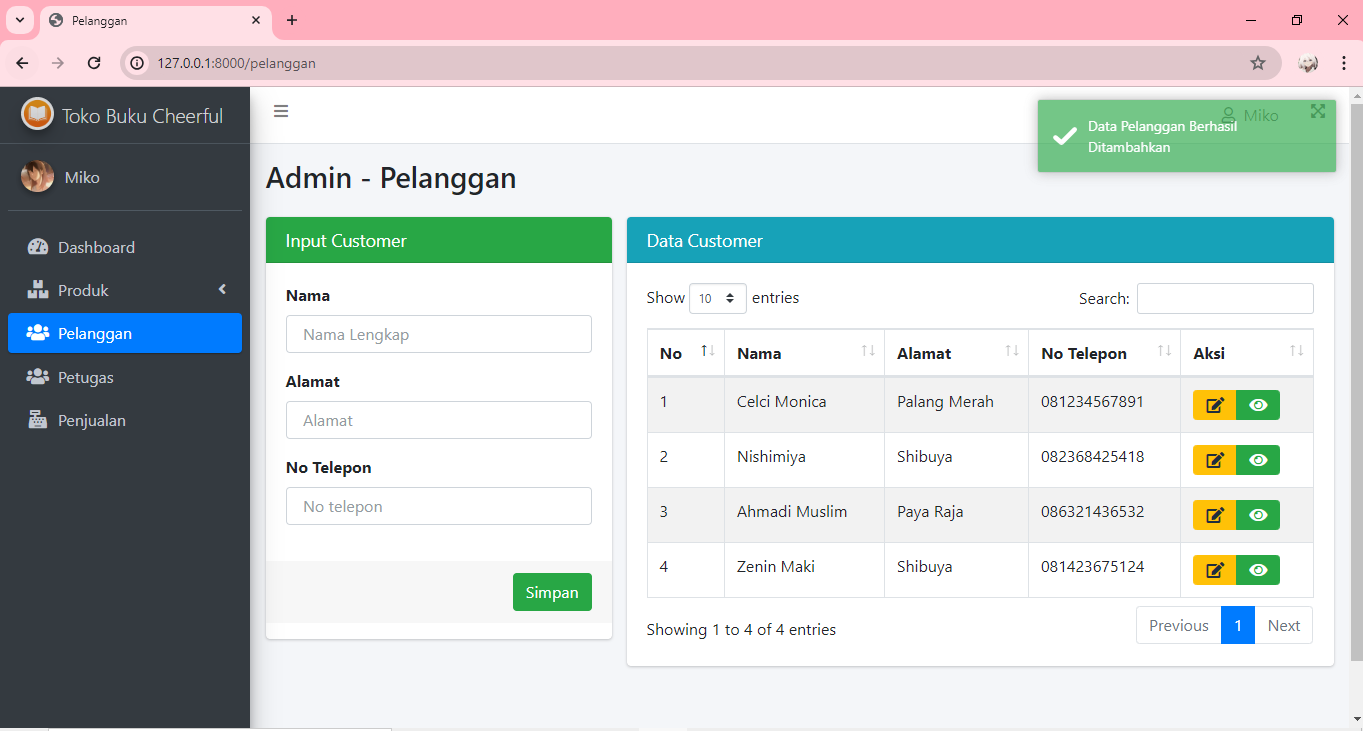
        </form>

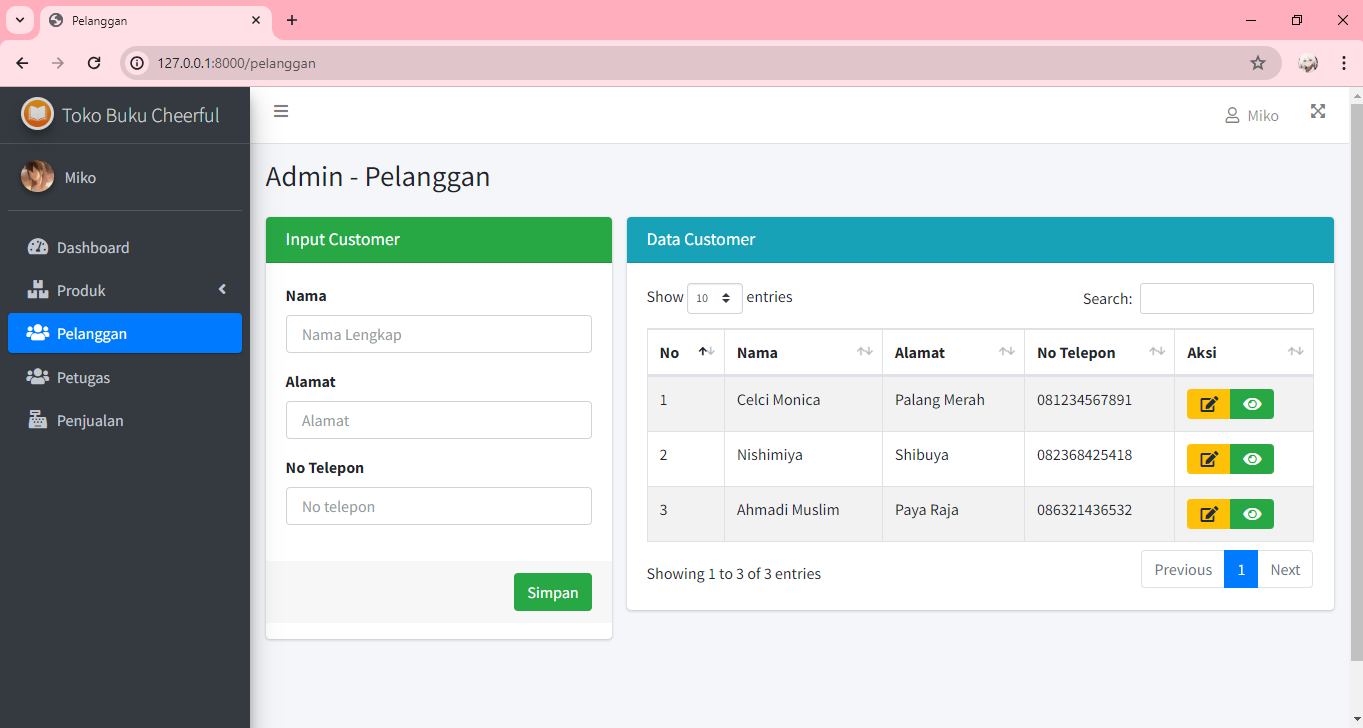
    </div>

</div>

@endsection

* Uji coba keseluruhan method yang telah dibuat
* Hasil: tampilan





* Penjelasan data **NULL**

public function update(Pelanggan $pelanggan, Request $request):RedirectResponse

    {

        $request->validate([

            "nama"=>"required",

        ]);

        if (empty($request['hp'])) {

            $request['hp']='null';

        if (empty($request['alamat']))

            $request['alamat']='null';

        }

Data yang di request akan menjadi Null jika form input tidak diisi atau dikosongkan. Hal ini bertujuan untuk mengantisipasi jika ada pelanggan yang tidak berkenan untuk memberikan data HP dan Alamat karena bersifat privasi

## Fitur User

## Manajemen User

### **Membuat Controller User**

* Membuat Cotroller User dengan prompt artisan pada terminal

php artisan make:controller UserController

* Tambahkan Method index() pada UserController

public function index()

    {

        return view('user.index',["title"=>"Data Pengguna","data"=>User::all() ]);

    }

* mengimport class model User pada UserController

use App\Models\User;

### **Membuat Route**

* Tambahkan script berikut pada routes/web.php

Route::resource('pengguna',UserController::class)->except('destroy','create','show','update','edit')->middleware('auth');

* Dan Tambahkan script berikut dibagian atas

use App\Http\Controllers\UserController;

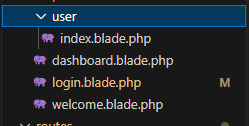
* Lalu cek daftar route menggunakan prompt artisan

php artisan route:list

### 

### **Membuat Tampilan Manajemen User**

* Buat folder baru di dalam views dengan nama user, lalu Buat file baru di dalam folder user dengan nama index.blade.php, lalu tambahkan script berikut pada file index.blade.php



@extends('layouts.template')

@section('tambahanCSS')

<!-- DataTables -->

<link rel="stylesheet" href="plugins/datatables-bs4/css/dataTables.bootstrap4.min.css">

<link rel="stylesheet" href="plugins/datatables-responsive/css/responsive.bootstrap4.min.css">

<link rel="stylesheet" href="plugins/datatables-buttons/css/buttons.bootstrap4.min.css">

<!-- Toastr -->

<link rel="stylesheet" type="text/css" href="//cdnjs.cloudflare.com/ajax/libs/toastr.js/latest/css/toastr.css">

@endsection

@section('judulh1','Admin - Users')

@section('judulh3','Users')

@section('konten')

<div class="col-md-4">

    <div class="card card-success">

        <div class="card-header">

            <h3 class="card-title">Input User</h3>

        </div>

        <!-- /.card-header -->

        <!-- form start -->

        <form action="{{ route('pengguna.store') }}" method="POST">

            @csrf

            <div class=" card-body">

                <div class="form-group">

                    <label for="name">Nama</label>

                    <input type="text" class="form-control" id="name" name="name" placeholder=" Nama Lengkap" required value="{{ old('name') }}">

                </div>

                <div class="form-group">

                    <label for="email">Email</label>

                    <input type="email" class="form-control @error('email') is-invalid @enderror" id="email" name="email" placeholder="JohnDoe@example.com" required value="{{ old('email') }}">

                    @error('email')

                    <div class="invalid-feedback">

                        {{ $message }}

                    </div>

                    @enderror

                </div>

                <div class="form-group">

                    <label for="password">Password</label>

                    <input type="password" class="form-control" id="password" name="password" required>

                </div>

            </div>

            <!-- /.card-body -->

            <div class="card-footer">

                <button type="submit" class="btn btn-success float-right">Simpan</button>

            </div>

        </form>

    </div>

</div>

<div class="col-md-8">

    <div class="card card-info">

        <div class="card-header">

            <h3 class="card-title">Data Category</h3>

        </div>

        <!-- /.card-header -->

        <div class="card-body">

            <table id="example1" class="table table-bordered table-striped ">

                <thead>

                    <tr>

                        <th>No</th>

                        <th>Nama </th>

                        <th>Email</th>

                    </tr>

                </thead>

                <tbody>

                    @foreach($data as $dt)

                    <tr>

                        <td>{{ $loop->iteration }}</td>

                        <td>{{ $dt->name }}</td>

                        <td>{{ $dt->email }}</td>

                    </tr>

                    @endforeach

                </tbody>

            </table>

        </div>

    </div>

</div>

@endsection

@section('tambahanJS')

<!-- DataTables  & Plugins -->

<script src="plugins/datatables/jquery.dataTables.min.js"></script>

<script src="plugins/datatables-bs4/js/dataTables.bootstrap4.min.js"></script>

<script src="plugins/datatables-responsive/js/dataTables.responsive.min.js"></script>

<script src="plugins/datatables-responsive/js/responsive.bootstrap4.min.js"></script>

<script src="plugins/datatables-buttons/js/dataTables.buttons.min.js"></script>

<script src="plugins/datatables-buttons/js/buttons.bootstrap4.min.js"></script>

<script src="plugins/jszip/jszip.min.js"></script>

<script src="plugins/pdfmake/pdfmake.min.js"></script>

<script src="plugins/pdfmake/vfs\_fonts.js"></script>

<script src="plugins/datatables-buttons/js/buttons.html5.min.js"></script>

<script src="plugins/datatables-buttons/js/buttons.print.min.js"></script>

<script src="plugins/datatables-buttons/js/buttons.colVis.min.js"></script>

<!-- Toastr -->

<script src="https://cdnjs.cloudflare.com/ajax/libs/toastr.js/latest/toastr.min.js"></script>

@endsection

@section('tambahScript')

<script>

    $(function() {

        $("#example1").DataTable({

            "responsive": true,

            "lengthChange": true,

            "autoWidth": false,

            "responsive": true,

        }).buttons().container().appendTo('#example1\_wrapper .col-md-6:eq(0)');

    });

    @if($message = Session::get('success'))

    toastr.success("{{ $message}}");

    @elseif($message = Session::get('updated'))

    toastr.warning("{{ $message}}");

    @endif

</script>

@endsection

* Perbaiki template.blade.php. Dengan menambahkan route pada sidebar user.

<li class="nav-item">

                            <a href="{{ route('pengguna.index')}}" class="nav-link {{ ($title==='Users')?'active':''}}">

                                <i class="nav-icon fas fa-users"></i>

                                <p>

                                    Petugas

                                </p>

                            </a>

                        </li>

* Pada manajemen user kita tidak menambahkan menu edit, karena nantinya kita akan membuat menu edit data user tidak di admin, namun di private menu masing-masing user. Sedangkan untuk menu hapus tidak kita buat juga karena User berkaitan dengan tabel order, sehingga jika data user dapat dihapus maka akan terjadi kecacatan data nantinya.

### **Membuat method store()**

* Untuk menyimpan data ke database, tambahkan method store pada UserController

<?php

namespace App\Http\Controllers;

use App\Models\User;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Hash;

use Illuminate\Http\RedirectResponse;

class UserController extends Controller

{

    //

    public function index()

    {

        return view('user.index',["title"=>"Data Pengguna","data"=>User::all() ]);

    }

    public function store(Request $request):RedirectResponse

    {

        $request->validate([

            "name"=>"required",

            "email"=>"required",

            "password"=>"required"

        ]);

        $password=Hash::make($request->password);

        $request->merge(["password"=>$password]);

        User::create($request->all());

        return redirect()->route('pengguna.index')->with('success','Data Berhasil Ditambah');

    }

}

### **Mengatur model User**

* Untuk menyimpan data ke database, tambahkan script berikut pada model User

<?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;

class User extends Authenticatable

{

    use HasApiTokens, HasFactory, Notifiable;

    /\*\*

     \* The attributes that are mass assignable.

     \*

     \* @var array<int, string>

     \*/

    protected $fillable = ['name','email','password'];

    /\*\*

     \* The attributes that should be hidden for serialization.

     \*

     \* @var array<int, string>

     \*/

    protected $hidden = [

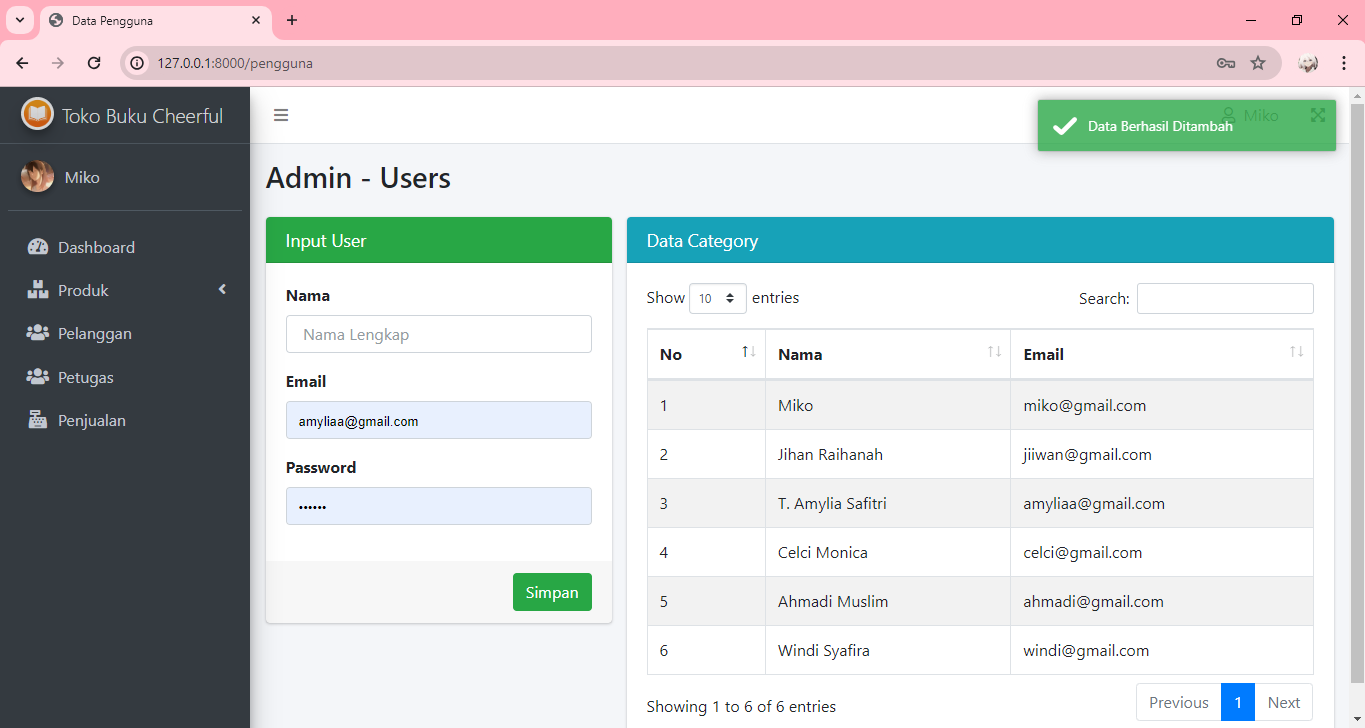
        'password',

        'remember\_token',

    ];

}

* Lalu uji coba menyimpan data user



## Autentikasi User

### **Menyiapkan Form Login**

* Buat file baru di dalam folder Views dengan nama login.blade.php.
* Lalu tambahkan script berikut:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>Login Pengguna</title>

    <!-- Google Font: Source Sans Pro -->

    <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback">

    <!-- Font Awesome -->

    <link rel="stylesheet" href="{{ asset('plugins/fontawesome-free/css/all.min.css') }}">

    <!-- icheck bootstrap -->

    <link rel="stylesheet" href="{{ asset('plugins/icheck-bootstrap/icheck-bootstrap.min.css') }}">

    <!-- Theme style -->

    <link rel="stylesheet" href="{{ asset('dist/css/adminlte.min.css') }}">

</head>

<body class="hold-transition login-page" style="background-image: url('bsss.jpg'); background-size: cover";>

    <div class="login-box">

        <!-- /.login-logo -->

        <div class="card card-outline card-primary" style="background:transparent; border: 2px solid rgba(270, 270, 270, .5); backdrop-filter:blur(0px); color:#fff; border-radius: 20px;">

            <div class="card-header text-center">

                <h2 class="h2">TokoBuku<b>Cheerful</b></h2>

            </div>

            <div class="card-body">

                <p class="login-box-msg">Please Login</p>

                @if(session()->has('loginError'))

                <div class="alert alert-danger alert-dismissible">

                    <button type="button" class="close" data-dismiss="alert" aria-hidden="true">×</button>

                    <h5><i class="icon fas fa-ban"></i> Alert!</h5>

                    {{ session('loginError') }}

                </div>

                @endif

                <form action="/login" method="post">

                    @csrf

                    <div class="input-group mb-3">

                        <input type="email" class="form-control @error('email') is-invalid @enderror" placeholder="Email" id="email" name="email" value="{{ old('email') }}" style="background: transparent; border: 2px solid rgba(255, 255, 255, .2)";>

                        <div class="input-group-append">

                            <div class="input-group-text">

                                <span class="fas fa-envelope"></span>

                            </div>

                        </div>

                        @error('email')

                        <div class="invalid-feedback">

                            {{ $message }}

                        </div>

                        @enderror

                    </div>

                    <div class="input-group mb-3">

                        <input type="password" class="form-control" placeholder="Password" id="password" name="password" style="background: transparent; border: 2px solid rgba(255, 255, 255, .2)";>

                        <div class="input-group-append">

                            <div class="input-group-text">

                                <span class="fas fa-lock"></span>

                            </div>

                        </div>

                    </div>

            </div>

            <div class="card-footer">

                <button type="reset" class="btn btn-warning">Cancel</button>

                <button type="submit" class="btn btn-primary float-right">Login</button>

            </div>

            <!-- /.card-body -->

        </div>

        </form>

        <!-- /.card -->

    </div>

    <!-- /.login-box -->

    <!-- jQuery -->

    <script src="{{asset('plugins/jquery/jquery.min.js')}}"></script>

    <!-- Bootstrap 4 -->

    <script src="{{asset('plugins/bootstrap/js/bootstrap.bundle.min.js')}}"></script>

    <!-- AdminLTE App -->

    <script src="{{asset('dist/js/adminlte.min.js')}}"></script>

</body>

</html>

### **Menambahkan Controller Login**

* Buat controller baru dengan nama LoginController

php artisan make:controller LoginController

* Tambahkan method loginView pada class LoginController

public function loginView()

    {

        return view('login');

    }

* Tambahkan Method authenticate pada class LoginController

public function authenticate(Request $request):RedirectResponse

    {

        $credentials=$request->validate([

            'email'=>['required'],

            'password'=>['required'],

        ]);

        if(Auth::attempt($credentials)) {

            $request->session()->regenerate();

            return redirect()->intended('/');

        }

        return back()->with('loginError','Login Failed');

    }

* Jangan lupa untuk mengimport 3 class berikut:

use Illuminate\Http\RedirectResponse;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Auth;

### **Menambahkan Route Login**

* Tambahkan 2 buah route berikut pada routes/web.php

Route::get('login',[LoginController::class,'loginView'])->name('login');

Route tersebut berguna untuk mengarahkan ke method loginView pada controller LoginController ketika mengakses login dengan fungsi get.

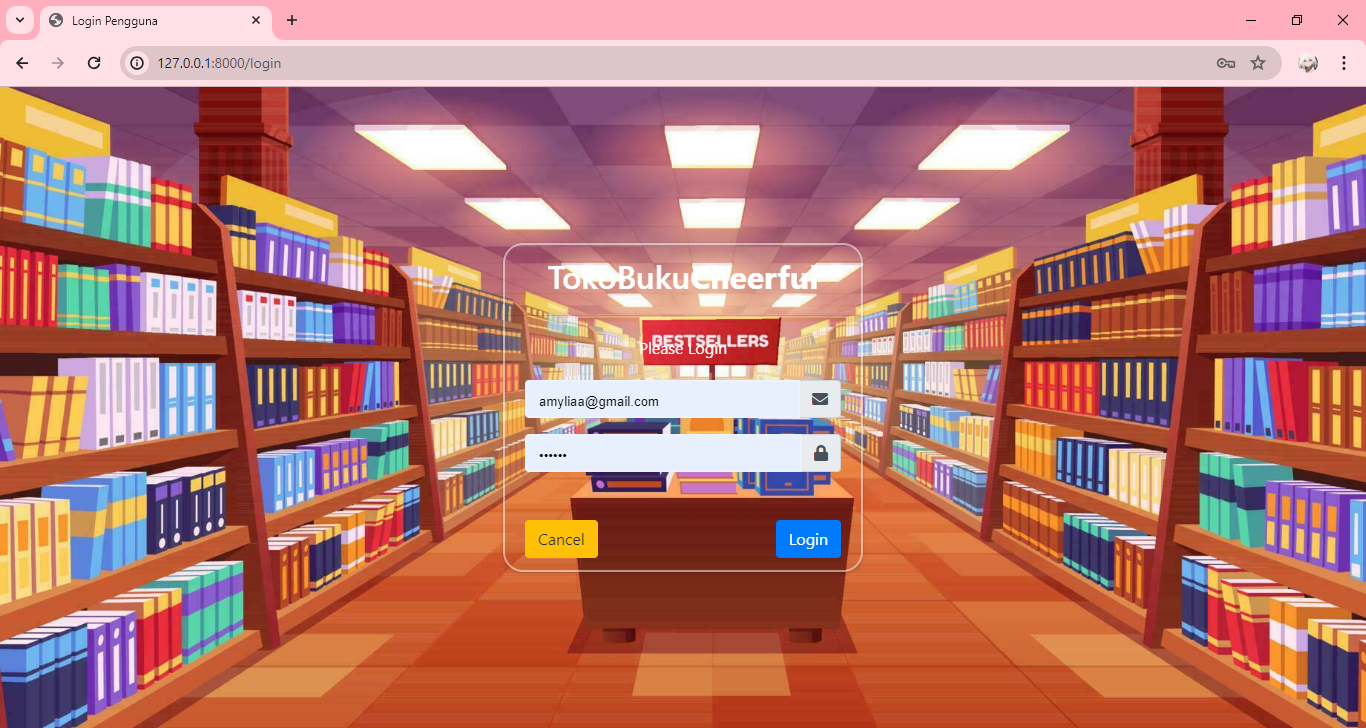
Berikutnya,

Route::post('login',[LoginController::class,'authenticate']);

Route tersebut berguna untuk mengarahkan ke method authenticate pada controller LoginController ketika mengakses login dengan fungsi post.

**POST dan GET keduanya adalah method atau fungsi yang digunakan untuk mengirimkan data ke halaman lain.**

* Silahkan ketikkan 127.0.0.1:8000/login maka seharusnya akan tampil form login seperti dibawah berikut:



* Masukkan email dan password yang sudah di daftarkan pada Manajemen User sebelumnya. Jika berhasilnya seharusnya akan mengarah ke Route ( / ) yang jika mengikuti Modul ini akan mengarah pada tampilan [view (dahsboard)](#1g78ehf8jgam).

### **Memperbaiki Route**

Saat ini jika user yang belum login mencoba mengakses route langsung dari browser address tetap akan bisa diakses, hal ini terjadi karena kita belum melindungi akses ke route kita yang lain menggunakan middleware. Middleware seperti namanya yaitu perangkat lunak yang berfungsi sebagai penengah. Dalam konteks laravel khususnya bagian Autentikasi, sebagai penengah antara request yang dilakukan oleh user dan Route yang tersedia. Jadi seandainya ada User atau user ilegal yang mencoba mengakses Route tertentu, maka aksesnya akan diblokir oleh middleware. Cara menggunakan middleware cukup gampang, secara default Laravel sudah menyediakan middleware autentikasi.

* Buka kembali routes/web.php
* Kemudian tambahkan ->middleware('auth'); pada setiap Route yang telah kita buat, kecuali Route yang menangani fungsi login. Berikut perubahan keseluruhan routes/web/php:

Route::get('/', function () {

    return view('dashboard', [

        "title" => "Dashboard"

    ]);

})->middleware('auth');

Route::resource('pelanggan',PelangganController::class)->except('destroy')->middleware('auth');

Route::resource('produk',ProdukController::class)->middleware('auth');

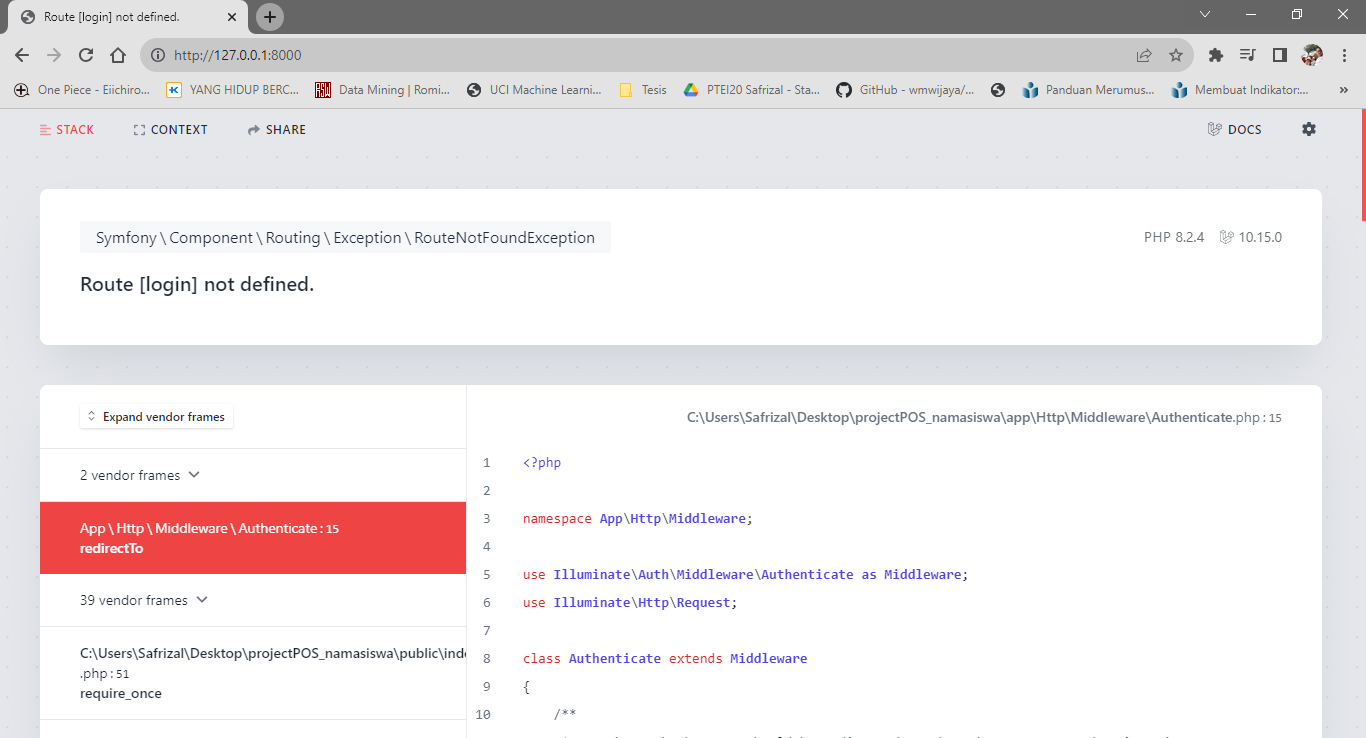
Route::resource('pengguna',UserController::class)->except('destroy','create','show','update','edit')->middleware('auth');

Route::get('login',[LoginController::class,'loginView'])->name('login');

Route::post('login',[LoginController::class,'authenticate']);

Route::post('/logout', [LoginController::class, 'logout'])->name('auth.logout');

* Saat ini ketika kita mengakses controller tertentu akan muncul pesan error berikut.



* Hal ini terjadi karena kita belum memberikan nama pada route login kita. Jadi perbaiki kembali routes/web.php. Perbaiki route login yang menuju ke tampilan login menjadi seperti berikut:

Route::get('login',[LoginController::class,'loginView'])->name('login');

* Sekarang seharusnya ketika kita coba mengakses routes tertentu yang kita batasi dengan middleware tanpa login maka akan diarahkan ke halaman login.

### 

### **Menambahkan Method Logout**

Saat ini kita sudah memiliki form login, dan fungsi login yang berjalan. Yang belum kita miliki adalah fungsi logout. Sebelum kita membuat fungsi logout, kita perbaiki dulu tampilan Aplikasi POS yang sedang kita kerjakan. Perbaikan ini bertujuan untuk menambahkan keterangan User yang sedang login dan menambahkan tombol logout.

* Buka template.blade.php Cari bagian Sidebar, lalu perbaiki script sesuai dengan lokasi yang ditandai pada gambar di bawah ini.

<!-- Sidebar -->

            <div class="sidebar">

                <!-- Sidebar user panel (optional) -->

                <div class="user-panel mt-3 pb-3 mb-3 d-flex">

                    <div class="image">

                        <img src="{{ asset('dist/img/ssakura.jpg')}}" class="img-circle elevation-2" alt="User Image">

                    </div>

                    <div class="info">

                        <a href="#" class="d-block">{{ Auth::user()->name}}</a>

                    </div>

                </div>

**Penjelasan:** {{ Auth::user()->name }} berguna untuk menampilkan nama user yang sedang login

* Kemudian cari bagian Right navbar links, tambahkan script <li> berikut di dalam tag <ul class=”navbar-nav ml-auto> </ul>

<!-- Right navbar links -->

            <ul class="navbar-nav ml-auto">

                <li class="nav-item dropdown">

                    <a class="nav-link" data-toggle="dropdown" href="#">

                        <i class="far fa-user mr-2"></i>{{ Auth::user()->name }}

                        <span class="badge badge-warning navbar-badge"></span>

                    </a>

                    <div class="dropdown-menu dropdown-menu-lg dropdown-menu-right">

                        <span class="dropdown-item dropdown-header">User Menu</span>

                        <div class="dropdown-divider"></div>

                        <a href="#" class="dropdown-item">

                            <i class="fas fa-user mr-2"></i> {{ Auth::user()->name }}

                            <span class="float-right text-muted text-sm"></span>

                        </a>

                        <div class="dropdown-divider"></div>

                        <form action="logout" method="POST">

                            @csrf

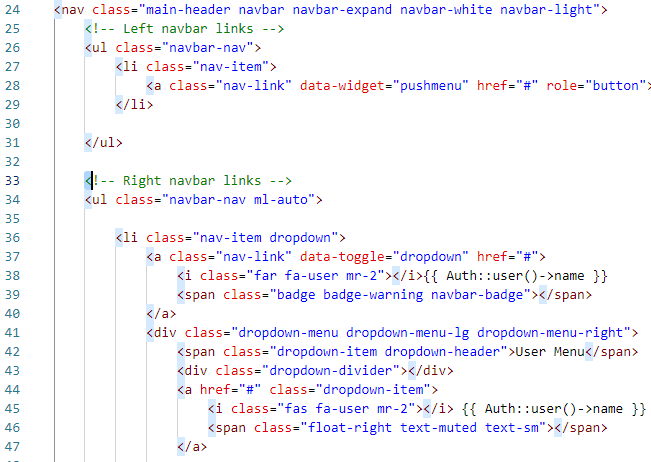
                            <button type="submit" class="dropdown-item">

                                <i class="fas fa-sign-out-alt mr-2"></i>Logout

                            </button>

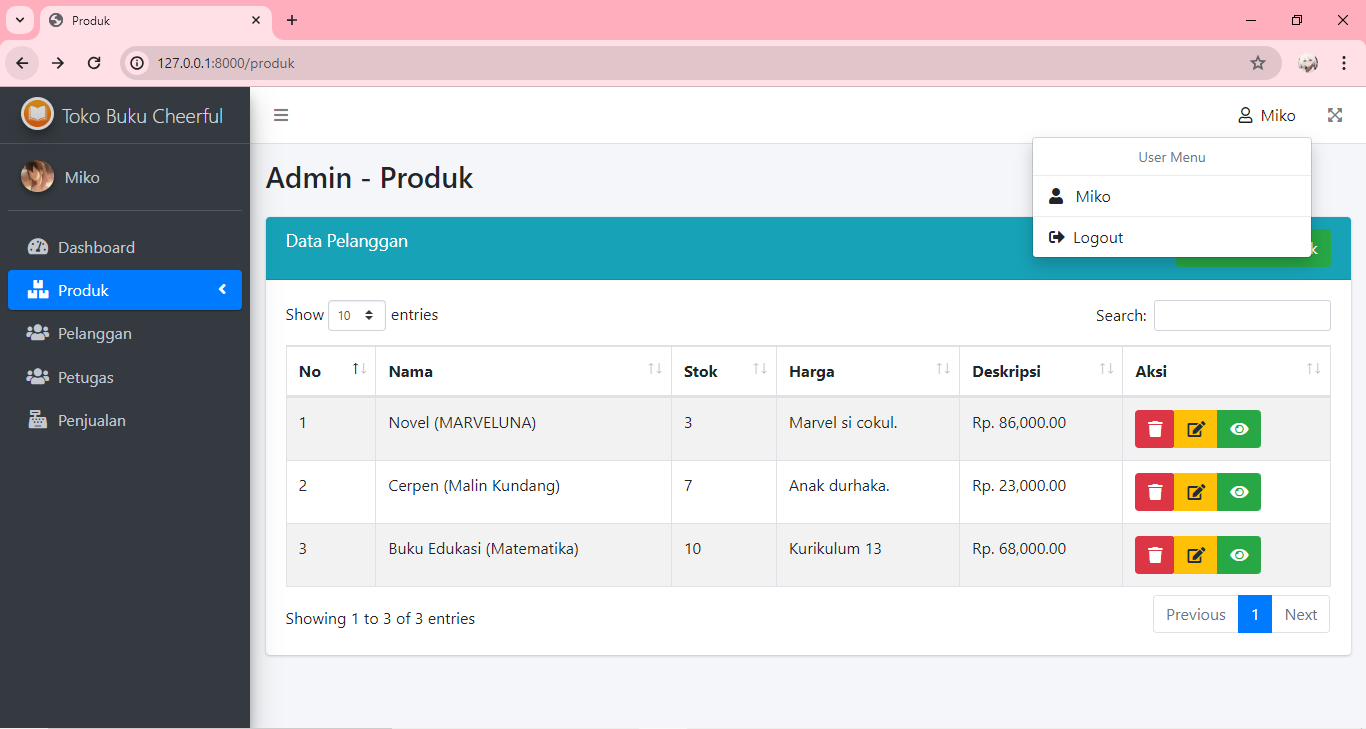
                        </form>

                    </div>



**Penjelasan:** jika diperhatikan kembali tombol logout diletakkan didalam sebuah form dengan method POST, tujuannya adalah untuk mencegah orang lain mengeluarkan kita dari aplikasi (logout).

* Jika penempatan script benar maka pada halaman utama akan muncul menu baru seperti berikut:



* Lalu buka kembali controller LoginController lalu tambahkan method logout di dalam class LoginController.

public function logout(Request $request): RedirectResponse

    {

        Auth::logout();

    $request->session()->invalidate();

    $request->session()->regenerateToken();

    return redirect('/login');

    }

* Silahkan uji coba tombol logout, lalu coba akses Routes tanpa melakukan login terlebih dahulu.

## Fitur Transaksi / Penjualan

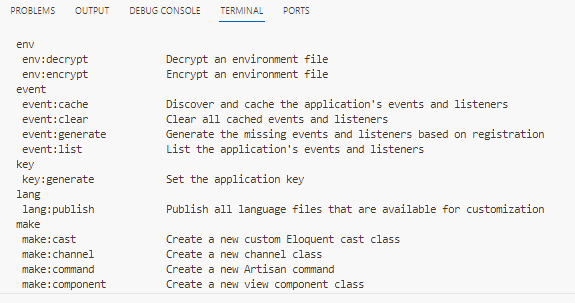
Untuk bagian penjualan, bagian ini agak spesial karena kita harus membuat halaman penjualan berubah tanpa me-refresh halaman, untuk melakukan hal tersebut kita akan gunakan **Livewire** karena livewire adalah framework fullstack untuk laravel yang dapat digunakan untuk membuat antar muka yang dinamis dengan cara yang sederhana tanpa meninggalkan kenyamanan dari laravel.

### **Instalasi Livewire**

1. Buat sebuah file baru didalam layouts dan beri nama **masterlw.blade.php**

php artisan

berikut tampilan jika livewire belum terinstal

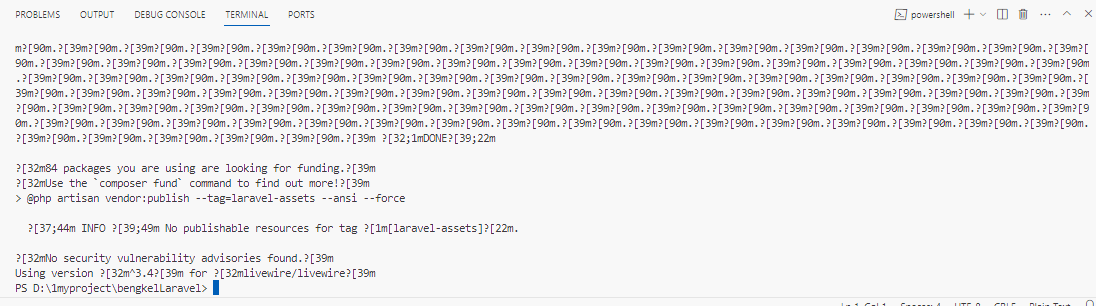


1. Buat 2 buah komponen baru dengan nama penjualan dan transaksis

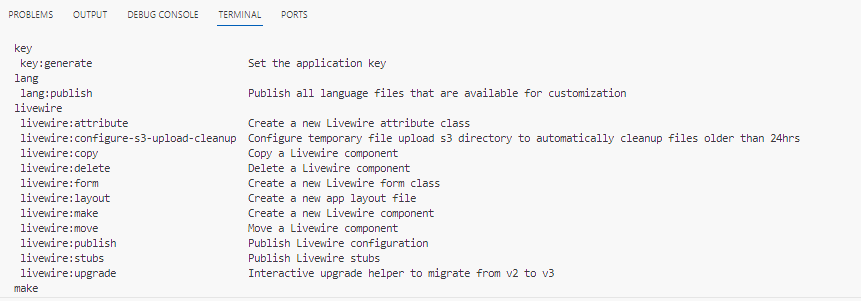


composer require livewire/livewire

berikut tampilan jika instalasi telah selesai



Jika di cek melalui php artisan, maka library livewire telihat sudah terinstal



1. Buat sebuah file baru didalam layouts dan beri nama **masterlw.blade.php**

### **Membuat Komponen**

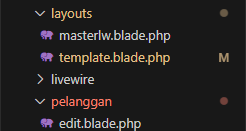
1. Buat 2 buah komponen baru dengan nama penjualan dan transaksis

php artisan livewire:make penjualan



php artisan livewire:make transaksis

1. Buat sebuah file baru didalam layouts dan beri nama **masterlw.blade.php**



1. Pada file **masterlw.blade.php**, tambahkan script berikut

<!doctype html>

<html lang="en">

<head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>Bootstrap demo</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2Kadq2F9CUG65" crossorigin="anonymous">

    <!-- Font Awesome Icons -->

    <link rel="stylesheet" href="{{ asset('plugins/fontawesome-free/css/all.min.css') }}">

    @livewireStyles

</head>

<body>

    <nav class="navbar navbar-expand-lg  navbar-dark bg-dark border-bottom border-body" data-bs-theme="dark">

        <div class="container-fluid">

            <a class="navbar-brand" href="#">Navbar</a>

            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle navigation">

                <span class="navbar-toggler-icon"></span>

            </button>

            <div class="collapse navbar-collapse" id="navbarNavAltMarkup">

                <div class="navbar-nav ms-auto">

                    <a class="nav-link active" aria-current="page" href="#">{{ Auth::user()->name }}</a>

                    <form action="logout" method="POST" class="d-flex">

                        @csrf

                        <button type="submit" class="btn btn-outline-secondary"><i class="fas fa-sign-out-alt mr-2"></i>Logout</button>

                    </form>

                </div>

            </div>

        </div>

    </nav>

    <div class="container-fluid py-3">

        @yield('konten')

    </div>

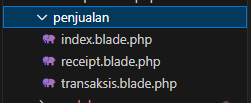
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-kenU1KFdBIe4zVF0s0G1M5b4hcpxyD9F7jL+jjXkk+Q2h455rYXK/7HAuoJl+0I4" crossorigin="anonymous"></script>

    @livewireScripts

</body>

</html>

1. Buat 1 buah folder baru didalam folder views dengan nama folder penjualan
2. Di dalam folder penjualan buat 2 buah file baru dengan nama **index.blade.php** dan **transaksis.blade.php**



1. Di dalam file **index.blade.php** tambahkan script berikut:

@extends('layouts.masterlw')

@section('konten')

  @livewire('penjualan')

@endsection

1. Di dalam file **transaksis.blade.php** tambahkan script berikut:

@extends('layouts.masterlw')

@section('konten')

  @livewire('transaksis')

@endsection

1. Sekarang pindah ke folder **views/livewire**, buka file **penjualan.blade.php** tambahkan script berikut.

<div>

<div class="row">

<div class="d-flex aligns-items-center justify-content-center">

<div class="col-6">

    <div class="card ">

        <div class="card-header">

            <h5 class="card-title">Invoice</h5>

        </div>

        <!-- /.card-header -->

        <!-- form start -->

        @if ($errors->any())

    <div class="alert alert-danger">

        <strong>Whoops!</strong> There were some problems with your input.<br><br>

        <ul>

            @foreach ($errors->all() as $error)

            <li>{{ $error }}</li>

            @endforeach

        </ul>

    </div>

    @endif

        <form wire:submit.prevent="store" method="POST">

            @csrf

            <div class=" card-body">

                <div class="form-group">

                    <label for="name">Pelanggan</label>

                    <select class="form-control" wire:model="pelanggan\_id">

                        <option hidden>--Pilih Pelanggan--</option>

                        @foreach($data as $dt )

                        <option value="{{ $dt->id }}">{{ $dt->nama }}</option>

                        @endforeach

                    </select>

                </div>

            </div>

            <!-- /.card-body -->

            <div class="card-footer text-end">

            <button type="submit" class="btn btn-success btn-sm">Submit</button>

            </div>

        </form>

    </div>

</div>

</div>

</div>

</div>

1. Buka file transaksis.blade.php pada folder views/livewire, lalu tambahkan script berikut:

<div>

    <div class="row">

        @if ($errors->any())

        <div class="alert alert-danger">

            <strong>Whoops!</strong> There were some problems with your input.<br><br>

            <ul>

                @foreach ($errors->all() as $error)

                <li>{{ $error }}</li>

                @endforeach

            </ul>

        </div>

        @endif

        <div class="d-flex aligns-items-center justify-content-center">

            <div class="col-lg-6">

                <div class="card">

                    <div class="card-header">Invoice</div>

                    <div class="card-body">

                        <table class="table table-sm">

                            <tr>

                                <th>Invoice</th>

                                <th>:</th>

                                <td>{{ $data->invoice }}</td>

                            </tr>

                            <tr>

                                <th>Pelanggan</th>

                                <th>:</th>

                                <td>{{ $data->Pelanggan->nama}}</td>

                            </tr>

                            <tr>

                                <th>Tanggal</th>

                                <th>:</th>

                                <td>{{ $data->created\_at->format('d M Y')}}</td>

                            </tr>

                            <tr>

                                <th>Total</th>

                                <th>:</th>

                                <td>@money($total)</td>

                            </tr>

                        </table>

                    </div>

                </div>

            </div>

            <div class="col-lg-6 ps-3">

                <div class="card">

                    <div class="card-header">Produk </div>

                    <div class="card-body">

                        <form wire:submit.prevent="store" method="POST">

                            @csrf

                            <div class=" card-body">

                                <input type="hidden" name="transaksi\_id" wire:model="transaksi\_id" value="{{ $data->id }}">

                                <div class="form-group">

                                    <label for="produk\_id">Produk</label>

                                    <select class="form-control" wire:model="produk\_id" name="produk\_id">

                                        <option hidden>--Pilih Produk--</option>

                                        @foreach($dataProduk as $dt )

                                        <option value="{{ $dt->id }}">{{ $dt->nama }}</option>

                                        @endforeach

                                    </select>

                                </div>

                                <div class="form-group">

                                    <label for="qty">Jumlah</label>

                                    <input type="number" class="form-control" wire:model="qty" name="qty">

                                </div>

                            </div>

                            <!-- /.card-body -->

                            <div class="card-footer text-end">

                                <button type="submit" class="btn btn-success btn-sm"><i class="fas fa-cart-plus"></i> Tambahkan</button>

                            </div>

                        </form>

                    </div>

                </div>

            </div>

        </div>

        <div class="d-flex aligns-items-center justify-content-center pt-3">

            <div class="col-lg-12">

                <div class="card">

                    <div class="card-header">Daftar Belanja</div>

                    <div class="card-body">

                        <table class="table table-sm">

                            <thead>

                                <tr>

                                    <th>No</th>

                                    <th>Nama Produk</th>

                                    <th>Harga</th>

                                    <th>Jumlah</th>

                                    <th>Sub Total</th>

                                    <th>Hapus</th>

                                </tr>

                            </thead>

                            <tbody>

                                @foreach($dataDetiltransaksi as $dt)

                                <tr>

                                    <td>{{ $loop->iteration }}</td>

                                    <td>{{ $dt->produk->nama }}</td>

                                    <td>@money($dt->harga)</td>

                                    <td>{{ $dt->qty}}</td>

                                    <td>@money($dt->harga \* $dt->qty)</td>

                                    <td><button class="btn btn-sm btn-danger" wire:click="delete({{ $dt->id }})"><i class=" fas fa-trash"></i></a></td>

                                </tr>

                                @endforeach

                            </tbody>

                        </table>

                    </div>

                    <div class="card-footer text-end">

                        <table class="table table-sm">

                            <tr>

                                <th>

                                    <h2>Total</h2>

                                </th>

                                <th>

                                    :

                                </th>

                                <th>

                                    <h2>@money($total)</h2>

                                </th>

                            </tr>

                            <tr>

                                <th>

                                    <h2>Uang</h2>

                                </th>

                                <th>

                                    :

                                </th>

                                <th>

                                    <input type="number" class="mt-2" wire:model.live="uang">

                                </th>

                            </tr>

                            <tr>

                                <th>

                                    <h2>Kembali</h2>

                                </th>

                                <th>

                                    :

                                </th>

                                <th>

                                    <h2>

                                        @money($kembali)</h2>

                                </th>

                            </tr>

                        </table>

                        <button class="btn btn-lg btn-success" wire:click="receipt({{$data->id}})"><i

                        class="fas fa-print"></i> Cetak</button>

                    </div>

                </div>

            </div>

        </div>

    </div>

</div>

### **Memperbaiki seluruh relasi**

1. Relasi pada Model **Pelanggan.php**

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

use Illuminate\Database\Eloquent\Relations\BelongsTo;

use Illuminate\Database\Eloquent\Relations\HasMany;

class Pelanggan extends Model

{

    use HasFactory;

    protected $fillable=['nama','telepon','alamat'];

    public function transaksi():HasMany

    {

        return $this->hasMany(Transaksi::class);

    }

}

1. Relasi pada model Detiltransaksi.php

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

use Illuminate\Database\Eloquent\Relations\BelongsTo;

class Detiltransaksi extends Model

{

    protected $fillable=['transaksi\_id','produk\_id','qty','harga'];

    use HasFactory;

    public function transaksi():BelongsTo

    {

        return $this->belongsTo(Transaksi::class);

    }

    public function produk():BelongsTo

    {

        return $this->belongsTo(Produk::class);

    }

}

1. Relasi pada Model Transaksi.php

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

use Illuminate\Database\Eloquent\Relations\BelongsTo;

use Illuminate\Database\Eloquent\Relations\HasMany;

class transaksi extends Model

{

    use HasFactory;

    protected $fillable=['invoice','pelanggan\_id','user\_id','total'];

    public function petugas():HasMany

    {

        return $this->hasMany(Petugas::class);

    }

    public function pelanggan():BelongsTo

    {

        return $this->belongsTo(Pelanggan::class);

    }

}

1. Relasi pada Model **Produk.php**

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

use Illuminate\Database\Eloquent\Relations\BelongsTo;

use Illuminate\Database\Eloquent\Relations\HasMany;

class Produk extends Model

{

    use HasFactory;

    protected $fillable=['nama','stok','harga','deskripsi'];

    public function detil\_transaksi():HasMany

    {

        return $this->hasMany(Detiltransaksi::class);

    }

}

### **Memperbaiki komponen livewire**

1. Buka file Penjualan.php yang terdapat pada folder Http\Livewire, kemudian tambahkan beberapa script berikut:

<?php

namespace App\Http\Livewire;

use App\Models\Transaksi;

use Livewire\Component;

use App\Models\Pelanggan;

use Illuminate\Support\Facades\Auth;

class Penjualan extends Component

{

    public $pelanggan\_id;

    public function render()

    {

        return view('livewire.penjualan',[

            'data'=>Pelanggan::orderBy('id','desc')->get()

        ]);

    }

    public function store()

    {

        $this->validate([

            'pelanggan\_id'=>'required'

        ]);

        Transaksi::create([

            'invoice'=>$this->invoice(),

            'pelanggan\_id'=>$this->pelanggan\_id,

            'user\_id'=>Auth::user()->id,

            'total'=>'0'

        ]);

        $this->pelanggan\_id=NULL;

        return redirect()->to('transaksi');

    }

    public function invoice()

    {

        $transaksi=Transaksi::orderBy('created\_at','DESC');

        if($transaksi->count()>0){

            $transaksi=$transaksi->first();

            $explode=explode('-',$transaksi->invoice);

            return 'INV-'.$explode[1]+1;

        }

        return 'INV-1';

    }

}

1. Pada file Transaksis.php yang terdapat di dalam folder Http\Livewire tambahkan property pada class.

<?php

namespace App\Http\Livewire;

use App\Models\Transaksi;

use App\Models\Detiltransaksi;

use App\Models\Produk;

use Exception;

use Illuminate\Support\Facades\Auth;

use Illuminate\Support\Facades\Redirect;

use Livewire\Component;

class Transaksis extends Component

{

    public $total;

    public $transaksi\_id;

    public $produk\_id;

    public $qty=1;

    public $uang;

    public $kembali;

    public function render()

    {

        $transaksi=Transaksi::select('\*')->where('user\_id','=',Auth::user()->id)->orderBy('id','desc')->first();

        $this->total=$transaksi->total;

        $this->kembali=$this->uang-$this->total;

        return view('livewire.transaksis')

        ->with("data",$transaksi)

        ->with("dataProduk",Produk::where('stok','>','0')->get())

        ->with("dataDetiltransaksi",Detiltransaksi::where('transaksi\_id','=',$transaksi->id)->get());

    }

    public function store()

    {

        $this->validate([

            'produk\_id'=>'required'

        ]);

        $transaksi=Transaksi::select('\*')->where('user\_id','=',Auth::user()->id)->orderBy('id','desc')->first();

        $this->transaksi\_id=$transaksi->id;

        $produk=Produk::where('id','=',$this->produk\_id)->get();

        $harga=$produk[0]->harga;

        Detiltransaksi::create([

            'transaksi\_id'=>$this->transaksi\_id,

            'produk\_id'=>$this->produk\_id,

            'qty'=>$this->qty,

            'harga'=>$harga

        ]);

        $total=$transaksi->total;

        $total=$total+($harga\*$this->qty);

        Transaksi::where('id','=',$this->transaksi\_id)->update([

            'total'=>$total

        ]);

        $this->produk\_id=NULL;

        $this->qty=1;

    }

        public function delete($Detiltransaksi\_id)

        {

            $Detiltransaksi=Detiltransaksi::find($Detiltransaksi\_id);

            $Detiltransaksi->delete();

            //update total

            $Detiltransaksi=Detiltransaksi::select('\*')->where('transaksi\_id','=',$this->transaksi\_id)->get();

            $total=0;

            foreach($Detiltransaksi as $od){

                $total+=$od->qty\*$od->harga;

            }

            try{

                Transaksi::where('id','=',$this->transaksi\_id)->update([

                    'total'=>$total

                ]);

            }catch(Exception $e){

                dd($e);

            }

        }

        public function receipt($id)

        {

            //update stok

            $Detiltransaksi = Detiltransaksi::select('\*')->where('transaksi\_id','=', $id)->get();

            //dd($Detiltransaksi);

            foreach ($Detiltransaksi as $od){

                $stoklama = Produk::select('stok')->where('id','=', $od->produk\_id)->sum('stok');

                $stok = $stoklama - $od->qty;

                try {

                    Produk::where('id','=', $od->produk\_id)->update([

                        'stok'=> $stok

                    ]);

                } catch (Exception $e){

                    dd($e);

                }

            }

            return Redirect::route('cetakReceipt')->with(['id' => $id]);

        }

}

### **Menambahkan route**

1. Buka file web.php pada folder routes
2. Tambahkan route penjualan

Route::get('penjualan',function(){

    return view('penjualan.index',[

        "title"=>"Penjualan"

    ]);

})->name('penjualan')->middleware('auth');

1. Tambahkan route transaksi

Route::get('transaksi',function(){

    return view('penjualan.transaksis',[

        "title"=>"Transaksi"

    ]);

})->middleware('auth');

1. Selanjutnya sebelum melakukan uji coba, buka file template.blade.php pada folder layouts, lalu perbaiki menu penjualan pada bagian side menu

<li class="nav-item">

                            <a href="penjualan" target="\_blank" class="nav-link {{ ($title==='Penjualan') ? 'active':'' }}">

                                <i class="nav-icon fas fa-cash-register"></i>

                                <p>

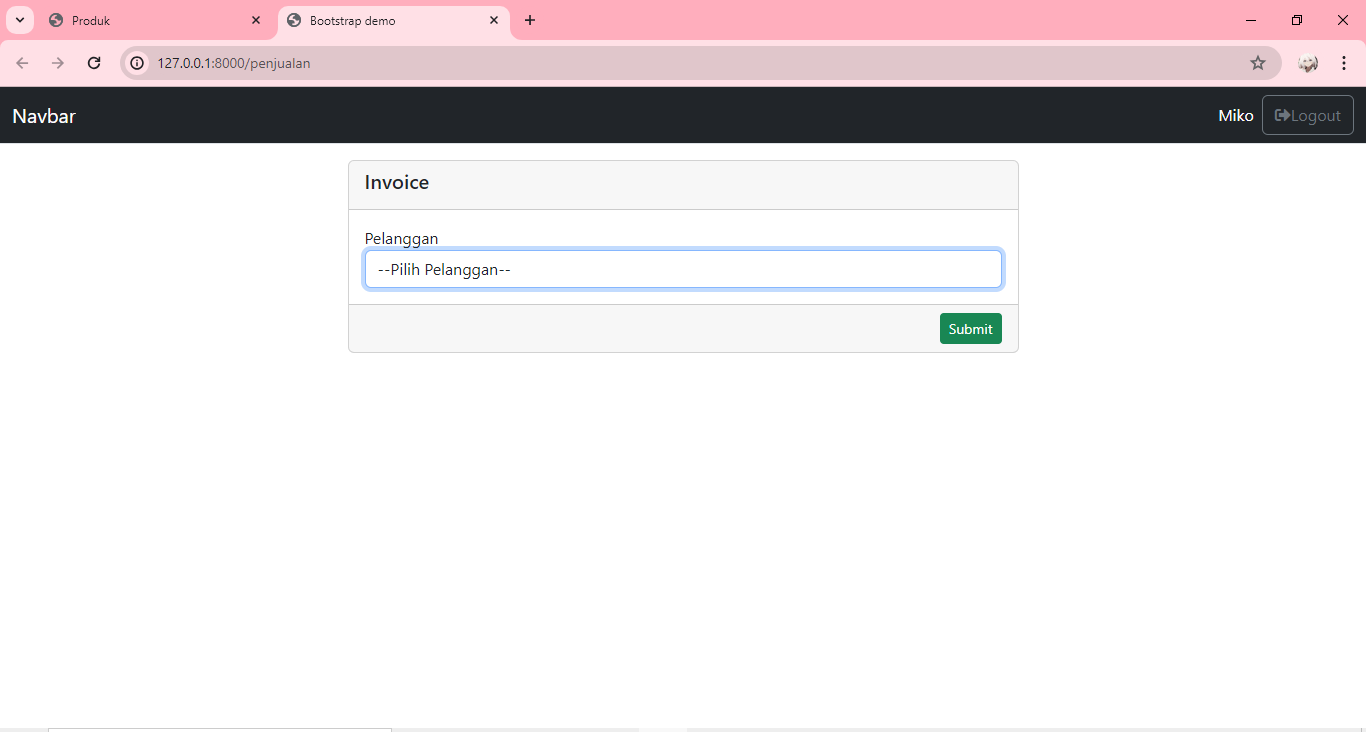
                                    Penjualan

                                </p>

                            </a>

                        </li>

1. Silahkan lakukan uji coba



### **Update Stok dan Cetak Receipt**

1. Perbaiki script tombol cetak pada file **views/livewire/transaksi.blade.php** seperti pada gambar berikut:

<div class="card-footer text-end">

                            <button type="submit" class="btn btn-success btn-sm"><i class="fas fa-cart-plus"></i> Tambahkan</button>

                        </div>

1. Selanjutnya buka file Transaksis.php pada folder App\Http\Livewire, lalu tambah function receipt berikut:

public function receipt($id)

    {

        $detiltransaksi = Detiltransaksi::select('\*')->where('transaksi\_id','=', $id)->get();

        //dd ($detiltransaksi);

        foreach ($detiltransaksi as $od) {

            $stocklama = Produk::select('stock')->where('id','=', $od->produk\_id)->sum('stock');

            $stock = $stocklama - $od->qty;

            try {

                Produk::where('id','=', $od->produk\_id)->update([

                    'stock' => $stock

                ]);

            } catch (Exception $e) {

                dd($e);

            }

        }

        return Redirect::route('cetakReceipt')->with(['id' => $id]);

    }

1. Buka file routes\web.php, tambahkan route berikut:

Route::get('cetakReceipt',[CetakController::class,'receipt'])->name('cetakReceipt')->middleware('auth');

1. Buat controlerr baru, dengan nama CetakController

php artisan make:Controller CetakController

1. Lalu jangan lupa import class CetakController ke dalam routes/web.php
2. Buka Controller CetakController lalu tambahkan script berikut:

<?php

namespace App\Http\Controllers;

use App\Models\Detiltransaksi;

use Illuminate\Http\Request;

use App\Models\Transaksi;

use Illuminate\View\View;

class CetakController extends Controller

{

    //

    public function receipt():View

    {

        $id=session()->get('id');

        $transaksi=Transaksi::find($id);

        //dd ($order)

        $Detiltransaksi=Detiltransaksi::where('transaksi\_id',$id)->get();

        return view('penjualan.receipt')->with([

            'dataTransaksi'=>$transaksi,

            'dataDetiltransaksi'=>$Detiltransaksi

        ]);

    }

}

1. Buat file baru didalam folder **views/penjualan** dengan nama **receipt.blade.php**
2. Tambahkan script berikut

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Cetak Receipt</title>

    <link rel="preconnect" href="https://fonts.googleapis.com">

    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

    <link href="https://fonts.googleapis.com/css2?family=Inconsolata:wght@300;400&di

splay=swap" rel="stylesheet">

    <style>

        @page {

            margin: 0;

            border: 0;

            size: 80mm, 100%;

        }

        body {

            font-size: 9pt;

            font-family: 'Inconsolata', monospace;

            text-align: justify;

        }

    </style>

</head>

<body>

    <p>

        invoice:{{ $dataTransaksi->invoice}}<br>

        Customer:{{ $dataTransaksi->pelanggan->nama}}<br>

        Cashier:{{ Auth::user()->nama}}<br>

        Tanggal:{{$dataTransaksi->created\_at->format('d M Y H:m')}}

    </p>

    <table>

        <thead>

            <tr>

                <td>Product</td>

                <td>Qty</td>

                <td>Price</td>

                <td>Amount</td>

            </tr>

        </thead>

        @foreach($dataDetiltransaksi as $dod)

        <tr>

            <td>{{ $dod->produk->nama }}</td>

            <td>{{$dod->qty}}</td>

            <td>@money($dod->harga)</td>

            <td>@money($dod->harga\*$dod->qty)</td>

        </tr>

        @endforeach

        <tr>

            <td colspan="3">Total:</td>

            <td>@money($dataTransaksi->total)</td>

        </tr>

    </table>

    <script src="{{ asset('plugins/jquery/jquery.min.js')}}"></script>

    <script>

        $(document).ready(function() {

            window.print();

            setInterval(myURL, 5000);

        });

        function myURL() {

            document.location.href = "{{ route('penjualan') }}";

            // clearInterval(interval);

        }

    </script>

</body>

</html>

1. Lakukan uji coba kembali, pastikan stok terupdate dan semua menu dan tombol berfungsi dengan baik.

## Fitur Laporan