# Setup Project

## Instalasi Laravel

### Apa itu Laragon, Composer dan Laravel

1. Laragon adalah sebuah servel local yang memudahkan pengembangan web. Laragon menyediakan lingkungan pengembangan yang terintegrasi dengan PHP, MySQL dan berbagai Tools lainnya.
2. Composer, adalah alat pengelola dependensi PHP yang digunakan untuk menginstal dan mengelola library atau framework PHP seperti Laravel
3. Laravel, adalah sebuah framework PHP yang popular digunakan untuk membangun aplikasi web modern.

### Instalasi Laragon

1. Kunjungi situs resmi Laragon (<https://laragon.org/download/>) dan unduh installer sesuai dengan sistem operasi Anda.
2. Jalankan installer dan ikuti petunjuk yang diberikan. Pilih lokasi instalasi yang diinginkan.
3. Setelah instalasi selesai, Anda bisa melakukan konfigurasi tambahan jika diperlukan.

### Instalasi Composer

1. Kunjungi [https://getcomposer.org/download/](https://www.google.com/url?sa=E&source=gmail&q=https://getcomposer.org/download/) dan unduh installer Composer.
2. Jalankan installer Composer dan ikuti petunjuknya. Pastikan Anda menambahkan path ke Composer ke dalam variabel environment PATH.
3. Buka terminal atau command prompt dan ketik composer -v. Jika Composer terinstal dengan benar, versi Composer akan ditampilkan.

### Instalasi Project Laravel dengan Laragon Shortcut

1. Pastikan Laragon sedang berjalan.
2. Klik tombol mulai atau logo laragon untuk menjalankan server php dan apachce.
3. Klik kanan di area dekstop/kerja laragon
4. Klik “quick app” , kemudian isi nama project yang akan di buat, lalu tunggu hingga proses instalasi selesai.
5. Untuk menjalankan project, klik kanan pada area dekstop laragon, lalu klik www>nama\_project, kamu akan diarahkan ke browser dengan alamat project yang kamu buat. Biasanya laragon sudah memberikan fitur pretty url dengan akhiran .test (“nama\_project.test”).

### Instalasi Project Laravel Perintah Terminal Composer

1. Buka terminal yang ada di laragon, untuk memudahkan kamu berada pada directory yang biasa di gunakan yaitu www/
2. Ketikan perintah

*composer create-project laravel/laravel nama-project*

1. Masuk ke direktori project

*cd nama-proyek*

1. Masukan perintah berikut untuk menjalankan project kamu :

*php artisan serve*

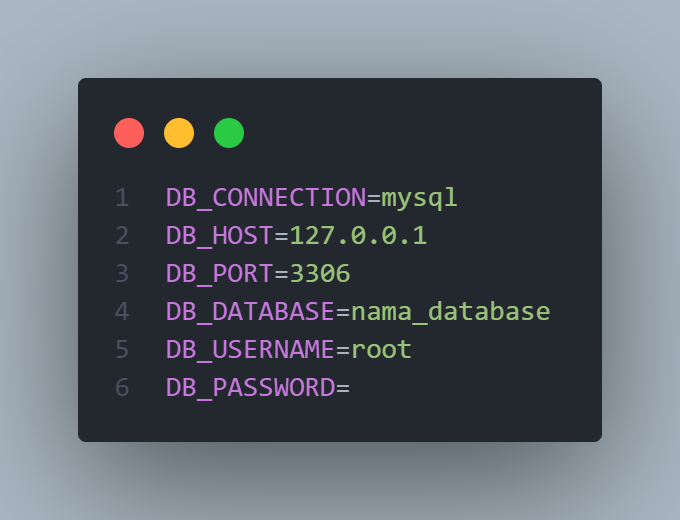
1. Akses proyek melalui browser dengan URL yang ditampilkan di terminal, biasanya <http://127.0.0.1:8000> atau <http://nama-project.test> .

## Database Struktur

Dalam pembelajaran kali ini, kita tidak menggunakan cara tradisional laravel, melainkan kita akan menggunakan third-party package yang disebut dengan blueprint. Berikut adalah langkah – langkah menggunakan package blueprint:

1. Buka terminal dan pastikan kamu sedang di dalam project nya. Lakukan instalasi nya dengan menjalankan perintah berikut:

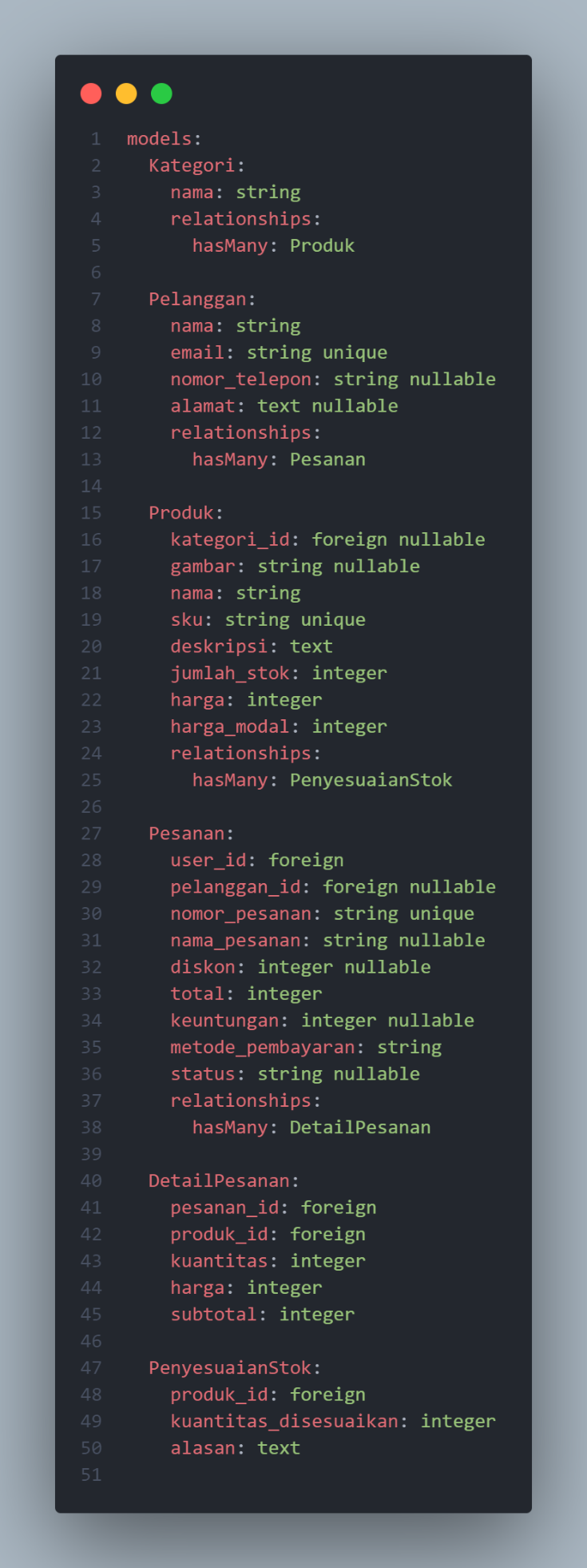
*composer require -W --dev laravel-shift/blueprint*

1. Setelah itu, maka sekarang kamu bisa buat satu database dengan nama project yang kamu buat pada saat instalasi laravel, jika sudah maka sekarang kamu bisa buka file .env dan memodifikasi bagian DB\_ seperti berikut:
2. Setelah itu, buka kembali terminal Anda untuk mengenerasi file draft.yaml

*php artisan blueprint:new*

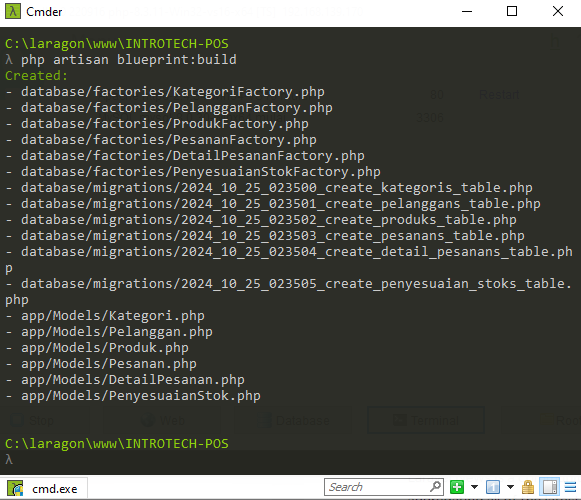
dengan begitu, maka akan ada file baru dengan nama draft.yaml di root directory.

1. Silakan buka file draft.yaml nya dan silakan buat struktur tabel seperti dibawah ini:

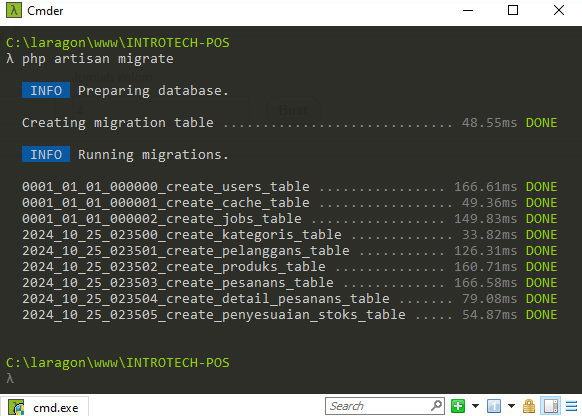


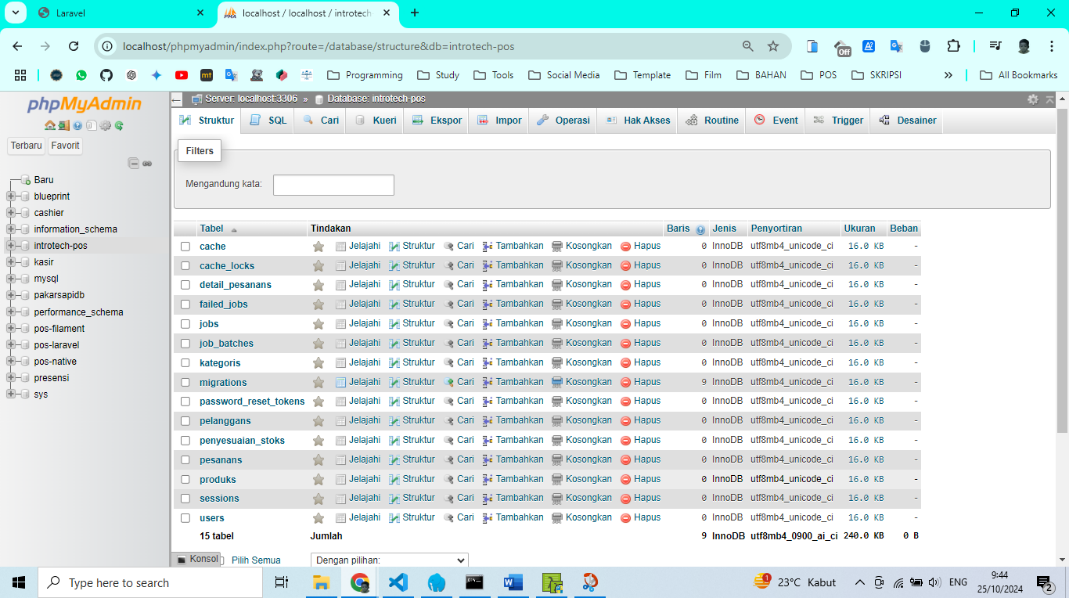
1. Setelah itu, maka sekarang Anda bisa buka kembali terminal untuk mengenerate migration, factory dan juga model nya dengan perintah:

*php artisan blueprint:build*

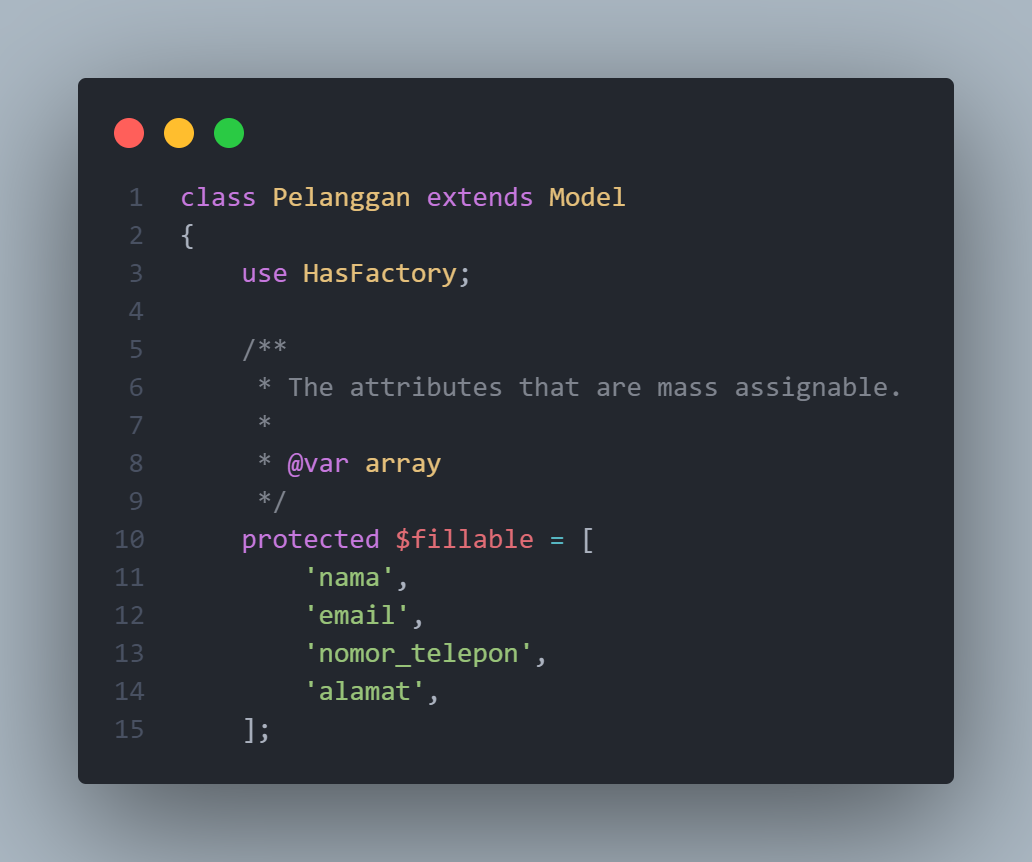
jika tidak ada masalah, harusnya output yang muncul akan seperti ini:

1. Setelah itu, maka sekarang silakan lakukan migrate dengan menjalankan perintah berikut:

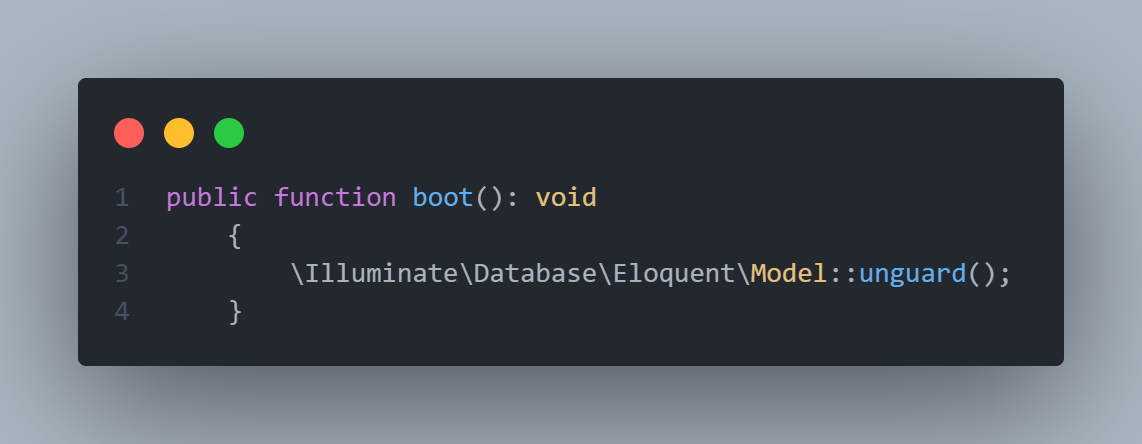
*php artisan migrate*



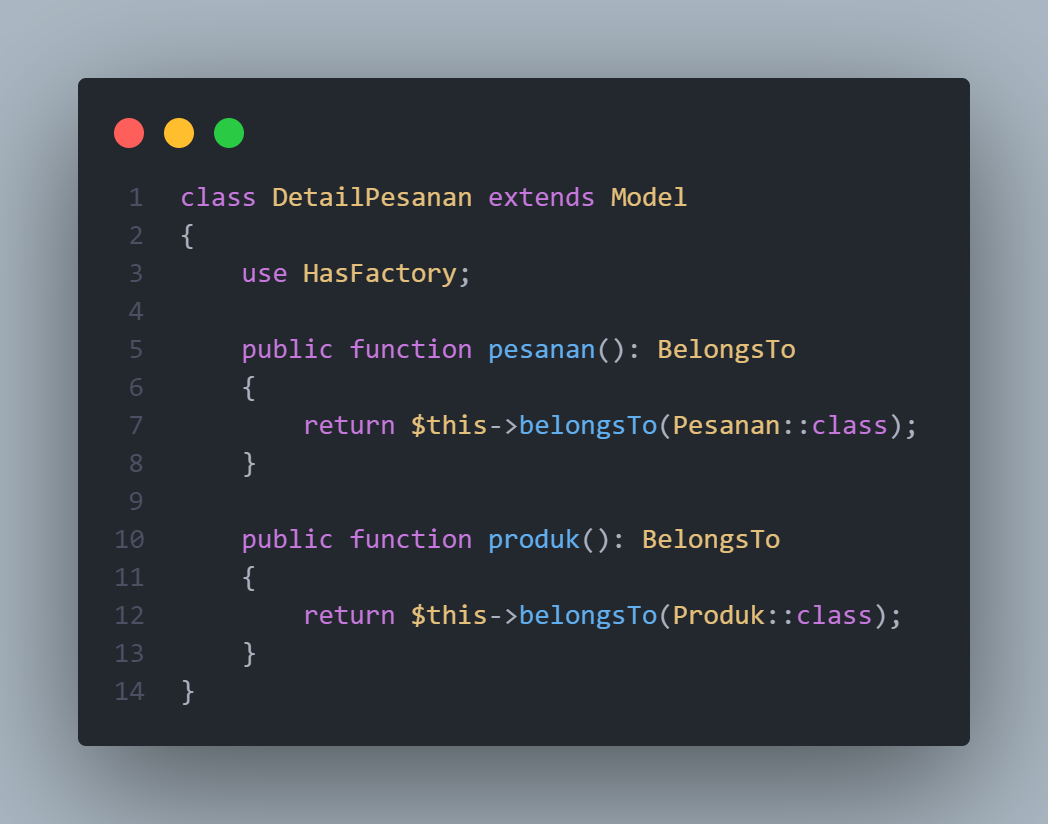
## Memodifikasi Model

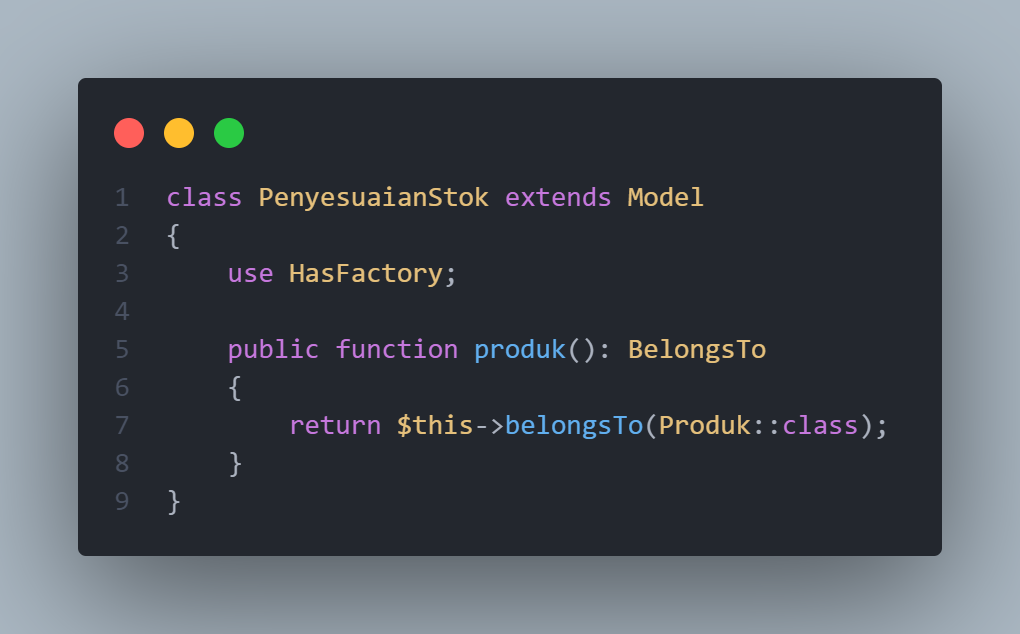
Jika kita perhatikan pada setiap model yang ada, semua pasti ada fillable fill nya seperti misalnya pada model **Pelanggan.php**

Yang mana semua itu akan kita modifikasi menjadi guarded, namun kita tidak akan melakukan nya di semua model, alih-alih melakukan itu satu persatu disetiap model kita akan melakukan nya sekaligus di file app/Providers/AppServiceProvider.php,

buka file nya dan silakan tambahkan ini pada method boot().

Kemudian, selanjutnya tugas kita adalah memodifikasi model yang kita punya yaitu membersihkan semua fillable yang tidak penting.

1. Kategori.php
2. Pelanggan.php
3. Pesanan.php
4. DetailPesanan.php
5. Produk.php
6. PenyesuaianStok.php



## Menerapkan Enums

Dalam tabel orders, terdapat field payment\_method dan status yang, seharusnya menggunakan enum, sengaja dibuat sebagai string. Hal ini dilakukan untuk memudahkan penambahan nilai baru ke dalam enum tersebut dari kode kita sendiri, tanpa perlu mengubah struktur database.

Untuk melakukan itu, silakan buat 1 folder di dalam folder app dengan nama Enums dan di dalam nya silakan buat 2 file sekaligus yaitu:

1. StatusPesanan.php
2. MetodePembayaran.php

### Status Pesanan

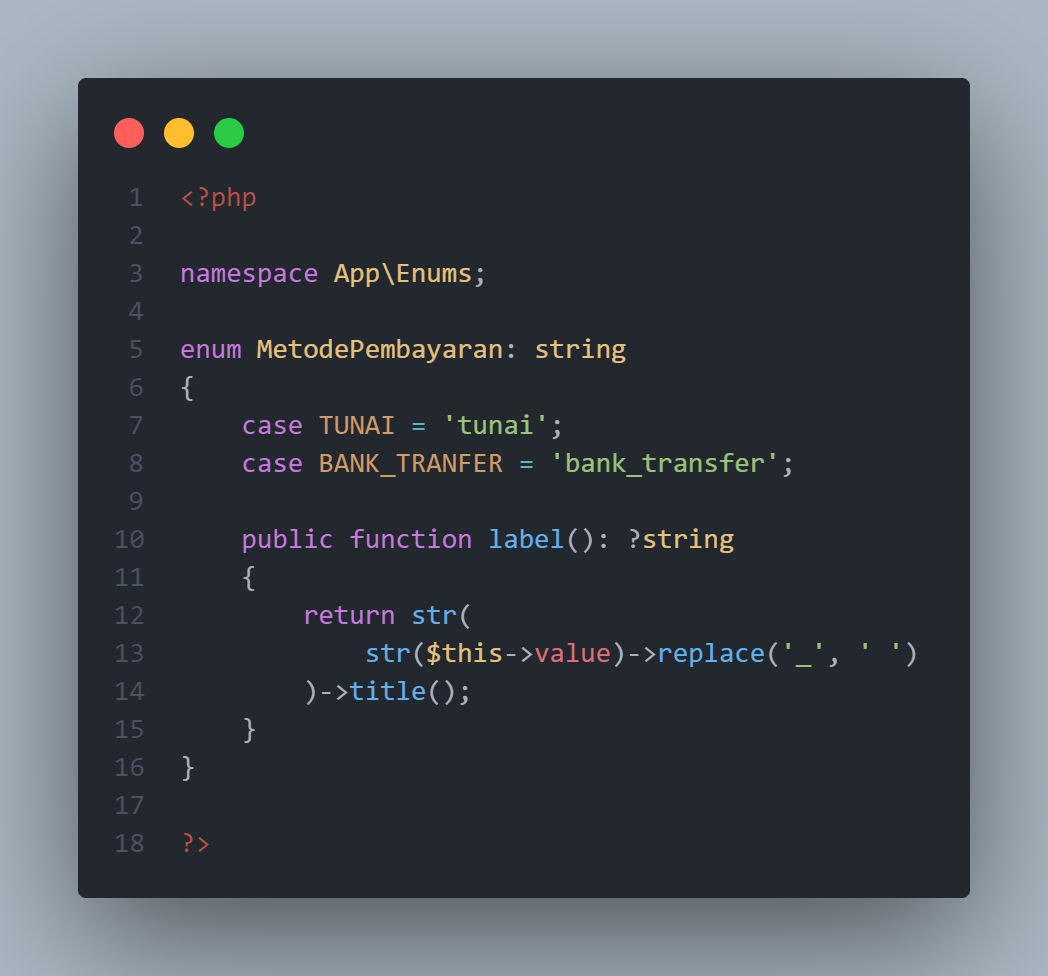
Buka file StatusPesanan.php dan masukan kode berikut:



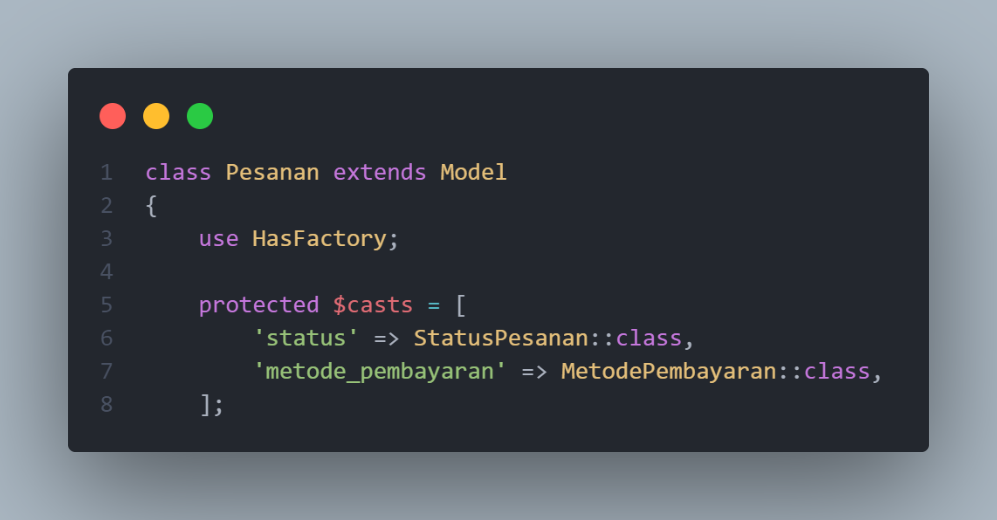
Jika Anda perhatikan baik-baik pada kode di atas, saya membuat 2 tambahan method yaitu label() dan color(). Itu saya lakukan agar opsi untuk select kita nanti terlihat bagus. Karena by default, jika kita masukkan enum ini ke select, secara otomatis akan mengambil nama nya itu huruf besar semua. Dengan method label() ini, maka dia akan lebih terlihat bagus dengan bantuan str dari laravel. Jika color(), itu hanya akan membantu kita nanti nya dalam tampilan view atau pun tabel.

### Metode Pembayaran

Untuk payment method ini sangat sederhana sekali, karena juga kita tidak perlu membuat fungsi color() nantinya.

Buka filenya dan silakan masukkan enum nya seperti:

### Cast Enums

Sekarang, waktunya kita untuk cast enum ini ke model Order. Buka file nya dan silkan buat cast seperti:

## Database Factory dan Seeder

Dalam hal ini, kita tentu butuh dummy data agar kita tidak perlu membuat data satu persatu. Seperti misalnya kategori, atau bahkan produk nya. Untuk itu, silakan buka terminal Anda dan silakan buat seeder baru dengan nama KategoriSeeder.

### Buat Seeder

Untuk membuat seeder kamu bisa menjalankan 2 perintah berikut:

*php artisan make:seeder CategorySeeder*

*php artisan make:seeder UserSeeder*

### Modifikasi function run() KategoriSeeder.php



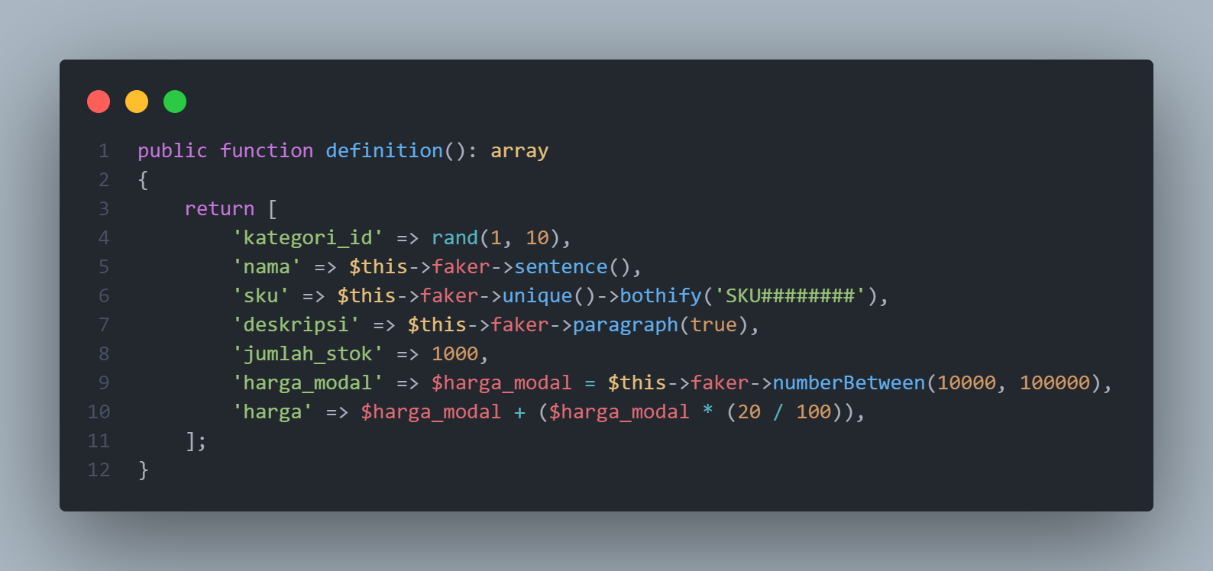
### Modifikasi function run() UserSeeder.php



Untuk credentials di atas bisa Kamu ganti dengan nama Kamu. Oia, mungkin Kamu bertanya juga, kenapa kita butuh user seeder ini, itu sengaja saya lakukan agar nanti kita tidak perlu menjalankan perintah make:filament-user setelah kita lakukan seed.

### Modifikasi function definition() ProdukFactory.php

karena kita perlu memodifikasi factory yang diberikan oleh blueprint by default. Pada method definition() silakan modifikasi seperti:



### Modifikasi function definition() PenyesuaianStokFactory.php

Kemudian selanjutnya, kita perlu membuat dummy data juga untuk stock adjustment, dimana kita ingin membuat semacam factory yang langsung mempunyai efek terhadap product yang telah kita buat. Buka file **PenyesuaianStokFactory.php**, dan silakan modifikasi method definition() dan tambahkan 1 method lagi yaitu configure() seperti:

Dengan begitu, setelah factory ini dibuat, maka otomatis produk akan langsung diperbarui. Makanya di sini jika Anda perhatikan baik-baik, kita menggunakan afterCreating().

### Modifikasi function run() DatabaseSeeder.php

Setelah semua nya, maka sekarang kita bisa lanjut untuk mendaftarkan factory dan seeder tepat pada file DatabaseSeeder.php. Buka filenya dan silakan modifikasi isi dari method run() menjadi seperti:



### Seeding

Kemudian sekarang, kita bisa lanjut untuk menjalakan seeder ini dengan menjalakan perintah seed, namun saya berfikir akan langsung membuat yang namanya migrate fresh, agar jika ada data default sekarang menjadi bersih. Buka terminal Kamu silakan jalankan perintah berikut:

*php artisan migrate:fresh --seed*