LARAVEL TUTORIAL

@Gwanghaegun

BAGIAN KE2

**MEMBUAT CRUD DENGAN LARAVEL**

Pada Tutorial seelumnya, kita sudah belajar bagaimana membuat halaman statis menggunakan Route -> view, dan Route->controller-> view, sekali lagi, halaman statis merupakan halaman tetap, yang kita nggak akan ubah unutk waktu lama, dan tidak sehingga kita nggak perlu menggunakan Database.

Nah, pada bagian ini, kita akan membuat Sistem CRUD sederhana dengan Laravel, dimana pada tutorial kali ini, kita akan mulai menggunakan Model dan juga akan memaksimalkan penggunaan si Controller.

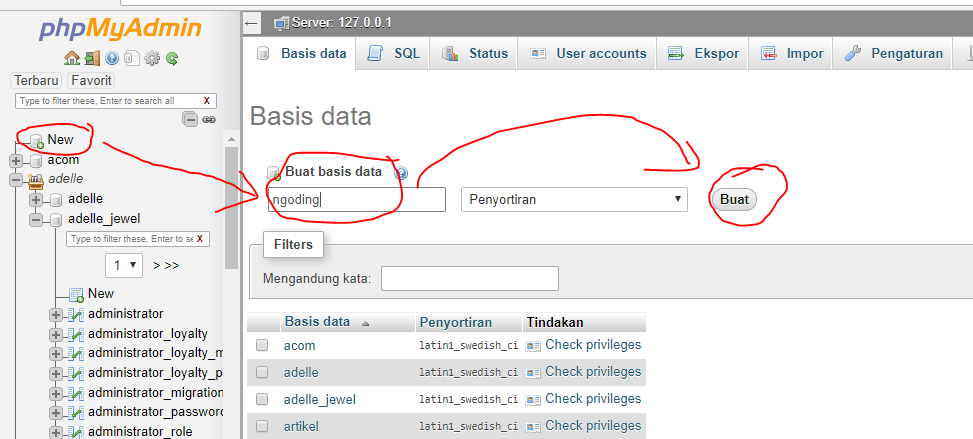
View

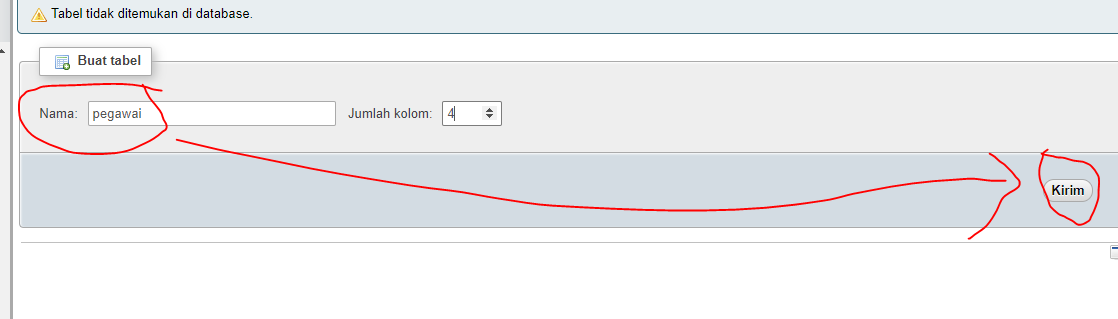
Controller

Route

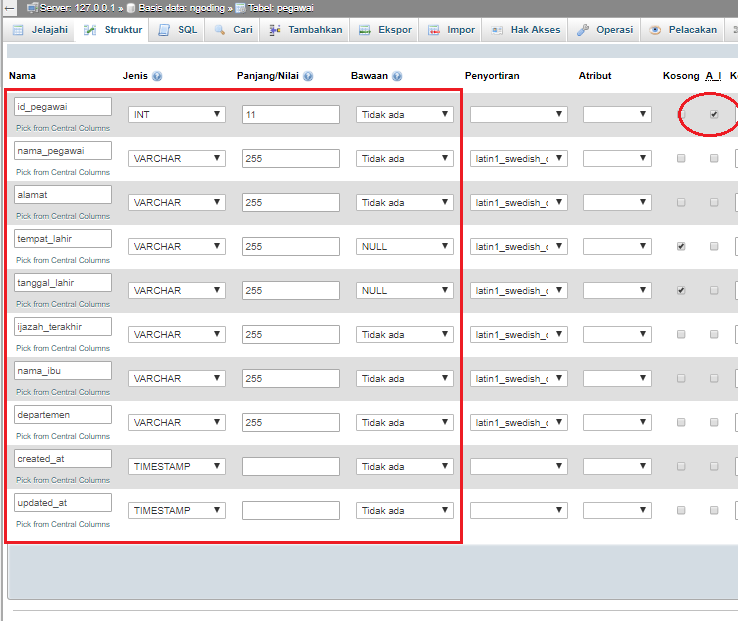
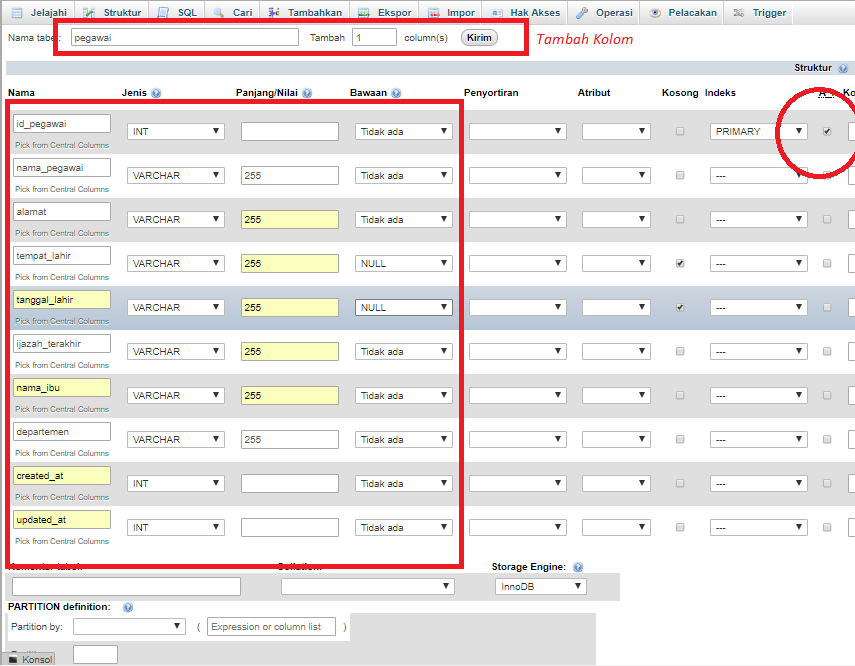
# Persiapan Dan Koneksi dengan Database

## Siapkan dulu database dan buat tabel ‘pegawai’.



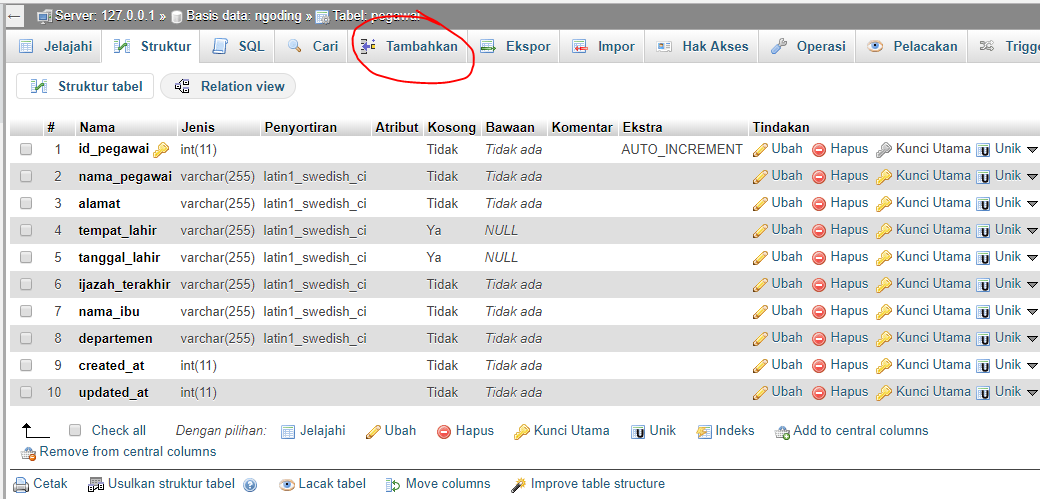


*Untuk jumlah kolom, bisa disesuaikan nanti, berikan nilai berapapun, bebas.*

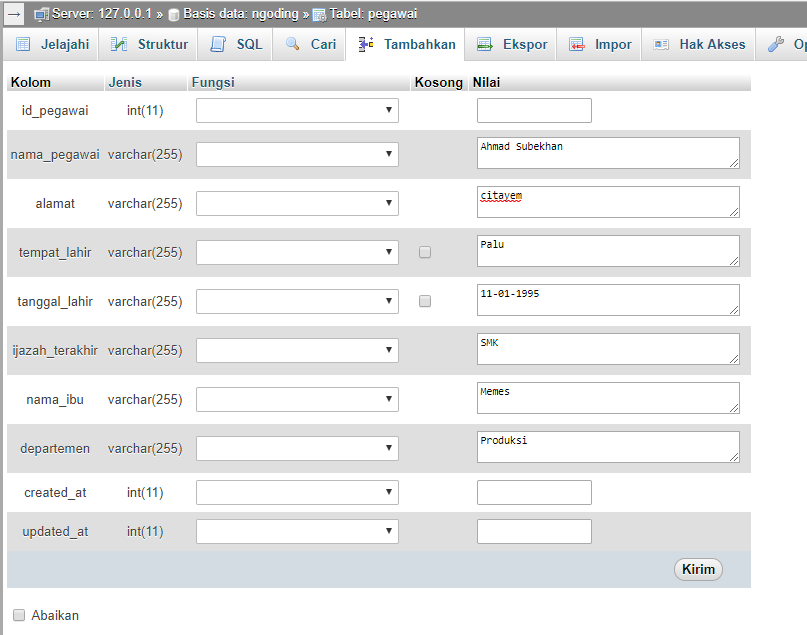


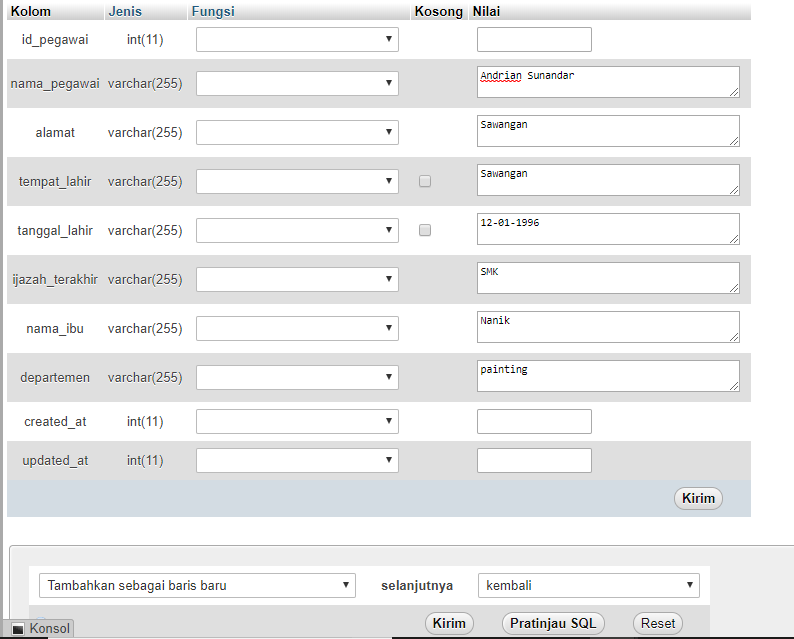
*Setelah itu klik simpan/ go*

## Selesai membuat tabel, kita bisa isi data dengan data dummy (bohongan), caranya klik tab tambahkan/ insert seperti gambar dibawah:



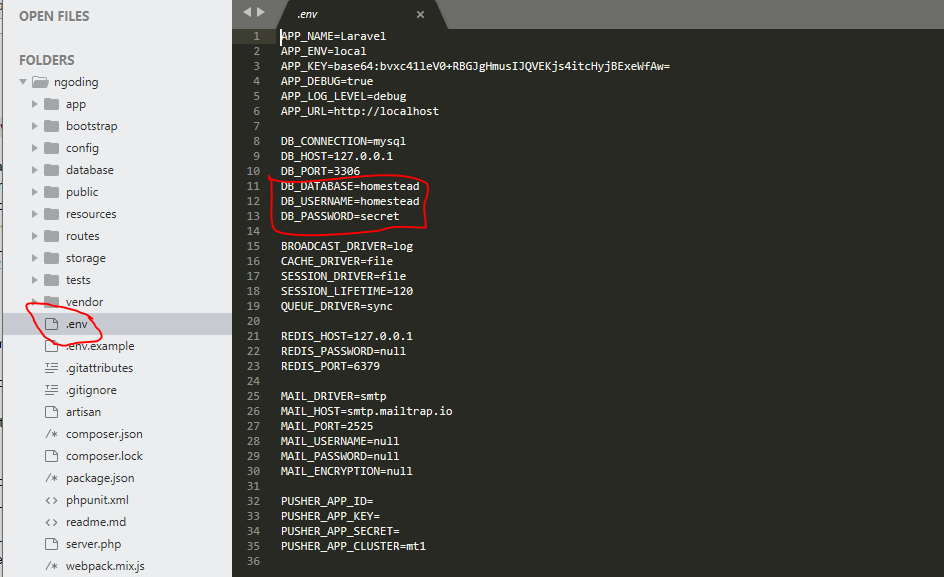
## Isi data seperti berikut:



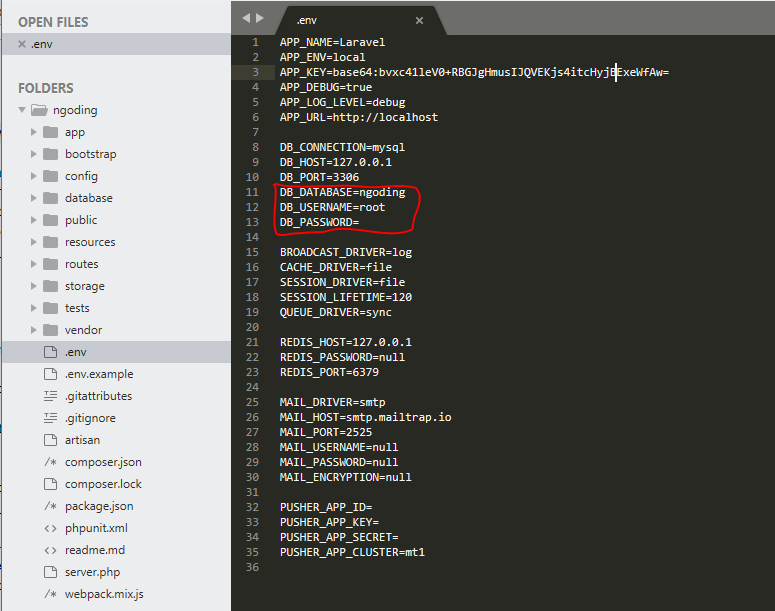


*Setelah itu klik kirim/ go*

## Setelah itu, buka project yang sudah ada, atau project yang baru di download. Buka file **.env** , ubah beberapa bagian.



Menjadi:



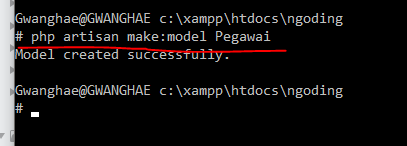
## Setelah mendaftarkan database, user dan password, kita insya allah sudah bisa konek dengan database, kenapa insya allah? Karena bisa aja kita Typo ngetik nama database atau passwordnya ngga sengaja kita isi, makanya disini kita butuh ketelitian

## Membuat Model Pegawai.

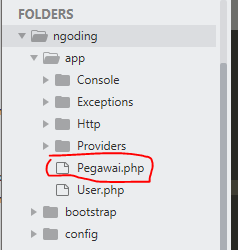
Setelah kita membuat database dan tabel, kemudian menghubungkannya dengan ENV, maka selanjutnya kita perlu membuat ‘Perwakilan’ tabel tersebut supaya kita bisa akses, nah di dalam Laravel, sebuah tabel di database dapat diwakili oleh 1 model, dalam hal ini, kita akan membuat perwakilan tabel pegawai, maka kita buat juga sebuah model dengan nama **Pegawai**, nama model itu menggunakan huruf kapital di awal, tidak harus, tapi untuk sekedar lebih rapih saja.

Untuk membuatnya kita buka kembali CMD, kemudian ketik :

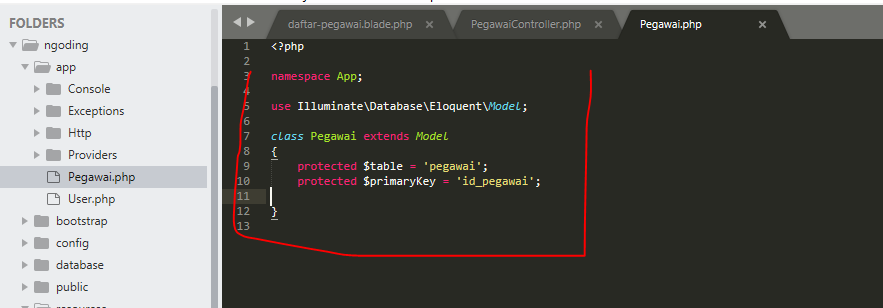
**php artisan make:model Pegawai** kemudian tekan enter



Untuk memeriksa apakah model sudah dibuat, buka folder app:



Buka file tersebut dan ubah menjadi seperti ini:

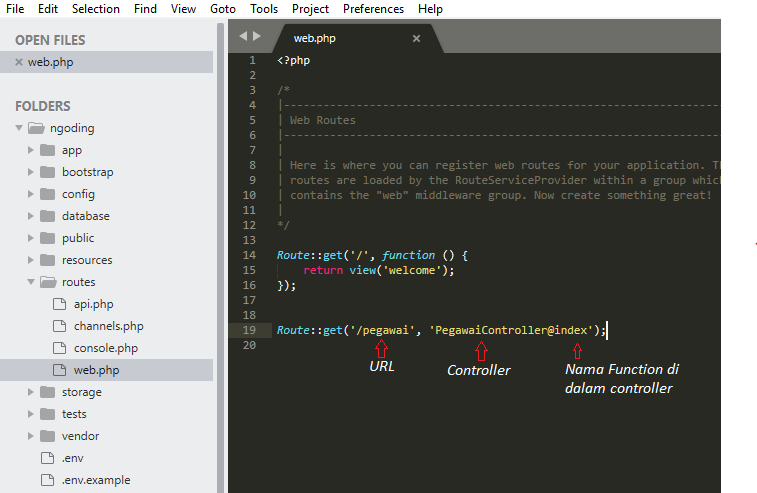


# MENAMPILKAN DATA

langkah pertama dari CRUD (Create Read Update Delete) yang akan kita mulai disini adalah **Read** data.

## Caranya, pertama tama, daftarkan URL untuk memanggil halaman View,

## buka file **web.php** di dalam folder **routes** :



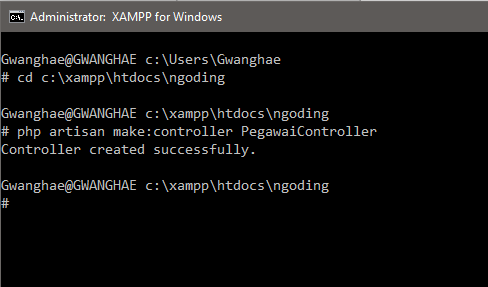
Dari gambar diatas, kita membuat URL yg kita beri nama: ‘/**pegawai’** boleh juga ‘**pegawai’** tanpa garis miring di depan, kemudian url tersebut akan memanggil file controller bernama **PegawaiController** (belum dibuat) yang di dalamnya kemudian akan mengeksekusi function yang bernama **index.**

## Membuat Controller dengan CMD

Buka CMD, kemudian masuk ke dalam folder project kita,

Ketik **cd c:\xampp\htdocs\nama\_project** kemudian tekan ENTER.

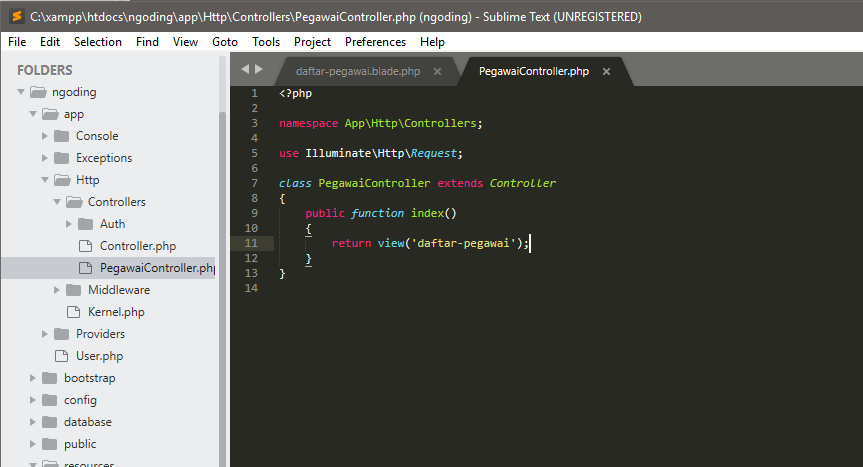
Setelah kita berhasil masuk ke dalam project folder kita, buat lah controllerny dengan cara mengetik **php artisan make:controller PegawaiController** lalu tekan enter.



Setelah berhasil, cek pada sublime/atom anda, buka folder **app/Http/Controllers** PegawaiController sudah dibuat otomatis oleh LARAVEL.

## Halaman Controller

Buka halaman PegawaiController yang sudah kita buat, kemudian isi halaman tersebut dengan:



Kita telah menambahkan satu fungsi untuk memanggil halaman view yang bernama **daftar-pegawai**, yang akan mencari halaman **daftar-pegawai.blade.php** dimana halaman tersebut yang nantinya akan menampilkan data yang kita bawa dari database.

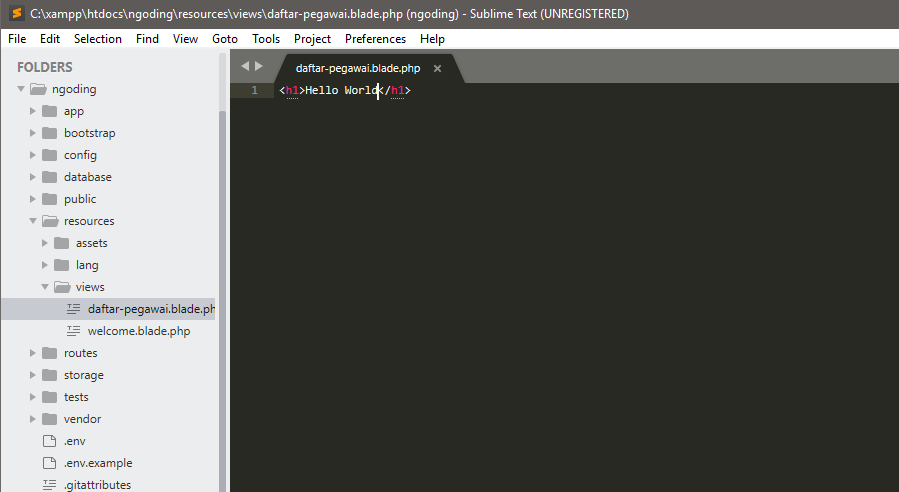
Tentunya file **daftar-pegawai.blade.php** belum kita buat, dan pada langkah selanjutnya akan kita buat.

## Membuat Halaman View.

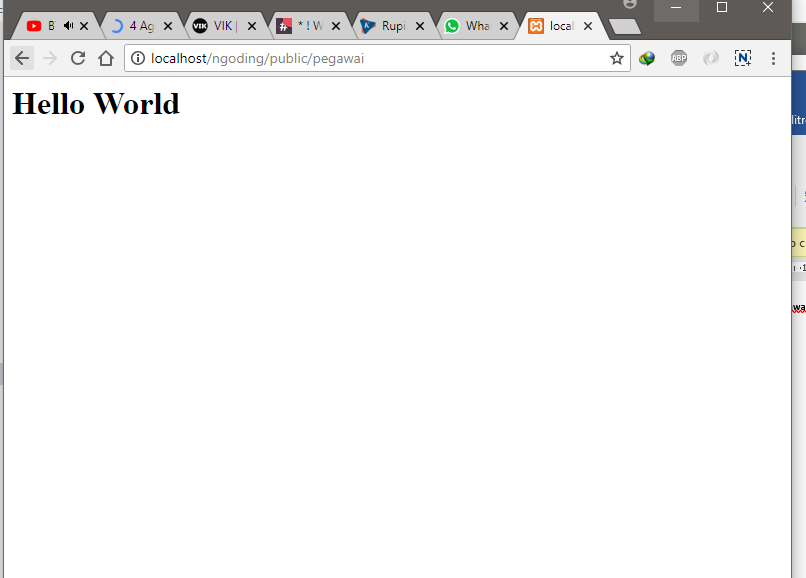
Pada langkah ini, kita kan membuat halaman daftar-pegawai.blade.php

Ngga ad acara otomatis untuk halaman view seperti kita membuat Controller tadi, karena kita cukup mengkik kanan pada folder **views** dan pilih add new, dimana letak folder views? Anda bisa menemukannya di dalam folder **Resource.**

Untuk sementara, kita isi di dalamnya dengan “Hello World” dan simpan dengan nama **daftar-pegawai.blade.php**



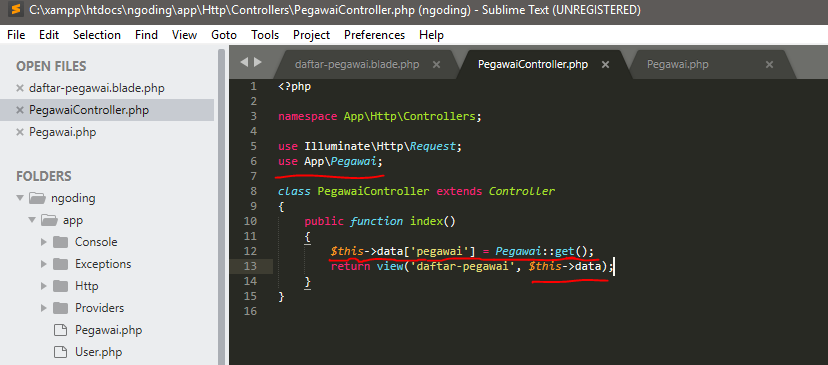
Test apakah halaman tersebut tampil ketika kita mengetik **‘pegawai’** setelah **public/**



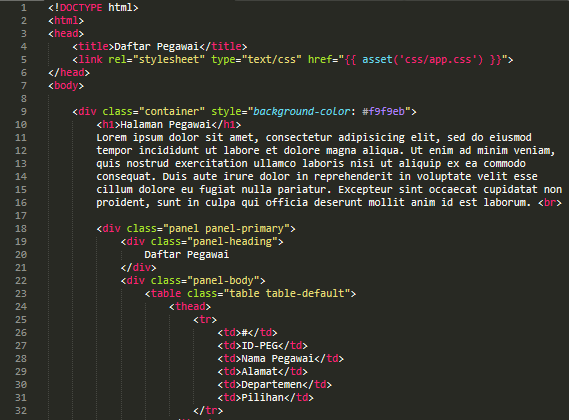
## Menampilkan Data Pegawai

Setelah kita berhasil menampilkan hello world, itu artinya ngga ada yang salah dengan **Route->Controller->views** nya . nah sekarang, kita bakal lanjut nampilin data dari tabel ke dalam halaman **daftar-pegawai.blade.php**

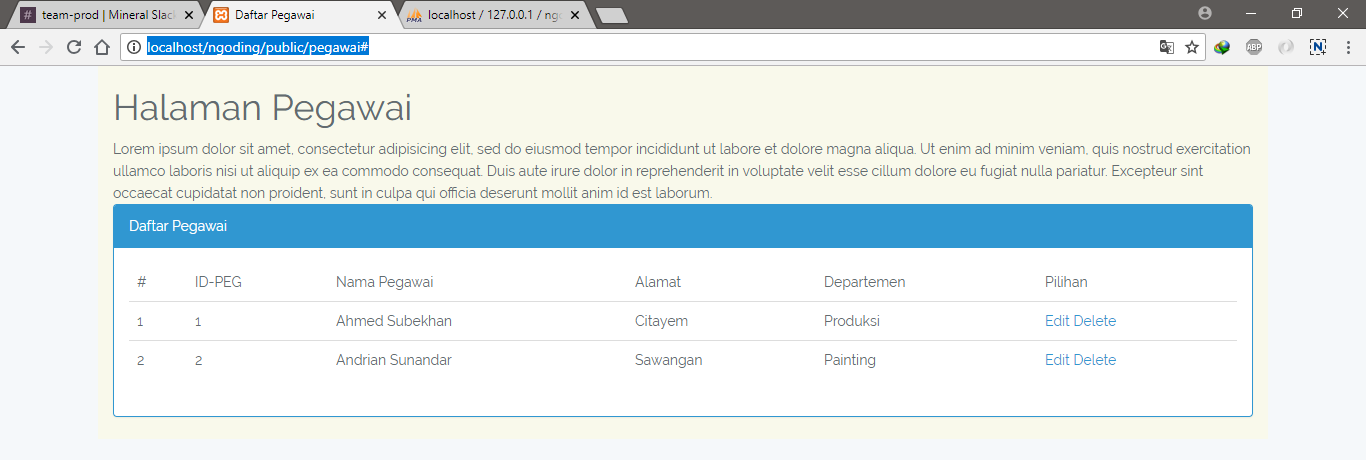
Balik lagi ke file **PegawaiController**, buka dan ubah seperti berikut:



Setelah itu buka halaman **daftar-pegawai.blade.php** dan ubah menjadi seperti berikut:



Save kemudian jalankan:



# Menambah Data

## Menyiapkan URL tambah

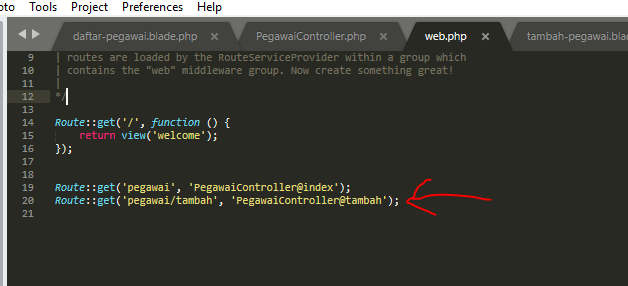
## Untuk bisa menuju halaman form Tambah Data, kita harus mendaftarkan Urlnya terlebih dahulu ke dalam **Routes.**

Sebelum itu, buka kembali file **daftar-pegawai.blade.php** kemudian tambahkan scrypt seperti di gambar

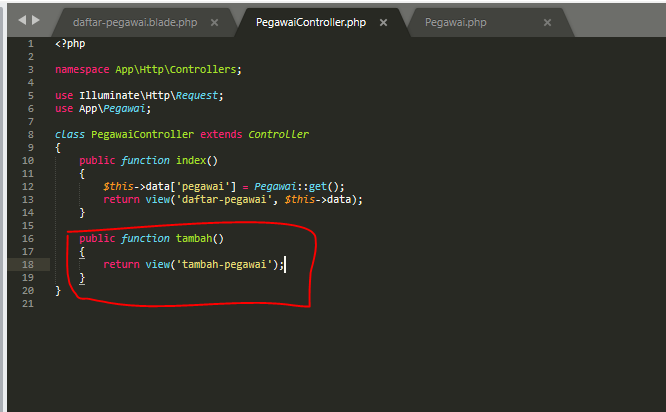


Scrypt diatas berfungsi untuk menambahkan TOmbol, dengan a href, dan disitu bisa dilihat kalau kita akan menggunakan url **pegawai-tambah**  , url tersebut tidak akan bisa kita akses sebelum kita tambahkan dia di fille **web.php** yang terletak di dalam folder **routes** .

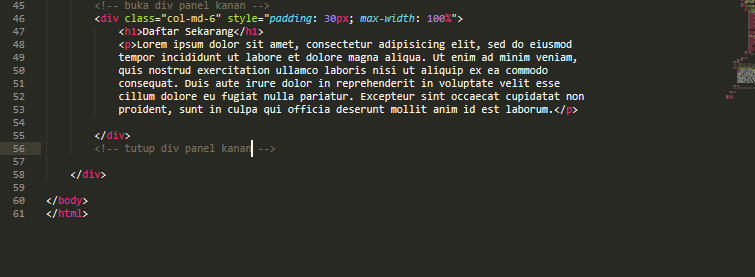
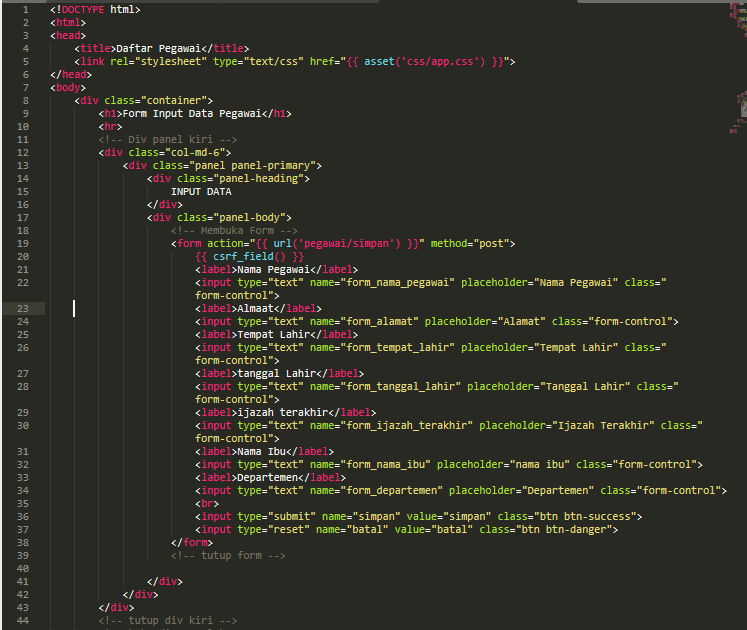
Buka file tersebut dan tambahkan seperti gambar berikut:



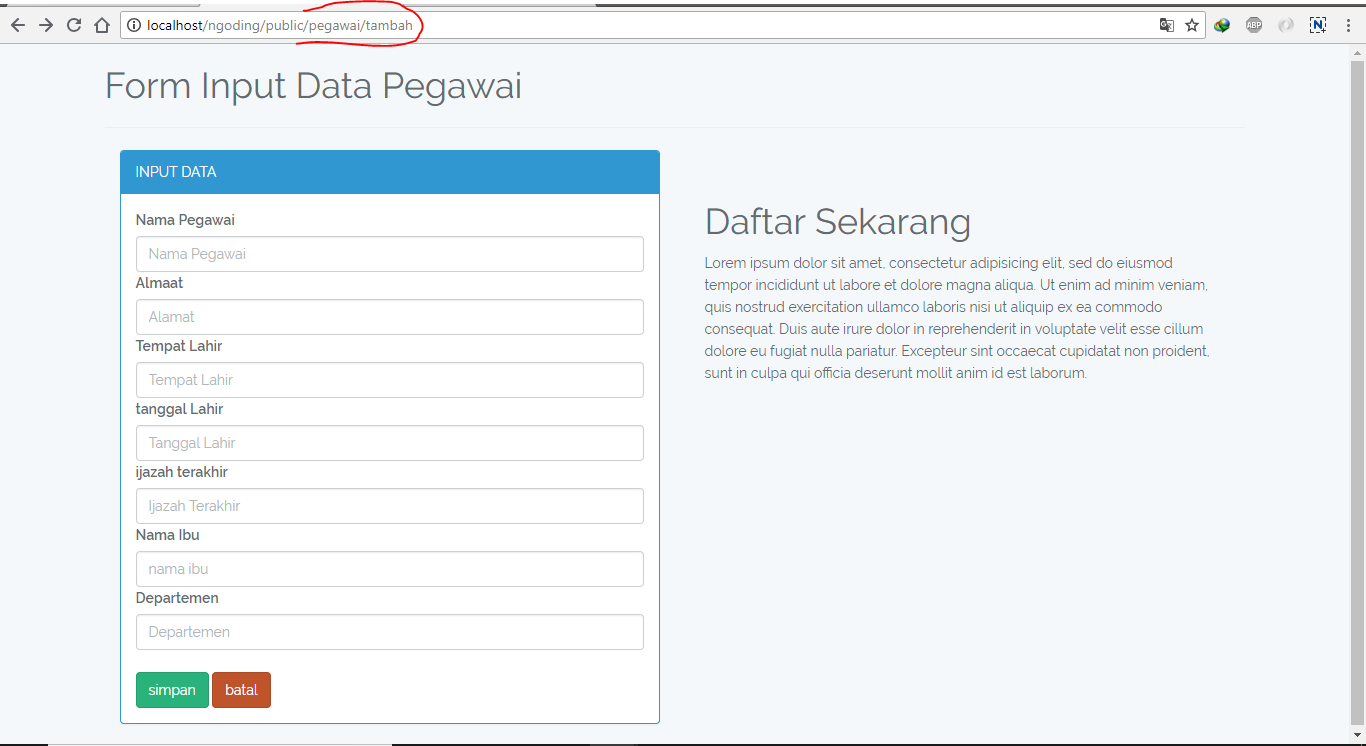
Setelah itu, buka file **PegawaiController** di dalam folder **app/Http/Controllers** dan tambahkan ubah menjadi seperti ini:



## Membuat File Tambah Pegawai

Klik kanan pada folder **views**, di dalam folder **resources,** pilih new file, lalu simpan dengan nama **tambah-pegawai.blade.php** kemudian isi seperti gambar berikut:

Kalau di run, hasilnya akan tampak seperti ini:

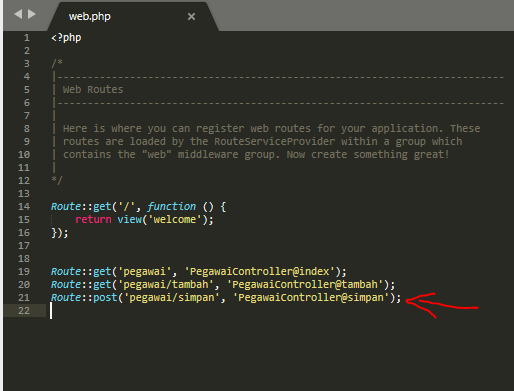


## Menyimpan inputan

setelah Form selesai, selanjutnya adalah bagaimana agar data tersebut bisa tersimpan ke dalam database,

pertama, buat url untuk menyimpan, url yg akan kita gunakan disini ialah **pegawai/simpan**

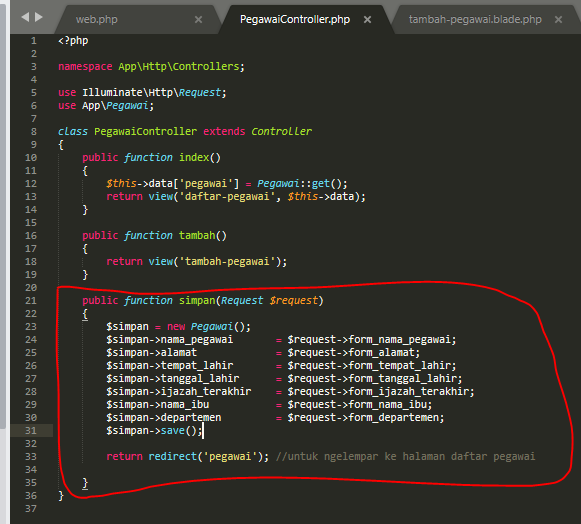
buka file **web.php** yang berada di dalam folder **routes**



Gambar diatas menunjukkan kita mendaftarkan url **pegawai/simpan** dengan method POST, berbeda dengan method sebelumnya, untuk mengirim data maka kita perlu method POST.

setelah itu, buka file **PegawaiController**

dan buat seperti berikut

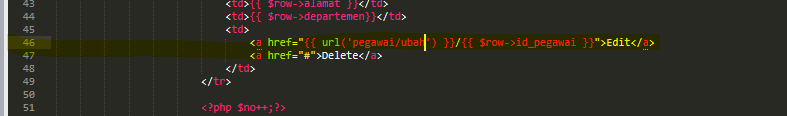


Setelah itu, isi form input pegawai tadi, lalu simpan, jika sesuai, maka data akan tersimpan

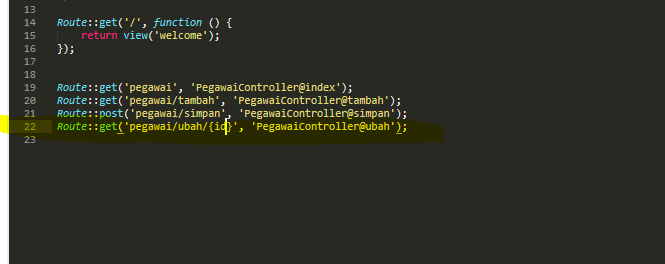
# Mengubah Data

## Siapkan URL

Siapkan url yang akan kita gunakan untuk mengubah, sebelum itu, buka kembali file **daftar-pegawai.blade.php** lalu ubah pada bagian tombol edit

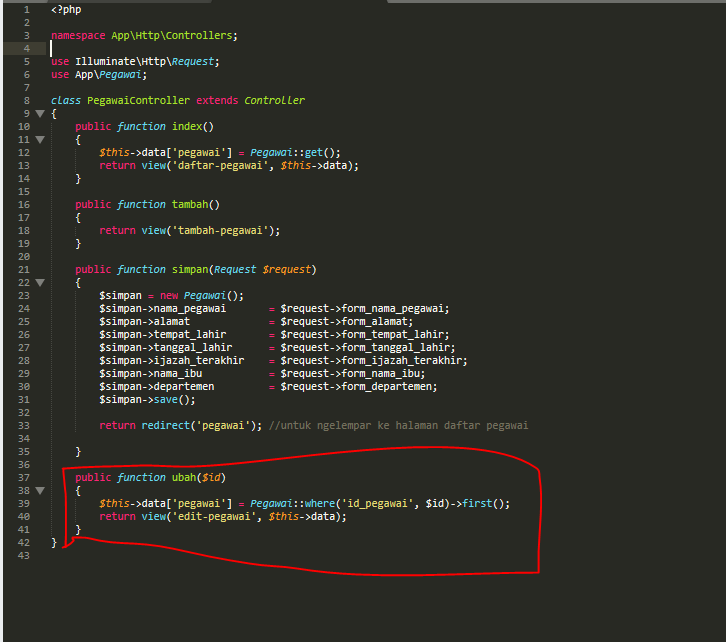


Selanjutnya daftarkan url tersebut ke dalam **web.php** di dalam folder **Routes**



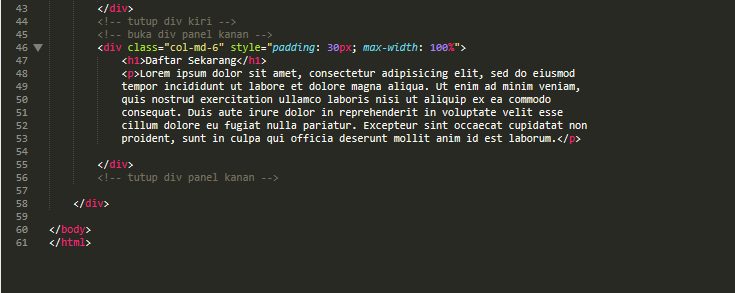
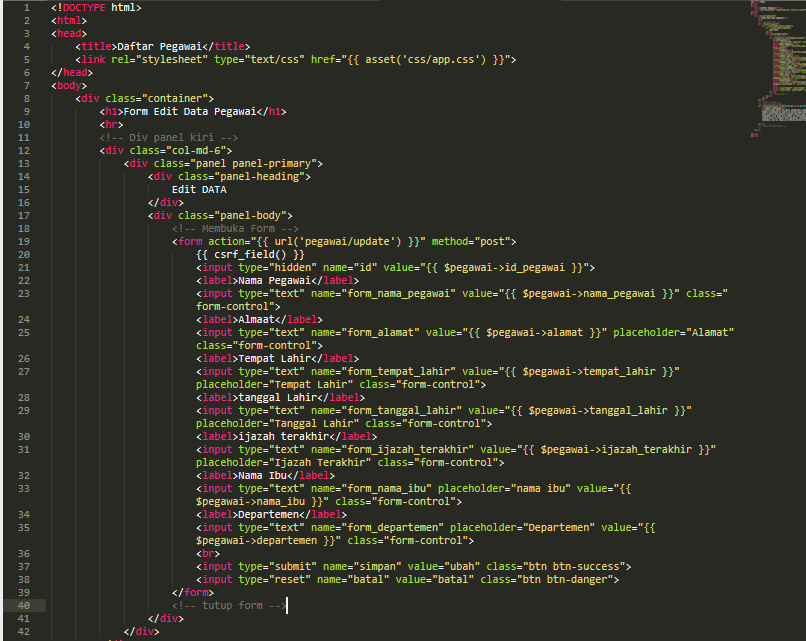
## Tambah Function Simpan di PegawaiController

Buka file **PegawaiController** di dalam folder **app/Http/Controllers**. Lalu tambahkan scrypt seperti di gambar:

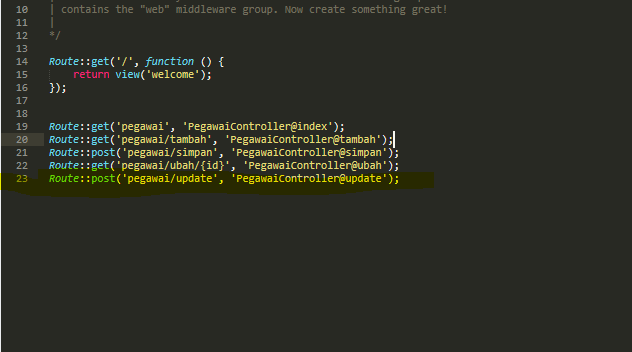


## Membuat Form Edit Pegawai

Klik kanan pada folder **views**, di dalam folder **resources,** pilih new file, lalu simpan dengan nama **edit-pegawai.blade.php** kemudian isi seperti gambar berikut:

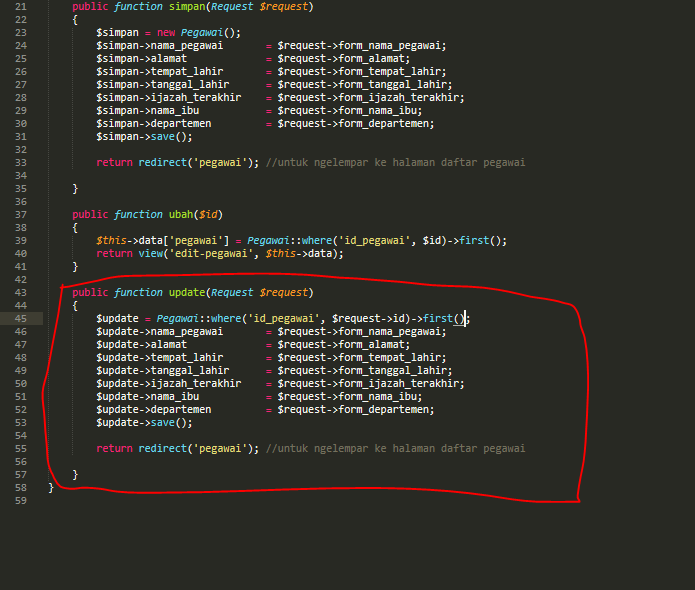


Selanjutya daftarkan juga URL untuk update data:



## Menambahkan Function Update di dalam PegawaiController

Tambahkan scrypt seperti di gambar ke dalam PegawaiController:

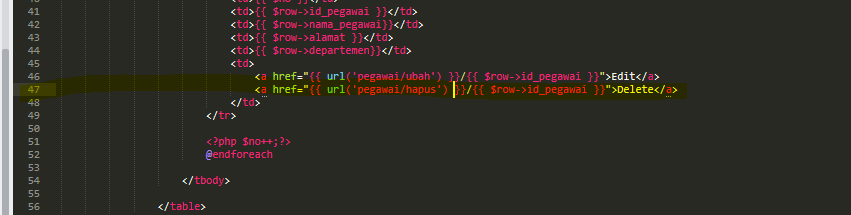


Kemudian jalankan, klik tombol edit di salah satu data pada halaman pegawai, lalu ubah.

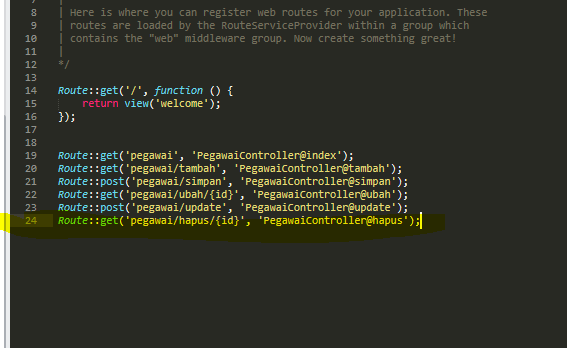
# Menghapus Data

## Siapkan URL

Siapkan url yang akan kita gunakan untuk mengubah, sebelum itu, buka kembali file **daftar-pegawai.blade.php** lalu ubah pada bagian tombol hapus



Selanjutnya daftarkan url tersebut ke dalam **web.php** di dalam folder **Routes**



## Menambhakan Functin Hapus pada PegawaiController

Buka controller PegawaiController dan tambahkan fungsi berikut:



Simpan kemudian jalankan.

Hapus salah satu data

selesai