|  |
| --- |
| **­­­­­­­­LAPORAN PRAKTIKUM PENGENALAN APLIKASI BERBASIS INTERNET**  **CRUD LARAVEL** |
| ­­­­ |
| **Agus Pranata Marpaung**  **13323033**  **DIII TEKNOLOGI KOMPUTER** |
| **INSTITUT TEKNOLOGI DEL**  **FAKULTAS VOKASI** |

**Judul Praktikum**

|  |  |  |
| --- | --- | --- |
| **Minggu/Sesi** | : | XII/3 |
| **Kode Mata Kuliah** | : | 1331204 |
| **Nama Mata Kuliah** | : | PENGENALAN APLIKASI BERBASIS INTERNET |
| **Setoran** | : | *Softcopy* |
| **Batas Waktu Setoran** | : | *23 April 2024 jam 18:00* |
| **Tujuan** | : | 1. *Mahasiswa dapat melakukan praktikum mengenai CRUD pada laravel 11* |

# Framework Laravel

Laravel merupakan salah satu framework Laravel yang bersifat opensource. Laravel dirancang agar proses pengembangan aplikasi web menjadi lebih mudah dan cepat karena berbagai fitur bawaan didalamnya. Fitur-fitur tersebut merupakan salah satu alasan mengapa banyak pengembangan memanfaatkan kerangka kerja Laravel yang menggunakan konsep Model- ViewController (MVC). Laravel berada di bawah lisensi MIT, dengan menggunakan GitHub sebagai tempat berbagi kode. Pada saat ini pengembangan Laravel berada di versi 11 dan baru dirilis pada 12 Maret 2024 dan hanya mendukung versi php dengan minimum 8.2 ke atas.

Dengan menggunakan pattern MVC ini, struktur kode yang dihasilkan menjadi lebih terstruktur dan memiliki standar yang jelas dengan memisahkan bisnis logic (alur pikir), data logic (penyimpanan data) dan presentation logic (antarmuka aplikasi) atau secara sederhana adalah memisahkan antara desain, data dan proses dimana komponen dari MVC antara lain adalah:

1. Model

Model berhubungan dengan data dan interaksi ke database atau webservice. Model juga merepresentasikan struktur data dari aplikasi yang bisa berupa basis data maupun data lain, misalnya dalam bentuk file teks, file XML maupun webservice. Biasanya di dalam model akan berisi class dan fungsi untuk mengambil, melakukan update dan menghapus data website. Sebuah aplikasi web biasanya menggunakan basis data dalam menyimpan data, maka pada bagian Model biasanya akan berhubungan dengan perintah-perintah query SQL

1. View

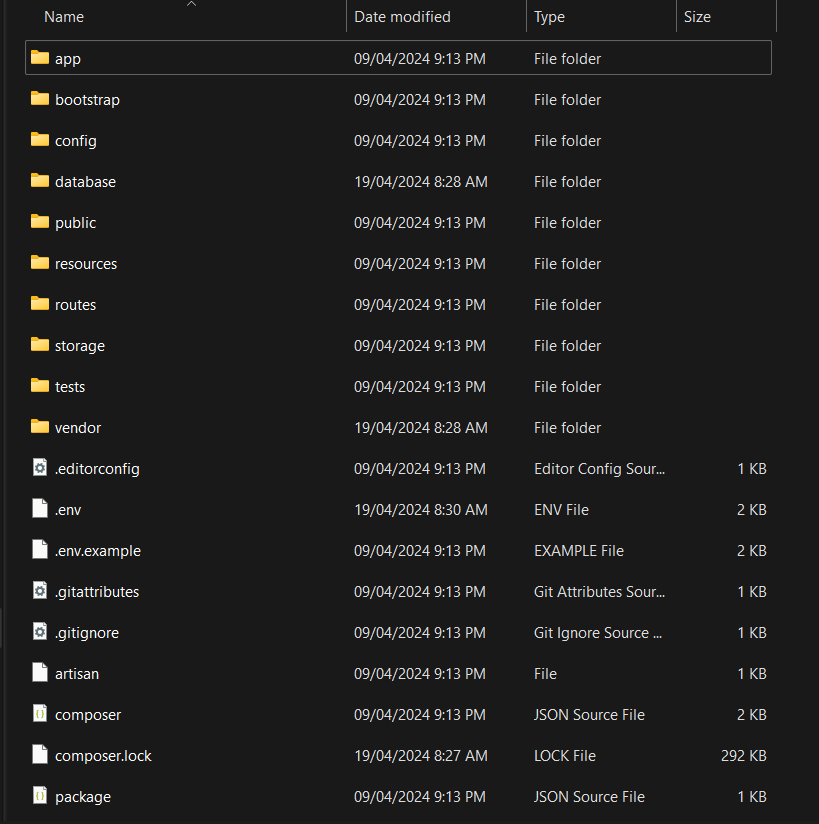
View berhubungan dengan segala sesuatu yang akan ditampilkan ke end-user. Bisa berupa halaman web, rss, javascript dan lain-lain. Kita harus menghindari adanya logika atau pemrosesan data di view. Di dalam view hanya berisi variabel-variabel yang berisi data yang siap ditampilkan. View dapat dikatakan sebagai halaman website yang dibuat dengan menggunakan HTML dan bantuan CSS atau JavaScript. Di dalam view jangan pernah ada kode untuk melakukan koneksi ke basisdata. View hanya dikhususkan untuk menampilkan data-data hasil dari model dan controller.

1. Controller

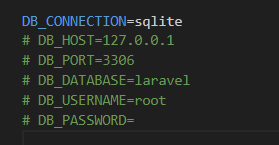
Controller bertindak sebagai penghubung data dan view. Di dalam Controller inilah terdapat class- class dan fungsi-fungsi yang memproses permintaan dari View ke dalam struktur data di dalam Model. Controller juga tidak boleh berisi kode untuk mengakses basis data karena tugas mengakses data telah diserahkan kepada model. Tugas controller adalah menyediakan berbagai variabel yang akan ditampilkan di view, memanggil model untuk melakukan akses ke basis data, menyediakan penanganan kesalahan/error, mengerjakan proses logika dari aplikasi serta melakukan validasi atau cek terhadap input.

# Membuat CRUD dengan Framework Laravel

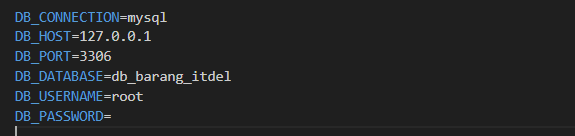
1. Buatlah folder baru dengan nama **Laravel,** Kemudian lakukan instalasi laravel dengan nama project **Barang** dengan menggunakan composer. Jika instalasi telah selesai, maka folder yang akan dihasilkan akan seperti berikut:



1. Kemudian aktifkan **Apache** dan **MySQL** melalui XAMPP.
2. Lakukan konfigurasi koneksi antara **MySQL** dengan laravel 11. Silahkan Anda membuka file **.env** lalu cari code seperti dibawah ini.



Setelah Anda menemukan codenya, silahkan Anda ubah code menjadi seperti code dibawah ini



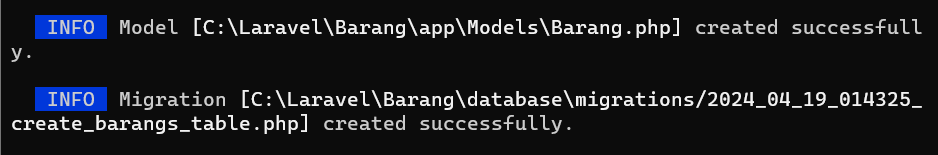
Pada code diatas kita pertama kali mengubah konfigurasi DB\_CONNECTION yang semula menggunakan sqlite menjadi mysql. Lalu kita hapus semua tanda # pada konfigurasi key agar variable nya diaktifkan. Lalu untuk DB\_DATABASE kita atur sesuai dengan database yang kita buat, disini kita buat nama databasenya yaitu db\_barang\_itdel.

1. Selanjutnya buat model dan migration pada project dengan menjalankan perintal dibawah di dalam terminal/CMD Anda.

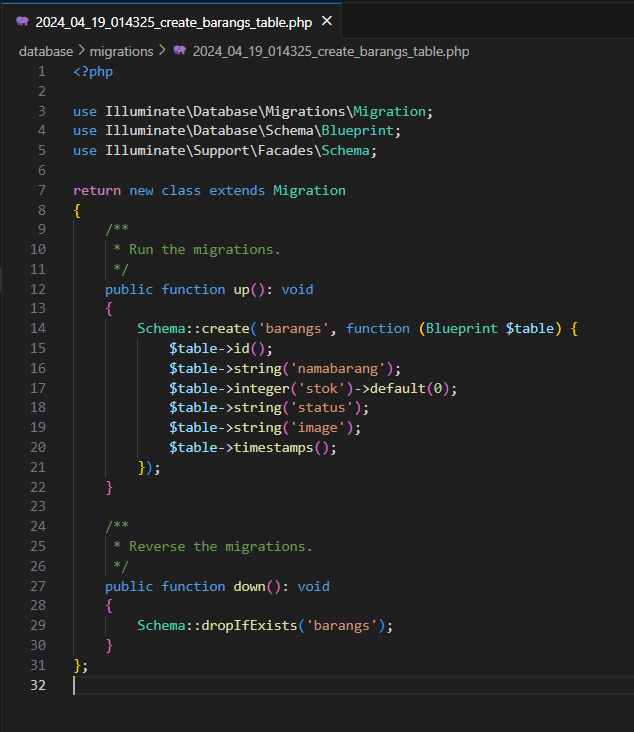


Jika berhasil maka pada project Anda akan bertambah 2 file baru yang berada di dalam folder yaitu:

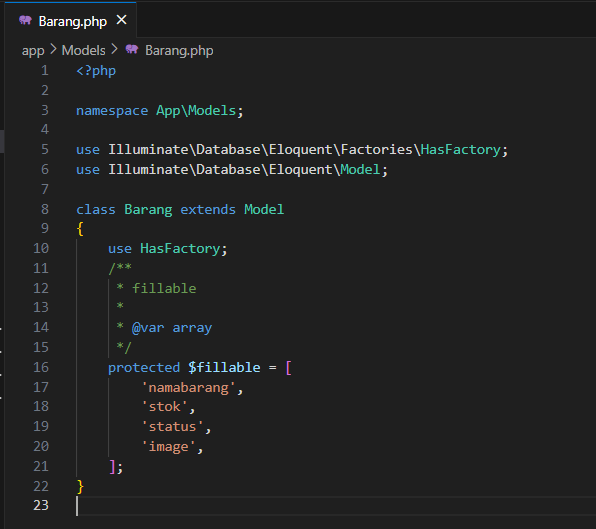
* app/Models/Barang.php
* database/migrations/2024\_04\_17\_025054\_create\_barangs\_table.php



1. Tambahkan field/kolom di dalam migration dengan membuka file migration lalu ketikkan code dibawah.



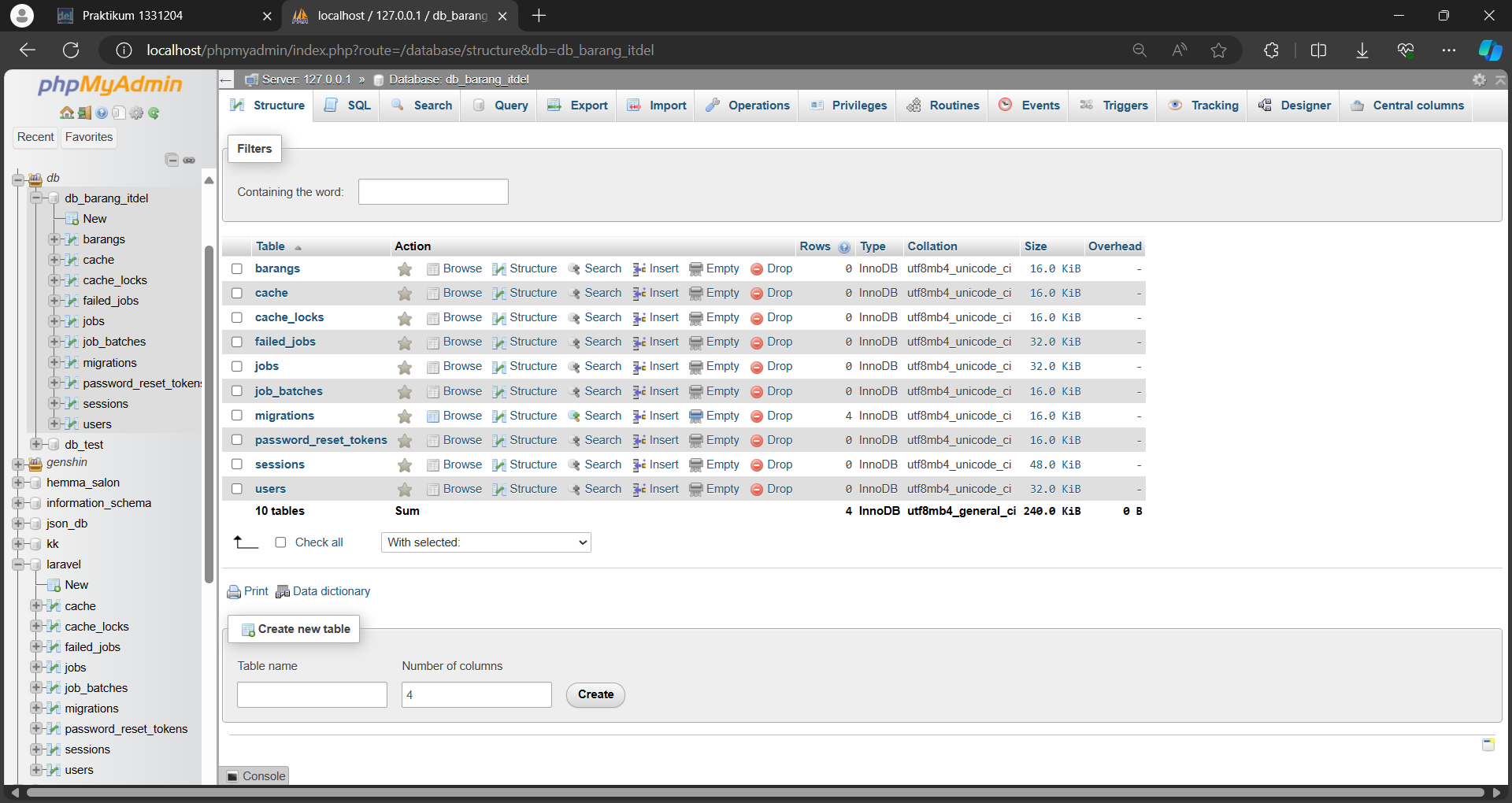
1. Tambahkan Mass Assignment agar field yang telah ditambahkan di dalam mingration dapat Melakukan manipulasi ke dalam database. Silahkan buka file Barang.php yang ada pada folder app/Models/Barang.php lalu ketikkan code berikut.



1. Jalankan proses migrate dalam project laravel, proses ini akan men-generate database dan table beserta field/kolom di dalam database. Silahkan jalankan perintah berikut di terminal/CMD Anda.



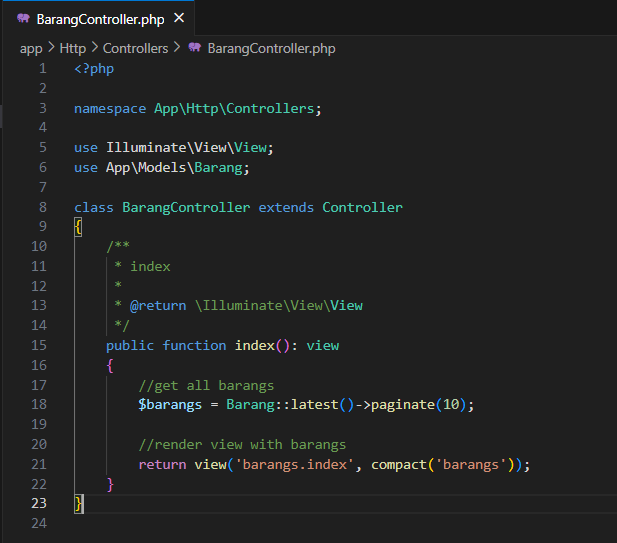
Jika keluar pertanyaan, silahkan pilih YES lalu enter dan tunggu prosesnya sampai selesai. Pastikan jaringan Anda tetap terhubung. Jika proses migrate telah selesai, silahkan untuk cek database pada PhpMyAdmin di [http://localhost/phpmyadmin.](http://localhost/phpmyadmin) Table-table serta field-field di dalamnya telah berhasil di generate seperti pada gambar.



1. Setelah berhasil migrate, Selanjutnnya membuat Controller Barang. Controller ini bertujuan untuk mengatur semua aksi-aksi di dalam project dan tentunya akan menghubungkan antara model, view dan juga route. Silahkan jalankan perintah di bawah pada terminal/CMD di dalam project Laravel.

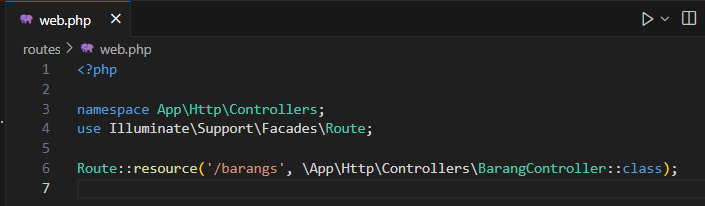


Jika perintah diatas berhasil dijalankan, maka pada project Anda akan bertambah 1 file controller baru yang berada di folder app/Http/Controllers/BarangController.php. Lalu silahkan Anda membuka BarangController yang baru saja dibuat dan ubah codenya seperti pada code dibawah.



Dari perubahan code diatas, kita Melakukan import Model **Barang** (use App\Models\Barang;) serta import return type **View**(use Illuminate\View\View;) dan menambahkan method baru yaitu index. Dalam method tersebut dilakukan pemanggilan data Barangs dari database melalui Model Barang ($barangs = Barang::latest()->paginate(10);). Jika data berhasil didapatkan dan ditampung di dalam variable $barangs lalu melakukan render view serta mengirim data Barang yang ada di dalam variable $barangs (return view('barangs.index', compact('barangs'));).

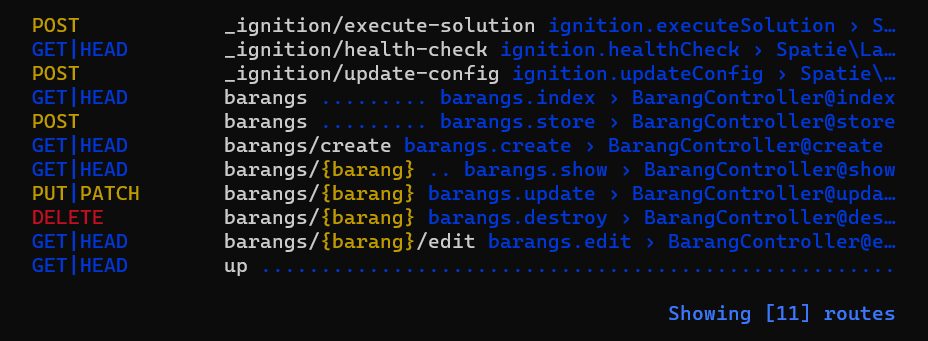
1. Selanjutnya membuat route untuk menghubungkan antara user dengan Aplikasi. Silahkan Anda buka file routes/web.php lalu ubah codenya menjadi code seperti dibawah.



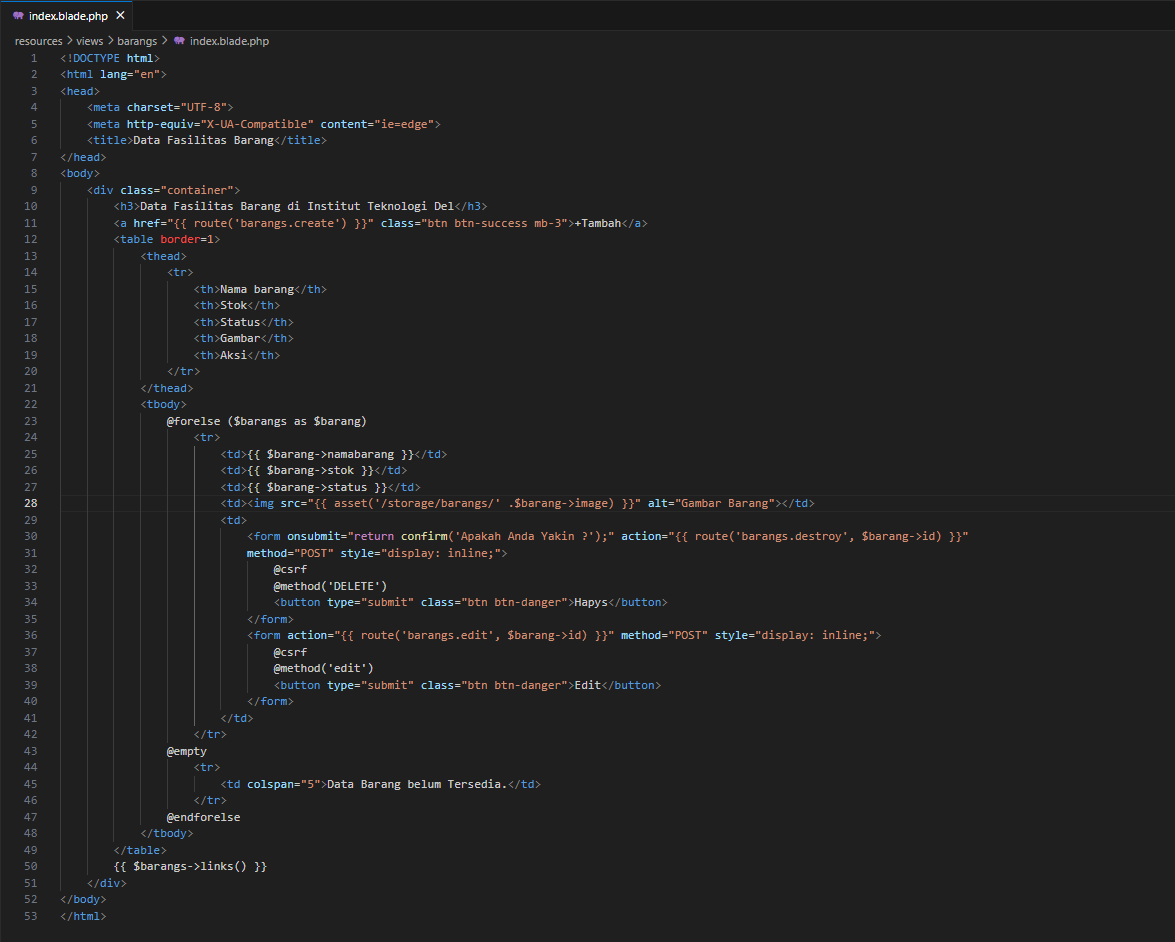
Mengapa route yang kita buat jenisnya resource? Agar laravel akan secara otomatis menghasilkan route-route untuk berbagai operasi CRUD seperti GET/ POST/ PUT/ PATCH/ DELETE. Untuk memastikan apakah route-route diatas telah di generate oleh laravel, silahkan jalankan perintah di bawah pada terminal/CMD project Anda dan tunggu sampai proses generate selesai.



Dan menghasilkan berikut:



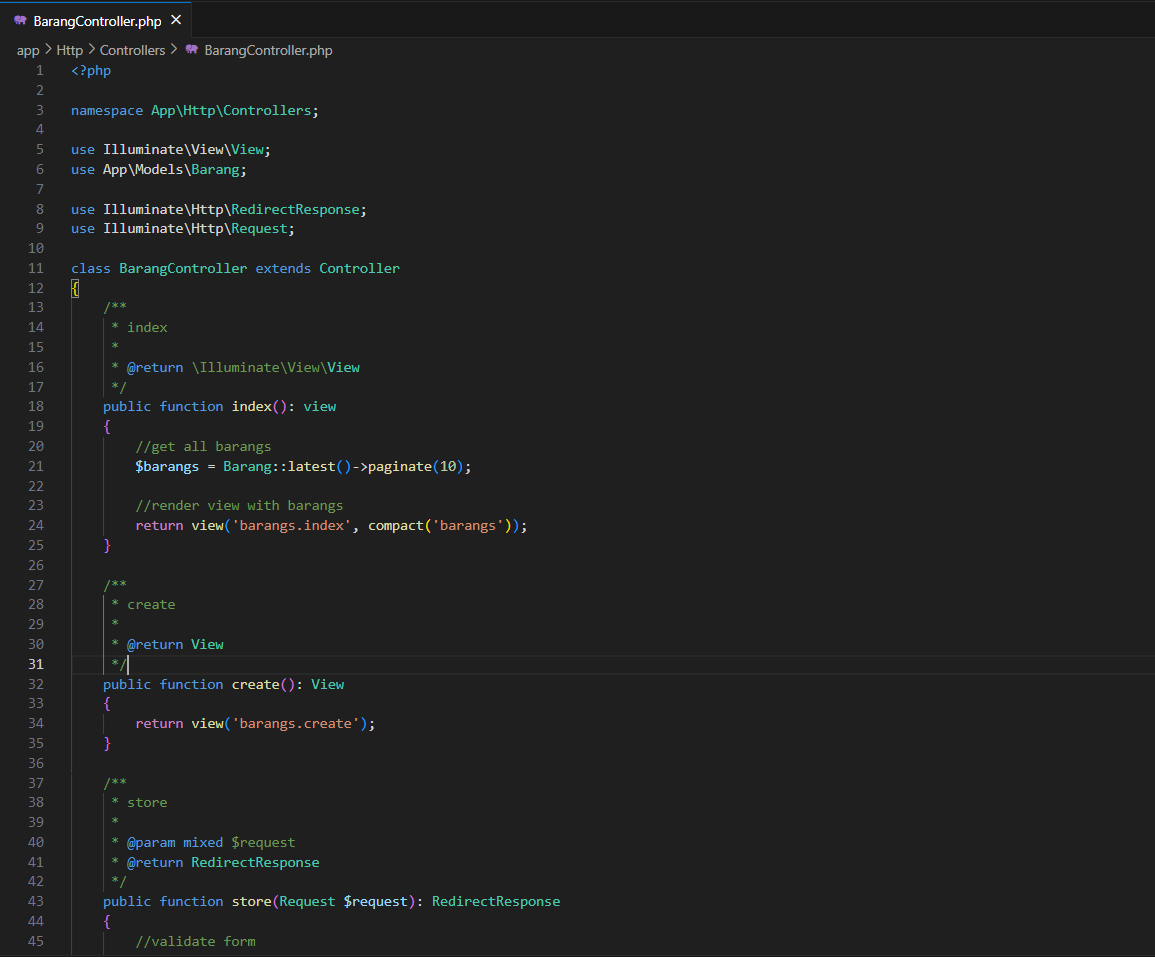
1. Setelah berhasil di generate, sekarang Anda membuat view untuk menampilkan Data. Silahkan Anda membuat folder baru dengan nama barangs di dalam folder resources/views. Dalam folder barangs silahkan buat file baru dengan nama index.blade.php, kemudian silahkan Anda ketikkan code dibawah ke dalam file index.

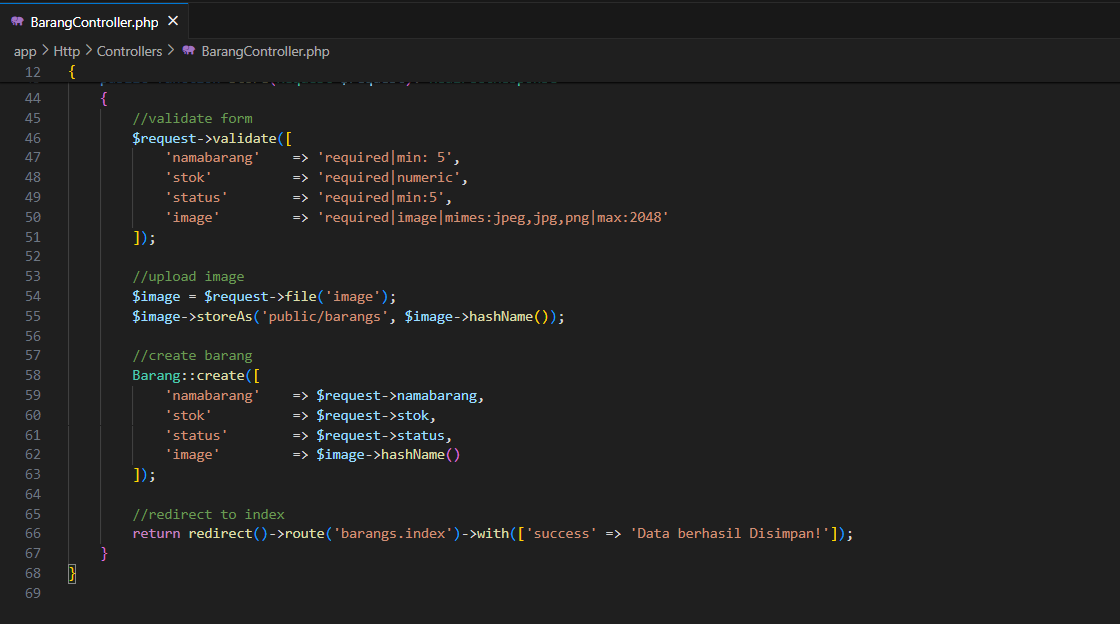


Selanjutnya coba jalankan project laravel yang telah Anda buat pada browser dengan mengakses <http://127.0.0.1:8000/barangs>maka tampilannya akan seperti di bawah.



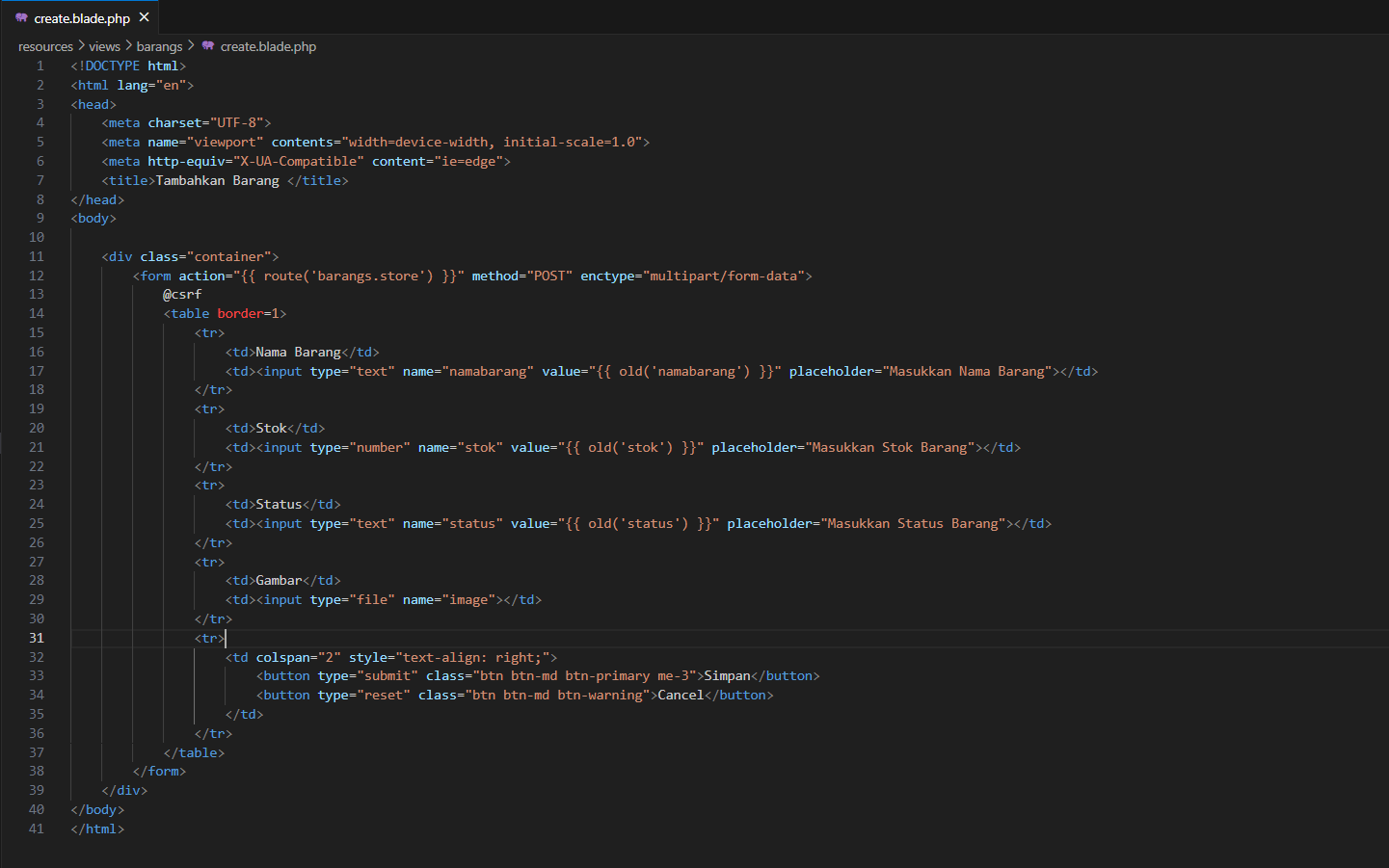
1. Setelah Anda dapat menampilkan Data pada project laravel Anda, selanjutkan kita akan membuat insert data ke dalam database. Langkah awal yang perlu dilakukan adalah menambahkan method create dan store ke dalam Controller. Method create ini digunakan untuk menampilkan halaman form tambah data sedangkan method store digunakan untuk insert data ke dalam database dan Melakukan upload gambar. Silahkan Anda buka kembali file BarangController.php lalu ubah codenya seperti pada code berikut.



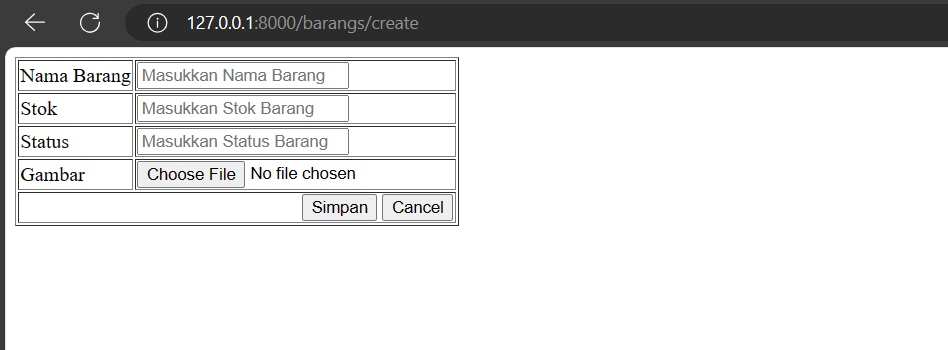


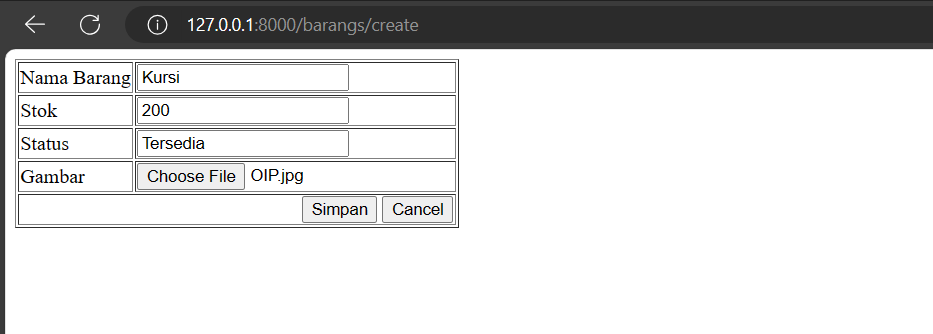
Dari code diatas, pertama kita import return type RedirectRespon dan Http Request pada line 8 dan 9. Dalam BarangController kita menambahkan 2 method baru yaitu function create(line 3235) dan function store(line 43-67).

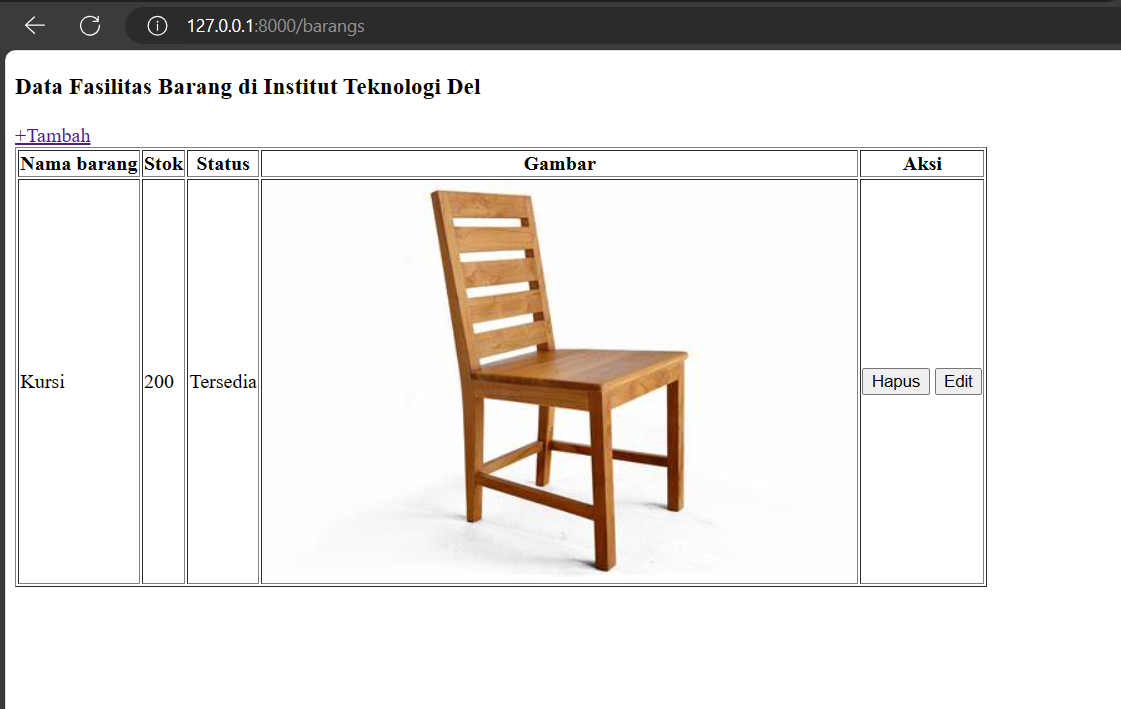
1. Silahkan mengetikkan code berikut untuk membuat view form create Barang. Silahkan buat terlebih dahulu file baru dengan penamaan create.blade.php di dalam folder resources/views/products.



1. Setelah selesai Anda ketik, silahkan jalankan project laravel Anda. Anda bisaklik button +Tambah pada halaman barangs index atau bisa juga mengakses link berikut pada browser Anda [http://127.0.0.1:8000/barangs/create.](http://127.0.0.1:8000/barangs/create) Berikut tampilan setelah Anda menjalankan project laravelnya.





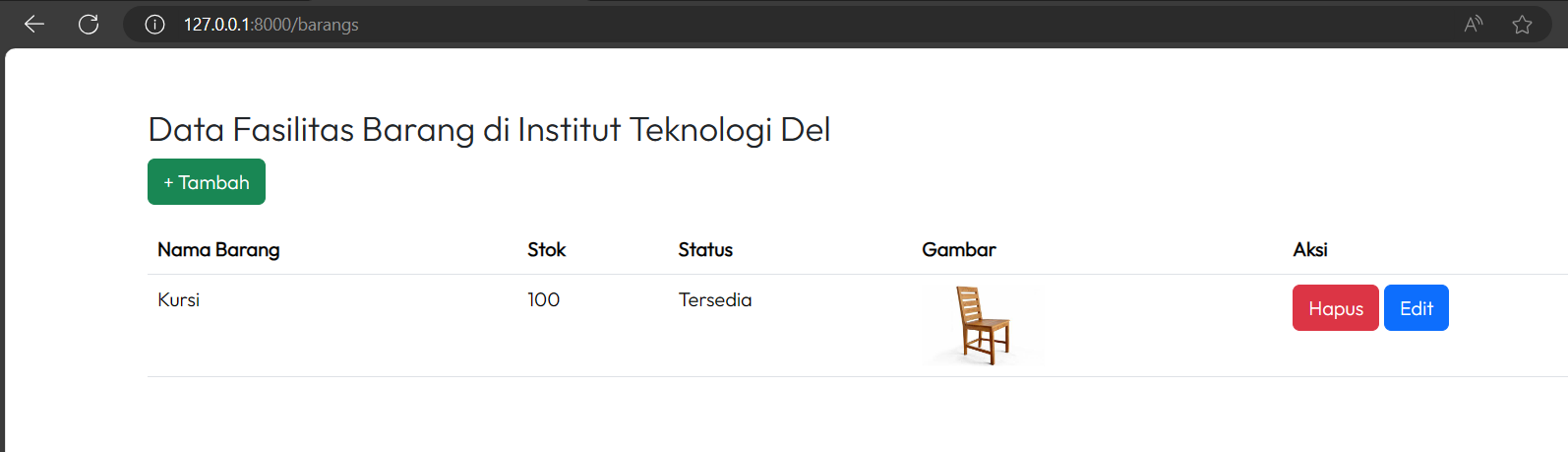


Tugas:

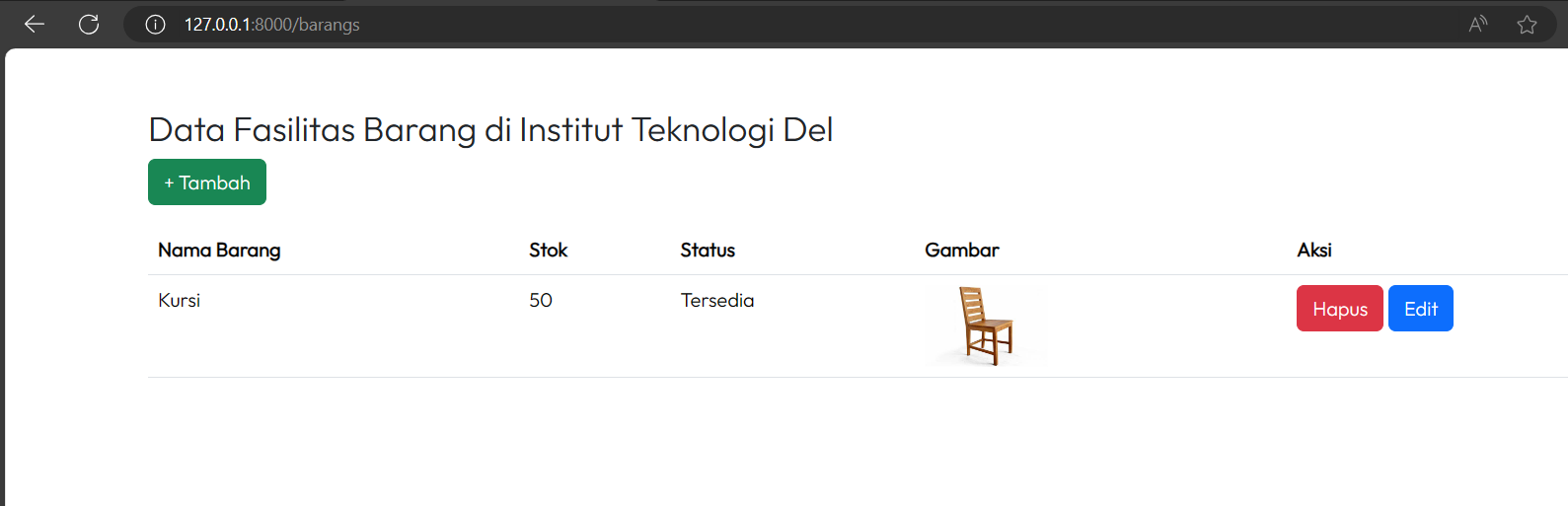
1. Silahkan Anda buatkan untuk Update dan Delete pada project laravel diatas.

**Jawab:**

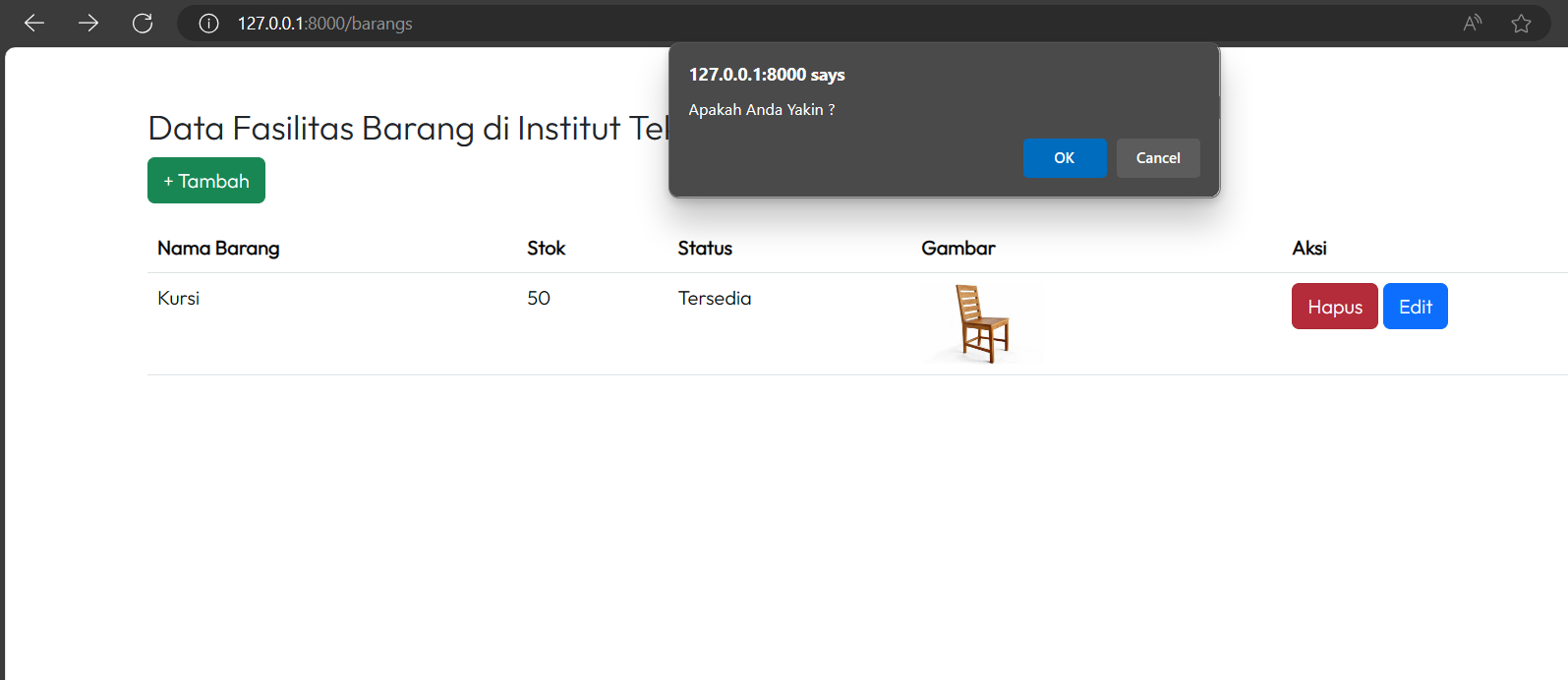
* **Update**

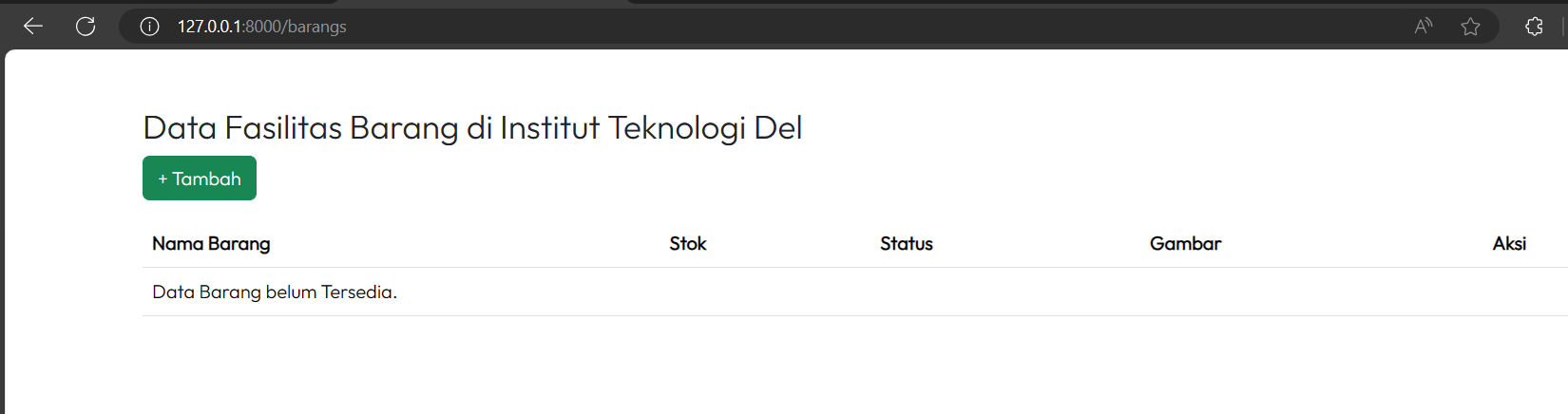
****

****

****

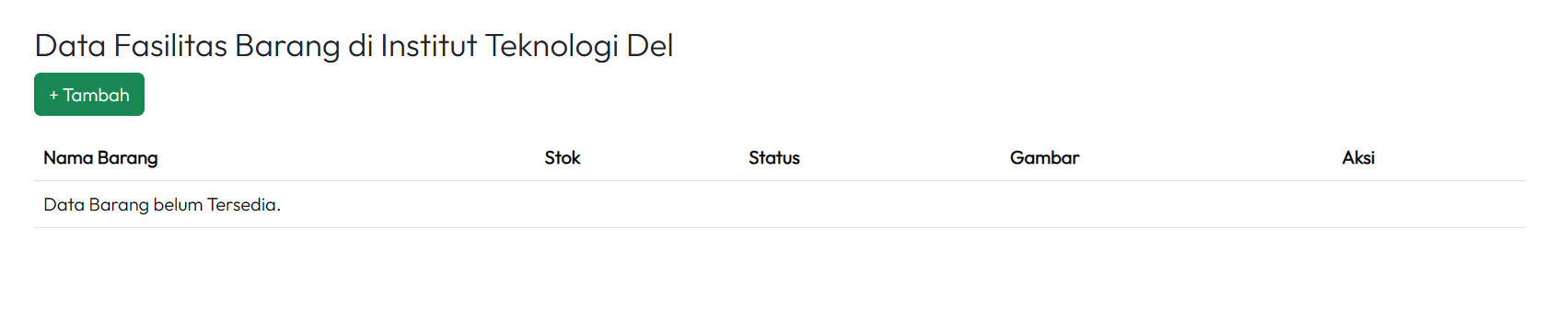
* **Delete**

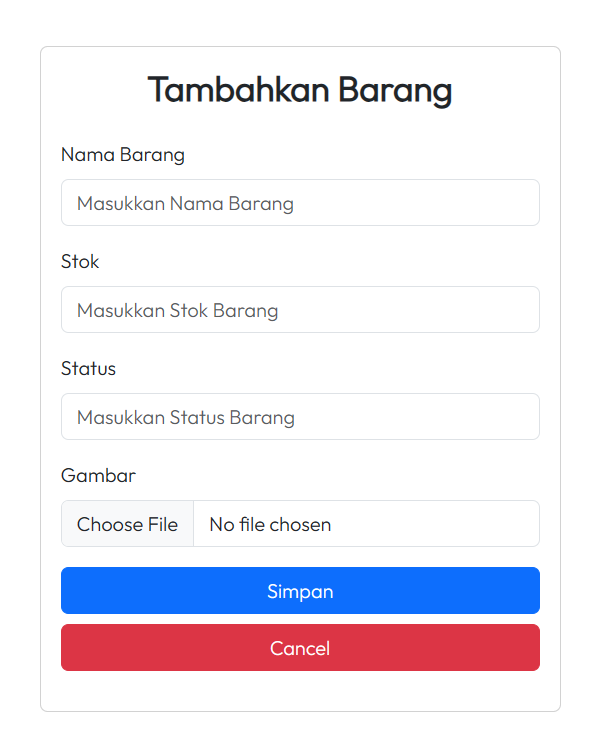
****

****

1. Silahkan Anda tambahkan script css pada project laravel Anda.

**Jawab:**

****

****

****

Praktikum sesi 3 dibuatkan ke dalam Laporan dalam bentuk pdf dengan penamaan CRUD\_Laravel\_NimLengkap.pdf. Tugas akan di submit ke ecourse dengan batas waktu Minggu, 21 April 2024 Pukul 22:00 WIB.