**LAPORAN BAB 6**

**PEMROGRAMAN WEB LANJUT**

Dosen Pengampu :

Lintang Setyo Kurniawati S.Kom, M.Pd



DISUSUN OLEH :

BAYU KRISNA (2010651095)

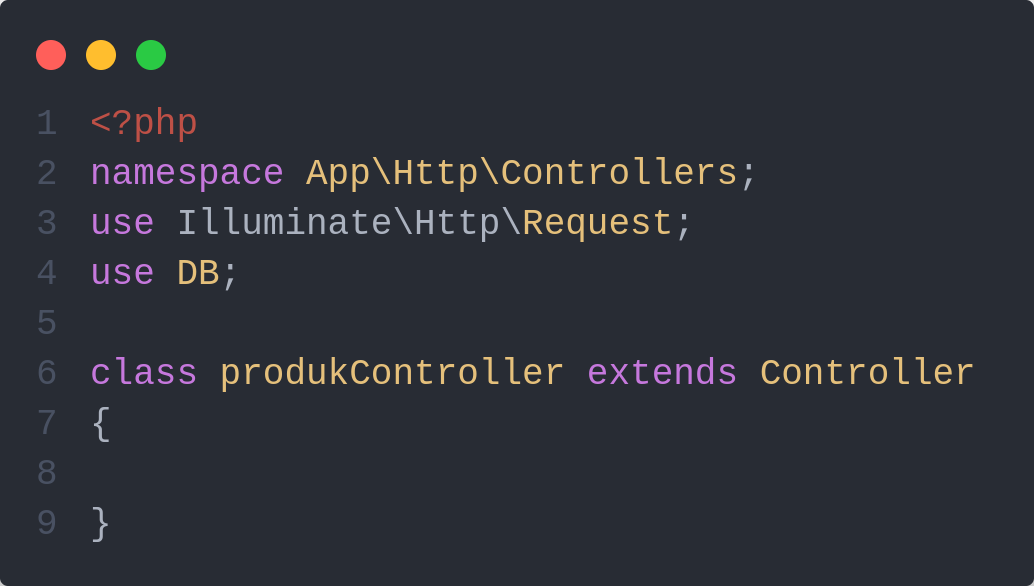
**TEKNIK INFORMATIKA**

**UNIVERSITAS MUHAMMADIYAH JEMBER**

**2022**

**INTERAKSI DATABASE DENGAH QUERY BUILDER**

Query Builder merupakan salah satu cara untuk menjalankan query database dengan lebih mudah, Query Builder juga telah dilengkapi dengan fitur keamanan untuk mencegah terjadinya SQL Injextion (adalah sebuah aksi hacking yang dilakukan di aplikasi client dengan cara memodifikasi perintah SQL yang ada di memori aplikasi client). Selain itu kita dapat menggunakan query builder tanpa harus membuat model terlebih dahulu. Sricpt query bulder cukup mudah dipahami, misalnya untuk menampilkan data dari tabel produk yang dibuat sebelumnya pertama-tama kita harus mendefinisikan penggunaan class dari QUERY BUILDER dengan cara menambahkan script use DB; pada bagian atas controller atau seperti contoh dibawah:



Dengan pendefinisian mengunakan script use DB; kita telah menambahkan class QUERY BUILDER ke dalam controller produkController. Lalu untuk menampilkan data dari tabel produks yang telah dibuat sebelumnya bisa menggunakan script berikut:

DB::table(‘produks’)->get();

KEGIATAB PRAKTIKUM 1 MENAMPILKAN DATA DENGAN QUERY BUILDER

Untuk mempraktekan contoh dari latihan menampilkan data dengan QUERY BUILDER pertama kita harus memeriksa file .env apakah projek sudah terhubung dengan database atau belum. File ini terletak pada bagian luar projek laravel yang dibuat, dan pastikan pada bagian mysql sudah terkonfigurasi seperti pada gambar dibawah:

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

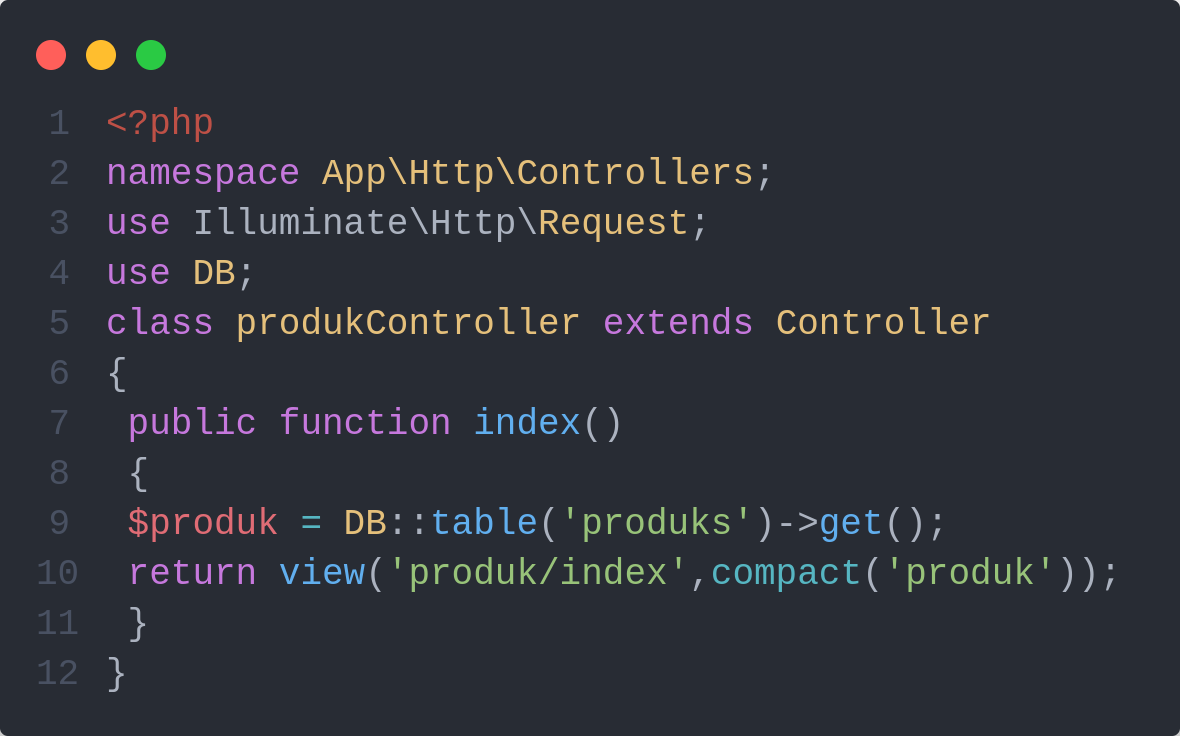
DB\_PORT=3306

DB\_DATABASE=penjualan

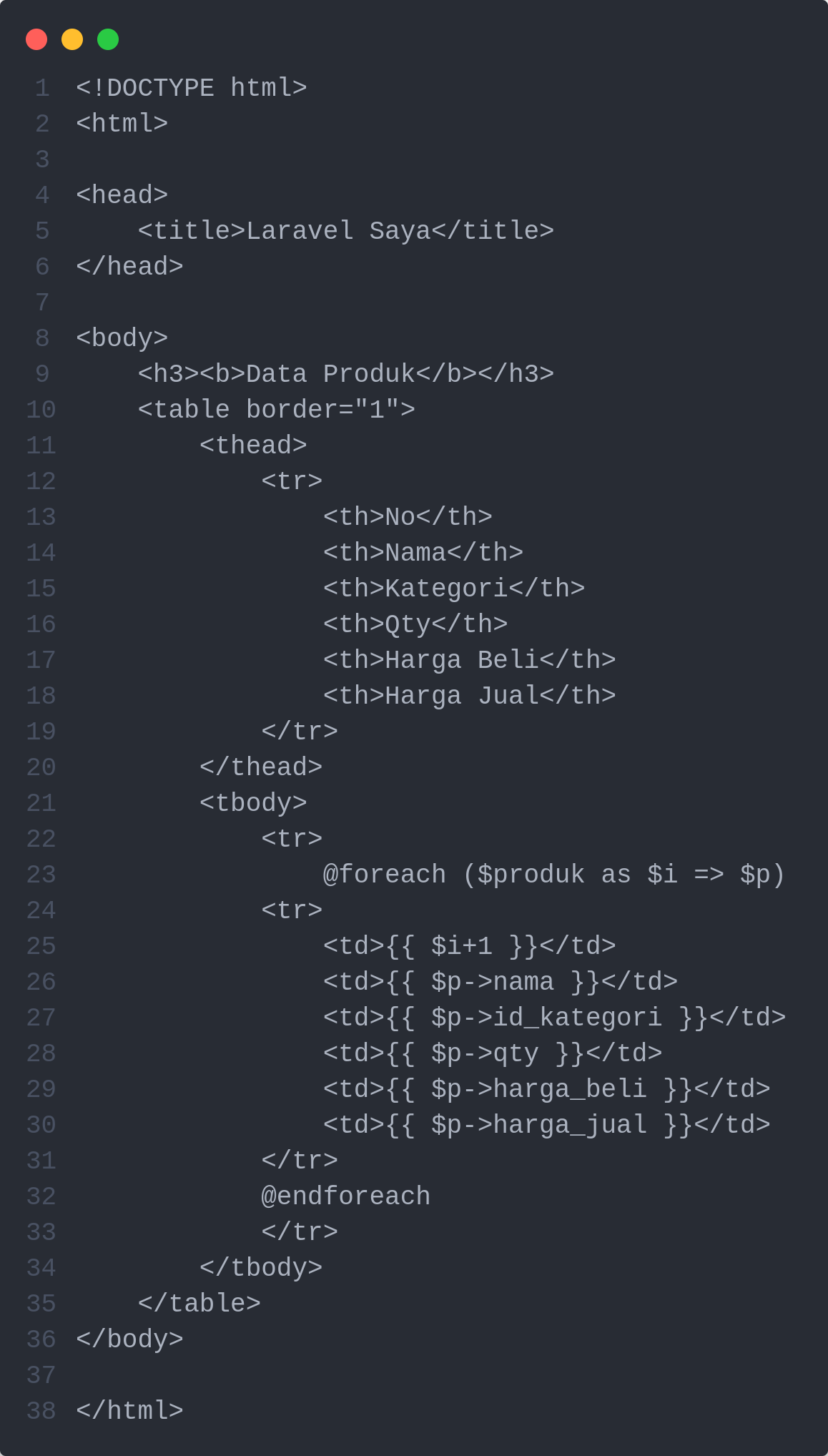
DB\_USERNAME=root

DB\_PASSWORD=

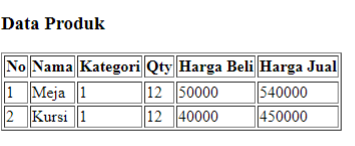
Database yang akan kita gunakan adalah database dengan nama penjualan, yang telah kita buat pada latihan pembuatan migration sebelumnya. Kemudian pada file produkController.php yang berada pada folder app\Http\Controller telah dibuat pada latihan sebelumnya buatlah satu fungsi yang bernama index() atau modifikasi jika sudah ada dan tambahkan script sebagai berikut:



Fungsi index() diatas akan mengambil semua data yang ada pada tabel produk dan kemudian akan mengirimkannya ke view index.blade.php yang ada di dalam folder resources/views/produk. Jadi pada view index.blade.php yang ada pada folder resourcess/views/produk (jika belum ada silahkan dibuat) modifikasi script nya menjadi seperti dibawah:



Sebelum mencoba menjalankan skript diatas pertama-tama cobalah untuk memeriksa route yang ada pada file web.php yang ada pada folder routes/web.php dan periksa apakah route Route::get('/produk', 'produkController@index'); sudah didefinisikan atau belum. Jika sudah maka aktifkan server artisan pada cmd lalu akses http://localhost:8000/produk maka hasilnya akan nampak seperti pada gambar dibawah:



**KEGIATAN PRAKTIKUM 2 JOIN TABEL DENGAN QUERY BUILDER**

Jika diperhatikan pada tampilan data produk pada tabel diatas pada kolom id\_kateogri kita masih menampilkan angka atau bukan data dari suatu kategori produk. Untuk memperbaikinya buatlah satu tabel baru dengan nama kateogri menggunakan fasilitas migration pada Laravel dengan tahapan sebagai berikut:

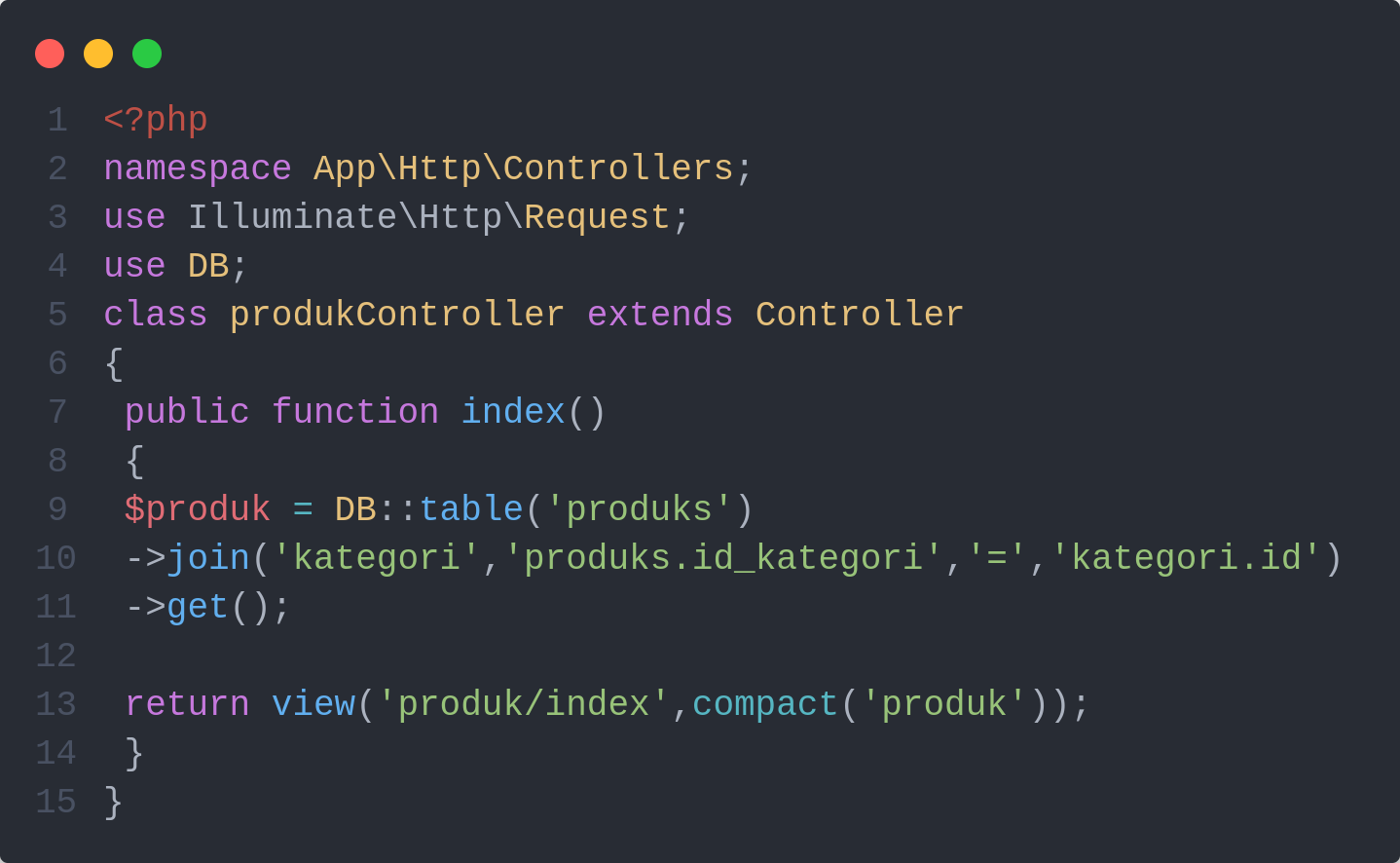
1. Jalankan perintah artisan berikut pada cmd atau comand promt yang sudah terarah ke dalam directori projek laravel yang digunakan
2. Langkah kedua bukalah file 2018\_09\_14\_015345\_create\_kategori\_table.php yang ada folder database/migration dan modifikasi koding di dalamnya menjadi seperti dibawah:



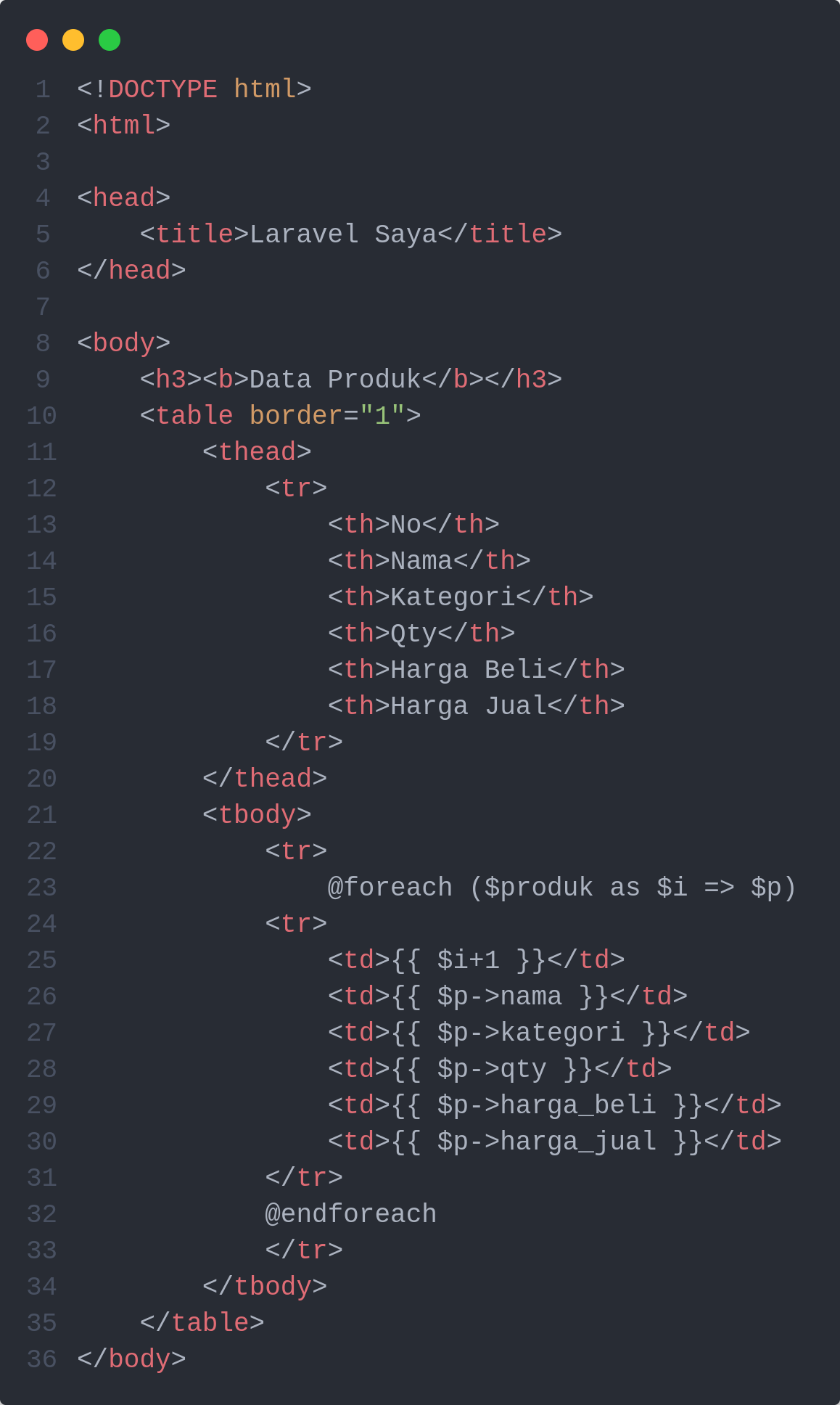
4. Jalankan perintah *php artisan migrate* dan tabel kan beruba seperti ini



Setelah membuat tabel kategori beserta dengan 1 contoh datanya sekarang bukalah class Controller produkController.php yang ada pada folder app\Http\Controller dan modifikasi fungsi index() yang ada di dalam controller tersebut menjadi seperti dibawah:



Kemudian pada view index.blade.php yang ada pada folder resources/views/produk/ modifikasi kodingnya menjadi seperti dibawah:



**MENAMBAH DATA DENGAN QUERY BUILDER**

Untuk menambahkan data menggunakan Query Builder buatlah satu route baru pada file web.php yang terletak pada folder routes/web.php dengan script sebagai berukut:

Route::get('/produk/store', 'produkController@store');

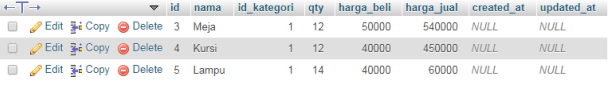
Pada route diatas akan menjalankan fungsi store yang ada di dalam class Controller produkController.php, untuk itu bukalah file produkController yang ada pada folder App/Http/Controller dan tambahkan satu fungsi yang bernama store() dan isikan script sebagai berikut



Pada script diatas adalah script untuk menambah data ke tabel yang ada di dalam database, pada kasus diatas tabel yang akan ditambah datanya adalah tabel produks. Laravel sendiri menggunakan array assosiatif untuk parameter kolom dan nilai yang akan ditambah dengan ketentuan index dari array akan menjadi nama kolom dan nilai dari array akan menjadi nilai yang akan disimapn ke kolom. Jika kita menjalankan localhost:8000/produk/store pada browser akan muncul tampilan seperti berikut:



Dan bila kita memerika tabel produk yang ada di dalam database penjualan maka akan ada satu data baru seperti pada gambar dibawah:



**MEMPERBAHRUI DATA QUERY BUILDER**

Untuk memperbaharui (update) data, kita harus menentukan terlebih dahulu data mana yang akan di update. Pada contoh ini kita akan memperbaharui data yang baru saja ditambahkan yaitu data dengan id 5, pertama buatlah satu route baru di fiile web.php di folder routes/web.php dengan skrip sebagai berikut:

Route::get('/produk/update', 'produkController@update');

Pada route diatas jika kita membuka localhost:8000/produk/update maka kita akan dibawa ke function update() yang ada pada file produkController.php, maka dari itu buatlah satu function baru bernama update di file produkController.php yang ada pada folder App/Http/Controller dengan script sebagai berikut:



Pada script diatas merupakan script untuk memperbaharui data menggunakan Query Builder. Pertama kita harus menentukan data mana yang akan diperbaharui dengan menggunakan fungsi where() dan diikuti dengan fungsi update() untuk memperbaharui data yang ada pada kolom. Dan jika dilihat pada tabel produk yang ada pada database penjualan data dengan id 3 akan berubah

**MENGHAPUS DATA QUERY BUILDER**

Untuk menghapus data kita perlu menentukan terlebih dahulu data mana yang akan dihapus lalu diikuti dengan fungsi delete(). Pertama buatlah route seperti berikut pada file web.php yang ada pada folder routes:

Route::get('/produk/delete', 'produkController@delete');

Pada route diatas jika kita mengakses localhost:8000/produk/delete pada browser maka kita akan diarahkan ke fungsi delete yang ada pada file produkController, maka dari itu buatlah fungsi yang bernama delete pada file produkController.php dan isikan script sebagai berikut



Pada script diatas adalah script untuk menghapus data menggunakan Query Builder. Pertama kita harus menentukan terlebih dahulu data mana yang akan dihapus menggunakan fungsi where() untuk menghapus data dengan syarat tertentu. Lalu diikuti dengan fungsi delete() untuk menghapus data.