Nama : Akmal Muhamad Firdaus

NIM : 1301204188

**Praktikum ABP Modul 6**

**Pendahuluan**

1. **Framework & MVC**

Framework adalah kumpulan dari aturan, standar, dan konvensi yang membantu dalam mengembangkan aplikasi web dengan lebih mudah dan efisien. Framework menyediakan kerangka kerja yang telah disediakan sebelumnya untuk memudahkan pengembangan aplikasi, sehingga pengembang dapat fokus pada masalah bisnis yang spesifik.

MVC (Model-View-Controller) adalah pola arsitektur yang sering digunakan dalam pengembangan aplikasi web. MVC memecah aplikasi menjadi tiga komponen utama:

* 1. Model - Merupakan bagian dari aplikasi yang bertanggung jawab untuk mengelola data dan aturan bisnis aplikasi.
  2. View - Merupakan bagian dari aplikasi yang bertanggung jawab untuk menampilkan data kepada pengguna akhir melalui antarmuka pengguna.
  3. Controller - Merupakan bagian dari aplikasi yang bertanggung jawab untuk menangani permintaan pengguna dan menghubungkan Model dan View.

Dengan menggunakan pola arsitektur MVC, pengembang dapat memisahkan tugas-tugas dalam pengembangan aplikasi web dan meningkatkan kemampuan aplikasi untuk diuji dan dikelola dengan lebih efisien. Banyak framework PHP yang mendukung pola arsitektur MVC, contohnya Laravel.

1. **Pengenalan Laravel**

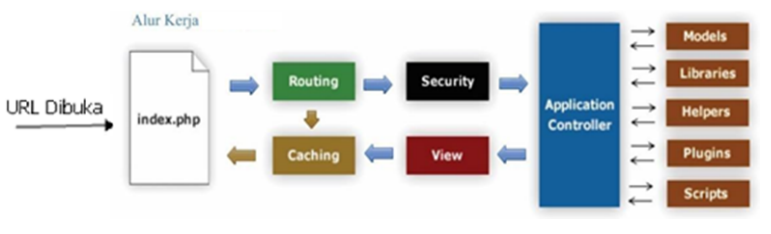
Laravel adalah sebuah framework PHP yang open source dan digunakan untuk membangun aplikasi web. Framework ini dikembangkan oleh Taylor Otwell pada tahun 2011 dan telah menjadi salah satu framework PHP yang paling populer digunakan oleh pengembang web.

Laravel didesain untuk memudahkan pengembangan aplikasi web dengan menggunakan pola arsitektur Model-View-Controller (MVC). Framework ini menyediakan banyak fitur yang membantu pengembang untuk membangun aplikasi web yang kompleks dengan cepat dan efisien, termasuk fitur-fitur seperti routing, migrasi database, ORM, templating engine, dan banyak lagi.

Beberapa kelebihan dari Laravel antara lain:

* Mudah dipelajari dan digunakan oleh pengembang web pemula maupun berpengalaman.
* Free, karena berada di bawah lisensi open source, kita dapat melakukan apa pun
* Memiliki dokumentasi yang lengkap dan dukungan komunitas yang besar.
* Memiliki sistem routing yang mudah digunakan dan dapat disesuaikan dengan kebutuhan aplikasi.
* Memiliki fitur bawaan yang kuat untuk otentikasi dan otorisasi pengguna.
* Memiliki sistem templating yang kuat untuk membangun tampilan aplikasi web.
* Mendukung banyak database management system (DBMS) seperti MySQL, PostgreSQL, dan MongoDB.

1. **Cara Kerja Laravel**



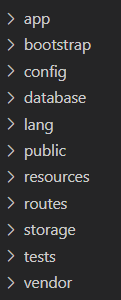
Soruce: Modul Praktikum ABP

Ketika suatu URL diakses, Laravel akan membaca file index.php dan memeriksa apakah URL tersebut terdaftar pada routing yang sudah dibuat. Jika URL tersebut terdaftar pada routing dan memenuhi persyaratan keamanan (middleware), seperti autentikasi **jika diperlukan**, maka Laravel akan memanggil controller yang sesuai dengan routing tersebut. Di dalam controller, semua logika bisnis yang diperlukan akan dieksekusi dan dapat mengembalikan view ke browser. Tampilan tersebut akan disimpan dalam cache agar pemrosesan selanjutnya lebih cepat.

Laravel memiliki banyak fitur bawaan yang memudahkan pengembang untuk membangun aplikasi web dengan lebih efisien, seperti sistem autentikasi dan otorisasi, migrasi database, dan masih banyak lagi.

Selain itu, Laravel juga memiliki fitur composer, yang merupakan manajer paket pada PHP. Fitur ini memudahkan pengembang untuk mengelola dependensi pada aplikasi web, seperti memasang library dan package pihak ketiga.

Secara keseluruhan, Laravel bekerja dengan menggabungkan pola arsitektur MVC, routing, dan fitur-fitur bawaan yang efisien dan mudah digunakan oleh pengembang web PHP.



Struktur Folder Laravel

1. app/

Folder ini berisi kode aplikasi Laravel, seperti model, controller, middleware, dan provider.

2. bootstrap/

Folder ini berisi file yang digunakan untuk memulai aplikasi Laravel, seperti file app.php dan autoload.php.

3. config/

Folder ini berisi konfigurasi untuk aplikasi Laravel, seperti konfigurasi database, cache, session, dan lain-lain.

4. database/

Folder ini berisi migrasi dan seed untuk database aplikasi Laravel. Migrasi digunakan untuk membuat struktur tabel database, sedangkan seed digunakan untuk memasukkan data awal ke dalam database.

5. lang/

Laravel adalah framework yang support localization (multi bahasa), untuk menerapkan berbagai bahasa, dapat disimpan pada folder lang ini.

6. public/

Folder ini berisi file yang dapat diakses secara publik melalui browser, seperti file index.php dan file gambar.

7. resources/

Folder ini berisi aset-aset yang digunakan dalam aplikasi Laravel, seperti file tampilan (view), file bahasa, dan file konfigurasi lainnya.

8. storage/

Folder ini berisi file sementara dan file yang digunakan oleh Laravel, seperti file log dan file cache.

9. tests/

Folder ini berisi file untuk melakukan pengujian aplikasi Laravel.

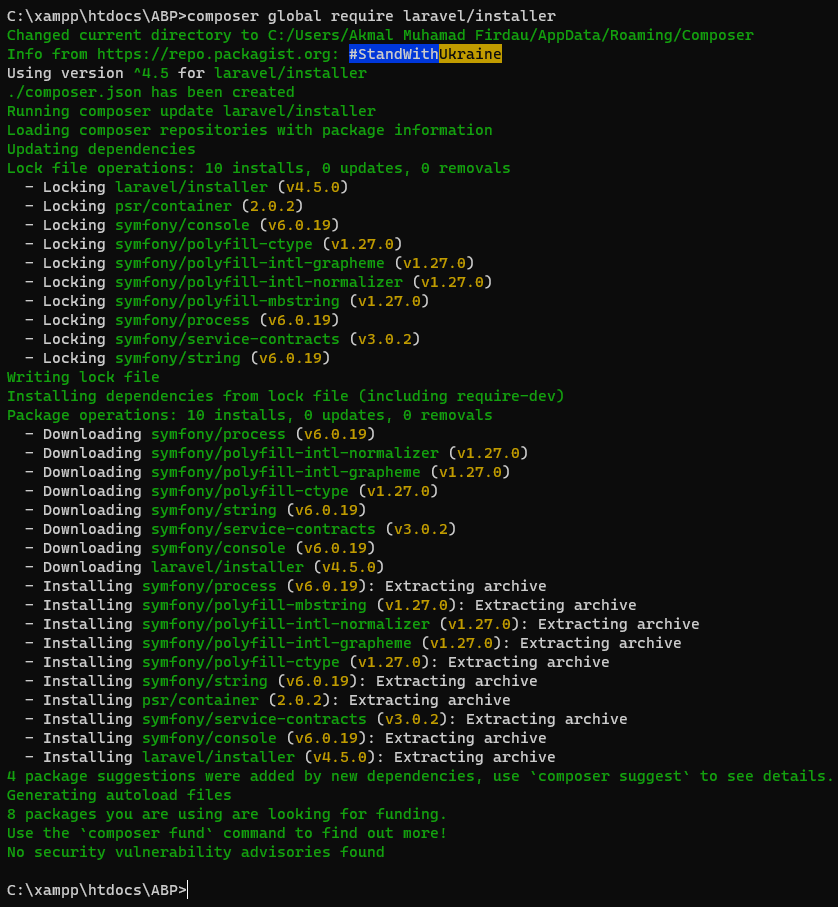
10. vendor/

Folder ini berisi paket-paket pihak ketiga yang digunakan oleh aplikasi Laravel.

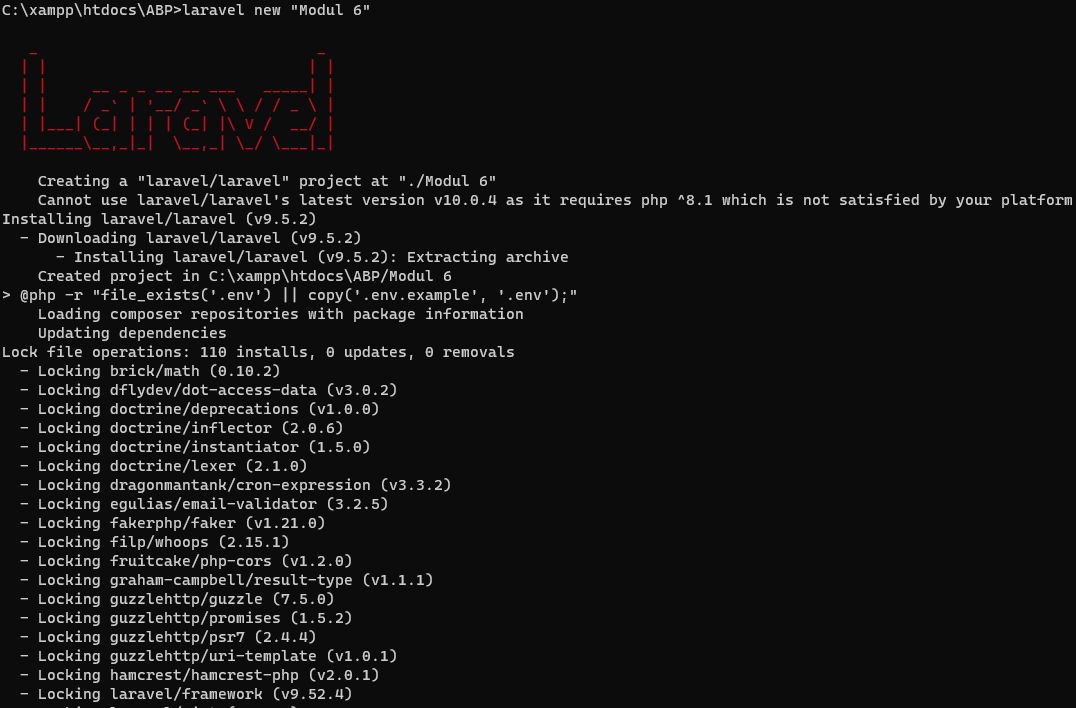
Secara keseluruhan, struktur folder Laravel dirancang agar mudah dipahami dan diorganisasi. Setiap folder memiliki tugas dan fungsinya masing-masing sehingga memudahkan pengembangan aplikasi Laravel.

1. **Instalasi Laravel**

Pastikan terlebih dahulu, bahwa [composer](https://getcomposer.org/) sudah terinstall. Lalu jalankan perintah “composer global require laravel/installer” untuk menginstall laravel secara global.

****

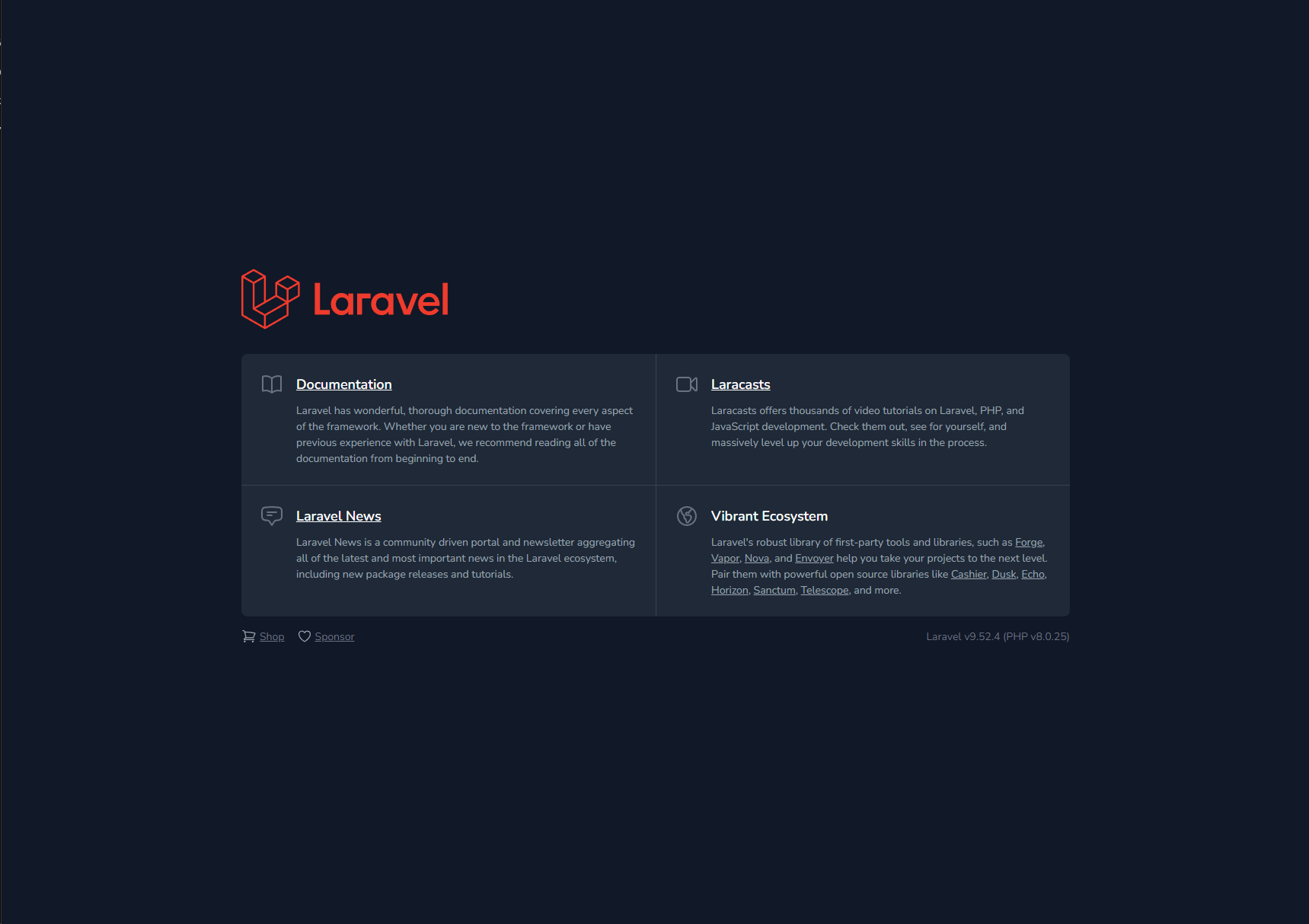
Jalankan perintah *laravel new “nama\_project”,* lalu tunggu hingga proses pembuatan projek baru selesai.



Lalu jalankan “php artisan ser” untuk menjalankan server

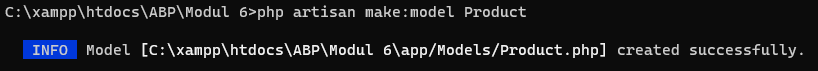


Welcome page Laravel



1. **Model**

Pembuatan model menggunakan CLI (Command Line Interface) pada Laravel sangat mudah dilakukan. Pertama, buka terminal atau command prompt dan pastikan sudah berada di dalam folder projek Laravel. Kemudian, ketik perintah *php artisan make:model NamaModel* dan tekan Enter.

****

Setelah itu, Laravel akan membuatkan file model baru pada folder "app/Models". Secara otomatis, nama file model tersebut akan mengikuti aturan penamaan konvensi Laravel, yaitu menggunakan format CamelCase dan singular .

Secara default, model yang baru dibuat akan memiliki beberapa konfigurasi awal seperti pengaturan nama tabel, kolom fillable, primary key, timestamp, dan sebagainya. Konfigurasi tersebut dapat dilihat pada file model yang baru dibuat pada bagian $table, $fillable, $primaryKey, $timestamps, dan sebagainya.

Selain itu, jika model tersebut akan terhubung ke database, maka Laravel akan memerlukan koneksi database yang sudah terkonfigurasi sebelumnya pada file .env di root folder projek. Dalam file .env, pastikan konfigurasi database sudah sesuai dengan server database yang digunakan. Jika belum terkonfigurasi, maka Laravel akan menghasilkan error saat menggunakan model tersebut.

Dikarenakan pada modul ini belum ada penggunakan database pada model, maka model dibuatkan data dummynya menggunakan array menjadi seperti berikut:



Product Model

Berikut adalah penjelasan mengenai method getAllProducts, getProductById, dan addProduct pada model:

1. getAllProducts():

Method ini berfungsi untuk mendapatkan semua data produk yang ada dalam database. Pada kasus nyata, method ini akan melakukan query SELECT \* FROM products pada tabel produk dan mengembalikan hasilnya dalam bentuk array atau collection. Dalam implementasinya, method ini dapat menggunakan method all() yang disediakan oleh Eloquent ORM untuk mengambil semua data dari tabel terkait. Tetapi dikarenakan pada modul 6 ini belum ada penggunaan database, maka return value dari array ini masih mengembalikan data static dari array pada variable $products

2. addProduct($data):

Method ini akan menambahkan data produk baru ke dalam array $products. Parameter $data berisi informasi produk seperti nama, deskripsi, harga, dan sebagainya. Dalam implementasinya, method ini akan membuat sebuah instance baru dari model Product, mengisi property-nya dengan data yang diberikan, dan menyimpannya ke dalam array $products.

1. **View**

View pada Laravel adalah bagian dari arsitektur Model-View-Controller (MVC) yang bertanggung jawab untuk menampilkan data ke user. Dalam Laravel, view diwakili oleh file-file blade template yang berisi kode HTML, PHP, dan bahasa template Laravel yang khusus, seperti directives dan expressions. View bertanggung jawab untuk menampilkan data yang diberikan oleh controller dan juga untuk menangani interaksi user seperti form input, button, dan hyperlink.

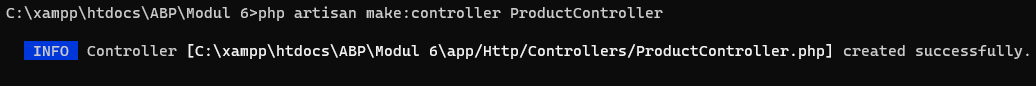
Dalam Laravel, view dapat dipecah menjadi beberapa bagian kecil yang disebut sebagai partials. Partial adalah file template yang digunakan kembali dalam beberapa halaman view yang berbeda. Partial dapat mempermudah pengelolaan kode dan meningkatkan kinerja aplikasi karena menghindari duplikasi kode.

Laravel juga menyediakan fitur-fitur yang memudahkan pembuatan view, seperti inheritance dan section. Inheritance memungkinkan developer untuk membuat view induk yang bisa digunakan kembali dalam beberapa view turunan, sedangkan section memungkinkan developer untuk memisahkan bagian-bagian dari view dan menggabungkannya dalam view lain.

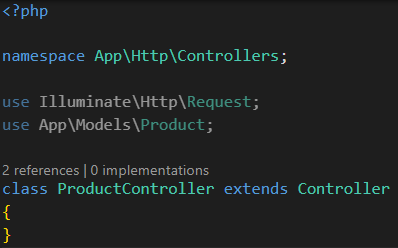


1. **Controller**

Pembuatan controller pada Laravel dapat dilakukan dengan menggunakan command-line interface (CLI) yang disediakan oleh framework ini. Caranya cukup mudah, yaitu dengan munuliskan command *php artisan make:controller NamaController* dapat juga diberi argument tambahan seperti --resource. Kelebihan lain dari menggunakan resource ini adalah kita dapat menambah satu Routing saja untuk menangani semua aksi dari satu Controller



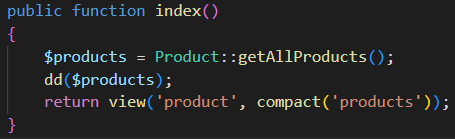
Setelah controller berhasil dibuat, kita dapat membuka file controller yang telah dibuat (biasanya berada di direktori App/Http/Controllers) dan menambahkan method-method yang dibutuhkan. Method-method tersebut biasanya digunakan untuk memproses request dari client dan memberikan response yang sesuai. Pada controller, kita juga dapat melakukan validasi data, melakukan pengolahan data, dan memanggil method-method pada model untuk melakukan query ke database.

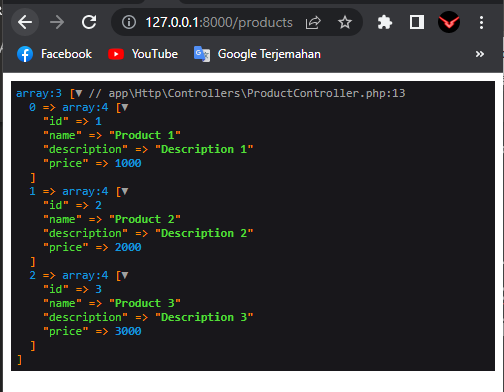


Setelah controller selesai dibuat, kita dapat menghubungkannya dengan routing pada file web.php untuk menentukan URL yang akan dipetakan ke dalam method-method pada controller. Dalam Laravel, kita dapat menggunakan resource routing untuk menghasilkan URL yang sesuai dengan kebutuhan CRUD pada controller.

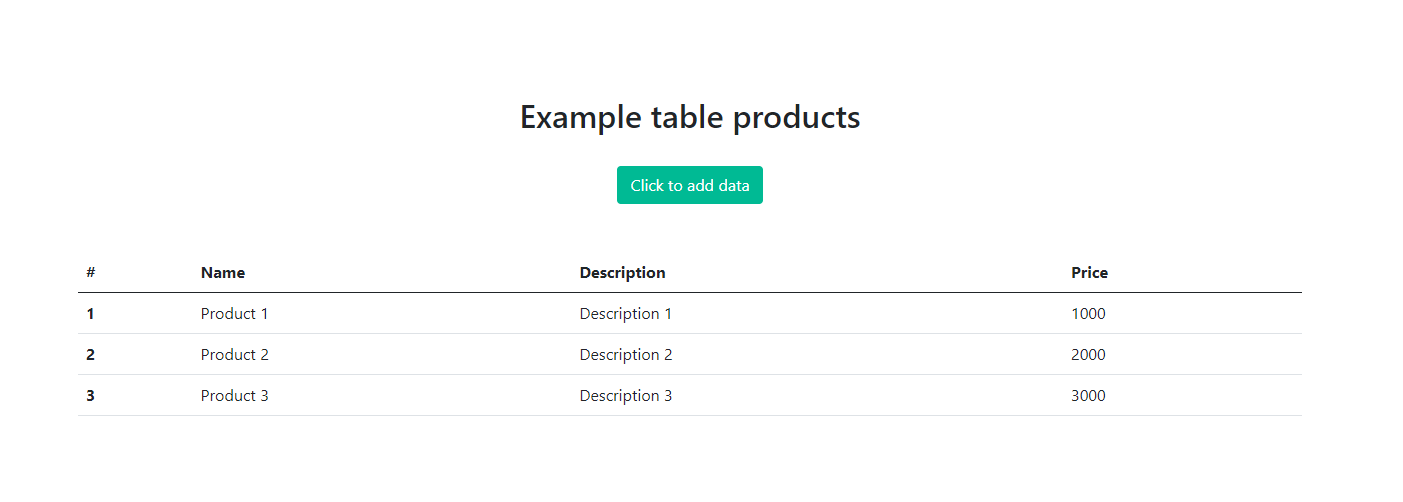
* 1. Membuat method index

Method index pada controller product ini digunakan untuk mengambil seluruh data products dari model. Mari kita coba dd(Dump Die) untuk melakukan pengecekan bahwa data tersebut sudah terambil dari model.



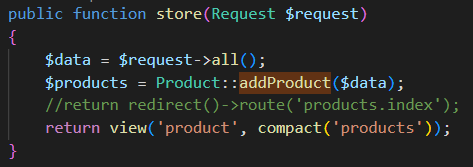


Data tersebut sudah sesuai dengan apa yang dikembalikan dari model yaitu array of product. Setelah itu beri comment pada *dd*, supaya nantinya yang ditampilkan adalah view dari halaman product.

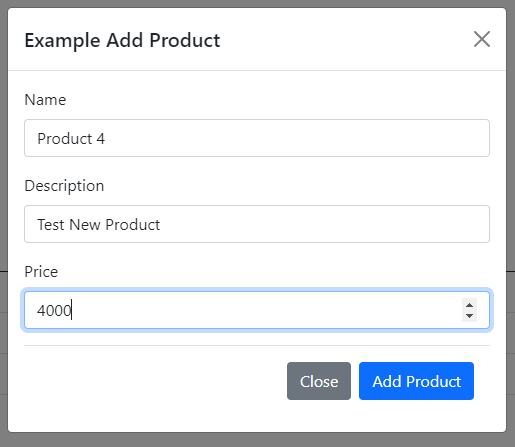


* 1. Membuat method store

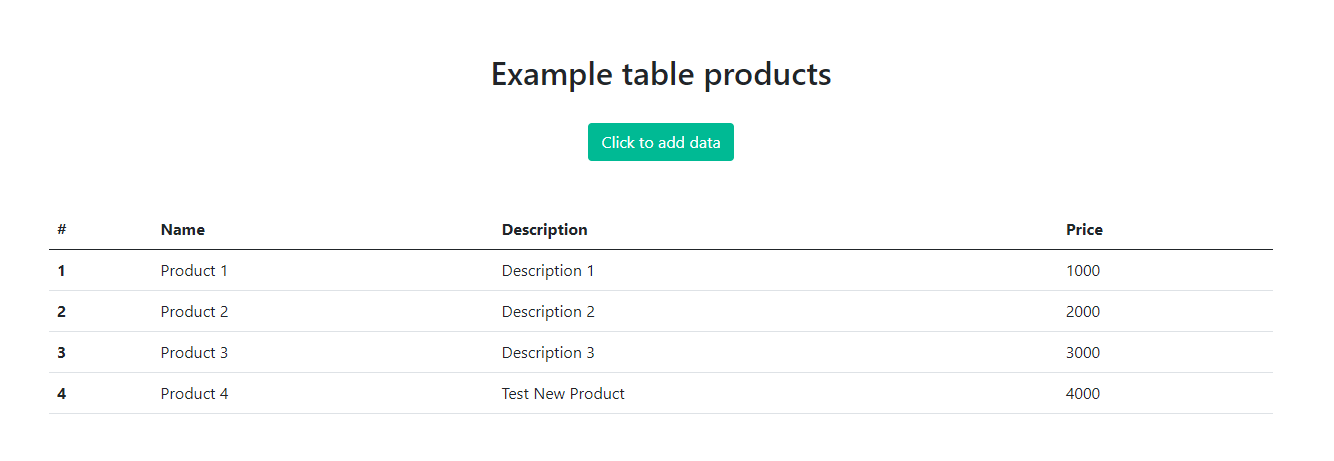
Method *store()* pada controller *ProductController* digunakan untuk menyimpan data produk baru (Dalam studi kasus ini menambahkan product baru kedalam array). Method ini dijalankan ketika user mengirimkan form untuk membuat produk baru melalui HTTP POST request.



Pada kasus nyata, method store biasanya akan me-return redirect ke halaman index (yang diberi comment), tetapi karena model kali ini menggunakan data dummy yang notabenenya disimpan dalam array, maka return adalah view dari product agar dapat terlihat perubahan yang terjadi pada model product.

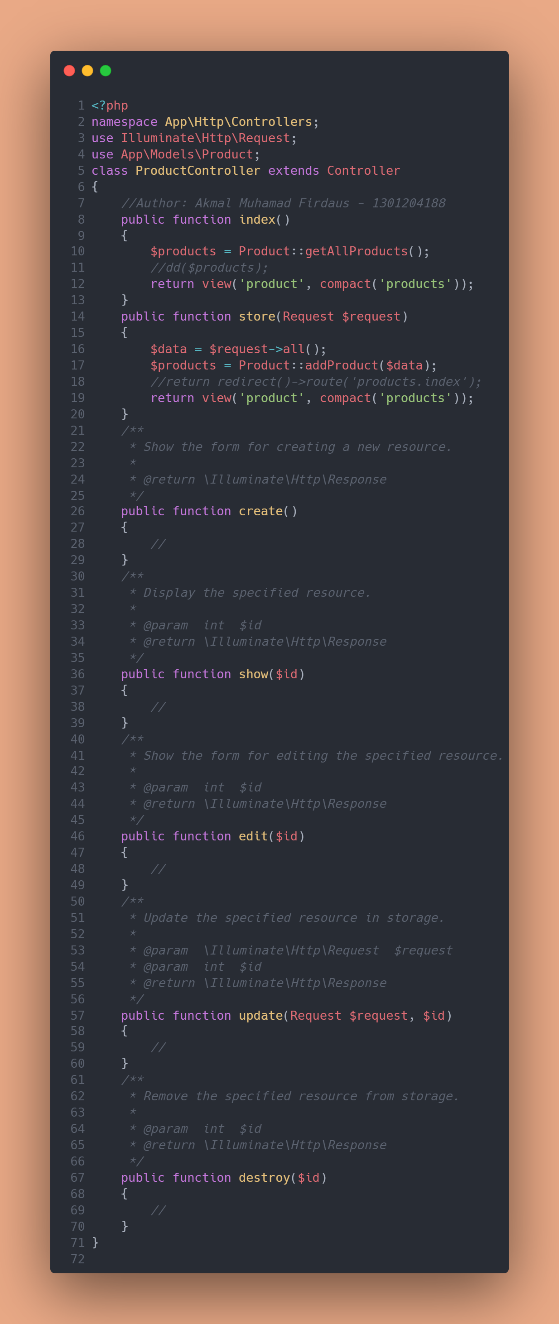


Tampilan dari tambah product yang ada pada view product menggunakan modal.



Tampilan setelah produk berhasil ditambahkan pada data dummy yang disimpan pada array.

Berikut adalah codingan lengkap dari ProductController:



1. **Routing**

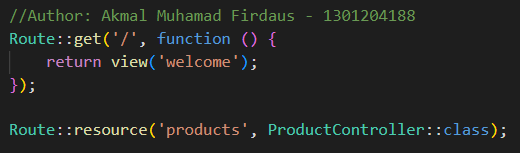
Routing pada Laravel adalah mekanisme untuk menentukan bagaimana aplikasi web menangani permintaan HTTP dari client. Dalam Laravel, routing digunakan untuk menentukan alamat URL (URI) apa yang akan diarahkan ke method tertentu dalam controller.

Routing dapat digunakan untuk menangani berbagai macam metode HTTP, seperti GET, POST, PUT, PATCH, dan DELETE. Setiap metode HTTP dapat ditangani oleh sebuah route yang berbeda.

Dalam Laravel, routing dapat didefinisikan dalam file routes/web.php atau routes/api.php. Untuk setiap route, kita bisa menentukan URI, controller method yang akan menangani route tersebut, dan middleware yang akan dijalankan sebelum atau setelah route dijalankan.

Laravel juga menyediakan fitur-fitur yang memudahkan pengelolaan routing, seperti route parameters, route groups, dan named routes. Route parameters memungkinkan kita untuk menentukan parameter pada URI, seperti id atau nama, yang kemudian dapat digunakan dalam method controller. Route groups memungkinkan kita untuk mengelompokkan beberapa route yang memiliki middleware atau prefix yang sama. Named routes memungkinkan kita untuk memberikan nama pada sebuah route yang kemudian dapat digunakan dalam view atau controller.

Pada routing cukup menambahkan Route::resource(“url”, namaController::class), maka secara otomatis routing pada laravel dapat mengenali GET, POST, PUT, PATCH, dan DELETE pada routes tersebut.



Contoh lain jika ingin menggunakan method HTTP tertentu adalah pada url, yaitu kita dapat menambahkan Route::method(“url”, aksi). Method dapat diganti sesuai dengan method apa yang ingin digunakan, lalu aksi dapat berupa mengembalikan view langsung atau memanggil salah satu function pada controller.

Berikut adalah Actions Handled By Resource ProductController

|  |  |  |  |
| --- | --- | --- | --- |
| **Verb/Method** | **URI** | **Action** | **Route Name** |
| GET | /products | index | products.index |
| GET | / products/create | create | products.create |
| POST | /products | store | products.store |
| GET | /products/{id} | show | products.show |
| GET | /products/{id}/edit | edit | products.edit |
| PUT/PATCH | /products/{id} | update | products.update |
| DELETE | /products/{id} | destroy | products.destory |

Full Source Code : <https://github.com/codernewbie04/CII3H4-ABP-Praktikum>