Oleh : Rahmat Fauzi, S.T.,M.T

Who Am I ?

- Graduated from:
  - ITS Surabaya (Bachelor of Electrical Engineering, 2008-2012)
  - ITS Surabaya (Master of Electrical Engineering, 2013-2015)
- Lecturer System Information Telkom University
- Member Profesional : IEEE and APTIKOM.
- Xsysware Academy Founder.
- Contact:
  - Room: Karang Building C.202 FRI Telkom University
  - Email: rahmatfauzi9013@gmail.com
  - Phone: 0812 2025 6700

Apa itu Laravel ?

Laravel adalah Kerangka kerja (Framework) Bahasa pemrograman PHP (Hypertext Prepocessor) berbasis open source dengan konsep MVC (Model View Controller).

Framework Laravel **bertujuan** meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan sintaks yang ekspresif, jelas dan menghemat waktu.

Laravel Menggunakan **GITHUB** sebagai tempat untuk berbagi kode.

MVC Sendiri adalah sebuah pendekatan software yang memisahkan beberapa komponen aplikasi yakni komponen **manipulasi data**, komponen **controller** dan komponen **user interface**.

Bagaimana Arsitektur Laravel ?

Setelah kita mengetahui konsep Laravel Menggunakan MVC (model view controller). Jika dirinci lebih detail fungsi dari masing – masing komponen tersebut adalah :

1. Model

Komponen yang berfungsi mengelola dengan sumber data dan logika data.

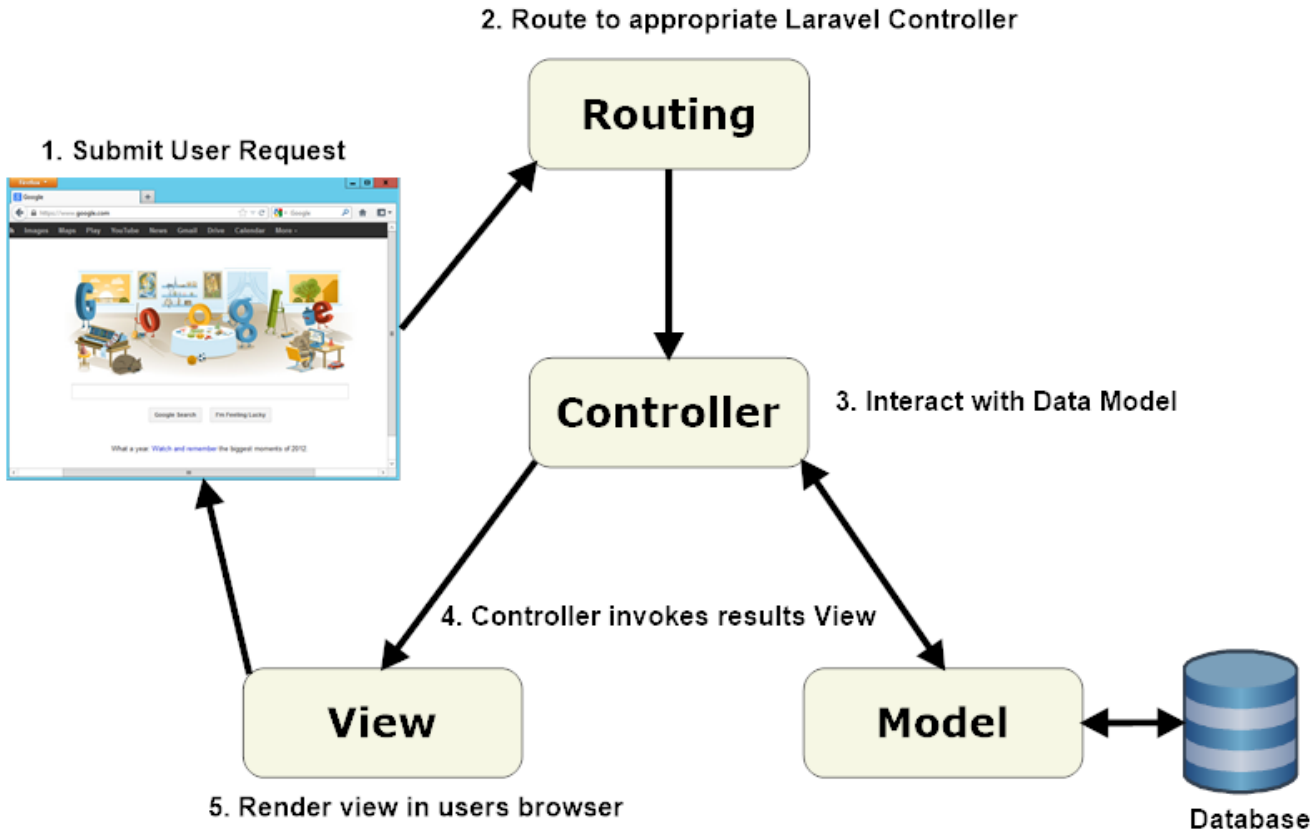
2. View

Komponen yang berfungsi membuat tampilan

3. Controller

Komponen yang berfungsi menerima input (request) dan memberikan output (response) data.

Visualisasi MVC Laravel



Penjelasan MVC

1. User mengakses aplikasi melalui route tertentu

2. Route tersebut oleh aplikasi telah dipetakan ke controller action

3. Controller action akan menggunakan model untuk mengakses data. Atau langsung mengembalikan view tanpa data (langsung ke step 5)

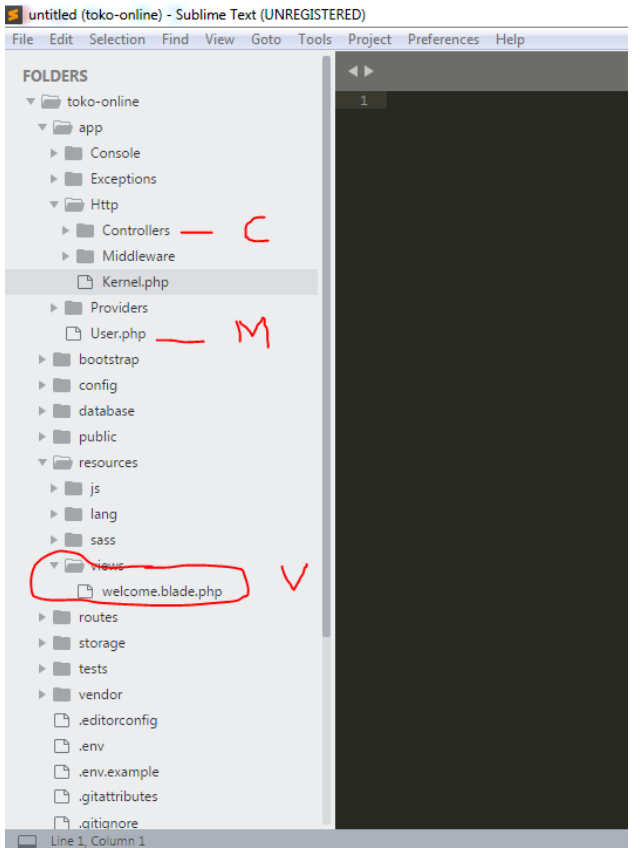
4. Model berinteraksi ke database untuk mendapatkan data atau menyimpan data

5. Setelah berhasil mendapatkan data melalui model, controller akan mengembalikan sebuah view

sekaligus data jika ada.

6. View tersebut pada akhirnya yang dilihat oleh user

MVC telah disediakan folder khusus di aplikasi laravel kecuali model.



1. Controller berada di folder app/Http/Controllers

2. View berada di folder resources/views

View berfungsi untuk menaruh kode tampilan ke pengguna aplikasi. Di file view ini lah kita letakkan kode html, css dan javascript bukan di controller, route atau model. File view bisa mengakses variable yang dilempar dari controller action seperti pada bahasan Controller.

Pada aplikasi Laravel baru, tersedia satu file view yaitu welcome.blade.php, silahkan dibuka pada resources/views/welcome.blade.php. File tersebut berisi kode html, css, javascript dan beberapa sintaks blade.

3. Khusus model tidak memiliki folder khusus, tetapi kita bisa meletakannya di folder app, atau di folder lainnya sesuai kebutuhan.

Pengenalan Routing

Route

Route jika diterjemahkan bebas memiliki makna **Rute / Jalur**.

Sedangkan dalam Laravel, Route bisa diartikan **jalur URL yang bisa diakses oleh pengguna aplikasi dan ke mana jalur itu diproses.**

Misalkan routing /helo yang bisa diakses di http:localhost/helo yang menampilkan string “Hello World”

Sintak paling dasar dari routing di Laravel adalah sebagai berikut:

```
Route::verb("/path", callback);
```

Route::verb("/path", callback);

Kode di atas dapat kita jelaskan sebagai berikut :

1. Facade Route.

2. Verb

Verb merupakan HTTP verb, terdiri dari get, post, put, delete, options, patch.

3. Path

Merupakan path URL setelah domain aplikasi kita yang kita izinkan untuk diakses oleh pengguna aplikasi.

4. Callback

Callback bisa berupa *closure* function atau Controller action yang ingin kita eksekusi ketika path tertentu diakses oleh pengguna aplikasi.

Closure function merupakan istilah lain untuk function yang tidak memiliki nama dan seringkali digunakan sebagai callback. Jangan bingung ya.

```
Route::get("/helo", function(){
return "Hello World";
});
```

Route di atas bisa kita baca seperti ini, ketika pengguna mengakses URL http://toko-online.test/helo, eksekusi closure function di parameter kedua, closure function tersebut menghasilkan string "Hello World". Sangat sederhana dan mudah dicerna, sehingga pengguna akan mendapatkan string Hello World di layar browser mereka.

Akan tetapi penggunaan closure function sebagai route callback jarang sekali kita pakai dalam pembuatan aplikasi sesungguhnya, karena kita memiliki controller.

Ingat, callback selain berupa closure function bisa juga kita isi dengan action dari controller. Sehingga misalnya kita memiliki WelcomeController dengan method beriSalam, kita bisa menulis ulang Route kita sebagai berikut

```
Route::get("/hello", "WelcomeController@beriSalam");
```

Dengan pemisalan method beriSalam akan menghasilkan “Hello World”, maka route di atas akan

menghasilkan string yang sama ke user seperti halnya ketika menggunakan definisi route yang kita tulis sebelumnya.

Route ke controller perlu diketahui terlebih dahulu, supaya kita paham saat membahas controller, untuk apa menulis controller dan action yang ada di dalamnya. Yaitu untuk kita isikan sebagai *callback* di router, sebagian besar penggunaan controller adalah untuk hal tersebut.

Di mana kita tuliskan definisi routing?

Definisi route : Untuk jalur Website Standar, kita bisa menulis di file **routes/web.php**

Sedangkan untuk penggunaan **web service / api service** bisa menuliskan **di routes/api.php**

Sedangkan, Jika kita ingin membuka jalur akses melalui command line kita bisa menggunakan **routes/console.php**

Adapun routes/channel.php dapat kita gunakan untuk jalur akses broadcast melalui websocket.

Umumnya aplikasi yang kita buat, cukup dengan **routes/web.php dan routes/api.php**. Bahkan jika aplikasi kita tidak perlu menyediakan web servie, kita hanya perlu menggunakan **routes/web.php** saja.

Pengenalan Model

Definisi Model ?

Model dalam laravel mempunyai makna **entitas yang berinteraksi dengan sumber data**.

Model **berfungsi** untuk query ke database, insert data baru, update, atau hapus record database.

Semua itu dilakukan dengan ORM (Object Relational Mapping) sehingga pada banyak kasus, kita tidak perlu menuliskan kode SQL secara langsung, akan tetapi langsung menggunakan method bawaan dari ORM seperti, **find, findOrFail, create, update**, dll.

Dan ORM bawaan yang dipakai oleh Laravel adalah **Eloquent.**

Membuat Model

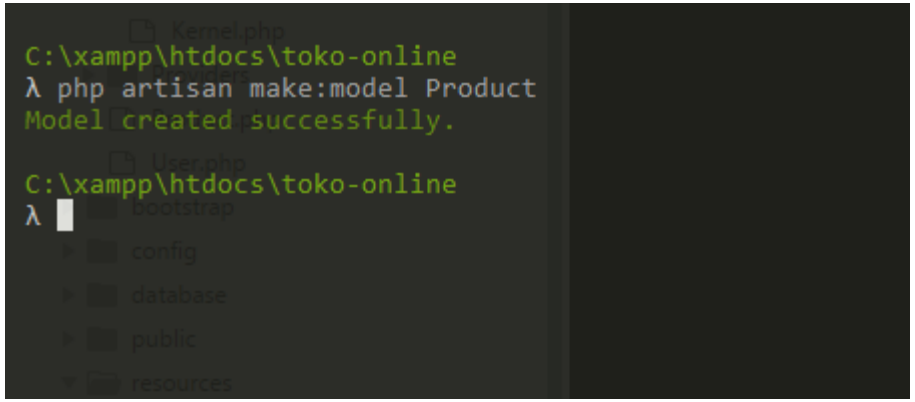
Untuk membuat model ketikan perintah berikut ini pada terminal

```
php artisan make:model NamaModel
```

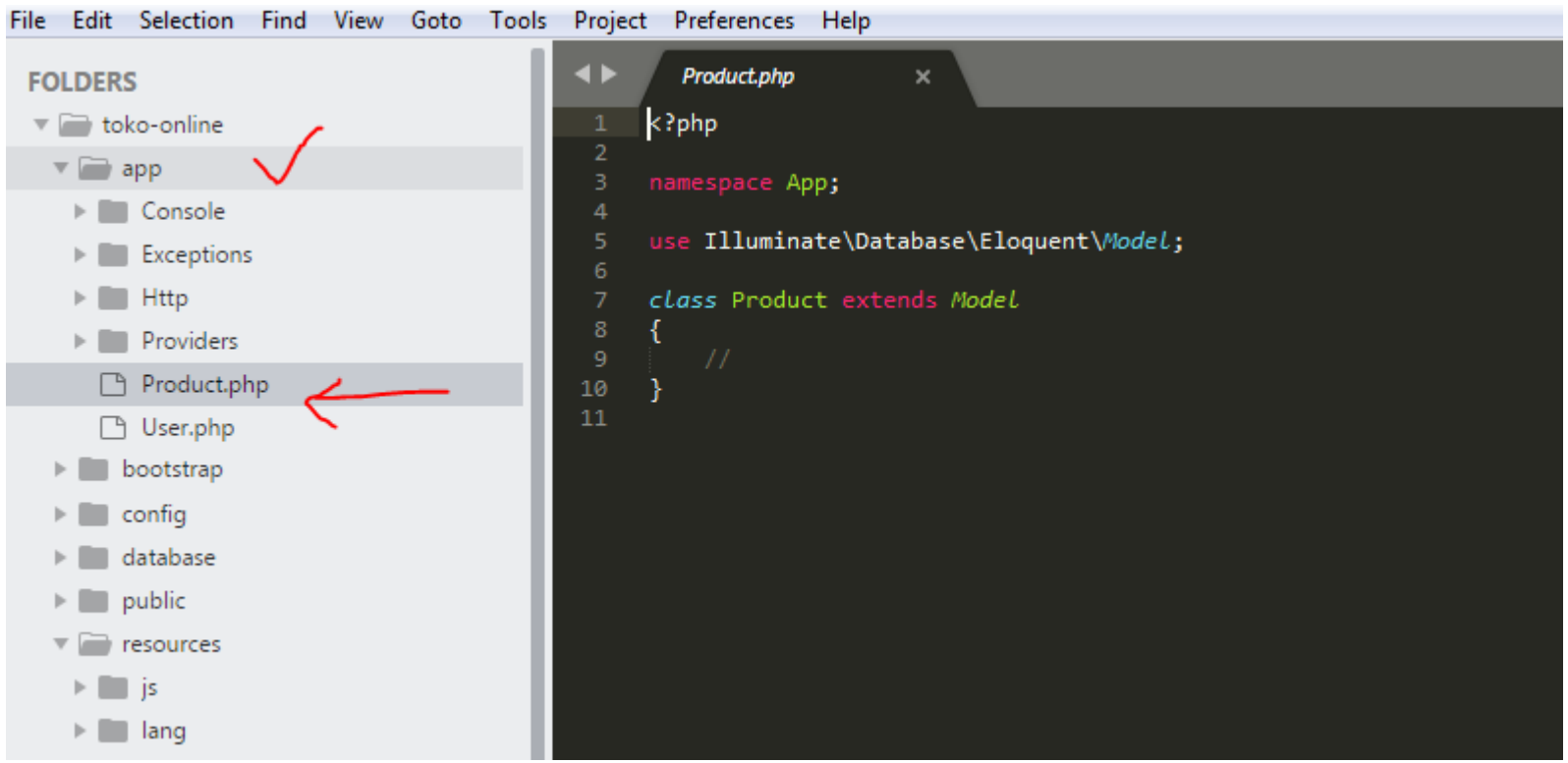
Misalkan

```
php artisan make:model Product
```

Bisa kita tulis seperti ini di terminal



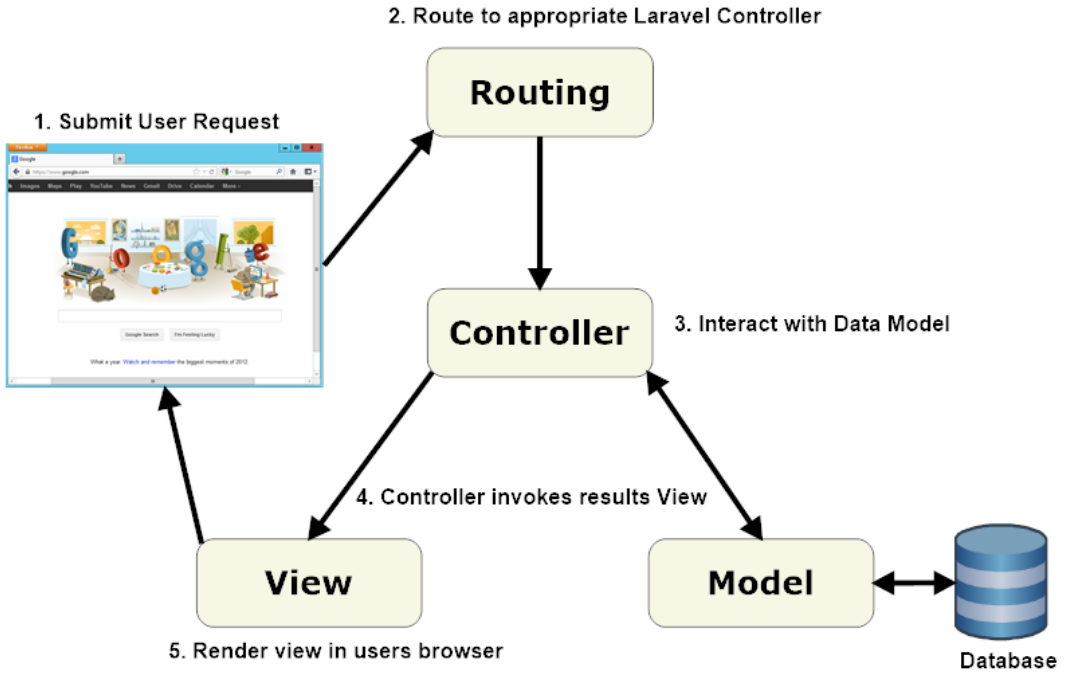
Akan muncul product.php di submenu app



Pengenalan Controller

Controller

Di bagian **Pengenalan Model,** kita sudah membuat Model **Product.** Akan tetapi Model Product belum **bisa diakses** oleh User karena belum ada **Controller.**



Controller

Di bagian **Route,** kita ingat ada dua method pemanggilan ada **Closure route** dan Method **Callback route.**

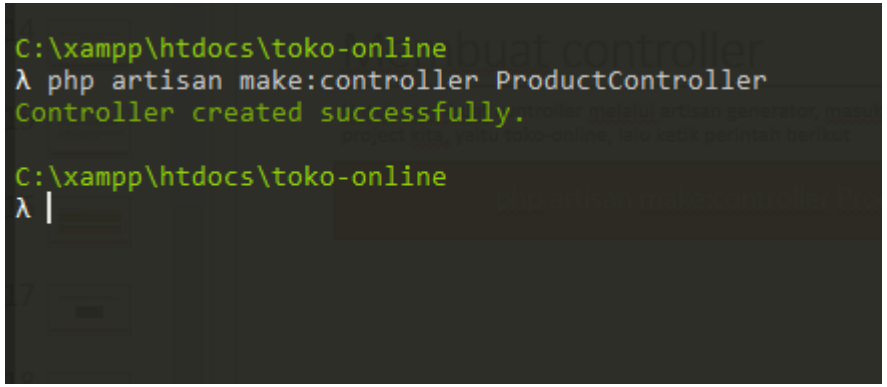
```
Route::get("/helo", function(){
return "Hello World";
});
```

```
Route::get("/products", "WelcomeController@index");
```

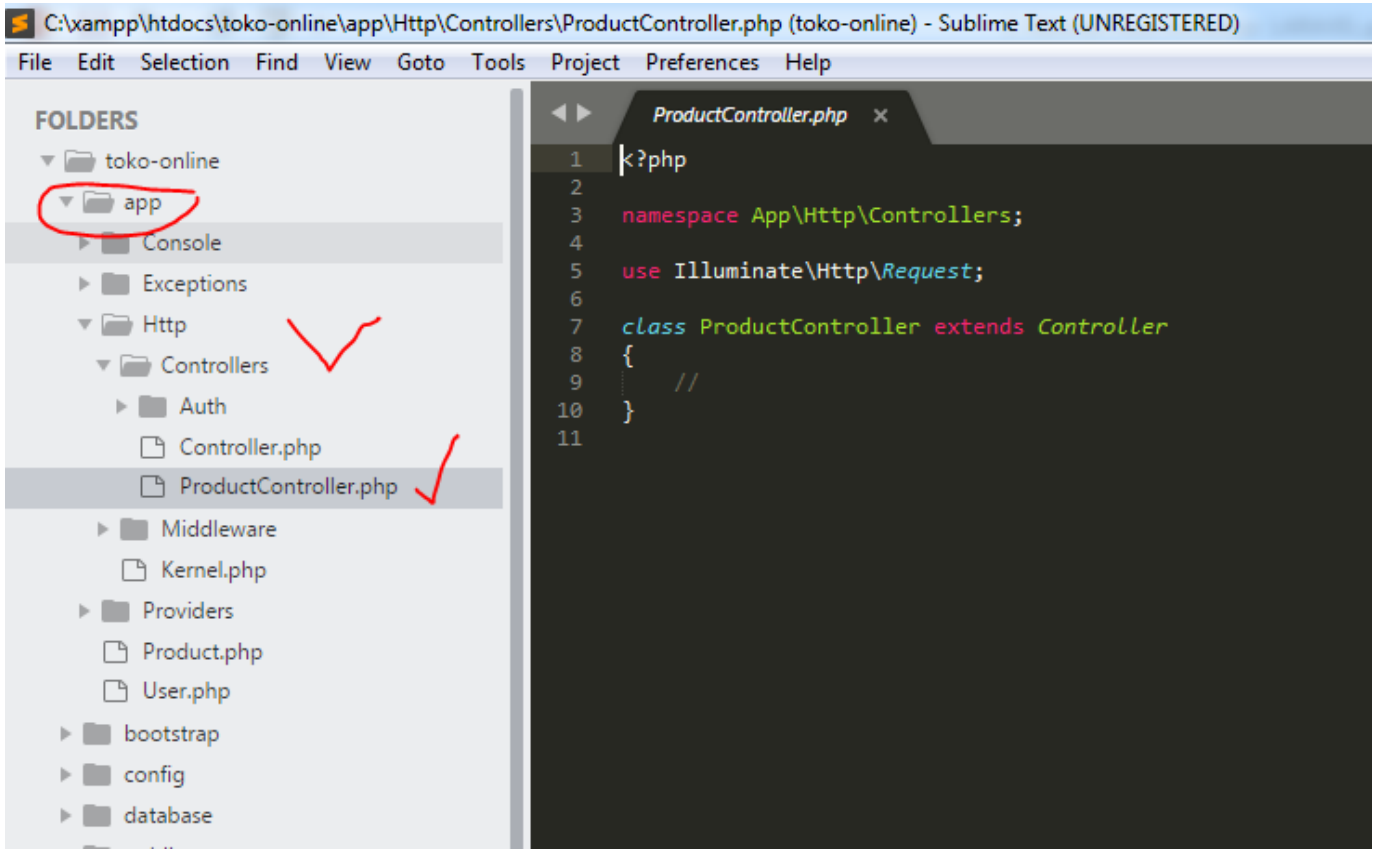
Membuat controller

Untuk membuat controller melalui artisan generator, masuk ke workspace, lalu masuk ke folder project kita, yaitu toko-online, lalu ketik perintah berikut

```
php artisan make:controller ProductController
```



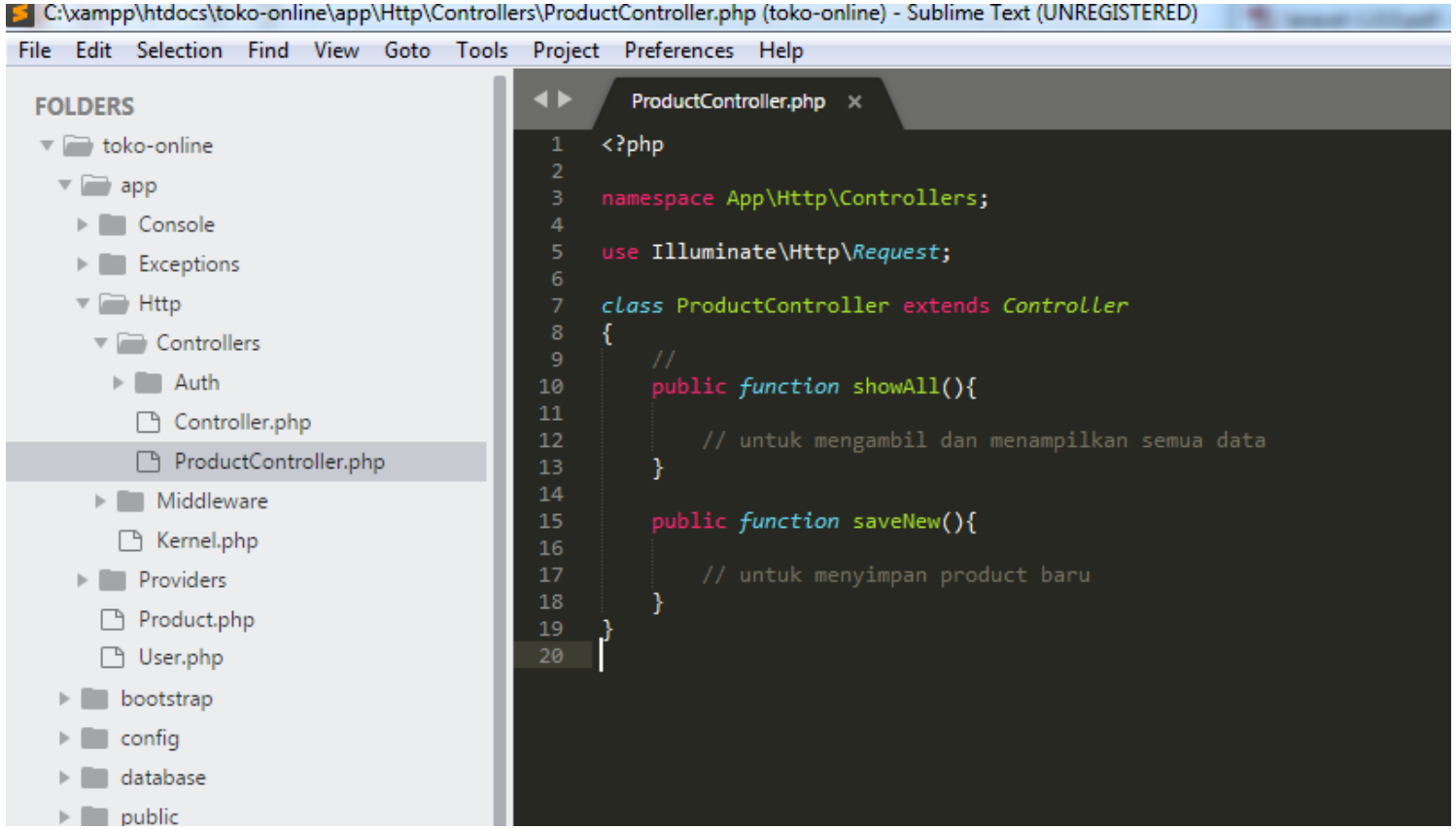
Sekarang kita memiliki file pada app/Http/Controllers/ProductController.php.



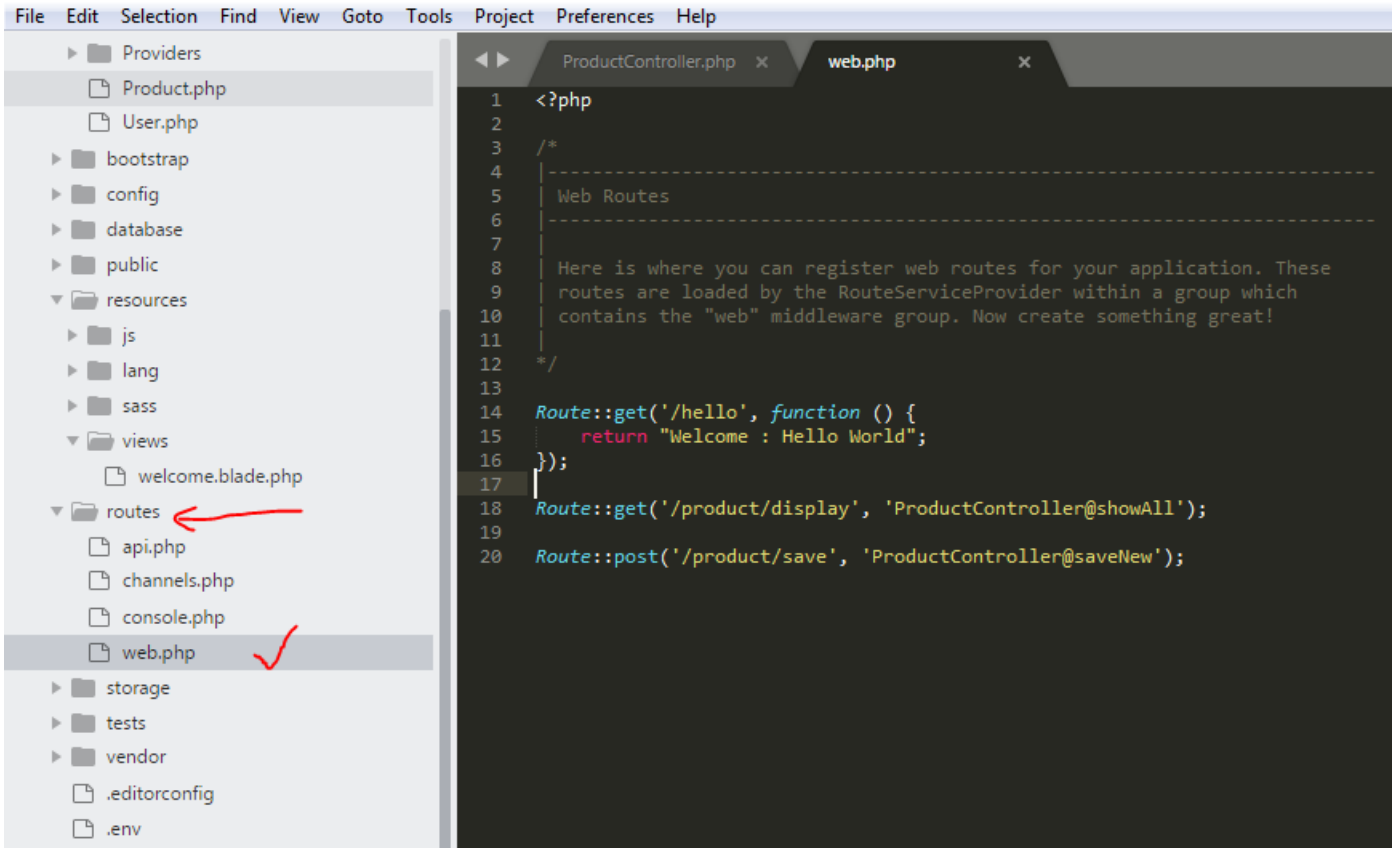
Menambahkan Controller action

Controller action merupakan method yang berfungsi untuk melakukan operasi logika. Dalam operasi logika tersebut kita bisa mengambil, mengupdate, menghapus data melalui model lalu mengembalikan sebuah response kepada pengguna aplikasi, baik berupa JSON atau berupa view:

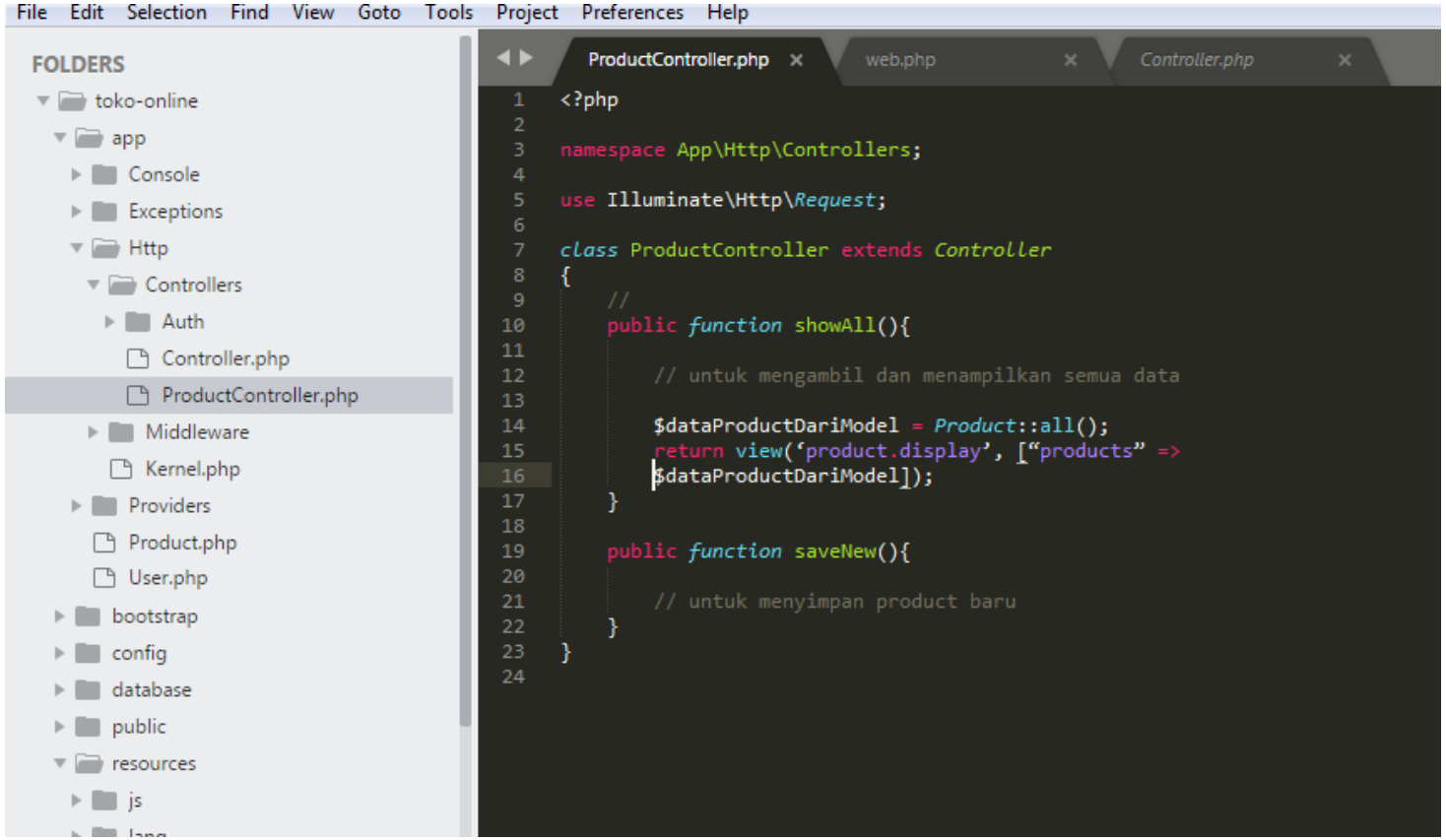
Menambahkan Controller action (Contoh)



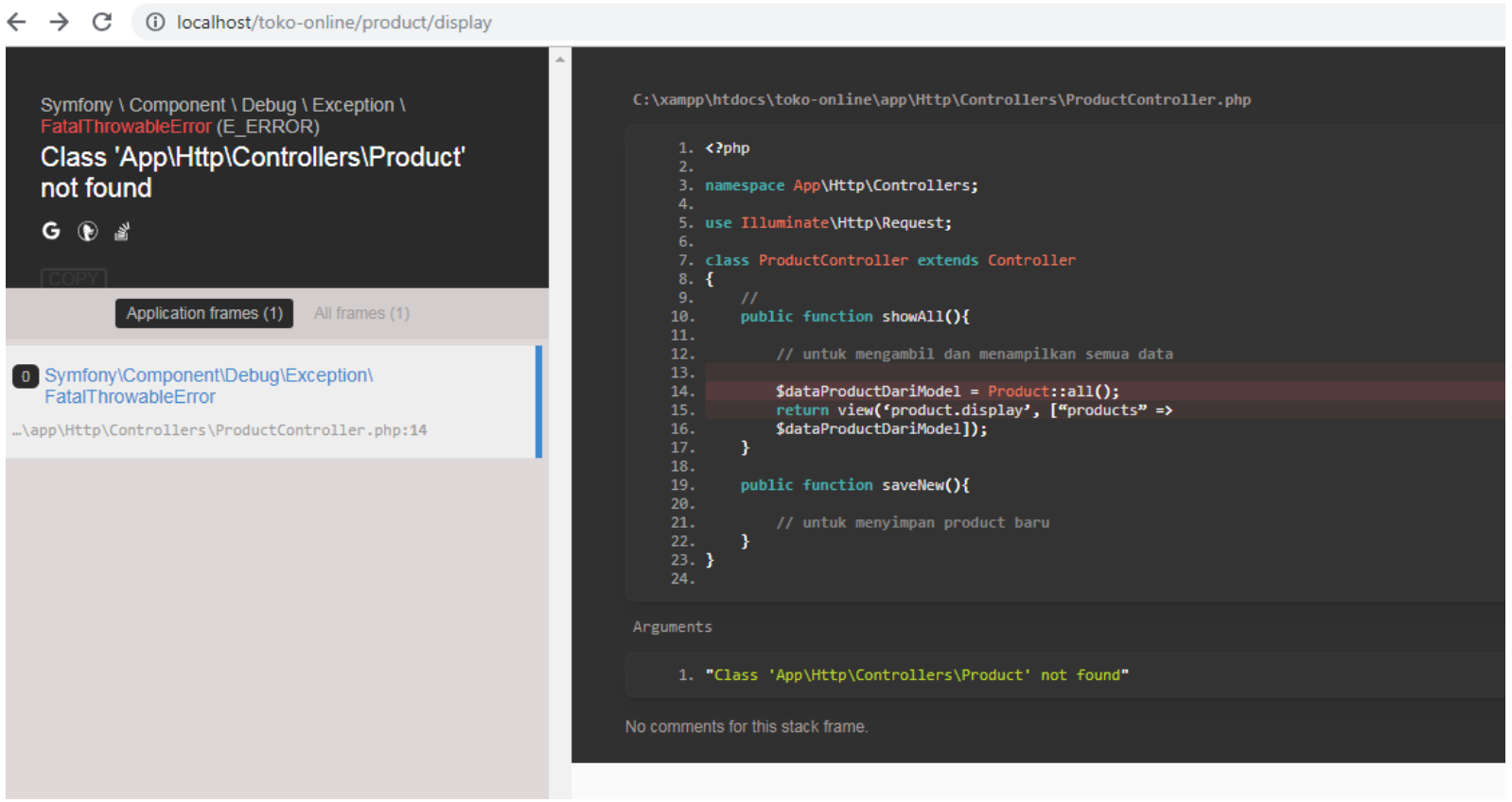
Menambahkan Route



Contoh Menambahkan code ProductController



Akan muncul Pesan Error



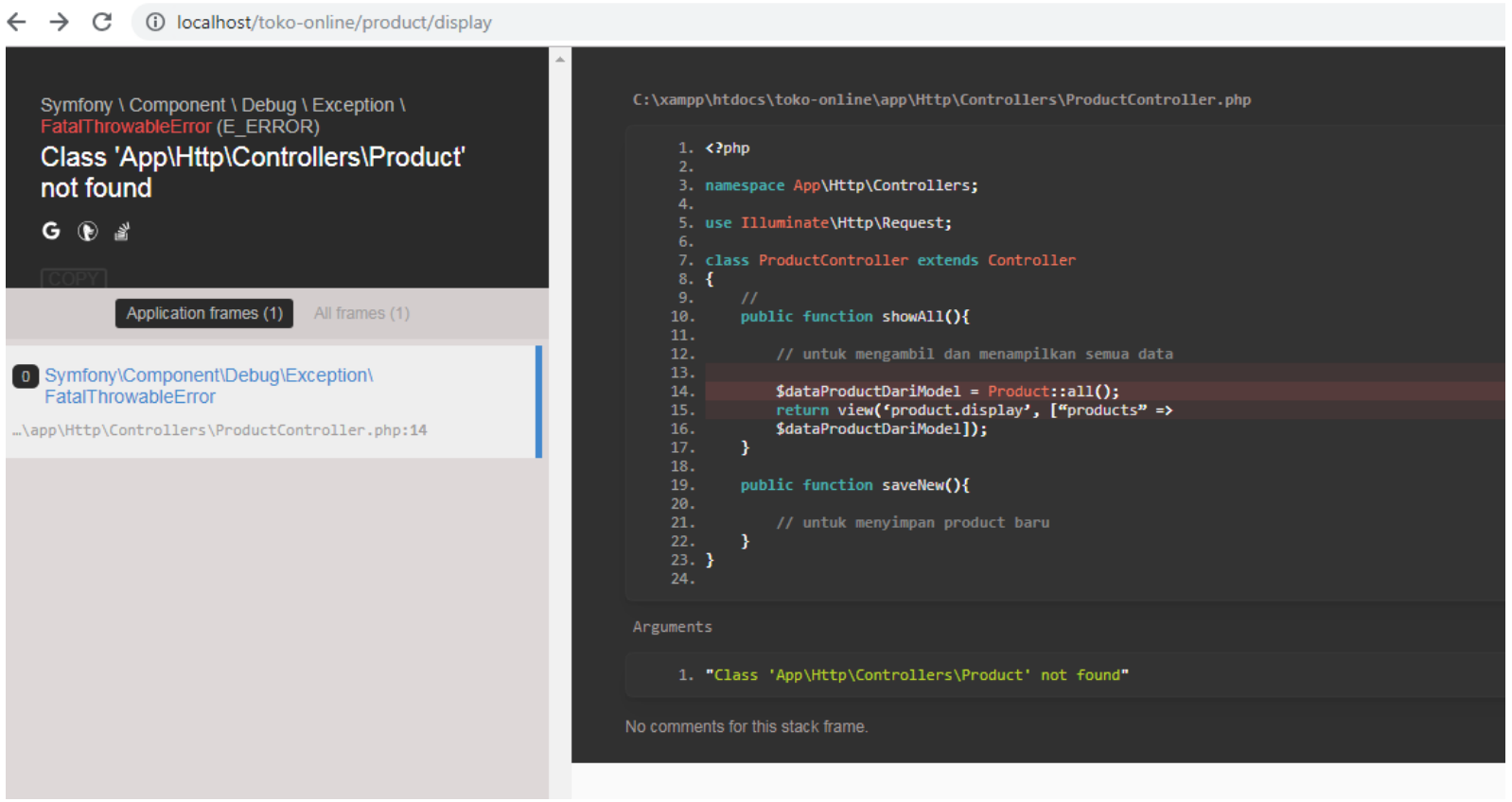
Pengenalan View

Pengenalan View

View bertanggungjawab untuk memberikan tampilan ke user. Jika kita ingin meletakkan kode html, css dan javascript di view lah tempatnya. Laravel membuat view lebih powerful dengan memanfaatkan templating engine.

Karena Laravel menggunakan templating engine bawaan Blade, maka file view diakhiri dengan .blade.php. Misal **product.blade.php, product-list.blade.php, product-detail.blade.php.**

Masih muncul Pesan Error-> Karena belum ada View



Maka kita buat sebuah file baru di **resources/views/product/display.blade.php**. Lalu coba tambahkan text atau kode html, misalnya

 View display product siap .

Jika sudah silahkan coba untuk mengakses route yang tadi error. Apa yang terjadi?

view berhasil menampilkan pesan.

Kesimpulan

Di bagian ini kita mempelajari alur kerja MVC Laravel, dengan memahami alur kerja MVC kita semakin mudah mencodding program selanjutnya.

Pertama kita definisikan jalur akses URL melalui route, route tersebut kita arahkan ke controller dan action tertentu, action tersebut akan menggunakan model dan query eloquent untuk mengambil data, lalu action tersebut mengembalikan views dengan data data dari model bila diperlukan.