**BUKU MEMASAK**

LARAVEL

Bill tanthowi jauhari

*bill.web.id*

**KATA PENGANTAR**

Puji syukur kepada Alloh SWT yang telah memberikan rahmat kepada kita semua sehingga buku ini dapat selesai, tidak lupa mengucapkan Solawat dan salam kepada junjungan nabi Muhammad SAW yang telah membimbing kita kepada jalan yang benar yaitu agama islam.

Tidak lupa saya ucapkan terimakasih kepada Ibu dan Ayah saya yang telah membesarkan saya dan telah mendidik saya sehingga sekarang, semoga beliau sehat selalu aamiin. Terimakasih juga kepada semua teman dan editor yang telah meluangkan waktu membantu saya untuk menyelesaikan buku ini.

Buku ini menggunakan laravel versi 5.4, bagi pembaca yang menggunakan laravel versi 5.3 kebawah di sarankan untuk membuat project baru untuk versi laravel 5.4. Buku ini masih jauh dari kata sempurna, saya sangat membutuhkan kritik dan saran dari pembaca demi memperbaiki buku ini kedepannya, karna kritik dan saran pembaca akan menjadi pemicu semangat saya untuk menulis lagi dan lagi. trimakasih

*Hormat saya, Pasuruan 2017*

*BILL TANTHOWI JAUHARI*

* **PENGENALAN**

****

**Laravel** adalah framework modern yang pada masa kini banyak di minati para web developer, terutama developer baru yang mau memulai belajar framework, perlu di ketahui bahwa laravel adalah framework yang mempunyai prinsip *hard to learn but easy to use.* Artinya adalah laravel akan banyak menyita waktu anda untuk belajar, karna konsep yang di suguhkan banyak yang berbeda dengan framework pada umumnya seperti **codeigniter**. Bagi para developer yang baru di sarankan untuk mengetahui konsep MVC terlebih dahulu sebelum melangkah ke tahap2 pembuatan website.

**ARTISAN**

Laravel memiliki generator console bernama **artisan**, dengan menggunakan generator ini kalian bisa membuat struktural/pondasi file **MVC** tanpa membuat atau menulis kodenya secara manual. Contoh perintahnya seperti berikut :

**php artisan make:controller BlogController**

Perintah di atas akan membuat file controller baru dengan nama “BlogController”.

**COMPOSER**

Composer adalah package manager yang di miliki oleh bahasa pemrograman php, apabila temen2 pernah mendengar nama : maven, pip, gem, npm, bower, yarn dll. Itu adalah segelintir nama dari package manager milik bahasa pemrograman lain. Composer berfungsi untuk melakukan manajemen terhadap library atau package php yang bisa di gunakan untuk tujuan tertentu dari website kita.

**MVC (MODEL VIEW CONTROLLER)**

**MVC** adalah konsep desain pola yang di buat oleh creator framework untuk memisahkan kode pemrograman website agar bisa di pecah menjadi layer2 yang bisa lebih mudah di baca.

**Model** perfungsi untuk berinteraksi dengan layer database. **Controller** berfungsi sebagai pengatur logika dan berada di layer logic dari kode website kita atau juga bisa di bilang sebagai jembatan antara model dan view. **View** berfungsi sebagai layer yang berinteraksi dengan user, atau bisa di bilang di bagian layer tampilan (user interface).

**MVC LARAVEL**

Laravel memiliki konsep **MVC** , sama seperti konsep framework php lain. Mvc pada laravel tersimpan di beberapa direktory sebagai berikut :

* **Model** terdapat tepat di dalam folder **app**
* **Controller** terdapat di dalam folder **app/Http/Controller**
* **View** terdapat di dalam folder **resources/views**

**BLADE**

Blade adalah template engine yang di miliki oleh laravel, template engine memudahkan programmer untuk melakukan pengaturan template dan menulis kode php dengan lebih rapi dengan menggunakan directive yang di miliki oleh template engine ini, selain blade framework lain juga memiliki template engine seperty twig yang di pakai oleh symfony.

* **MEMBUAT PROJECT LARAVEL**

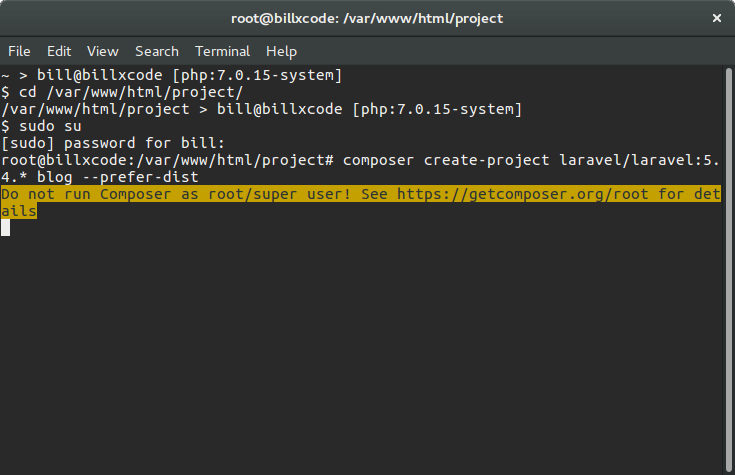
Untuk memulai project laravel ada dua cara yaitu menggunakan composer dan menggunakan laravel builtin installer.

**COMPOSER PROJECT**

Selain di pakai sebagai dependency manager, composer juga dapat di gunakan untuk melakukan perintah2 lainnya, seperti membuat project. Langkah langkah membuat project menggunakan composer di laravel adalh sebagai berikut :

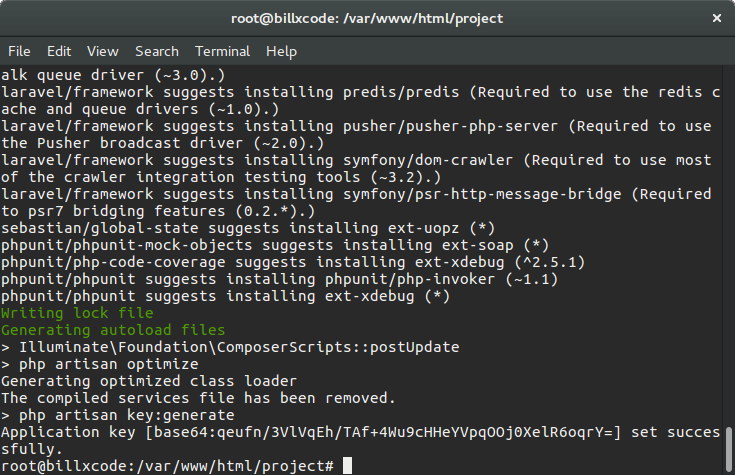
1. Buka terminal
2. Ketik perintah berikut di terminal

**composer create-project laravel/laravel:5.4.\* blog --prefer-dist**



Note : perintah di atas akan membuat folder blog dengan file project laravel di dalamnya

1. Done

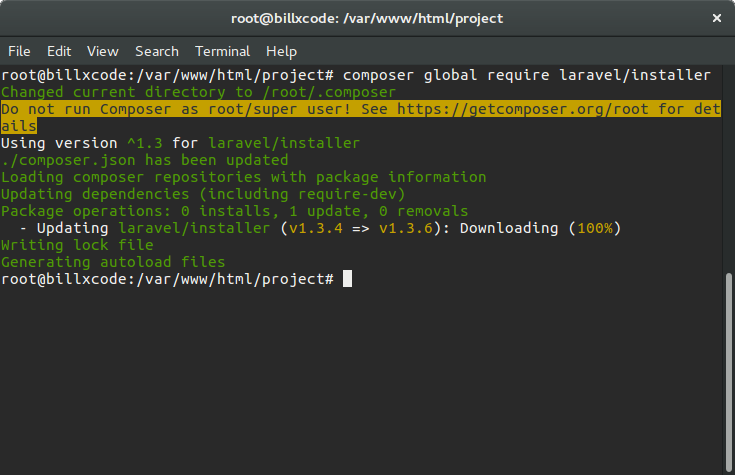


**LARAVEL INSTALLER**

Laravel installer adalah project generator yang di miliki oleh laravel, dengan menggunakan laravel installer kita bisa membuat project dengan sangat mudah. Sebelum menggunakan laravel installer, kita harus menginstall laravel installer terlebih dahulu.

1. Ketikkan perintah berikut di terminal

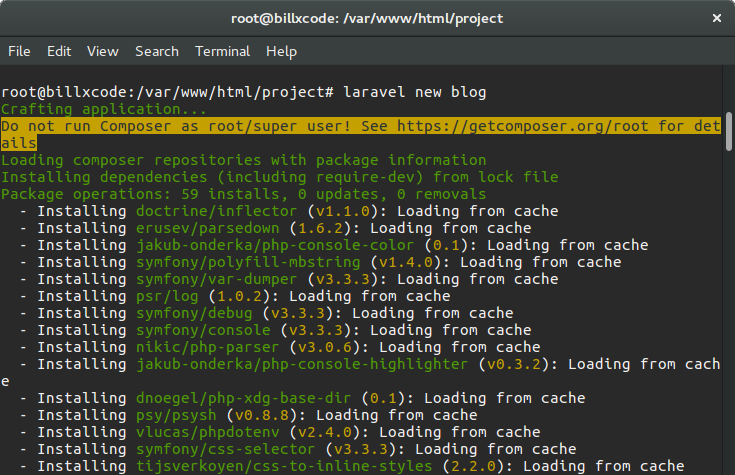
**composer global require laravel/installer**

**

*Note : perintah di atas akan menginstall laravel installer pada global mode*

1. Setelah proses selesai, buka termanal baru atau tab baru, lalu ketikkan perintah berikut

**laravel new blog**

**

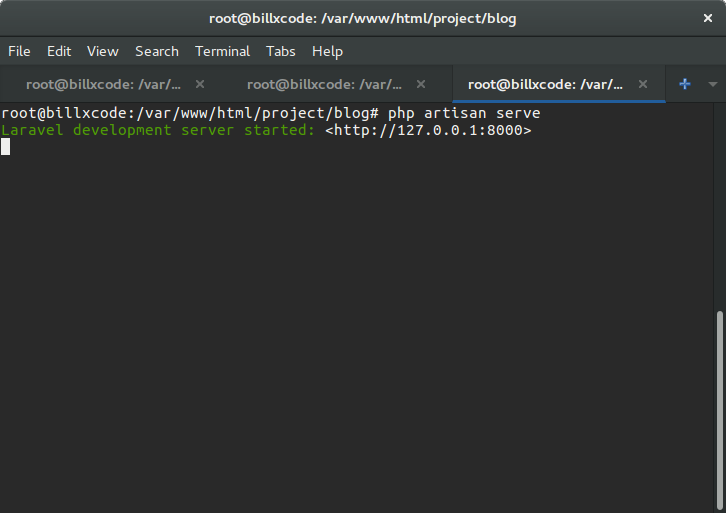
*Note : perintah di atas akan membuat project laravel baru dengan nama blog*

1. Done

**RUNNING LARAVEL SITE**

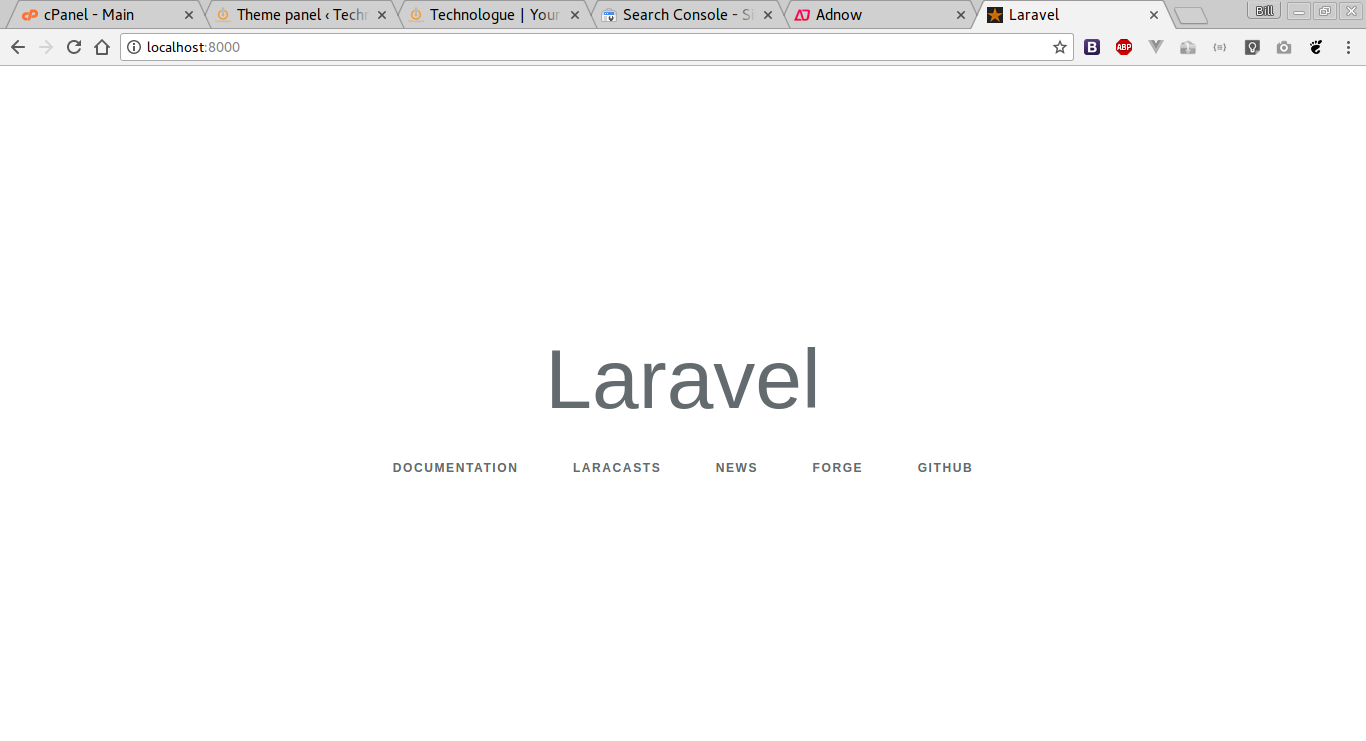
Cara menjalankan website laravel cukup dengan menggunakan laravel builtin server memanfaatkan menggunakan artisan console. Buka terminal lalu ketikkan perintah di bawah ini:

***php artisan serve***



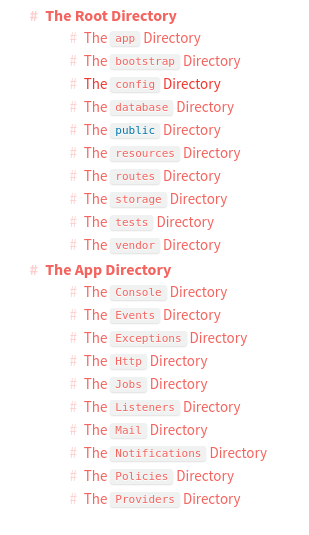
Note : laravel telah berada pada keadaan listening untuk menunggu request dari user melalui browser, kunjungi webnya pada <http://localhost:8000>

Hasilnya akan seperti berikut :



**MENGENAL PROJECT**

Project laravel yang telah kita buat memiliki banyak skali folder, setiap folder memiliki peran masing2 dan memiliki kekhususan masing2, mulai dari folder yang menyimpan MVC, sampai ke folder untuk melakukan testing. Berikut adalah struktur folder yang di miliki oleh laravel



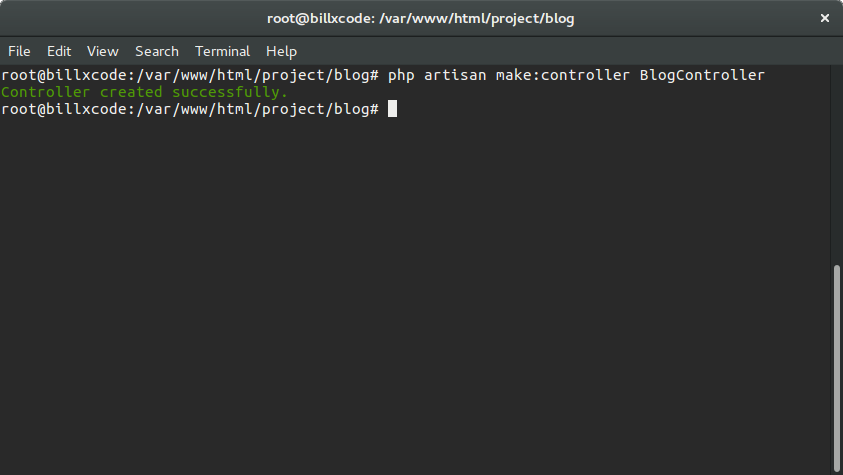
Pada bab selanjutnya kita akan memasuki bagian LARAVEL MVC, untuk mengenal pattern dari Model View Controller yang ada di framework laravel.

* **LARAVEL MVC**

**MEMBUAT CONTROLLER**

Pembuatan controller kita lakukan menggunakan generator laravel yaitu artisan . Ketikkan perintah berikut pada terminal anda

**php artisan make:controller blogcontroller**

****

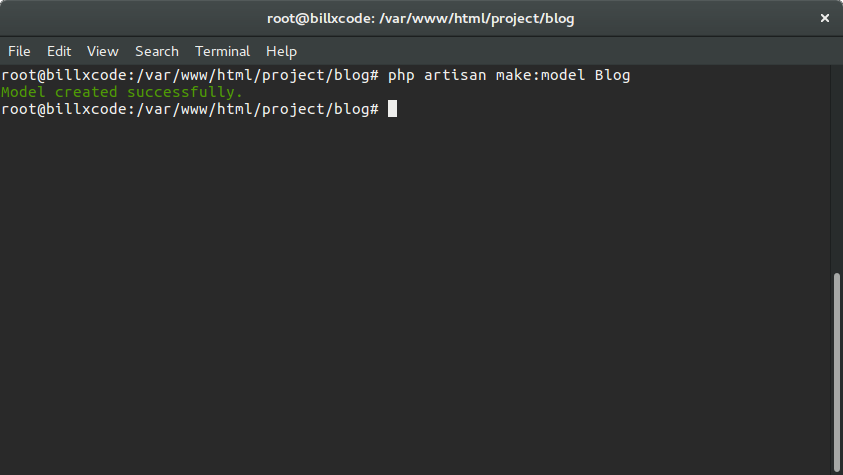
**Gambar 2.0**

Hasil yang di dapat dari perintah di atas adalah file baru bernama blogcontroller yang terdapat pada folder **app/http/controller.**

**MEMBUAT MODEL**

Pembuatan model juga kita lakukan menggunakan generator laravel. Ketikkan perintah berikut di terminal anda.

**php artisan make:model blog**

****

Hasil yang di dapat dari perintah di atas adalah file baru bernama **blog** di dalam folder **app.**

**MEMBUAT VIEW**

Pembuatan view tidak bisa dilakukan menggunakan artisan seperti ada pada model dan controller, view di laravel harus di buat secara manual, hal ini di sengaja oleh taylor ottwel, tetapi sampai saat ini pun saya belom tau *“kenapa kq seperti ini”.*

Untuk membuat view di laravel, anda buka folder **resources/views** lalu create new file dengan nama “blog” yang berarti anda harus membuat file dengan nama seperti berikut **blog.blade.php**.Kenapa kq ekstensi sebelum php ada kata **blade**? Karna jika kita ingin menggunakan directive yang di miliki blade, kita harus menambahkan additional extension dengan nama blade.

Pada bab selanjutnya kita akan memasuki bagian LARAVEL ROUTER, untuk mengetahui penamaan dan pengalamatan url pada website laravel.

* **LARAVEL ROUTER**

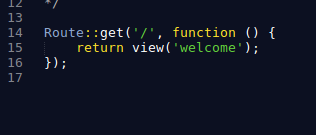
**ROUTER**

Router adalah salah satu bagian terpenting dalam menggunakan framework laravel, router berfungsi sebagai pengatur url dari website kita, hampir sama dengan router pada jaringan yaitu berfungsi untuk memberikan jalan ke pada suatu aliran data, router pada laravel memberikan jalan request data untuk mendapatkan response dari server melalui URL, contoh apabila kita ingin memberikan url pada website kita seperti ini <http://yourwebsite.com/your/url/site> . Maka di router kita tuliskan *“/your/url/site”* sebagai url yang nantinya akan bisa mengeksekusi perintah di website kita.

Sebelum kita masuk ke kodingan controller model dan view, kita akan belajar router terlebih dahulu. Router pada laravel 5.4 tersimpan pada folder **routes** disana terdapat file web.php, app.php, auth.php dll. Semua file tersebut adalah file router, file router utama yang akan kita pakai adalah file web.php. Untuk lebih lanjut akan masuk ke subbab berikutnya.

**ROUTER WELCOME**

Pada saat pertama install laravel kalian pasti mendapati halaman laravel welcome, nah halaman tersebut sebenernya adalah halaman yang ada di folder view, yang telah di render oleh router laravel. Kode router yang memunculkan laravel welcome terdapat pada file web.php di folder routes.



kode router di atas artinya url yang kita akses adalah root, lalu mengeksekusi fungsi yang isinya merender view bernama welcome.

**ROUTER PART I**

Ketikkan kode router pada langkah langkah di bawah ini

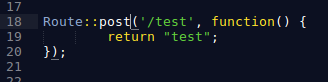
**Router level 1**

Router method GET [http://localhost:8000/test](http://blog.com/test)



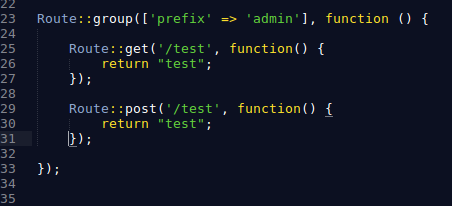
**Router level 2**

Router method POST



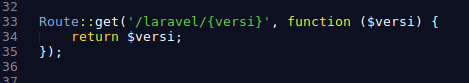
**Router level 3**

Router group yaitu router dengan banyak router di dalamnya



**Router level 4**

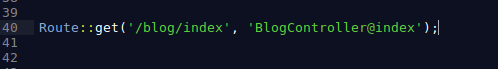
Parameter router, yaitu memberikan custom parameter di router kita



*Note : versi adalah nama parameter yang terdapat pada laravel, sehingga apabila kita memasukkan parameter pada url seperti ini* ***<http://localhost:8000/laravel/bill>*** *maka yang muncul adalah kata bill.*

**Router level 5**

Router untuk memanggil controller yang ada



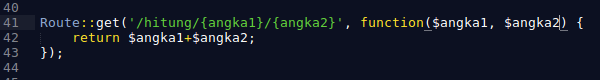
*Note : pada parameter kedua di dalam route terdiri dari nama class controller dan method di dalamnya, karna tersusun dengan* ***BlogController@index****, maka classnya adalah* ***BlogController*** *, methodnya* ***index***

**ROUTER PART II**

Setelah memahami subbab router part I, kita akan mulai sedikit mencoba eksploitasi ilmu para router ini, agar kita semakin faham guna dan fungsi dari router.

**Aritmatik dengan router**

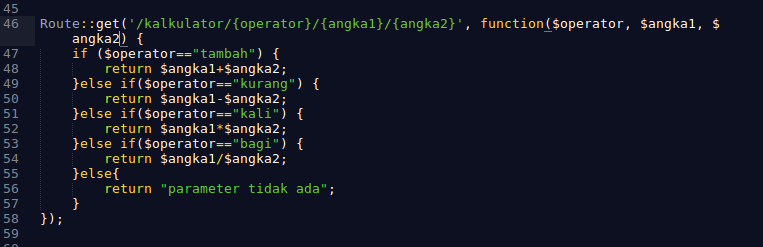
Menghitung menggunakan router dan parameter



**Aritmatik lanjut**

Pada bab selanjutnya kita akan memasuki bagian pemrograman php, untuk mengetahui fungsi dari file yang telah kita buat dari artisan sebelumnya.

Kalkulator menggunakan router

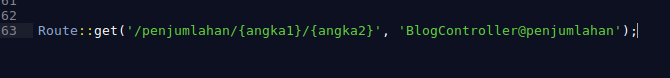


*Note : parameter operator akan menjadi pointer untuk menentukan operator apa yang akan di gunakan untuk mendapatkan hasil dari perhitungan angka1 dan angka2*

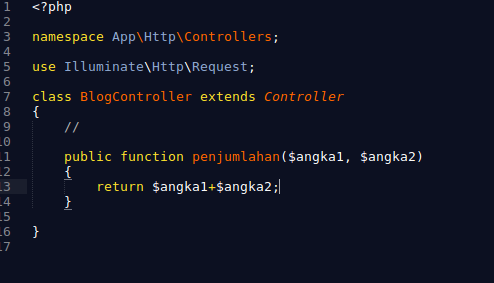
**Router memanggil controller**

Sama seperti pada router part I, kita akan mencoba memanggil controller untuk menghitung penjumlahan dari parameter yang kita kirimkan.

* **web.php**

****

* **BlogController.php**

****

*Note : alur yang terjadi di atas di mulai dari router dengan alamat penjumlahan/{parameter}/{parameter}, lalu di koneksikan ke BlogController lalu di tangkap oleh method penjumlahan .*

**Router dengan POST Method**

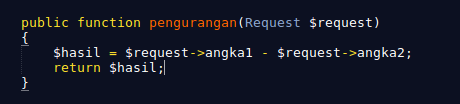
Router dengan post method adalah router yang di gunakan untuk menghandle request yang menggunakan metode POST, atau biasanya di gunakan oleh form2 website contoh : register, login, profile dll.

Jika ingin menggunakan post method pada router kita tinggal mengganti kata GET menjadi POST pada router yang telah kita pelajari di atas.

* **Web.php**

Screenshot from 2017-07-06 16-36-01

* + **BlogController.php**



*Note : method di atas di taruh tepat di bawah method penjumlahan*

Pada subbab ini tidak perlu di lakukan tes pada kode di atas, cukup pahami lalu lanjut ke bab selanjutnya. Untuk melakukan percobaan pada kode di atas kita harus menggunakan aplikasi pihak ke tiga yaitu **postman.**

Pada bab selanjutnya kita akan memasuki bagian LARAVEL BLADE, untuk mengetahui tentang fungsi directive dan mengatur halaman pada laravel.

* **LARAVEL BLADE**

Blade adalah template engine milik laravel, seperti yang sudah di sebutkan pada bab sebelumnya, blade terdiri dari banyak directive, yang akan kita pelajari pada bab ini.

Directive pada laravel sebenarnya adalah representasi dari kode php yang di miliki oleh php, guna directive sendiri adalah untuk mempermudah penggunaan notasi dan embeding code php dalam template atau file view yang kita buat. Blade juga membuat kode view kita lebih bersih dan mudah di baca oleh programmer lain apabila kita kolaboratif.

Pada subbab di bawah ini akan di bahas tentang beberapa notasi atau directive yang sering di pakai di blade.

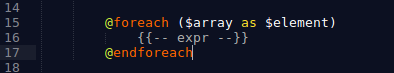
**PRINT**

Php menyediakan banyak sekali metode untuk melakukan perintah cetak, pada blade semua itu di persempit lagi hanya dengan menggunakan *double curly branch,* berikut penulisannya

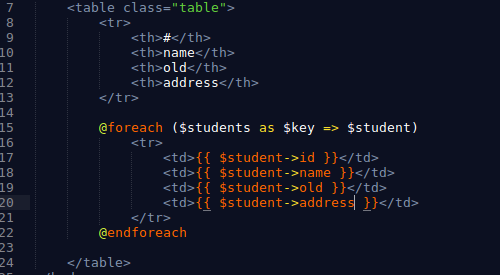
Screenshot from 2017-07-07 20-11-50

**INTERATION**

Perulangan di dalam php terdapat 3 cara yaitu for, while, do while. Dan ada satu tambahan iterasi untuk data array yaitu foreach. Pada blade perulangan yang akan di pakai adalah foreach, tetapi dengan directive yang di tetapkan oleh blade, penulisannya seperti berikut

****

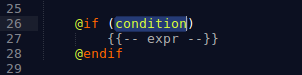
Contoh kasus yang sering di pakai adalah pada iterasi suatu data table, berikut contoh kode programnya

****

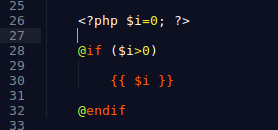
Kode di atas berarti bahwa, data pada variable $**students** akan di masukkan ke variable $**key** dan di lemparkan ke variable $**student**, lalu variable $**student** akan di pecah mendari banyak object seperti pada kode program di atas.

**DECISION**

Percabangan pada bahasa pemrograman PHP memiliki 2 tipe yaitu, *if else* dan *switch case*, walaupun terdapat dua tipe pada php, blade hanya menyediakan directive pada **if else**, berikut penulisan kode programnya.

****

Contoh kasusnya seperti berikut

****

**INHERITANCE**

Penurunan sifat atau inheritance adalah salah satu jenis fungsi dari OOP untuk dapat menurunkan metode atau variable yang ada di class parent ke class childs, secara menakjuban blade menyediakan fitur tersebut untuk dapat di gunakan di bagian template atau view. Skema yang di jalankan oleh blade sama saja yaitu penurunan atribute, penurunan content template parent ke template childs. Lebih mudahnya lihat diagram di bawah ini.

**Parent**

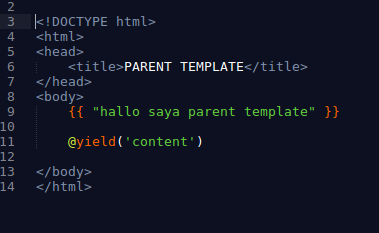
**Child**

**Child**

**Child**

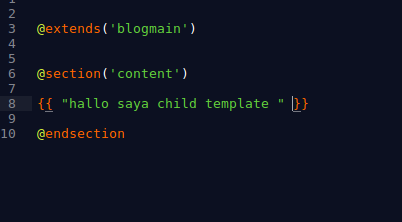
Setelah memahami diagram di atas, kalian bisa lanjut untuk membaca kode programnya di bawah ini. Parent template di representasikan oleh file **blogmain.blade.php** lalu child template bernama **dashboardblog.blade.php**

* + **PARENT TEMPLATE**

****

@**yield(‘content’)** berarti bahwa kita menyediakan bidang pada **parent template** untuk di miliki oleh **child template**

* + **CHILD TEMPLATE**

****

@**extends(‘blogmain’)** berarti kita melakukan penurunan dari template blogmain ke template child yang sekarang yaitu dashboardblog.

@**section(‘content’)** berarti kita mengambil bidang content yang ada di dalam parent template untuk di isi sesuai kebutuhan child template.

@**endsection** berfungsi untuk mengahiri scope section

Pada bab selanjutnya kita akan memasuki bagian koding CRUD LARAVEL , untuk memulai project kecil pertama menggunakan laravel.

Pada bab selanjutnya kita akan memasuki bagian RESTFUL METHOD LARAVEL, untuk memulai pengerjaan project pada laravel framework.

Pada bab selanjutnya kita akan memasuki bagian koding CRUD LARAVEL , untuk memulai project kecil pertama menggunakan laravel.

Pada bab selanjutnya kita akan memasuki bagian CRUD @ LARAVEL, untuk memulai pengerjaan project pada laravel framework.

* **RESTFUL METHOD LARAVEL**

Bab kali ini kita akan memulai untuk membuat project yang bisa di pakai oleh calon pengguna, project yang akan kita buat adalah project *to do list .* Project ini akan memiliki basic CRUD (Create Read Update Delete), dimana crud adalah basic concept yang harus ada di suatu website dinamis. Berikut definisi CRUD:

|  |  |
| --- | --- |
| **METHOD** | **DEFINITION** |
| CREATE | metode yang di gunakan untuk memasukkan data kedalam table database melalui kode program. |
| READ | metode yang di gunakan untuk membaca data pada suatu table untuk di munculkan di halaman website. |
| UPDATE | metode yang di gunakan untuk melakukan pembaharuan data pada suatu table. |
| DELETE | metode yang di gunakan untuk menghapus data pada table database. |

Laravel mendukung penuh dalam pembuatan website dengan basic CRUD, mulai dari dukungan *HTTP Verb* untuk membedakan setiap metode CRUD pada request data, lalu Restful Generator, yang memudahkan kita untuk membuat kode program berbasis Restful pada website kita, sehingga kode program lebih readable dan simple.

HTTP Verb adalah metode request data pada protocol HTTP yang di gunakan untuk membedakan proses CRUD terjadi, jadi skema yang di buat oleh HTTP Verb adalah apabila kita menggunakan verb tertentu maka akan merujuk pada response tertentu juga pada website kita. Berikut adalah table http verb.

|  |  |
| --- | --- |
| **METHOD** | **DEFINITION** |
| GET | Request data yang bertujuan mengambil data |
| POST | Request data yang bertujuan mengirim data |
| PUT/PATCH | Request data yang bertujuan memperbarui data |
| DELETE | Request data yang bertujuan menghapus data |

Setelah membaca table di atas, coba kalian perhatikan dan bandingkan table crud dan table http verb, setiap point pada crud berasosiasi dengan http verb, kedua jenis method ini memiliki tujuan yang sama tetapi berbeda level layer. Method crud bertempat di layer aplikasi, sementara http verb bertempat di layer network.

Berikut adalah table asosiasi dari table crud dan table http verb

|  |  |
| --- | --- |
| **METHOD CRUD** | **METHOD HTTP VERB** |
| CREATE | POST |
| READ | GET |
| UPDATE | PUT/PATCH |
| DELETE | DELETE |

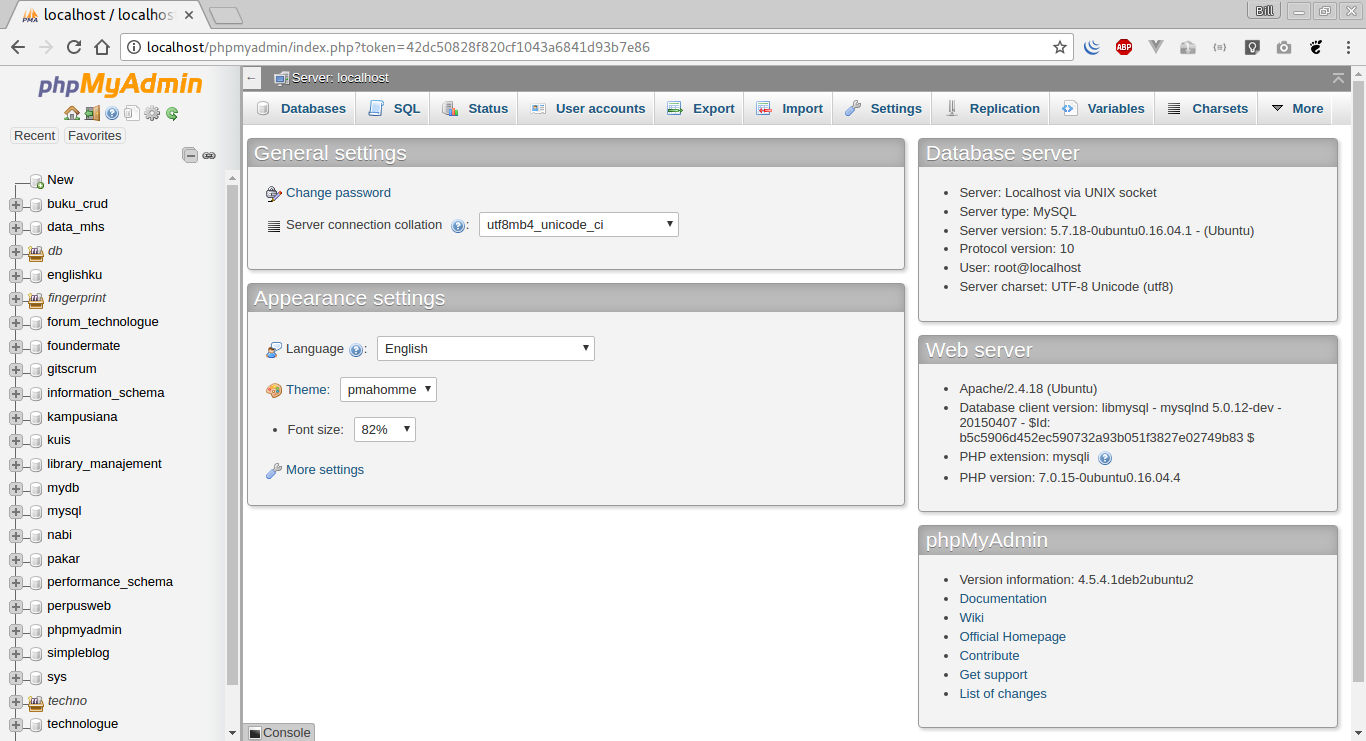
Setelah memahami tiap method di atas, selanjutnya kita akan mencoba untuk membuat project laravel berbasis CRUD dan HTTP Verb.

**CREATE DATABASE**

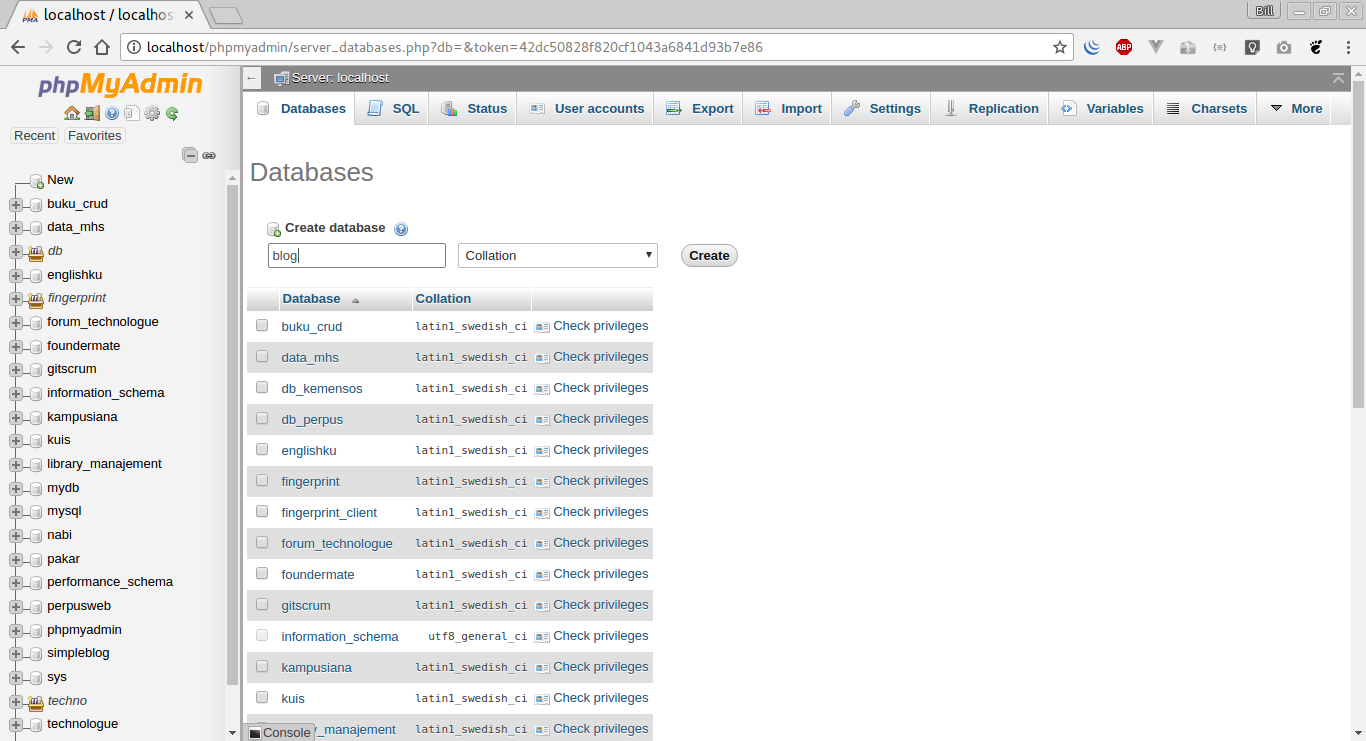
Pada module ini DBMS yang akan kita pakai adalah mysql, laravel juga sudah mendukung banyak dbms seperti sqlite, postgresql, dan sql server. Untuk membuat database Pertama tama kalian harus membuka tools RDBMS mysql yaitu PHPMYADMIN. Atau jika kalian lebih suka menggunakan console, kalian bisa langsung membuatnya dengan perintah pada console mysql yaitu CREATE DATABASE <name>.

Langkah langkahnya sebagai berikut:

1. Buka <http://localhost/phpmyadmin/> pada browser chrome



1. Goto databases tab, then create database name **blog**

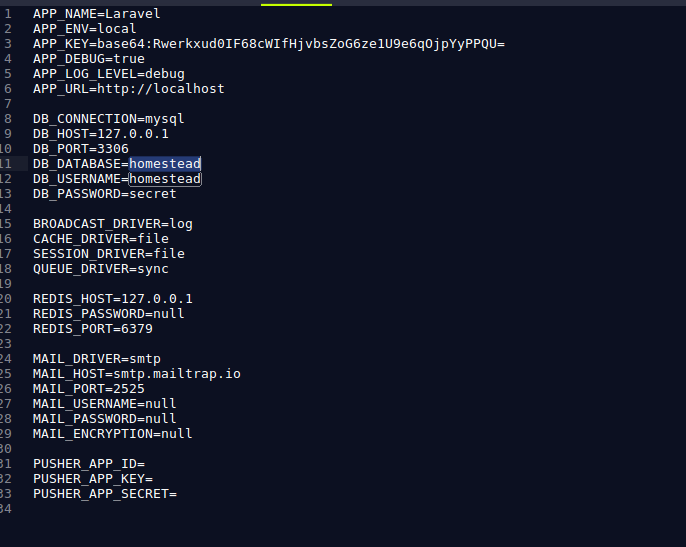


1. Klik tombol **create**

*Note : Setelah klik tombol create maka database telah kita buat, selanjutnya kita akan membuat file MVC dan Router sesuai CRUD dan HTTP Verb.*

Apabila database sudah di buat kita selanjutnya akan melakukan konfigurasi terlebih dahulu pada aplikasi laravel kita. Konfigurasi yang akan kita lakukan semua berpusat pada file .env, file ini adalah file private yang hanya boleh di ketahui oleh developer dan tidak boleh di taruh di sembarang folder, isi dari file .env adalah semua informasi sensitif yang berkaitan dengan authentification, name apps, password dll. Pada kali ini kita akan melakukan konfigurasi pada beberapa point saja untuk kebutuhan website kita.

Seperti yang telah di jelaskan di atas, .env file berisi informasi sensitif, informasi sensitif ini di bungkus dalam sebuah variable global yang nantinya akan di akses oleh website yang kita buat. Berikut adalah variable variable global yang terdapat pada .env file.



Pada file .env di atas kita akan melakukan konfigurasi pada blok database yaitu pada variable ber prefix **DB** , variable yang akan kita ganti adalah DB\_DATABASE, DB\_USERNAME dan DB\_PASSWORD. Berikut konfigurasi database yang akan saya buat.

|  |  |
| --- | --- |
| **VARIABLE GLOBAL** | **VALUE** |
| DB\_DATABASE | blog |
| DB\_USERNAME | root |
| DB\_PASSWORD | root |

*Note : konfigurasi di atas tidak selalu sama dengan konfigurasi yang ada pada kondisi kalian, terkadang value pada DB\_PASSWORD kosong karna kebanyakan orang tidak menggunakan password pada mysql nya.*

**CREATE MIGRATION**

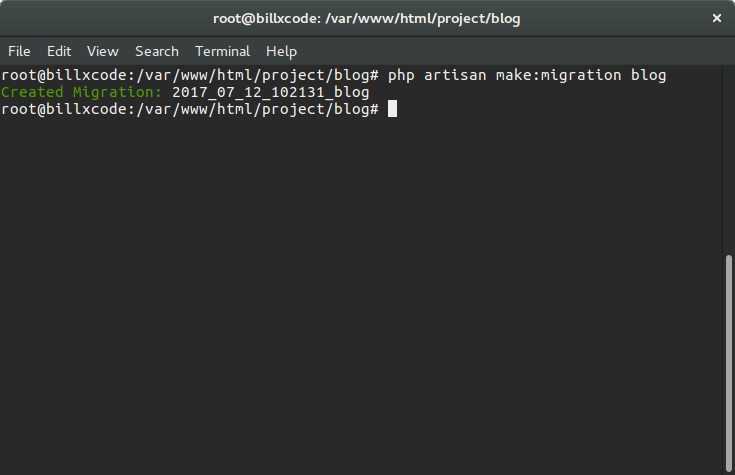
Migration adalah kode program yang berfungsi untuk melakukan manipulasi table, berbeda dengan model yang mengatur isi dari table. Kegunaan paling dasar dari migration adalah membuat struktur dari table, mulai dari name column, sampai relational table. Setiap file migration mewakili satu table tetapi satu table bisa di wakilkan beberapa migration.

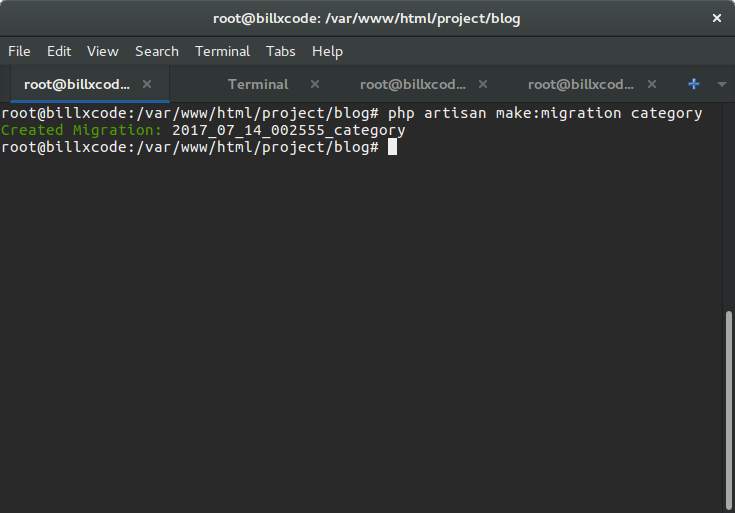
Pada kali ini kita akan membuat migration sesuai dengan kebutuhan CRUD yang akan kita buat.migration yang akan kita buat adalah blog dan category Ikuti langkah langkah di bawah ini

1. Buka terminal
2. Ketikkan perintah berikut

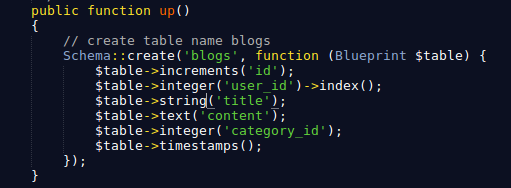
**#php artisan make:migration blog**

**#php artisan make:migration category**

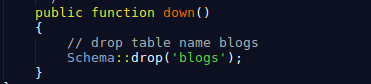




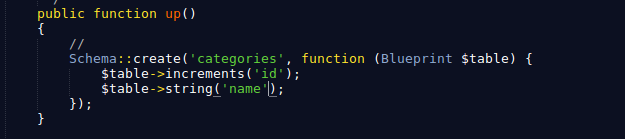
1. Setelah itu buka file migration tersebut pada folder **database/migrations** , di dalam folder tersebut sudah ada beberapa migration selain file yang sudah kita buat tadi, yaitu migration folder dan password\_reset. Buka file migration **blog**, lalu ketikkan kode berikut pada method **up()**.



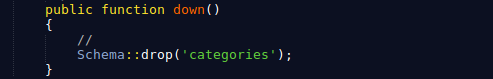
1. Lalu ketikkan kode berikut pada method **down()**



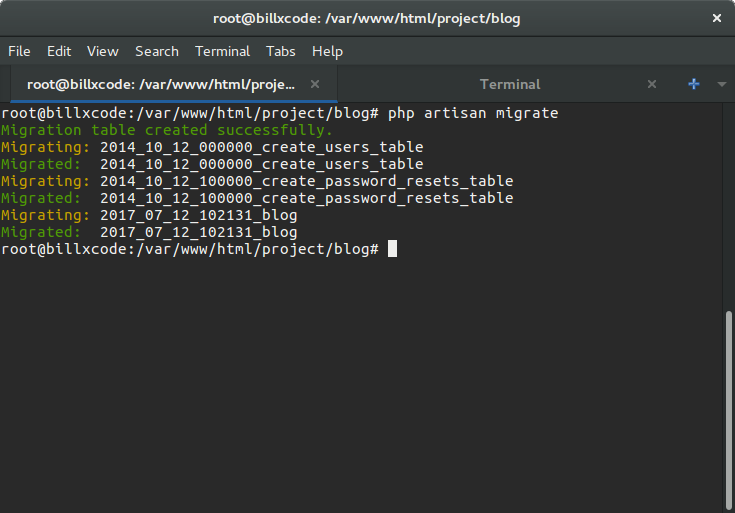
1. Stelah itu buka file migration category lalu ketikkan kode berikut di method **up()**

****

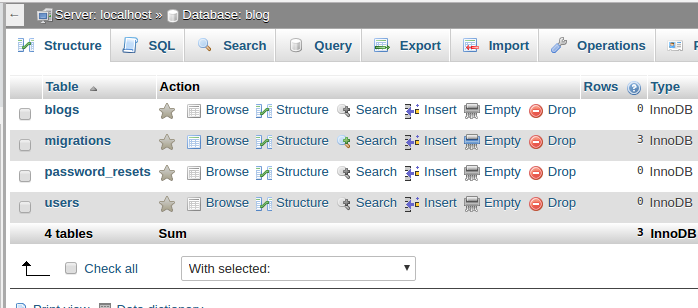
1. Lalu ketikkan kode berikut di method **down()**

****

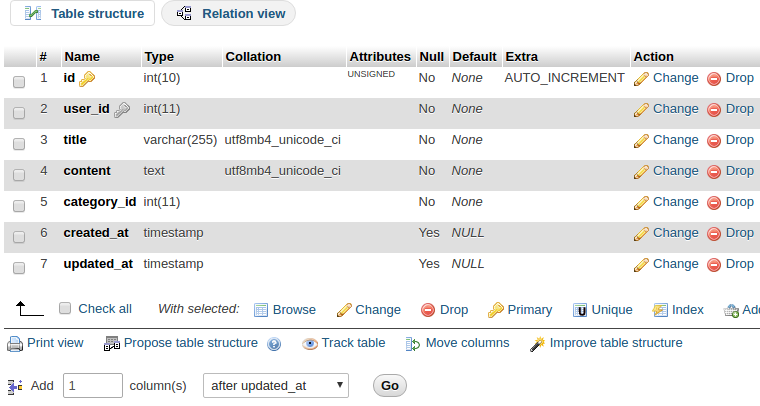
1. Setelah itu kita akan melakukan eksekusi untuk kode program yang telah kita tulis tadi, ketikkan perintah berikut pada terminal



1. Setelah selesai, coba periksa apakah table **blogs** sudah terbuat pada database **blog**.



1. Lalu priksa apakah kolomnya juga sudah terbuat juga.



Seperti yang kita lihat table dan kolom yang telah kita buat yaitu blogs, telah muncul, tetapi jika kita perhatikan pada database blog muncul table selain table blog, yaitu users dan password\_reset, table ini adalah table bawaan migration pada website laravel yang di buat pada saat kita memulai projectnya, jadi tidak perlu khawatir karna kita nanti juga akan menggunakan table tersebut.

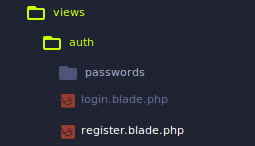
**CREATE AUTHENTIFICATION**

Authentification di butuhkan untuk membatasi suatu fungsi agar hanya bisa di akses oleh orang yang terdata dalam website kita, contohnya adalah login dan register, pada kali ini kita akan membuat login dan register dengan memanfaatkan instan command dari php partisan. Auth ini nantinya akan membatasi aksi CRUD hanya bisa di lakukan oleh user yang sudah login, Seperti hal kalian membuat blog dari blogger, kalian harus login dulu sebelum mulai untuk menulis.

Ketikkan perintah di bawah ini di terminal

**php artisan make:auth**

Perintah di atas, akan langsung melakukan generate controller, dan view yang di butuhkan untuk melaksanakan authentification, contohnya login form, register form, home page, dan controller login register.



**CREATE MODEL**

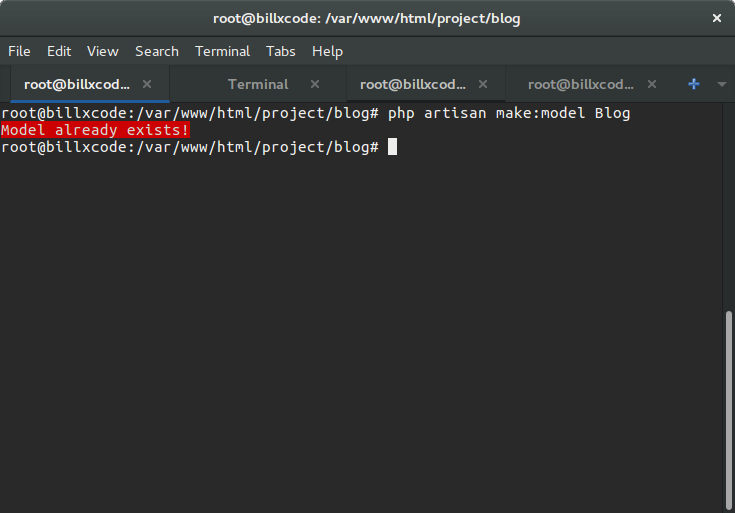
Model adalah bagian yang berfungsi untuk melakukan manipulasi data pada table database. Operasional yang bisa di lakukan oleh model adalah terbatas pada layer DML (Data Manipulation Language), pada layer ini model laravel hanya bisa melakukan modification data yang ada di dalam table, seperti INSERT, SELECT, UPDATE, DELETE.

Walaupun terbatas di layer DML, model laravel dapat melakukan relational yang notabene hanya bisa di lakukan pada layer DCL (Data Control Language) yaitu menggunakan fitur Eloquent ORM (Object Relational Model), dengan menggunakan Eloquent ORM model laravel dapat merelasikan setiap table dengan table lain baik itu one to one, one to many atau many to many. Tetapi perlu di ketahui bahwa relational pada Eloquent ORM tidak tertanam pada layer database, Eloquent ORM hanya tertanam pada layer aplikasi, jad bisa di ketahui bahwa ORM dapat melakukan relational pada table tetapi tidak permanen di dalam databasenya. Apabila table yang tidak ada relasinya tadi di pakai oleh website lain maka relasinya tetap tidak ada.

Manipulation pada laravel bisa di lakukan dengan dua cara, yaitu dengan menggunakan Eloquent ORM atau menggunakan query bulider yaitu DB Facade. Walaupun penggunaan DB Facade pada laravel lebih mudah di mengerti, tetapi sangat tidak di rekomendasikan untuk di pakai pada project besar, apabila kalian ingin mempelajari tentang DB Facade, silahkan buka dokumentasi berikut.

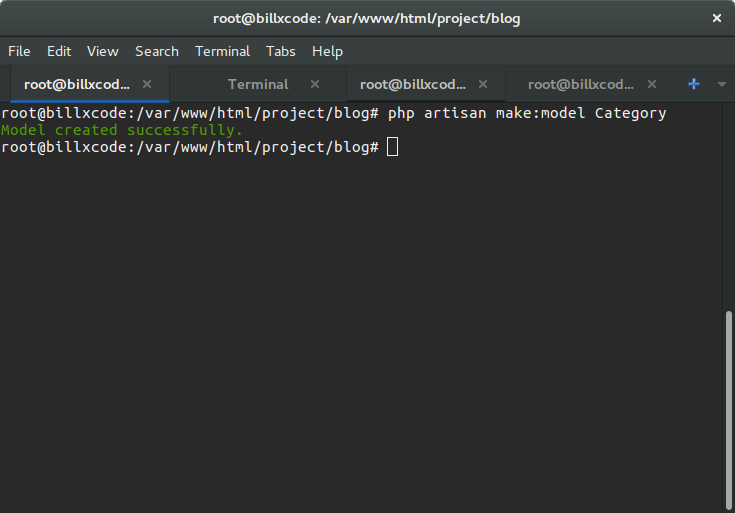
Pada project ini kita akan sedikit menyinggung tentang model, menggunakan Eloquent ORM dengan beberapa relasinya. Ikuti langkah langkah di bawah ini untuk membuat model yang sesuai dengan migration yang telah kita buat tadi.

1. Buka terminal
2. Buat model blog dengan perintah berikut



*Note : apabila kalian mengalami error di atas, anda tidak perlu khawatir karna kita telah membuat model blog pada langkah di bab sebelumnya*

1. Setelah itu buat model Category dengan perintah berikut

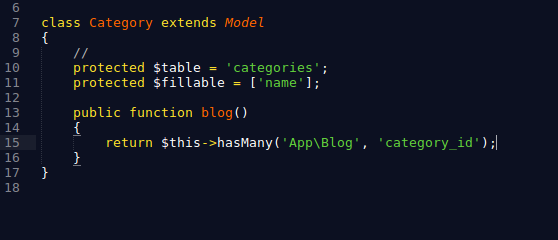


*Note : apabila anda mendapatkan hasil seperti di atas, maka model telah sukses di buat*

1. Setelah semua model di buat, kita akan mulai memberikan beberapa baris kode seperti konfigurasi table dan sedikit relasi. Ketikkan kode berikut di dalam model Blog.



1. Lalu ketikkan kode berikut di model **Category**



Setelah melakukan langkah langkah di atas berarti kalian telah selesai membuat model untuk project kita kali ini, kita akan memanfaatkan model yang telah kita buat untuk melakukan operasional berbasis database, yang nantinya kita panggil dari sisi controller.

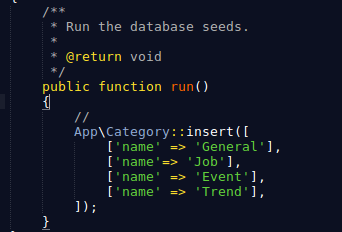
**CREATE SEEDER**

Seeder adalah file yang berfungsi untuk memberikan konten pada setiap migration yang telah kita buat, syarat agar seeder sukses di eksekusi adalah migration harus sudah ada dan harus sesuai dengan migration yang akan di tuju.

Ketikkan perintah berikut pada terminal untuk membuat seeder pada migration category.

**php artisan make:seeder categoryseeder**

Setelah perintah di atas sukses di jalankan, maka file categoryseeder akan muncul di direktori *database/seed,* buka file seedernya lalu ketikkan kode di bawah ini pada method run().



Kode di atas berfungsi untuk memasukkan data general, job, event, dan trend kedalam table categories ketika perintah seeding di eksekusi. Setelah kalian menuliskan kode di atas, maka lanjukan dengan menuliskan perintah di bawah ini di terminal untuk mengeksekusi seeding.

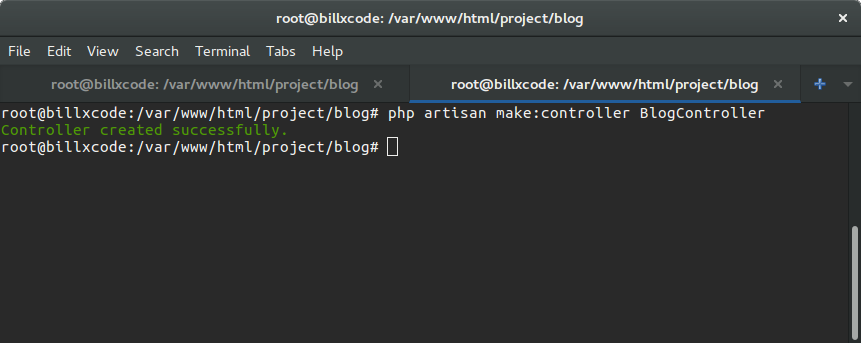
**php artisan db:seed --class=categoryseeder**

Perintah di atas, akan mengeksekusi seeding pada class categoryseeder, hasilnya adalah table categories akan berisi data general, job, event, trend pada kolom name.

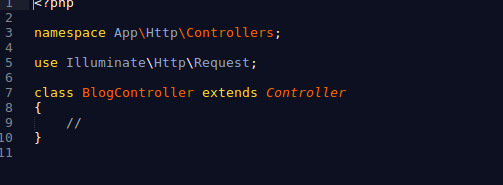
**CREATE CONTROLLER**

Controller bertugas sebagai pengatur logika dan jembatan dari model ke view. Pada kali ini controller yang akan saya buat berfungsi menangani perintah CRUD dari model Blog. Kita namakan controller blog kita sebagai **BlogController**, ketikkan perintah berikut ini di terminal untuk membuat controller menggunakan artisan.

**php artisan make:controller BlogController**

****

Hasil dari perintah artisan di atas akan langsung ada di dalam directory **app/http/controller**, dengan nama file BlogController.php , berikut adalah isi dari file **BlogController.php**.

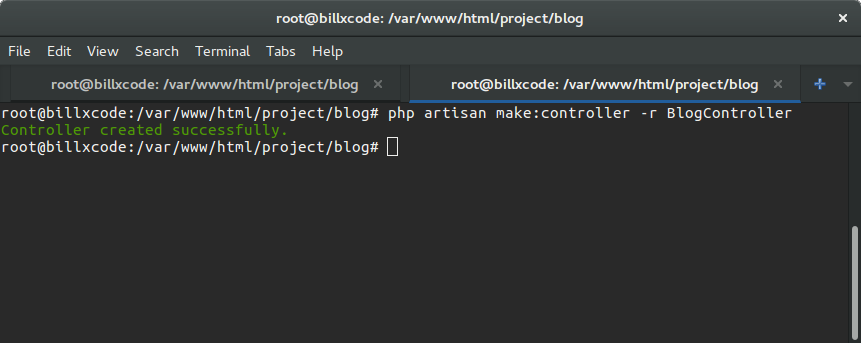


Setelah selesai membuat file BlogController maka selanjutnya kita akan membuat method berbasis CRUD di dalamnya. Perlu di perhatikan lebih lanjut bahwa method yang akan kita buat pada controller blog ini akan berasosiasi dengan method HTTP Verb, dan beberapa akan menangani view saja. Berikut adalah method yang akan kita buat pada controller blog.

|  |  |
| --- | --- |
| **METHOD** | **KEGUNAAN** |
| INDEX | Berfungsi memunculkan data dari table blog |
| CREATE | Berfungsi memunculkan form posting blog |
| STORE | Berfungsi menyimpan data yang telah di kirimkan melalui form posting |
| EDIT | Berfungsi memunculkan form posting blog dengan data spesifik |
| UPDATE | Berfungsi untuk mengupdate data yang di kirimkan oleh form edit posting |
| DELETE | Berfungsi untuk menghapus data spesifik |

Method di atas adalah method berbasis Restful yang di standardkan oleh laravel, untuk membuat controller berbasis restful dengan method yang sudah terisi semua kita tinggal menambahkan argument (-r) pada perintah artisan sebelumnya, hasilnya akan menjad seperti berikut.

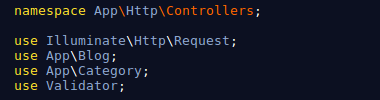
**php artisan make:controller -r blogcontroller**

****

Hasil dari perintah artisan di atas adalah file **BlogController.php** tetapi dengan keadaan berisi baris baris method restful. Berikut hasilnya

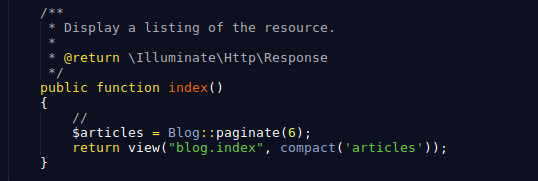


Sebelum menuliskan kode pada tiap methodnya tambahkan beberapa library terlebih dahulu di atas class blogcontroller, seperti pada gambar di bawah ini.

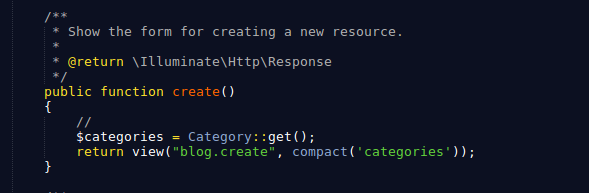


Apabila langkah di atas telah berhasil kalian lakukan, maka selanjutnya kita akan menuliskan baris kode di setiap method yang sudah ada. Ketikkan kode di bawah ini sesuai dengan nama methodnya.

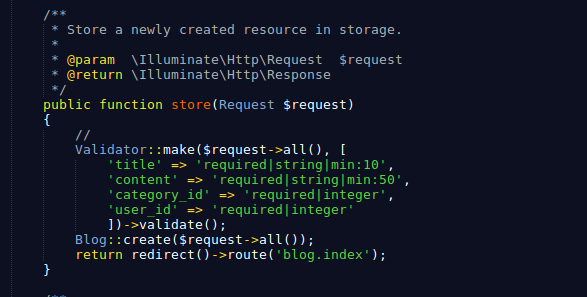
* INDEX



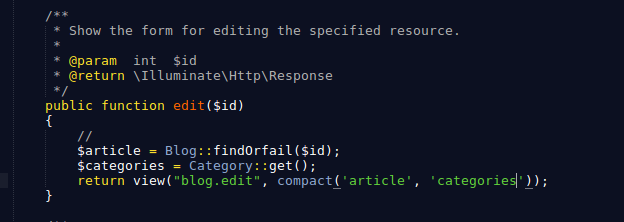
* CREATE



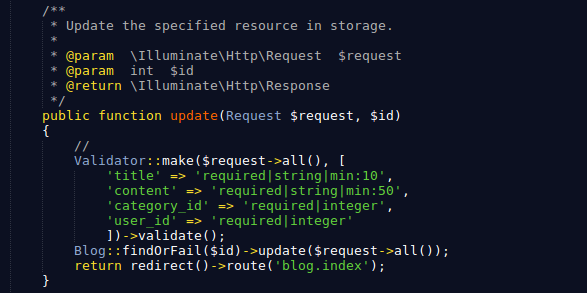
* STORE



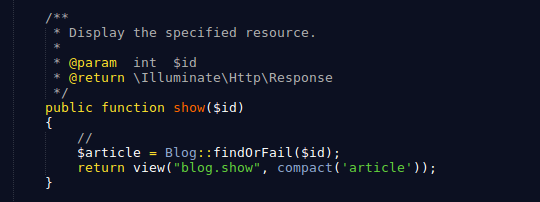
* EDIT



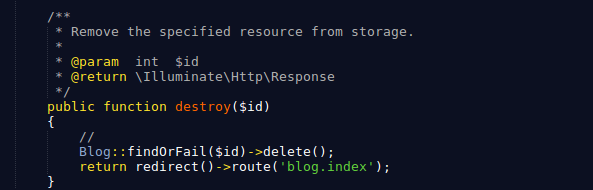
* UPDATE



* SHOW



* DELETE



**CREATE VIEW**

View pada laravel akan memanfaatkan template engine bernama blade dengan ketentuan harus menyisipkan extensi blade pada setiap file nya. File blade akan di panggil oleh controller agar dapat di akses, setiap file blade yang di panggil controller akan di compile menjadi file dengan nama yang telah di enkripsi tipe md5 dan isinya telah di compile menjadi tag html dan kode embeded php. Setiap file blade yang telah di compile akan di tempatkan di folder storage, maka dari itu folder storage harus bisa di akses dengan permission read and write agar setiap compile bisa berjalan lancar.

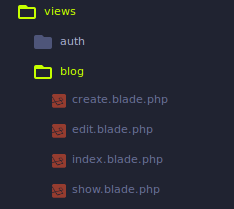
File view yang akan kita buat semuanya merujuk pada pemanggilan yang ada di controller, semua penamaan file yang ada di view akan menyesuaikan dengan apa yang di panggli oleh method di controller. Pemanggilan file view pada controller memanfaatkan library view(), contohnya pada method INDEX pada class BlogController, kalian akan menjumpai kode berikut

return view(“blog.index”, compact(‘articles’));

Perhatikan kode tersebut pada method view, terdapat 2 statement kode yang di pisahkan oleh koma, statement pertama adalah nama file view, statement kedua adalah data yang akan di lempar ke dalam file view. Perhatikan pada nama file view yaitu “blog.index”, nama tersebut berarti bahwa file view bernama index.blade.php yang berada di dalam folder blog pada resources/view. Jadi kalian harus membuat folder baru dengan nama blog terlebih dahulu, baru kalian membuat file dengan nama index.blade.php.

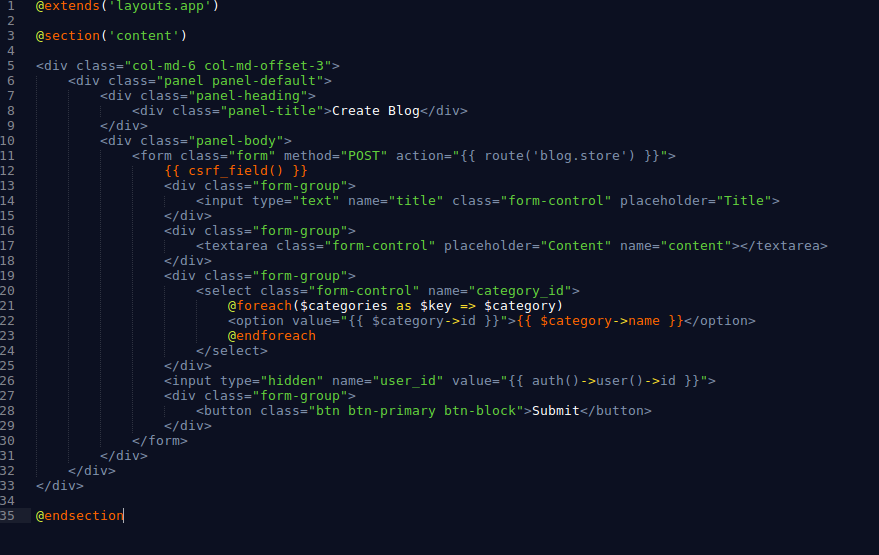
Setelah kalian sukses membuat file index, lakukan hal yang sama pada method lain yang memanggil method view, yaitu method create, edit, dan show.

Hasilnya akan seperti berikut :

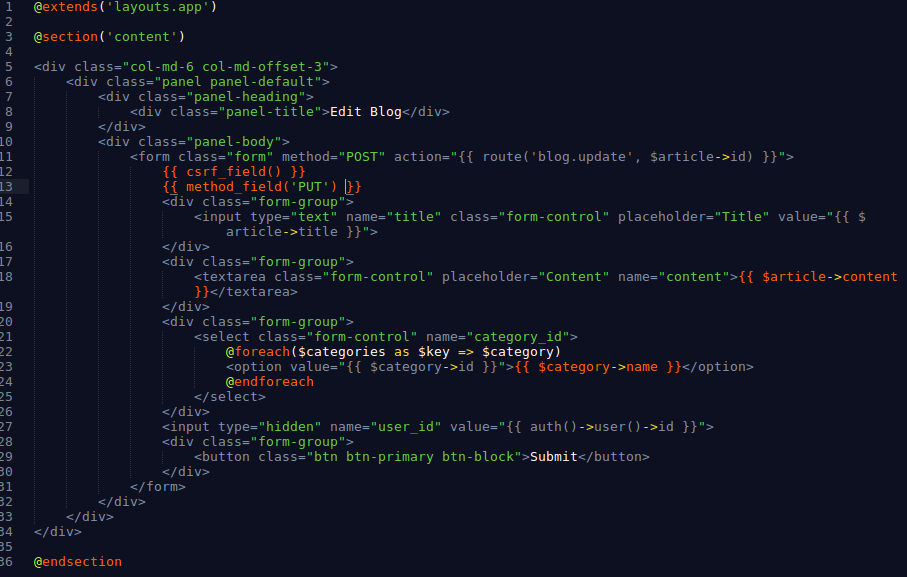


Setelah kalian sukses membuat semua filenya, maka selanjutnya kalian harus menulis kode programnya pada setiap filenya, berikut kode program tiap filenya.

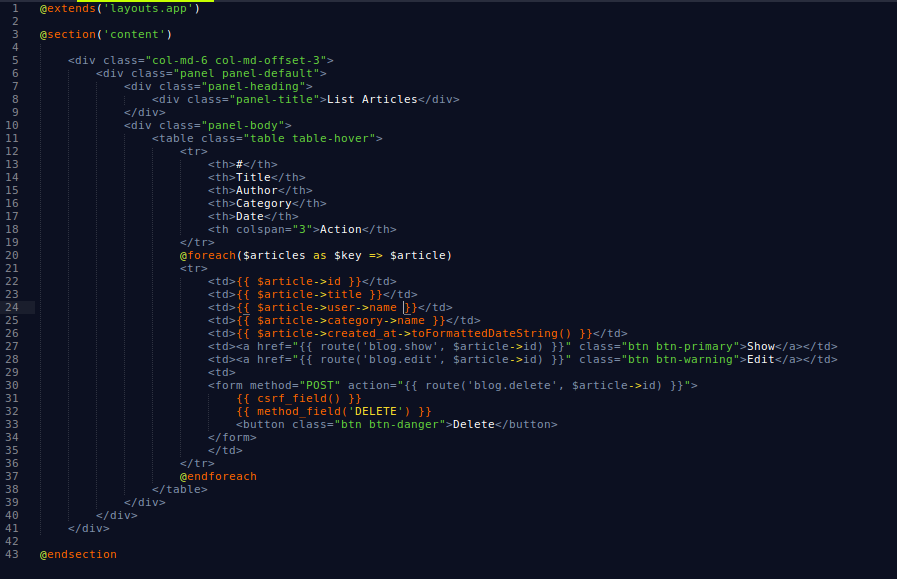
* CREATE



* EDIT



* INDEX

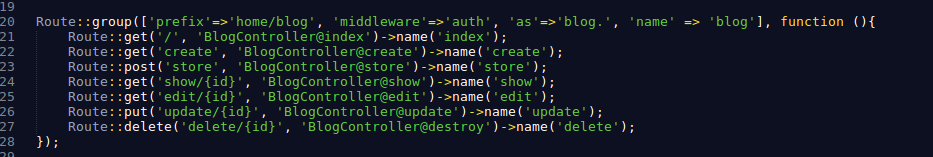


* SHOW



**CREATE ROUTER**

Router berfungsi untuk memberikan jalan agar controller yang kita buat bisa di akses oleh user dengan menggunakan url yang kita tentukan di router, pada project ini, router yang akan kita pakai semuanya berbasis RESTFUL, yaitu memakai HTTP Verb (POST, GET, PUT/PATCH, DELETE). Berikut baris kode dari dari router yang akan kita buat.



Kriteria dari router di atas adalah sebagai berikut

|  |  |  |  |
| --- | --- | --- | --- |
| # | **Parameter** | **Value** | **Description** |
| 1 | prefix | home/blog | Berarti url router yang akan kita pakai semuanya di awali dengan home/blog |
| 2 | middleware | auth | Berarti router dalam home/blog hanya bisa di akses oleh user yang sudah login |
| 3 | as | blog. | Berarti router dalam home/blog memiliki nama dengan awalan blog. |
| 4 | name | blog | Berarti router home/blog memiliki nama blog |

Seperti yang bisa kita lihat pada tiap kode router di atas, router group berfungsi untuk mengelompokkan router router yang sejenis, keunggulannya jika kita menggunakan router group maka kode kita lebih rapi dan mudah di baca, dan lebih efisien untuk penulisan kodenya.

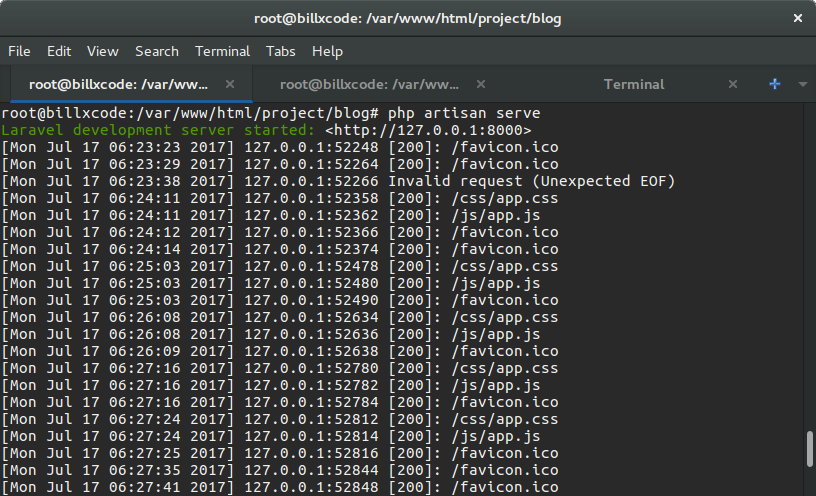
Tiap router di dalam router group blog menggunakan method sesuai HTTP Verb dan berasosiasi dengan method CRUD di dalam class BlogController, setelah kalian menuliskan semua router yang ada, maka kalian telah menyelesaikan project CRUD website blog pada kali ini. Silahkan masuk ke bab berikutnya untuk melakukan testing terhadap website yang telah kalian buat.

Pada bab selanjutnya kita akan melakukan testing terhadap web yang sudah kita buat. **LAUNCH THE KRAKEN!!**

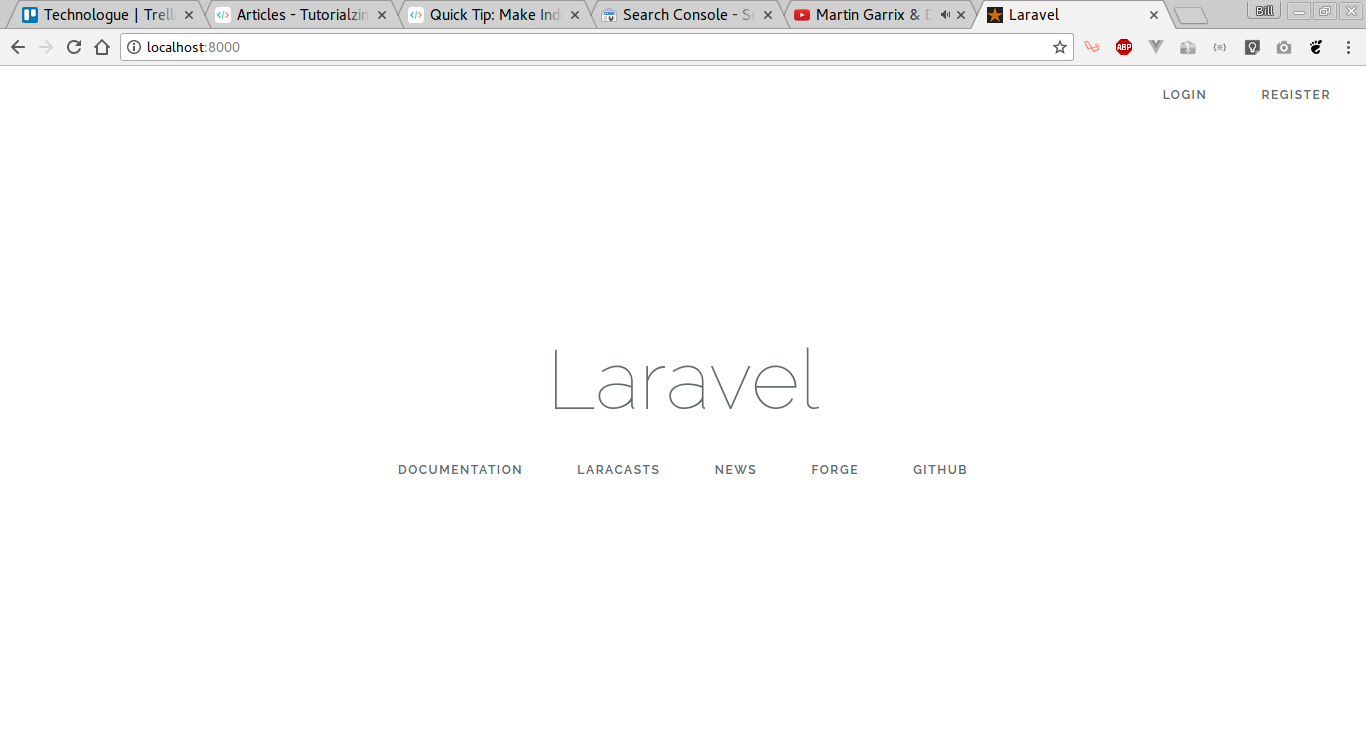
* **LAUNCH THE KRAKEN**

Setelah kalian melakukan semua langkah di tiap bab sebelumnya dengan sukses maka tandanya kalian telah menyelesaikan pembuatan website ini dengan baik, apabila masih ada error maka periksa dulu langkah mana yang belom kalian lakukan. Berikut adalah perintah untuk menjalankan website kita, **ITS TIME TO LAUNCH THE KRAKEN!!!**

**php artisan serve**

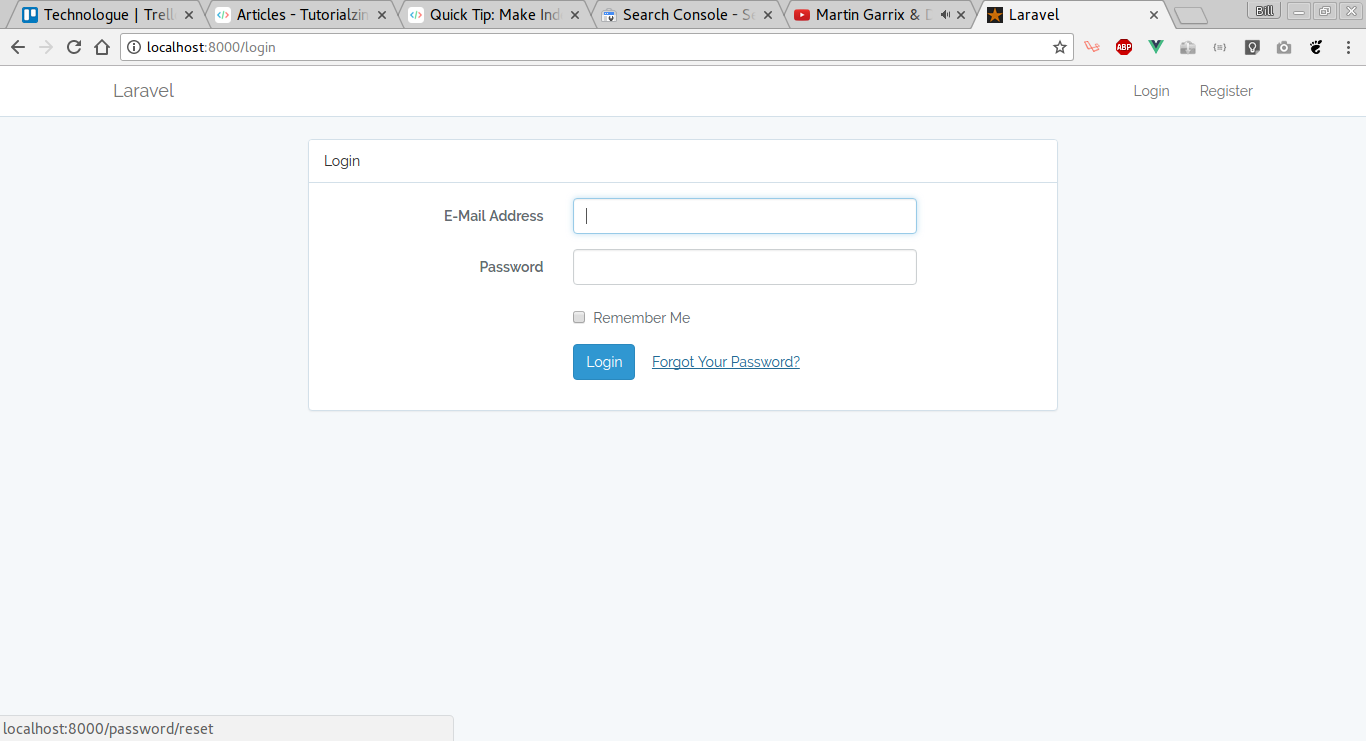
****

Buka browser dan kunjungi url localhost:8000, maka kalian akan menjumpai tampilan seperti berikut

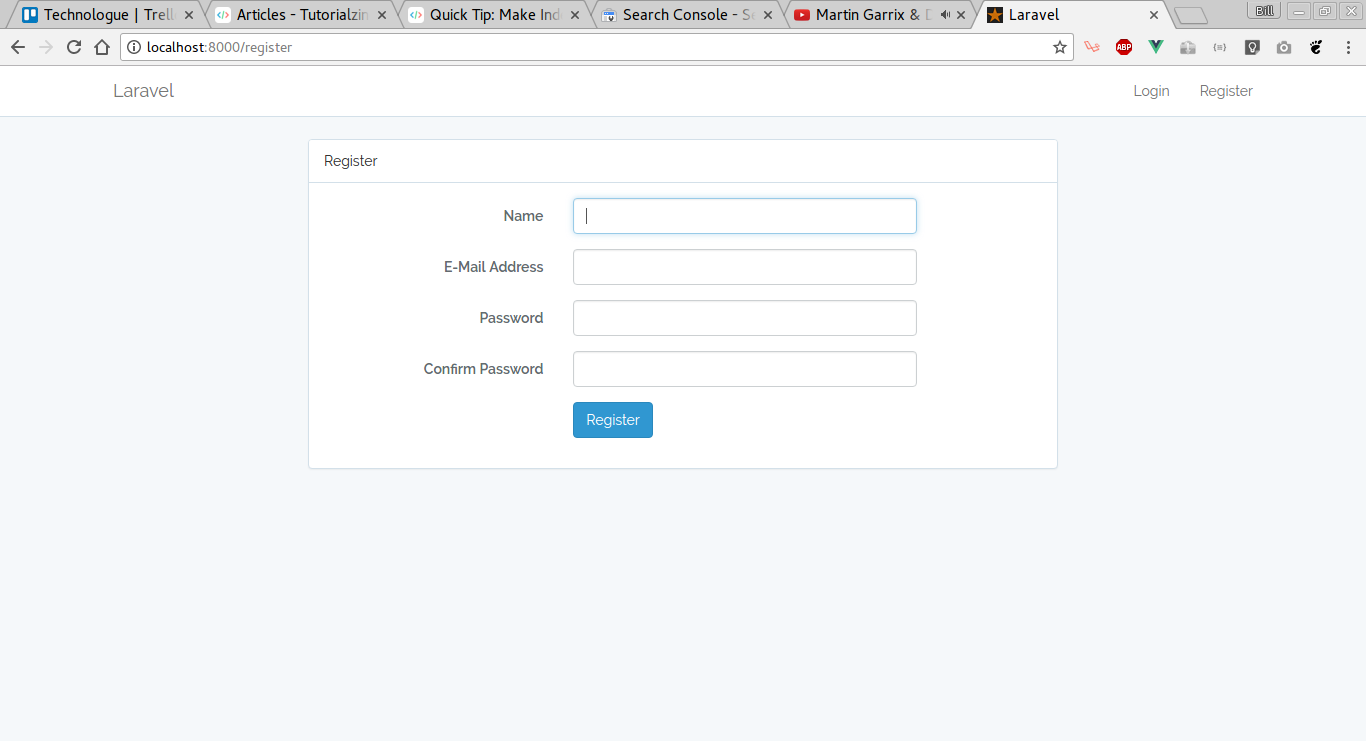


Terdapat button login dan register pada pojok kanan atas, apabila kalian klik maka akan masuk ke laman login dan register seperti berikut

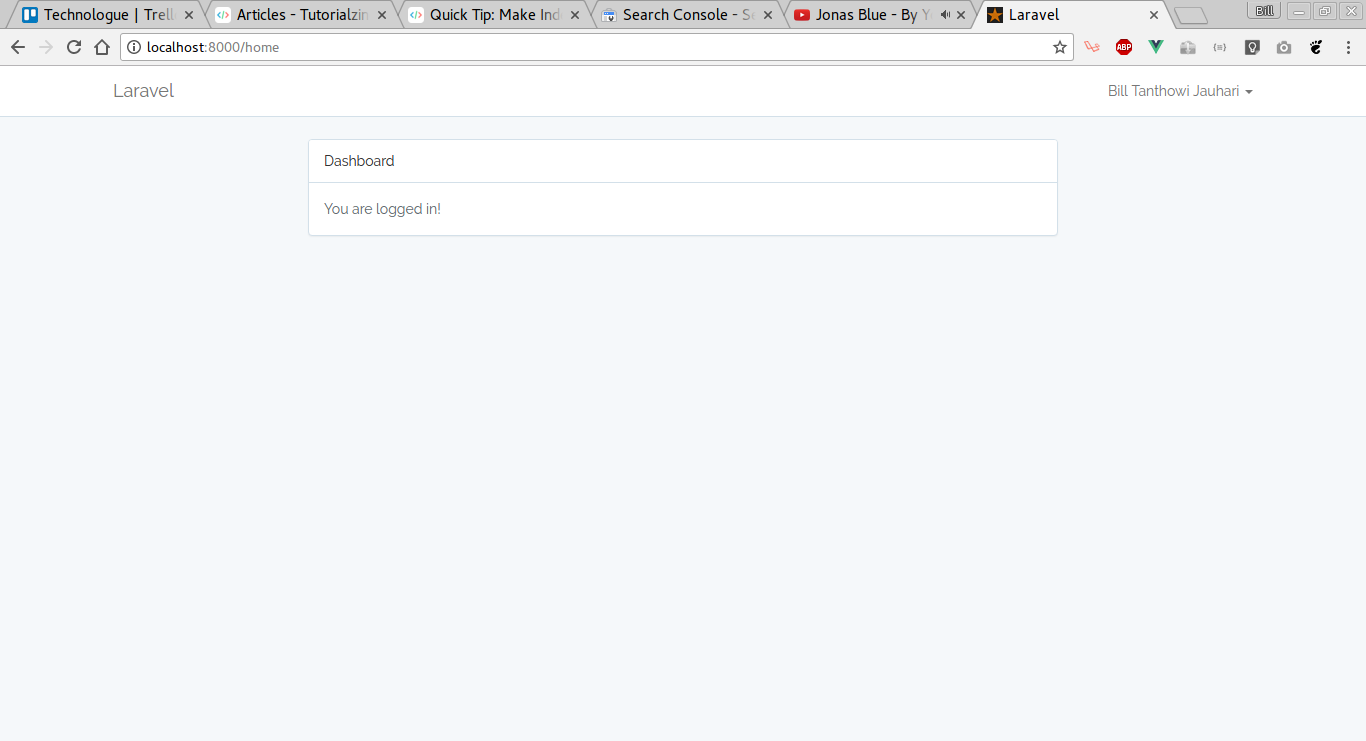
* LOGIN



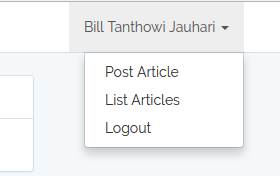
* REGISTER



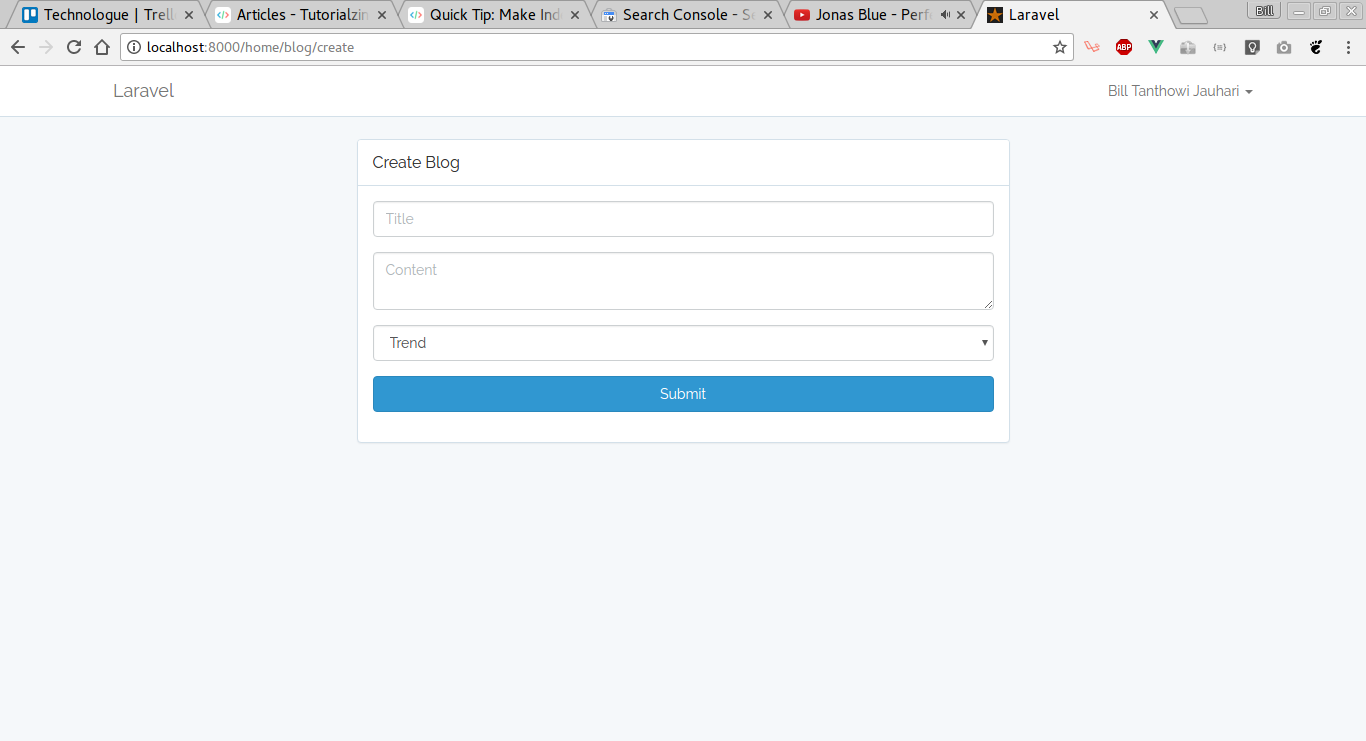
Silahkan buat akun dan login agar bisa masuk halaman home. Setelah kalian bisa login maka akan masuk home seperti berikut



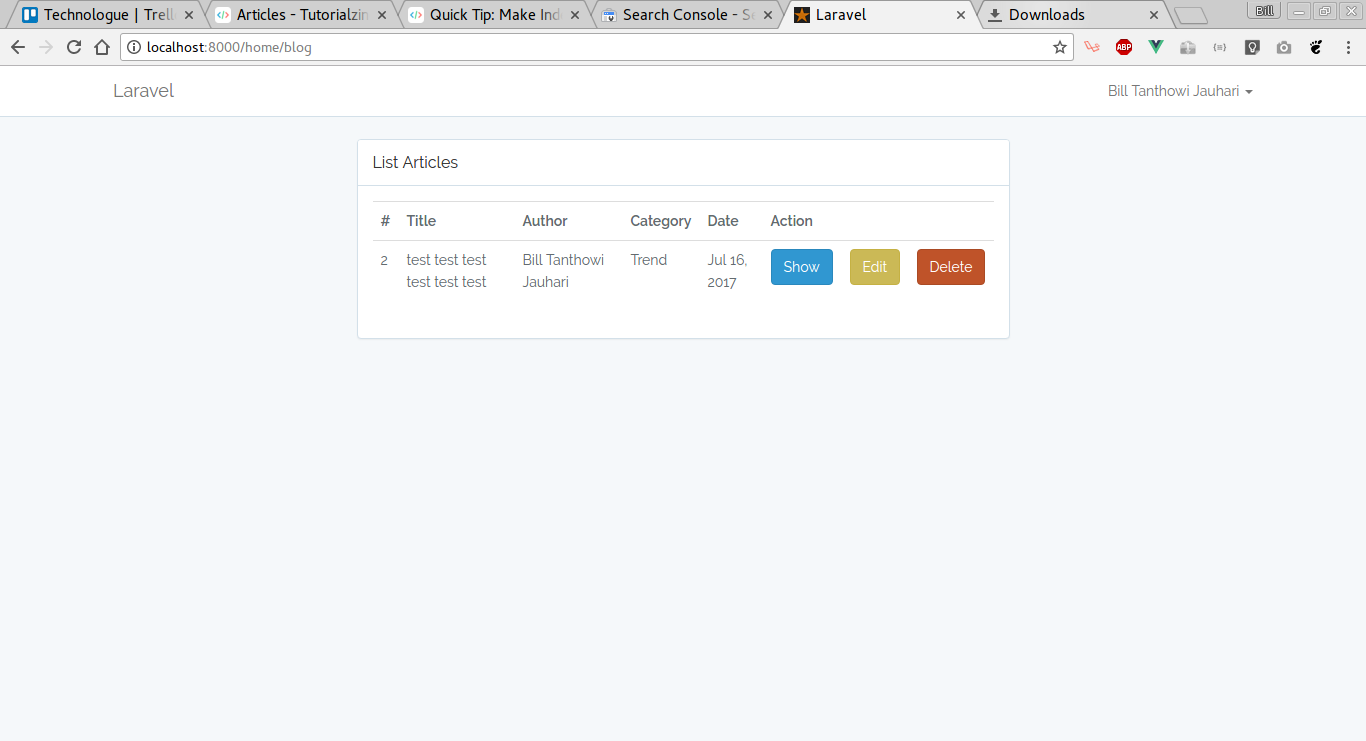
Setelah berhasil masuk home, silahkan kalian coba membuat postingan baru, arahkan pointer ke pojok kanan atas dimana ada nama kalian pada dropdown menu.



Pilih post article, setelah itu kalian akan masuk ke halaman pembuatan post article dengan url home/blog/create.



Apabila kalian berhasil maka akan muncul datanya seperti berikut



Setelah itu coba lakukan beberapa aksi pada button yang sudah di sediakan di daftar postingan di atas. Happy blogging guys. Selanjutnya kalian tinggal mencoba untuk mengupload projectnya ke hosting atau server sebenarnya.

* **KESIMPULAN**

Inti dari pembuatan project pada umumnya adalah berbasis CRUD, tetapi tidak semuanya di pakai, tetapi tidak mungkin lebih dari CRUD. Jadi apabila kalian ingin menulis baris kode yang baik dan benar, tentukan kode program CRUD pada setiap table.

**KRITIK DAN SARAN**

Saya sangat mengharapkan kritik dan saran yang membangun dari para pembaca, buku ini sangat jauh dari kesempurnaan. Silahkan kirim kritik dan saran ke email saya **bill.tanthowi.j@gmail.com**.

* **MENGENAL PENULIS**

Saya adalah seorang web developer di bidang backend dengan menggunakan bahasa PHP sebagai bahasa utama dalam bekerja. Framework php yang telah saya pelajari adalah codeigniter, laravel, slim, lumen dan symfony. Dalam bekerja saya cenderung menggunakan framework dari laravel base, seperti laravel framework dan lumen. Bahasa pemrograman yang saya suka adalah php, ruby, python dan arduino.

Saya mulai mendalami di dunia web developing sejak tahun 2014, dan mulai mengenal framework di tahun 2015-2016. Framework pertama yang saya pelajari adalah codeigniter, dari framework tersebut saya mulai mengenal dunia framework, composer, packagist dll.

Sejak module ini di tulis saya sudah mulai berkarir di perusahaan digital newsletter di jakarta, bekerja remote dan sambil menyelesaikan perkuliahan saya. Sebelumnya juga sempat bekerja di perusahaan software house di surabaya selama kurang lebih 1 tahun.

Moto hidup saya adalah *pantang menyerah dalam berusaha, tetap bersyukur akan rezeki yang Alloh SWT berikan.*