CATATAN PEMBELAJARAN

TEKNIK INFORMATIKA (TI)



DISUSUN OLEH :

WIDI ARROHMAN

3042019006

TEKNIK INFORMATIKA

POLITEKNIK NEGERI KETAPANG

2020

PRAKATA

Puji dan syukur penulis panjatkan kepada Allah SWT atas selesainya makalah yang berjudul " KONSEP BASIS DATA (KBD)". Atas dukungan moral dan materi yang diberikan dalam penyusunan makalah ini, maka penulis mengucapkan banyak terima kasih kepada :

1. Bapak Eka Wahyudi, S.Pd., M.Cs selaku ketua jurusan Teknik Informatika, yang memberikan bimbingan, saran, dan nasehat kepada kami selaku mahasiswa.

2. Ibu Rizqia Lestika Atimi, S.T,M.T, selaku ketua program studi Teknik Informatika, yang memberikan semangat kepada penulis.

3. Teman-teman kelas Teknik Informatika A, yang banyak memberikan bantuan kepada sang penulis.

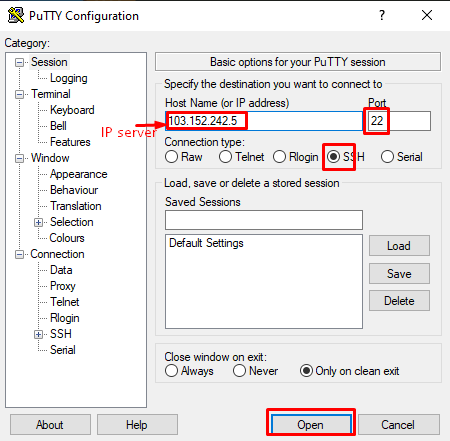
Penulis menyadari bahwa makalah ini belumlah sempurna. Oleh karena itu, saran dan kritik yang membangun dari dosen dan rekan-rekan sangat dibutuhkan untuk penyempurnaan makalah ini.

Ketapang , 3 November 2020

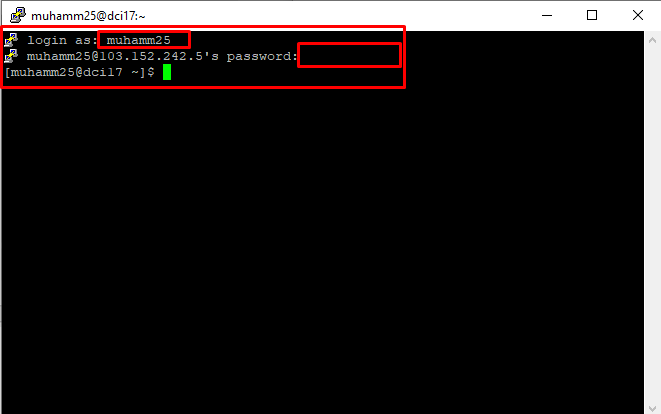
Widi Arrohman

BAB I SERVER

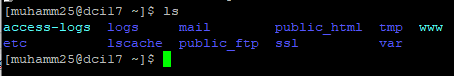
1. Koneksi SSH server dengan aplikasi Putty
2. Koneksi ke server



1. Masukkan username dan password untuk melakukan remote



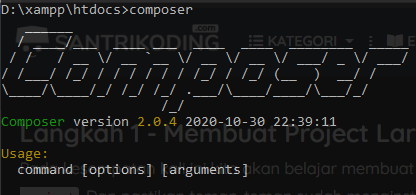
1. Koneksi berhasil



1. Judul 2
2. Judul 3

BAB II CRUD LARAVEL 8

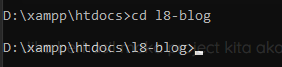
1. Instalasi laravel 8
2. Pastikan sudah meng-install composer



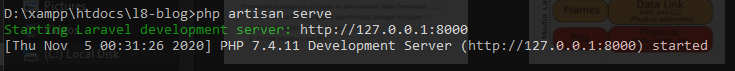
1. Ketik “**composer create-project –prefer-dist laravel/laravel l8-blog**”

C:\Users\Widi Arrohman\Documents\Lightshot\Screenshot_83.png

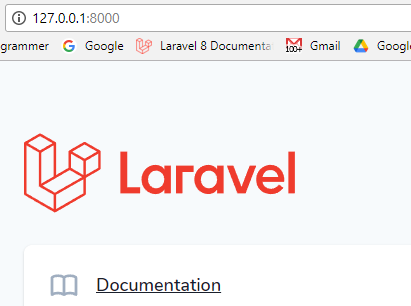
1. Masuk kedalam project yang telah di buat tadi “**cd l8-blog**”



1. Menjalankan project laravel dengan perintah “**php artisan serve**”



1. Buka web browser dan masukkan alamat “**localhost:8000**” atau bisa menggunakan “**127.0.0.1:8000**”, Jika tampilan sudah seperti dibawah. Maka berhasil.

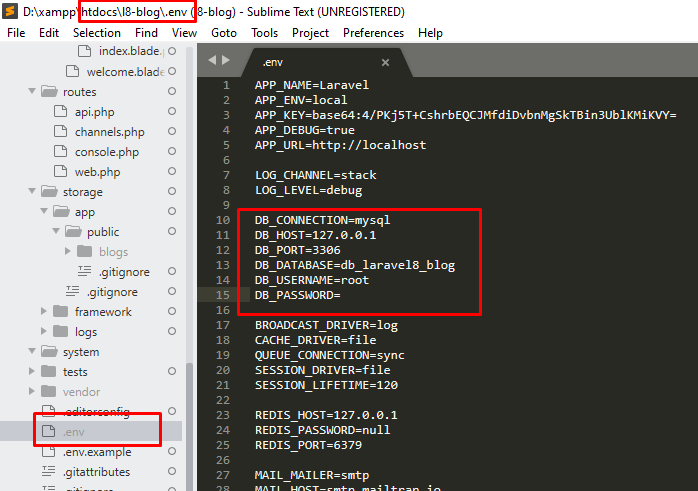


1. Membuat dan menjalankan Migration
2. Pengertian Migration

Migration adalah sebuah version control database, dimana akan membantu team untuk mengubah dan membagikan sebuah schema database dari aplikasi yang dibangun. Jika sebelumnya teman-teman biasanya membuat table-table secara manual di dalam database, maka dengan migration hal itu sudah tidak perlu dilakukan lagi.

1. Koneksi database

-pastikan sudah membuat database dengan nama “**db\_laravel8\_blog**”

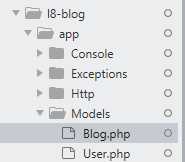


1. Membuat Model & Migration

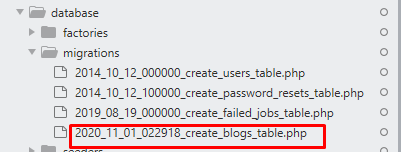
-Dengan cara ketik “**php artisan make:model Blog -m**”.

C:\Users\Widi Arrohman\Documents\Lightshot\Screenshot_89.png

-Perintah dibawah akan membuat sebuah model baru dengan nama Blog di dalam folder app/Models/Blog.php

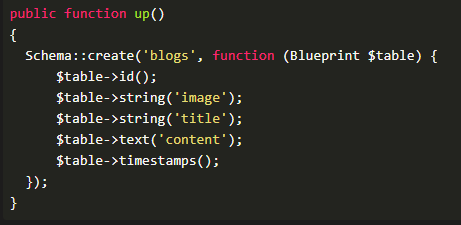


- kita juga menambahkan flag “**-m**” yang artinya sekaligus membuat untuk file migrationnya. Kita bisa melihatnya di “**database/migrations/2020\_11\_01\_022918\_create\_blogs\_table.php**”. Untuk nama file depan dari migration akan random sesuai dengan tanggal saat dibuat.



1. Membuat Field Table Migration

Sekarang kita lanjutkan untuk menambahkan beberapa filed yang akan kita gunakan nanti, silahkan buka file “**database/migrations/2020\_11\_01\_022918\_create\_blogs\_table.php**” dan silahkan ubah pada function up menjadi seperti berikut ini :



Di atas kita menambahkan 3 filed baru, yaitu :

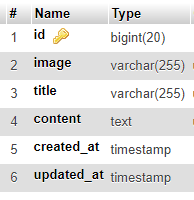
* image dengan tipe data string
* title dengan tipe data string
* content dengan tipe data text.

1. Menjalankan Migration

Ketik “**php artisan migrate**” pada CMD

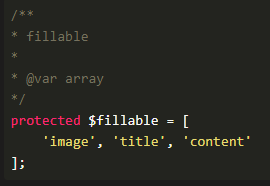
C:\Users\Widi Arrohman\Documents\Lightshot\Screenshot_94.png

1. Jika di database sudah terdapat kolom seperti dibawah, maka migration telah berhasil



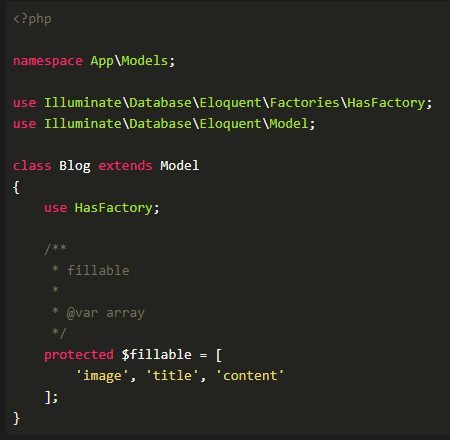
1. Menambahkan Mass Assigment

Sekarang kita lanjutkan untuk menambahkan mass assigment di dalam model Blog. Mass Assigment merupakan method yang digunakan untuk mengizinkan sebuah field dari table agar dapat menyimpan sebuah data. Kita bisa membuat mass assignment dengan properti $fillable di dalam model Blog. Silahkan buka file “**app/Models/Blog.php”** kemudian tambahkan kode berikut ini :



-Di atas kita menambahkan 3 field yang kita izinkan untuk menyimpan data di dalam table blogs, yaitu **(images, title, content)**.

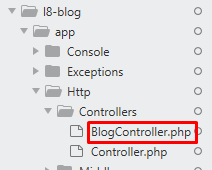
1. Tampilan Akhir



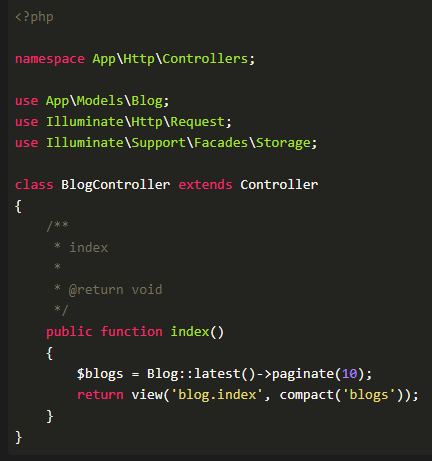
1. Menampilkan Data dari Database
2. Membuat Controller Blog
3. Ketika **“php artisan make:controller BlogController”**

C:\Users\Widi Arrohman\Documents\Lightshot\Screenshot_98.png

1. Jika berhasil teman-teman akan mendapatkan 1 file baru di dalam folder **“app/Http/Controllers/BlogController.php”**



1. Silahkan buka file tersebut kemudian ubah kode-nya menjadi seperti berikut ini :



- **“use App\Models\Blog;”**

Di atas pertama kita import Model **Blog** terlebih dahulu, karena kita akan menggunakan model ini untuk menampilkan data, input data dan yang lain-nya.

**- “use Illuminate\Http\Request;”**

Kemudian kita juga import Http **Request** dari Laravel, ini akan kita manfaatkan untuk mendapatkan data request dari sisi client untuk di masukkan di dalam database saat proses input data.

**-“use Illuminate\Support\Facades\Storage;”**

Dan kita juga import **Facades Storage** dari Laravel, ini akan kita manfaatkan untuk melakukan store atau upload data gambar ke dalam server.

**-**Kemudian di atas kita menambahkan 1 **method/function** baru yaitu **index**, ini akan kita manfaatkan untuk menampilkan data blog di dalam project kita.

**-“$blogs = Blog::latest()->paginate(10);”**

Di dalam **function index** kita memanggil Model Blog kemudian mengurutkan datanya berdasarkan terbaru dengan method **latest()** dan membatasi data yang ditampilkan sejumlah 10 data perhalaman.

-**“return view('blog.index', compact('blogs'));”**

Setelah itu kita memanggil sebuah view dengan nama **index.blade.php** di dalam folder **blog** yang mana folder tersebut berada di dalam folder **resources/views/.** Dan kita parsing data blog tersebut ke dalam view menggunakan helper **compact**.

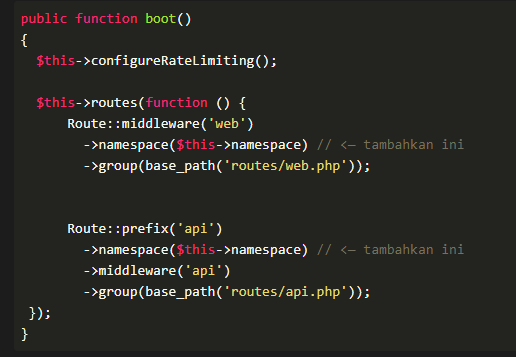
1. Konfigurasi Namespace Route
2. Silahkan buka file **app/Providers/RouteServiceProvider.php** kemudian cari kode berikut ini :

C:\Users\Widi Arrohman\Documents\Lightshot\Screenshot_101.png

Kemudian ganti menjadi seperti berikut ini :

C:\Users\Widi Arrohman\Documents\Lightshot\Screenshot_102.png

Di atas kita membuat agar namespace dari controller kita arahkan ke dalam folder app/Http/Controllers. Dan setelah itu kita juga harus melakukan register namespace di dalam method boot.

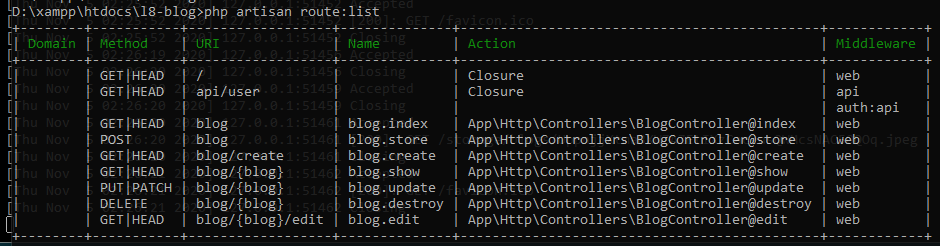


1. Membuat Route
2. Silahkan buka file **routes/web.php** kemudian tambahkan kode berikut ini :

C:\Users\Widi Arrohman\Documents\Lightshot\Screenshot_104.png

-Di atas kita menggunakan route dengan jenis resources, yang artinya route tersebut sudah complete, mulai dari (**index, create, store, edit, update dan destroy**). Jadi kita tidak perlu define route satu-satu.

1. (**php artisan route:list**)

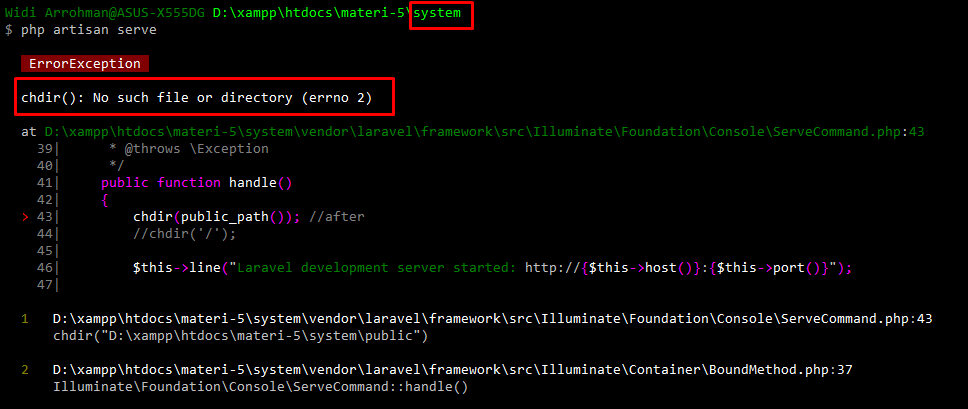
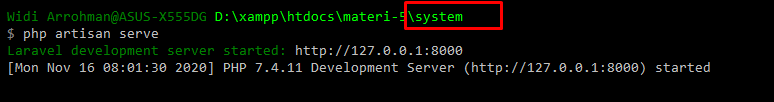


1. Membuat View
2. Silahkan buat folder baru dengan nama **blog** di dalam folder **resources/views/** dan kemudian di dalam folder blog silahkan buat file baru dengan nama **index.blade.php** dan massukkan kode berikut ini :

|  |
| --- |
|  |

1. Asdf
2. asdf

BAB III PHP ARTISAN SERVE

1. 
2. Masuk ke alamat “D:\xampp\htdocs\materi-5\system\vendor\laravel\framework\src\Illuminate\Foundation\Console\ServeCommand.php”
3. 
4. 

BAB IV UPLOAD GITHUB

1. **git init**

untuk meng-set folder yang digunakan tersebut sebagai repo local git. bisa di bilang ini instalasi git pertama kali

1. **git add .**

untuk menambah file project yang mau di upload sebelum di commit, tanda titik setelah kata “add” pada perintah tersebut adalah keseluruhan file dan folder project tersebut, saat awal upload kalian bisa menggunakan perintah tersebut. Namun saat commit atau upload ke repository selanjutnya bisa menggunakan perintah add dengan “nama file” untuk memberikan status commit. Ini adalah contohnya apabila ingin menggunakan “git add” :

// mengupload keseluruhan file pada repo (hanya digunakan saat upload pertama saja)  
**git add .**

// mengupload file sesuai dengan nama file  
**git add index.html**

// mengupload file pada dalam folder  
**git add pages/index.html**

1. **git commit -m "first commit"**

untuk menambah keterangan/status perubahaan saat upload ke repo online, untuk memasukkan keterangan tersebut setelah “git commit -m” ditambah tanda petik lalu komentar(lihat di list perintah untuk contoh).

1. **git remote add origin** [**https://github.com/aisy/js-belajar-react.git**](https://github.com/aisy/js-belajar-react.git)

untuk meng-setting remote origin dari repo online, repo online bisa dilihat pada link yang tersedia di bagian atas Project dengan format “.git”, diperlukan ini untuk mengakses ke repo tersebut sehingga kita bisa melakukan apapun di repo online tersebut.

1. **git push -u origin master**

Perintah untuk mengupload file yang ada pada repo lokal ke repo online yang diletakkan pada branch yang sudah tersedia di repo online.

“ <https://medium.com/aisy-rozsidhy/tutorial-penggunaan-github-untuk-pemula-part-1-upload-file-ke-github-e807df4e9ecc> ”

BAB IV UPDATE GITHUB

1. **git pull**

“ <https://medium.com/@sarascahya/bagaimana-cara-mendapatkan-pembaharuan-dari-repository-github-yang-saya-clone-sebelumnya-7c582d97fc74> ”

BAB IV JUDUL