

DOKUMENTASI CRUD LARAVEL MONGODB  
(BLOG ARTICLE)

NAMA : KELVIN FEBRIAN GO  
KELAS : XII – RPL

Untuk pertama – tama, install laravel terlebih dahulu

```
composer create-project --prefer-dist laravel/laravel laravelmongodb
```

Setelah terinstall, pastikan tidak ada masalah dalam proses penginstallan. Lalu selanjutnya kita mengonfig agar bisa membuat laravel dan mongodb terhubung

pergi ke .env dan ubah menjadi seperti ini

```
//.env

MONGO_DB_HOST=127.0.0.1
MONGO_DB_PORT=27017
MONGO_DB_DATABASE=mongocrud
MONGO_DB_USERNAME=
MONGO_DB_PASSWORD=
```

dan pergi ke database.php, tambahkan di bagian connections seperti ini

```
//database.php

'connections' => [

    .....

    'mongodb' => [
        'driver'   => 'mongodb',
        'host'     => env('MONGO_DB_HOST', 'localhost'),
        'port'     => env('MONGO_DB_PORT', 27017),
        'database' => env('MONGO_DB_DATABASE'),
        'username' => env('MONGO_DB_USERNAME'),
        'password' => env('MONGO_DB_PASSWORD'),
        'options'  => []
    ],
],
]
```

Setelah ditambahi, sekarang kita install packagenya dengan cara

```
“composer require jenssegers/mongodb “
(tidak pakai tanpa “”)
```

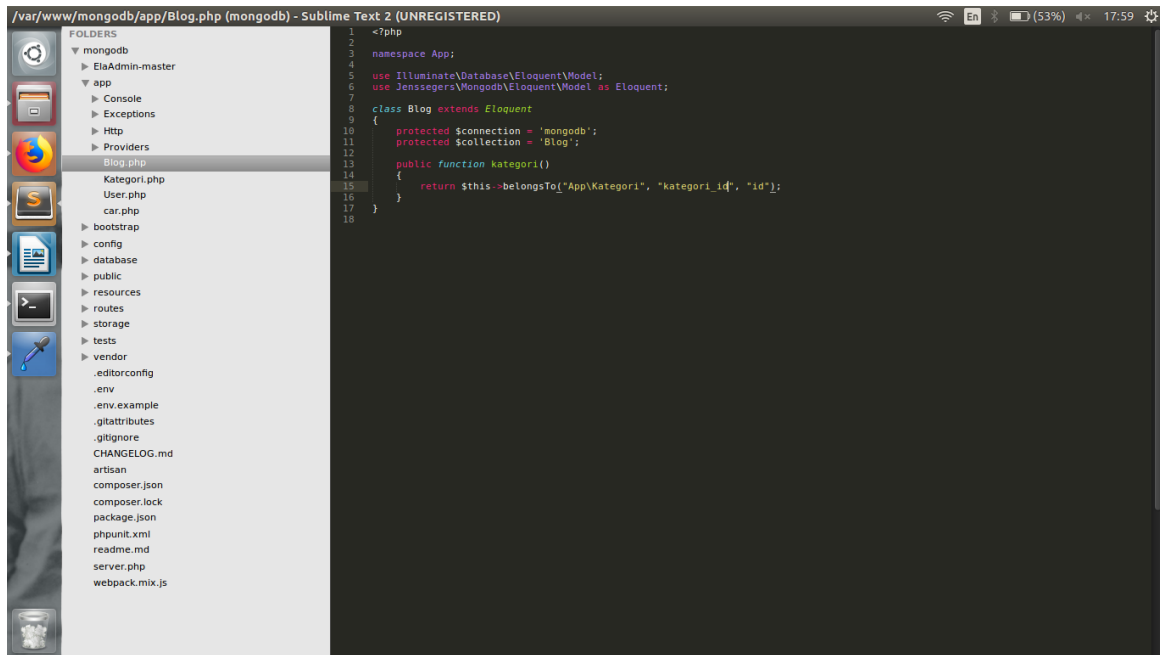
selanjutnya, kita menambahkan providernya, tambahkan di file config → app.php di bagian providersnya, tambahkan seperti ini

```
“ Jenssegers\Mongodb\MongodbServiceProvider::class, “
```

Setelah koneksi berhasil, kita buat modelnya untuk membuat database

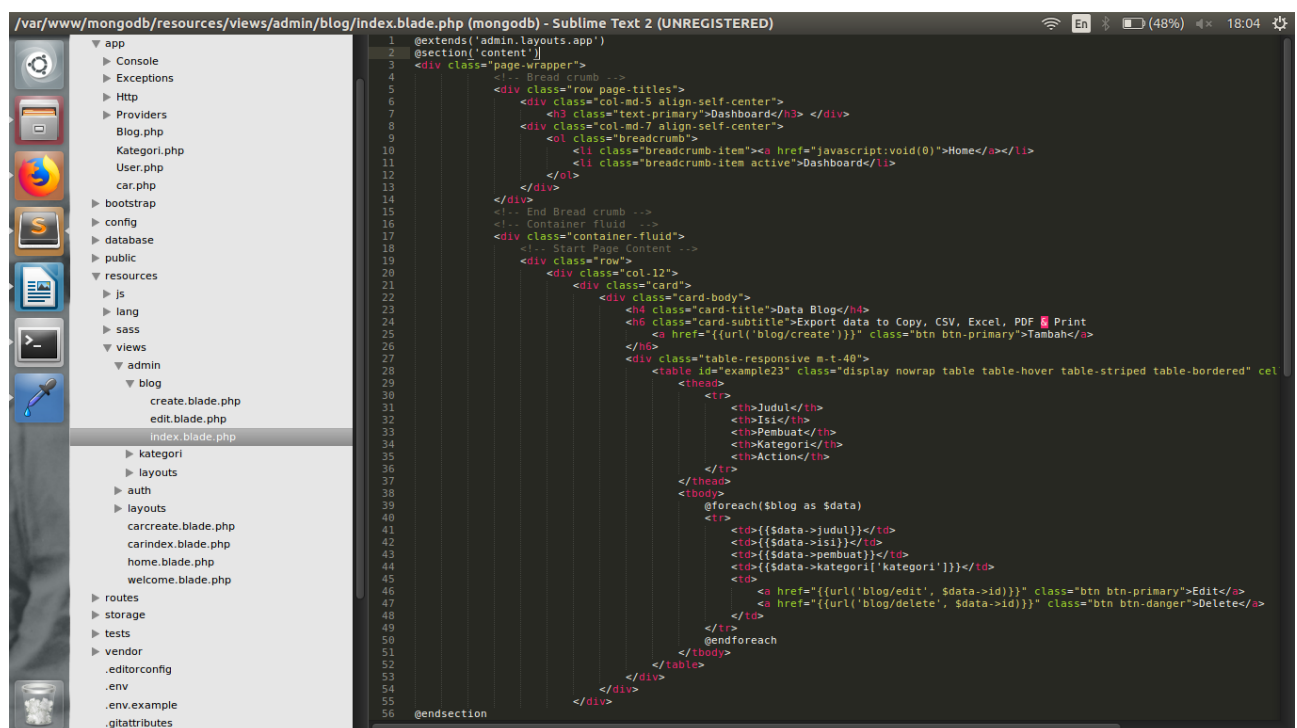
```
“ php artisan make:model Blog “
```

lalu tambahkan seperti ini:



```
1 <?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6 use Jenssegers\Mongodb\Eloquent\Model as Eloquent;
7
8 class Blog extends Eloquent
9 {
10     protected $connection = 'mongodb';
11     protected $collection = 'Blog';
12
13     public function kategori()
14     {
15         return $this->belongsTo('App\Kategori', 'kategori_id', 'id');
16     }
17 }
18
```

Buat view untuk show data di resources → views bebas mau pakai template/native.kalau saya seperti ini

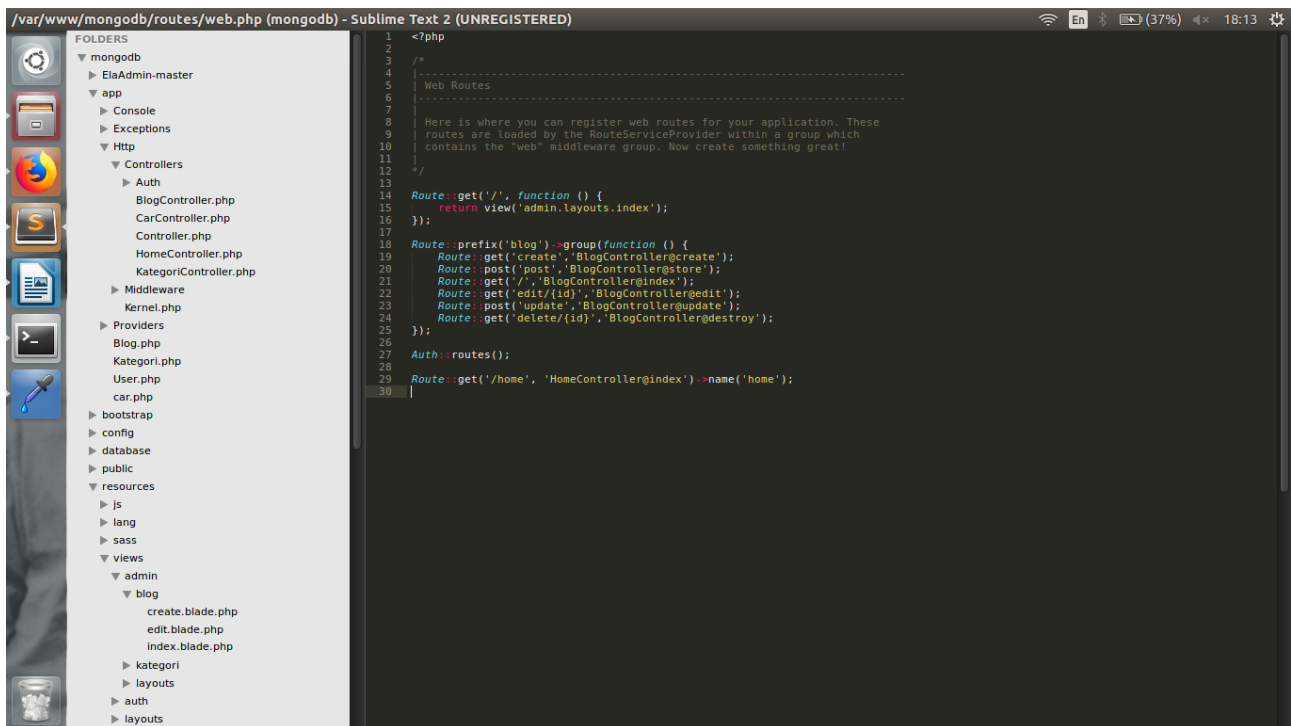


```
1 @extends('admin.layouts.app')
2 @section('content')
3 <div class="page-wrapper">
4     <!-- Breadcrumb -->
5     <div class="row page-titles">
6         <div class="col-md-5 align-self-center">Dashboard</div>
7         <div class="col-md-7 align-self-center">
8             <ol class="breadcrumb">
9                 <li class="breadcrumb-item"><a href="javascript:void(0)">Home</a></li>
10                 <li class="breadcrumb-item active">Dashboard</li>
11             </ol>
12         </div>
13     </div>
14     <!-- End Breadcrumb -->
15     <!-- Container fluid -->
16     <div class="container-fluid">
17         <!-- Start Page Content -->
18         <div class="row">
19             <div class="col-12">
20                 <div class="card">
21                     <div class="card-body">
22                         <h4 class="card-title">Data Blog</h4>
23                         <h6 class="card-subtitle">Export data to Copy, CSV, Excel, PDF & Print
24                         <a href="{{url('blog/create')}}" class="btn btn-primary">Tambah</a>
25                     </h6>
26                     <div class="table-responsive m-t-40">
27                         <table id="example23" class="display nowrap table table-hover table-striped table-bordered" >
28                             <thead>
29                                 <tr>
30                                     <th>Judul</th>
31                                     <th>Isi</th>
32                                     <th>Pembuat</th>
33                                     <th>Kategori</th>
34                                     <th>Action</th>
35                                 </tr>
36                             </thead>
37                             <tbody>
38                                 @foreach($blog as $data)
39                                     <tr>
40                                         <td>{{ $data->judul }}</td>
41                                         <td>{{ $data->isi }}</td>
42                                         <td>{{ $data->pembuat }}</td>
43                                         <td>{{ $data->kategori['kategori'] }}</td>
44                                         <td>
45                                             <a href="{{url('blog/edit', $data->id)}}" class="btn btn-primary">Edit</a>
46                                             <a href="{{url('blog/delete', $data->id)}}" class="btn btn-danger">Delete</a>
47                                         </td>
48                                     </tr>
49                                 @endforeach
50                             </tbody>
51                         </table>
52                     </div>
53                 </div>
54             </div>
55         </div>
56     @endsection
```

pasti ada error disini,karena kita belum membuat controller.Kita membuat controllernya dulu

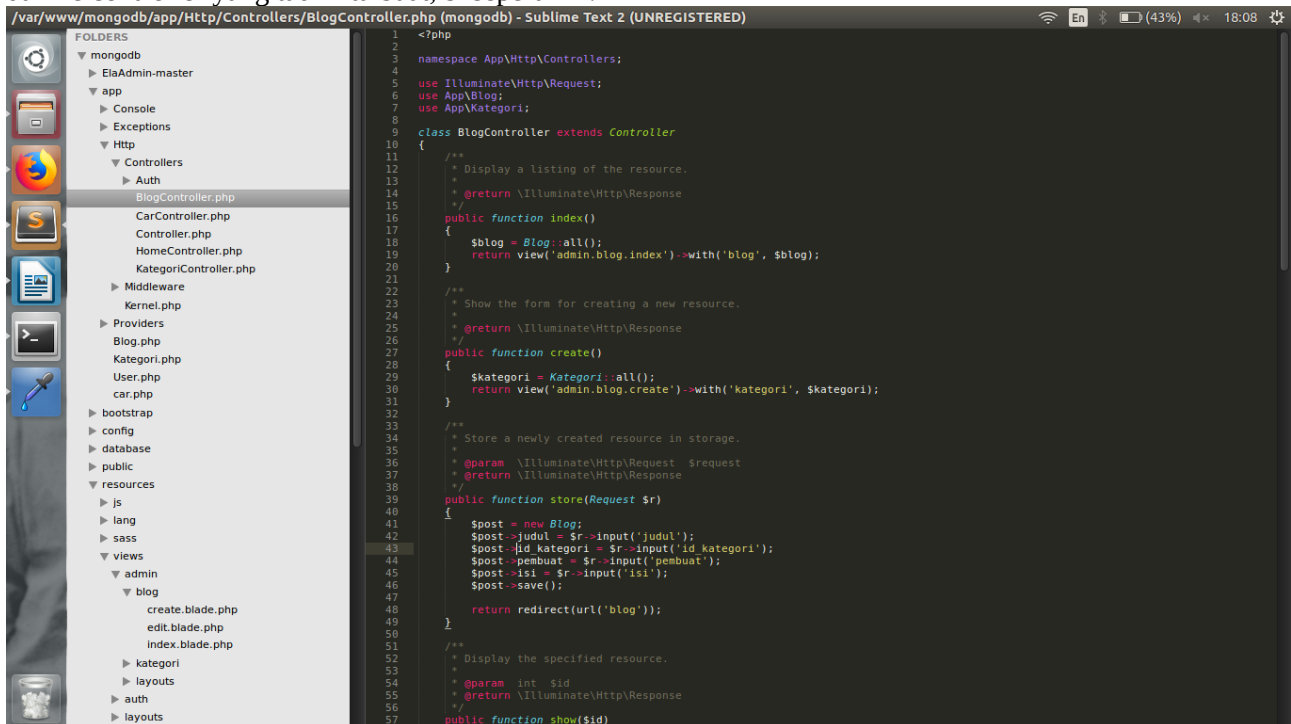
“ php artisan make:controller BlogController --resource “

setelah itu, buat routesnya terlebih dahulu. Pergi ke Routes → web.php dan isi seperti ini



```
1 <?php
2
3 /*
4  * Web Routes
5  */
6
7
8 Here is where you can register web routes for your application. These
9 routes are loaded by the RouteServiceProvider within a group which
10 contains the "web" middleware group. Now create something great!
11
12
13
14 Route::get('/', function () {
15     return view('admin.layouts.index');
16 });
17
18 Route::prefix('blog')->group(function () {
19     Route::get('create', 'BlogController@create');
20     Route::post('store', 'BlogController@store');
21     Route::get('/', 'BlogController@index');
22     Route::get('edit/{id}', 'BlogController@edit');
23     Route::post('update', 'BlogController@update');
24     Route::get('delete/{id}', 'BlogController@destroy');
25 });
26
27 Auth::routes();
28
29 Route::get('/home', 'HomeController@index')->name('home');
30
```

dan ke controller yang tadi kita buat, isi seperti ini:



```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Blog;
7 use App\Kategori;
8
9 class BlogController extends Controller
10 {
11     /**
12      * Display a listing of the resource.
13      *
14      * @return \Illuminate\Http\Response
15      */
16     public function index()
17     {
18         $blog = Blog::all();
19         return view('admin.blog.index')->with('blog', $blog);
20     }
21
22     /**
23      * Show the form for creating a new resource.
24      *
25      * @return \Illuminate\Http\Response
26      */
27     public function create()
28     {
29         $kategori = Kategori::all();
30         return view('admin.blog.create')->with('kategori', $kategori);
31     }
32
33     /**
34      * Store a newly created resource in storage.
35      *
36      * @param \Illuminate\Http\Request $request
37      * @return \Illuminate\Http\Response
38      */
39     public function store(Request $r)
40     {
41         $post = new Blog;
42         $post->judul = $r->input('judul');
43         $post->id_kategori = $r->input('id_kategori');
44         $post->pembuat = $r->input('pembuat');
45         $post->isi = $r->input('isi');
46         $post->save();
47
48         return redirect(url('blog'));
49     }
50
51     /**
52      * Display the specified resource.
53      *
54      * @param int $id
55      * @return \Illuminate\Http\Response
56      */
57     public function show($id)
58     {
59
60     }
61 }
```

(1)

```
56 * @return \Illuminate\Http\Response
57
58 public function show($id)
59 {
60     //
61 }
62
63 /**
64  * Show the form for editing the specified resource.
65  *
66  * @param int $id
67  * @return \Illuminate\Http\Response
68  */
69 public function edit($id)
70 {
71     $edit = Blog::find($id);
72     $kategori = Kategori::all();
73     return view('admin.blog.edit')->with('edit', $edit)->with('kategori', $kategori);
74 }
75
76 /**
77  * Update the specified resource in storage.
78  *
79  * @param \Illuminate\Http\Request $request
80  * @param int $id
81  * @return \Illuminate\Http\Response
82  */
83 public function update(Request $r)
84 {
85     $update = Blog::find($r->input('id'));
86     $update->judul = $r->input('judul');
87     $update->id_kategori = $r->input('id_kategori');
88     $update->pembuat = $r->input('pembuat');
89     $update->isi = $r->input('isi');
90     $update->update();
91     return redirect(url('blog'));
92 }
93
94 /**
95  * Remove the specified resource from storage.
96  *
97  * @param int $id
98  * @return \Illuminate\Http\Response
99  */
100 public function destroy($id)
101 {
102     $delete = Blog::find($id);
103     $delete->delete();
104     return redirect(url('blog'));
105 }
106
107 }
```

(2)

coba cek di web bisa atau tidak.kalau berhasil,tandanya laravel crud mongodb anda sudah bisa dijalankan dengan baik

=== Terima Kasih ===