

# WebDev UMM Tutorial

## LARAVEL 5.6

### Instalasi Laravel 5.6

#### 1. Persiapan

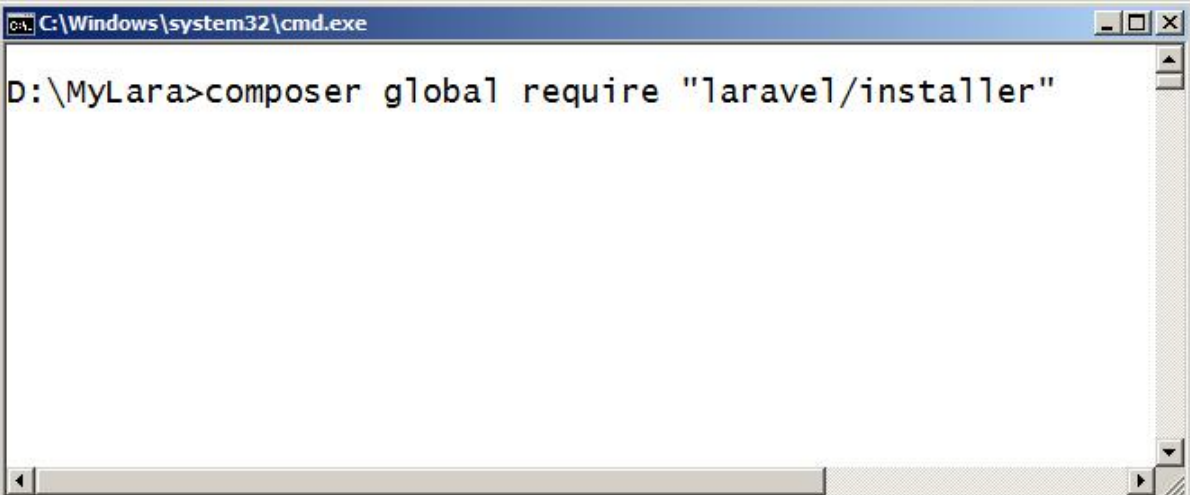
Windows

- XAMPP PHP Versi 7.1 Keatas
- Composer

Linux

- XAMPP Linux
- Composer

#### 2. Instalasi



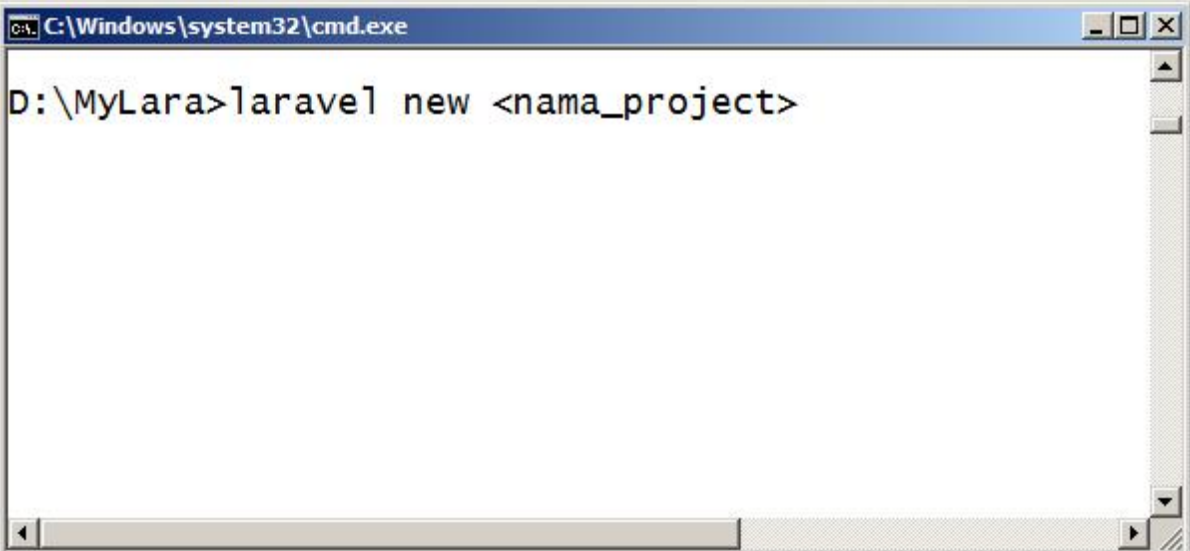
```
C:\Windows\system32\cmd.exe

D:\MyLara>composer global require "laravel/installer"
```

**Gambar 1. Download Laravel Installer Melalui Composer**

NB : Proses ini dilakukan sekali jika ingin membuat project laravel lagi tinggal melakukan

**Gambar 2**

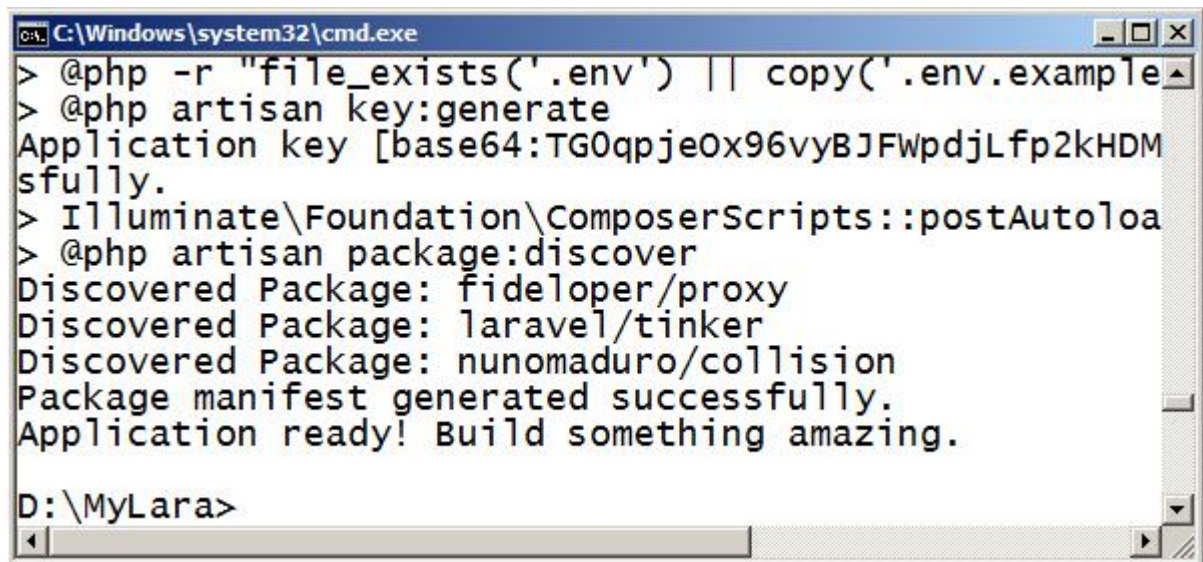


```
C:\Windows\system32\cmd.exe

D:\MyLara>laravel new <nama_project>
```

**Gambar 2. Membuat Project Laravel**

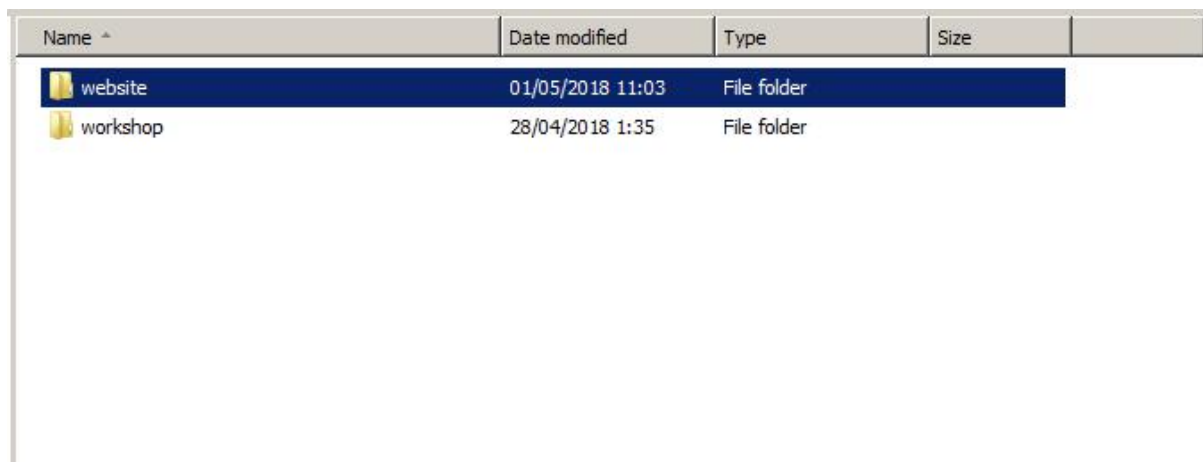
# WebDev UMM Tutorial



```
C:\Windows\system32\cmd.exe
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
> @php artisan key:generate
Application key [base64:TG0qpjeOx96vyBJFWpdjLfp2kHDM] generated successfully.
> Illuminate\Foundation\ComposerScripts::postAutoloader
> @php artisan package:discover
Discovered Package: fideloper/proxy
Discovered Package: laravel/tinker
Discovered Package: nunomaduro/collision
Package manifest generated successfully.
Application ready! Build something amazing.

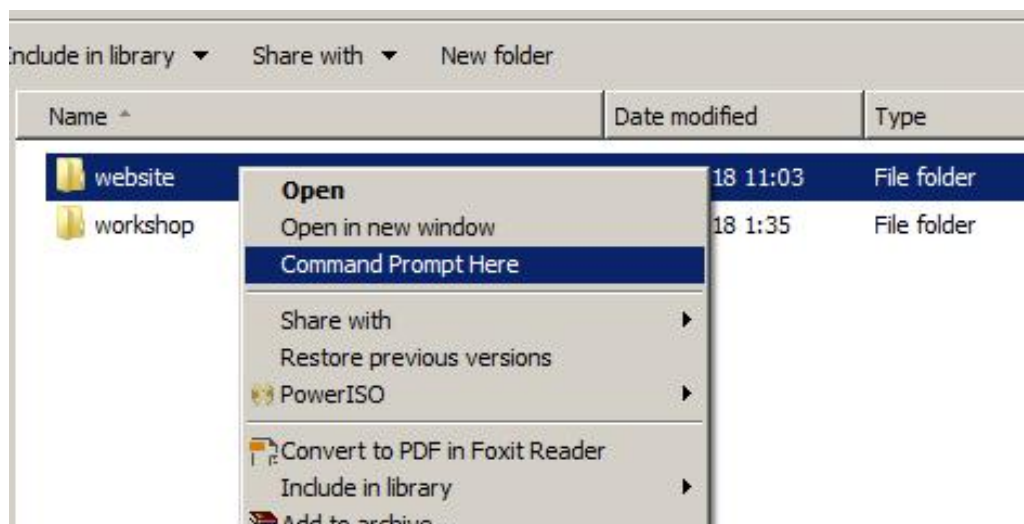
D:\MyLara>
```

Gambar 3. Project Laravel Berhasil Dibuat



Name ^	Date modified	Type	Size
website	01/05/2018 11:03	File folder	
workshop	28/04/2018 1:35	File folder	

Gambar 4. Folder Project Laravel yang sudah dibuat

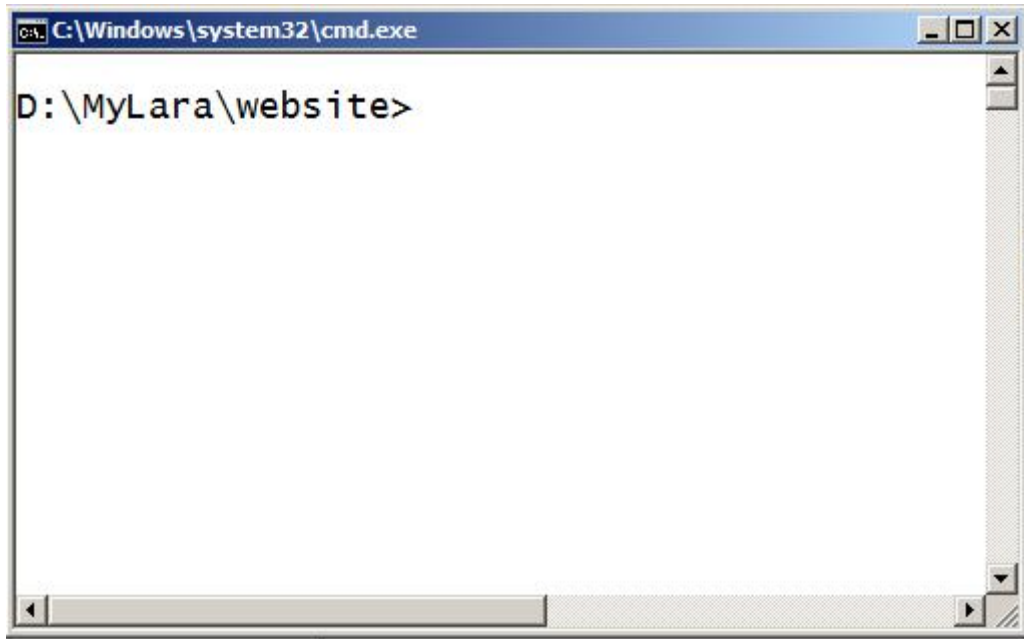


Gambar 5. Tips Simple Untuk membuka Command Prompt Tanpa Harus Search Folder Melalui CMD  
Terkhusus di Windows 7

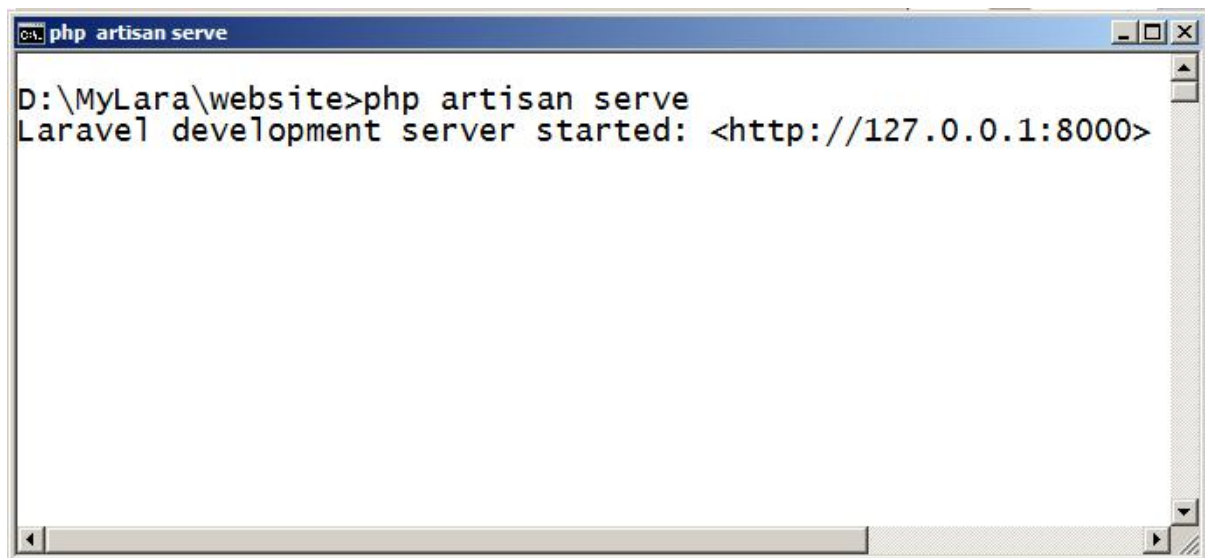
Link :

<https://www.howtogeek.com/howto/windows/add-the-command-prompt-to-the-windows-explorer-right-click-menu/>

# WebDev UMM Tutorial



Gambar 6. Folder Project Laravel



Gambar 7. Start Laravel Project

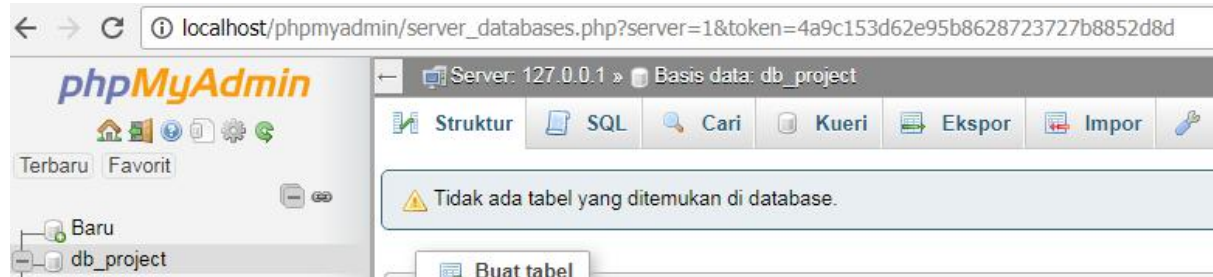


Gambar 8. Laravel Project Di Browser

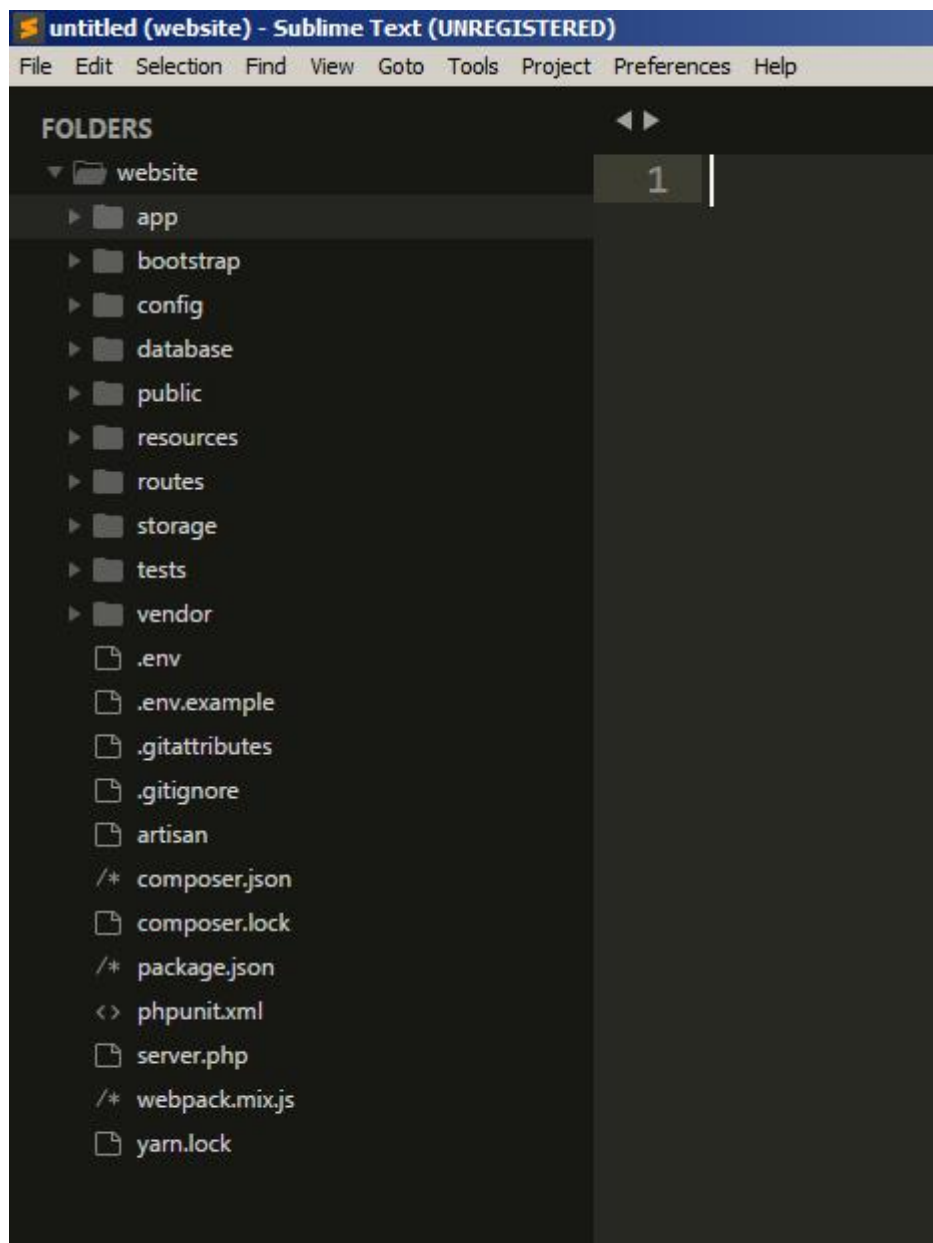
# WebDev UMM Tutorial

## Migration

Migration adalah salah satu fitur dari laravel dimana pengelolaan database ( membuat table dan lain-lain menggunakan script php. Istilah migration sendiri mengartikan bahwa laravel mudah untuk berpindah kemana-mana. Semisal dari satu developer ke developer lain.

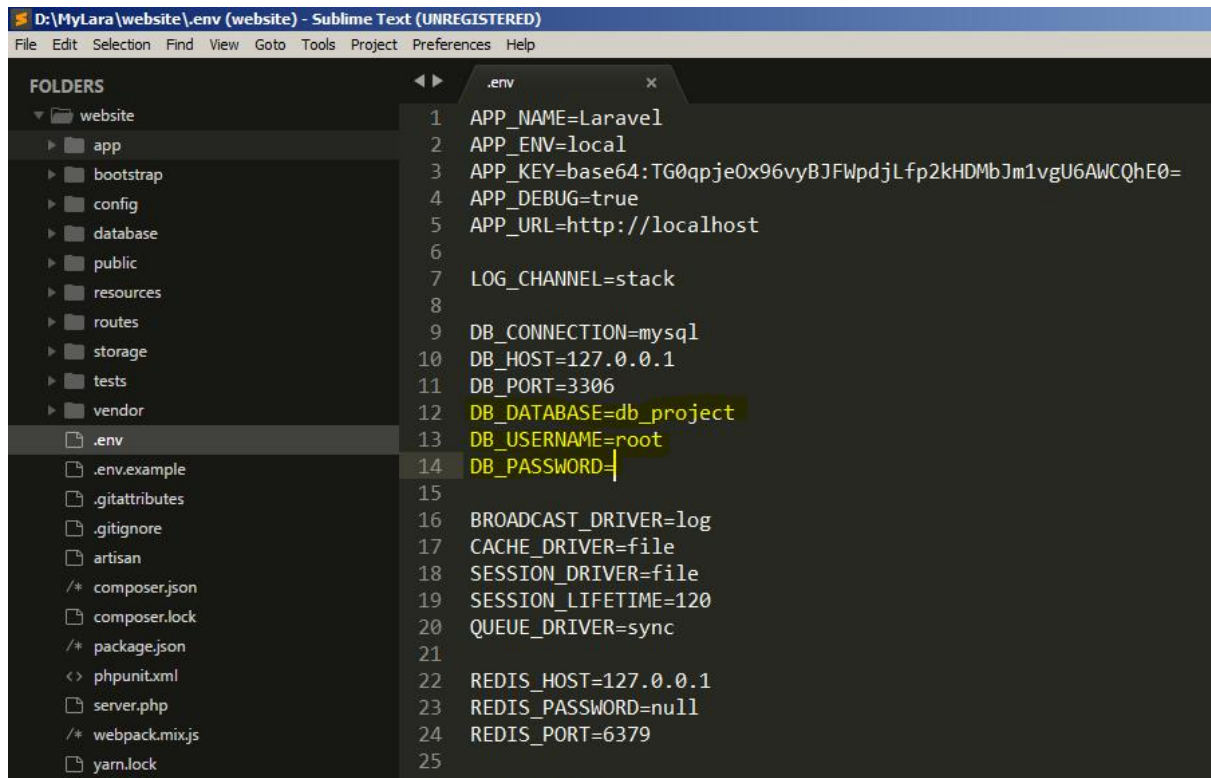


Gambar 9. Buat Database Kosong untuk Project Laravel



Gambar 10. Buka Project Laravel di Sublime

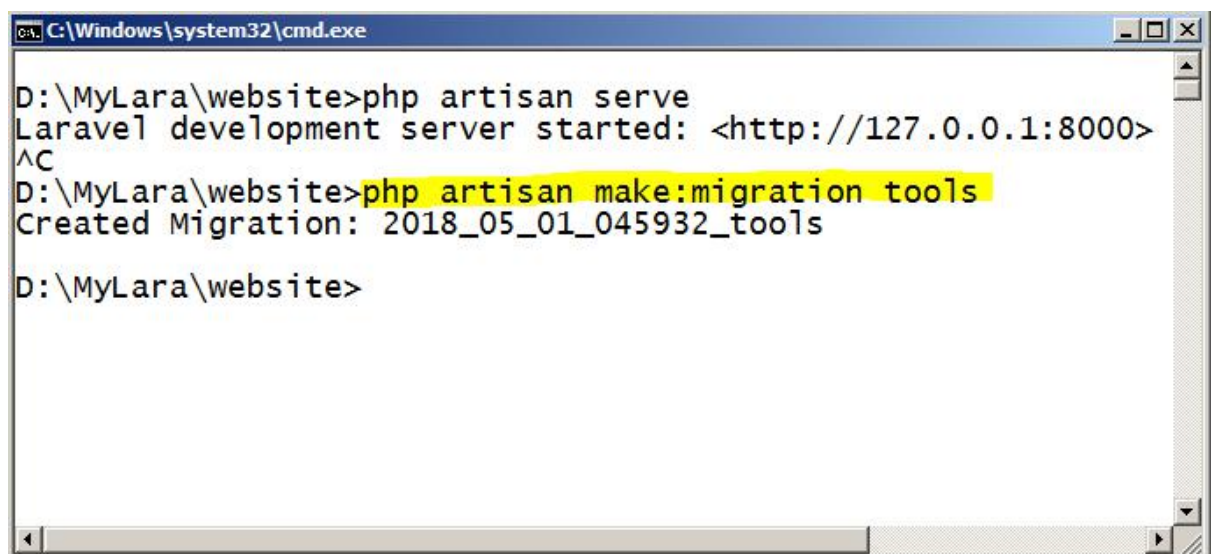
# WebDev UMM Tutorial



Gambar 11. Buka .Env File edit db\_name, db\_username, db\_password sesuai dengan Database yang anda miliki



Gambar 12. Struktur Folder Database



Gambar 13. Membuat File Migration Melalui PHP Artisan

NB : Gunakan Ctrl+C untuk menghentikan project laravel yang masih berjalan



# WebDev UMM Tutorial

Sebagai contoh untuk membuat sebuah migration pada tutorial ini saya akan mencontohkan untuk membuat sebuah database untuk mengelola daftar barang

tools	
tool_id	INT UNSIGNED
tool_name	VARCHAR 255
tool_count	INT
created_at	Timestamp
updated_at	Timestamp

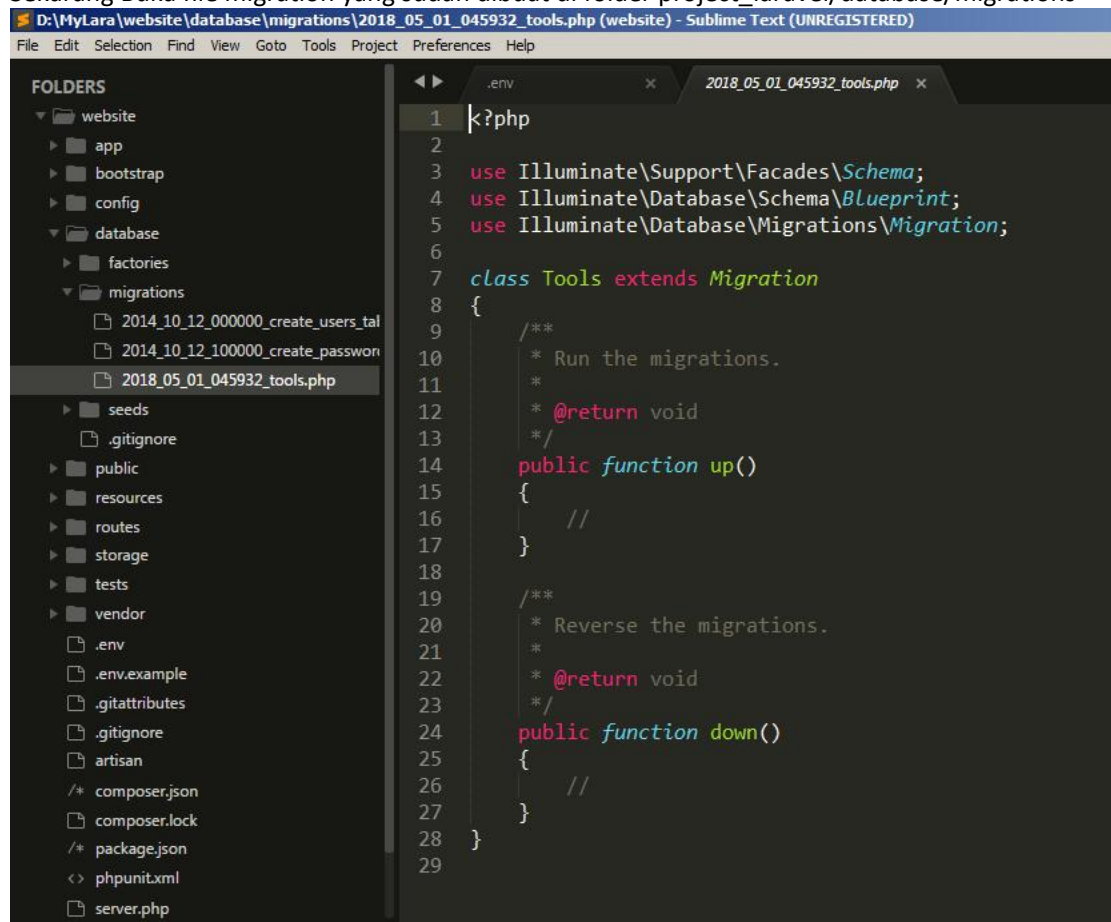
Gambar 14. Contoh Table Tools

Secara Default Laravel menggunakan konsep singular dan plural yang ada di penulisan bahasa inggris, tapi dengan konfigurasi sebenarnya bisa kita atur menggunakan bahasa indonesia. Hubungannya nanti adalah pada Model. Nama Model Laravel menggunakan Singular Form dan Table yang dikontrol menggunakan plural. <https://www.grammarly.com/blog/plural-nouns/>

Model	Table Database / Migration
Book	Books
User	Users
City	Cities
Category	Categories
Tax	Taxes

Gambar 15. Table Contoh Nama Model dan Migrationnya / tabel database

Sekarang Buka file migration yang sudah dibuat di folder project\_laravel/database/migrations

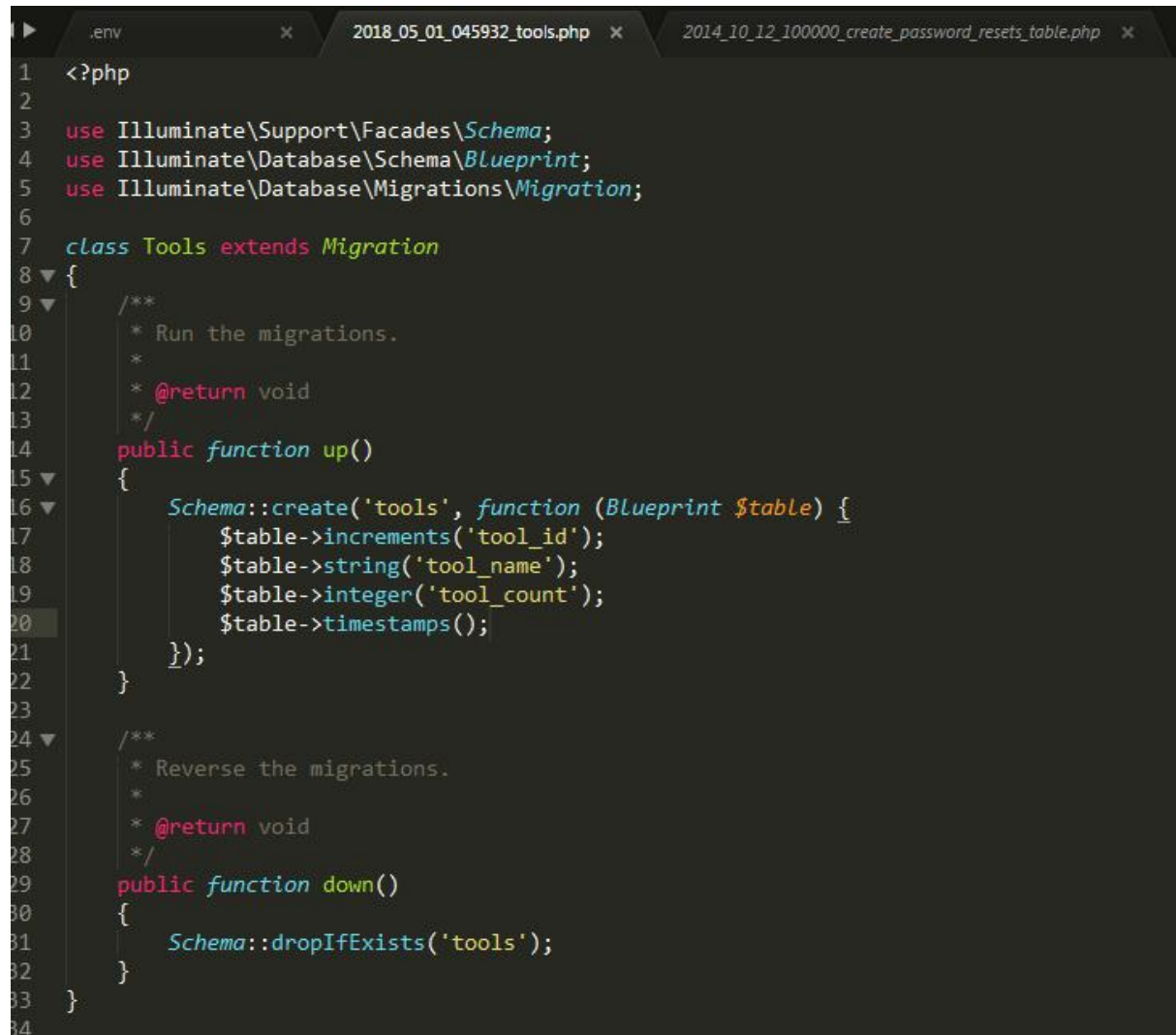


```
1 k?php
2
3 use Illuminate\Support\Facades\Schema;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Database\Migrations\Migration;
6
7 class Tools extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         //
17     }
18
19     /**
20      * Reverse the migrations.
21      *
22      * @return void
23      */
24     public function down()
25     {
26         //
27     }
28 }
29
```

Gambar 15. File migration tools

# WebDev UMM Tutorial

Edit file sebagai berikut, sesuaikan dengan desain database yang sudah ada pada kasus ini yang digunakan adalah desain database tools.



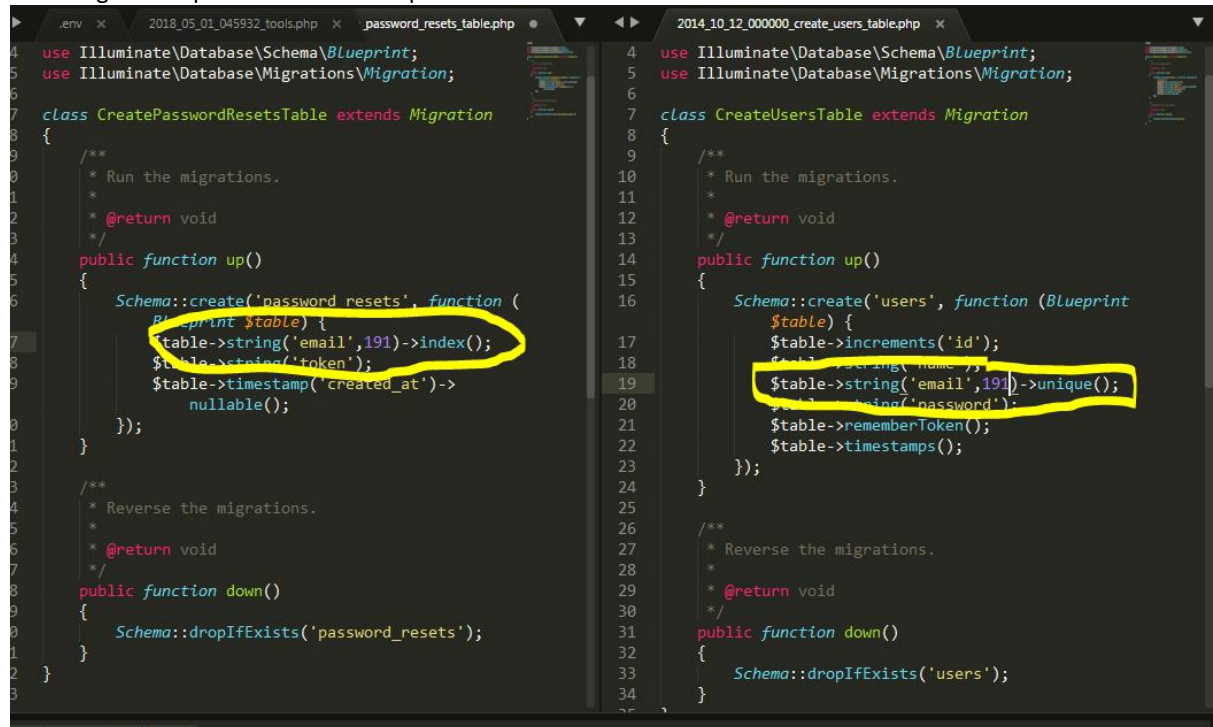
```
1 <?php
2
3 use Illuminate\Support\Facades\Schema;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Database\Migrations\Migration;
6
7 class Tools extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('tools', function (Blueprint $table) {
17             $table->increments('tool_id');
18             $table->string('tool_name');
19             $table->integer('tool_count');
20             $table->timestamps();
21         });
22     }
23
24     /**
25      * Reverse the migrations.
26      *
27      * @return void
28      */
29     public function down()
30     {
31         Schema::dropIfExists('tools');
32     }
33 }
34
```

Gambar 16. Migration untuk table tools

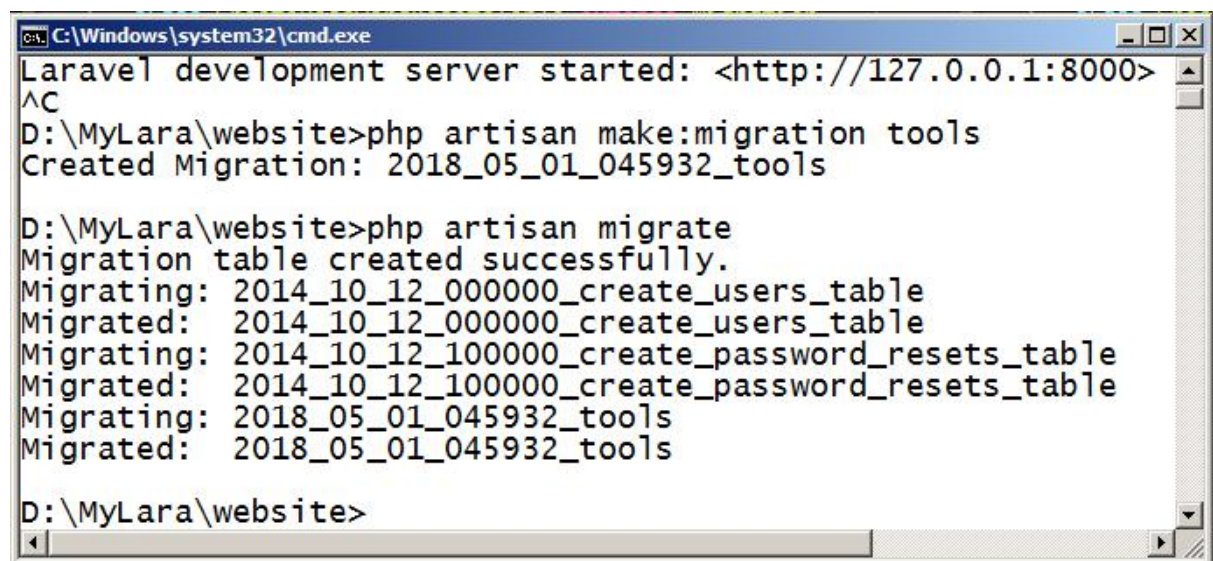
# WebDev UMM Tutorial

Pada MariaDB biasanya terjadi error untuk pembuatan unique key, silahkan cek file `2014_10_12_000000_create_users_table.php` dan `2014_10_12_100000_create_password_resets_table.php`.

Pada Bagian Unique di email ubah seperti ini



Gambar 17. Edit Migration users dan reset\_password

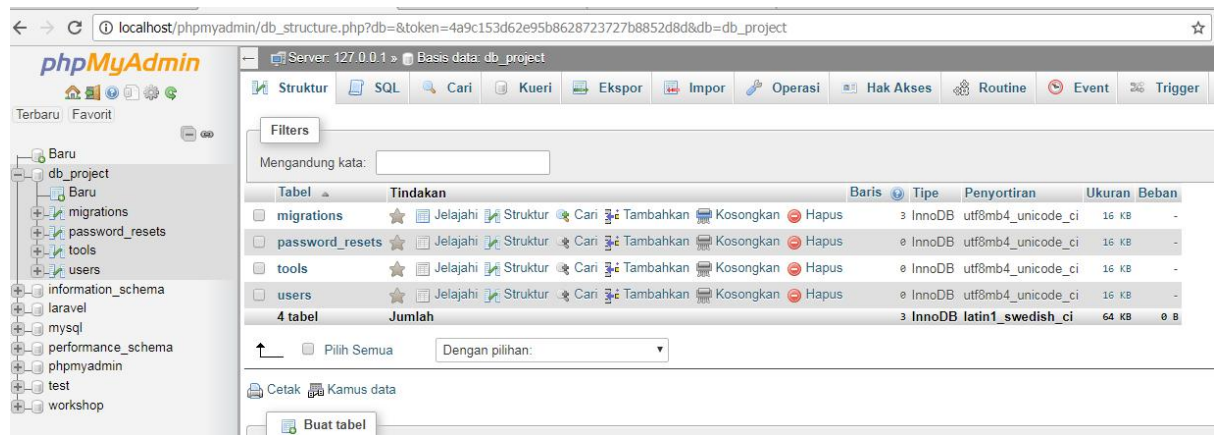


Gambar 18. Jalankan php artisan Migrate

Jika tidak terjadi error maka akan menjadi seperti diatas



# WebDev UMM Tutorial



The screenshot shows the phpMyAdmin interface for a database named 'db\_project'. The left sidebar lists various databases, including 'migrations', 'password\_resets', 'tools', 'users', 'information\_schema', 'laravel', 'mysql', 'performance\_schema', 'phpmyadmin', 'test', and 'workshop'. The main panel displays the structure of the 'users' table, which is highlighted. The table has 4 columns: 'id', 'name', 'email', and 'password'. The 'id' column is the primary key. The 'email' column is indexed. The 'password' column is indexed. The table is using the InnoDB engine and the utf8mb4\_unicode\_ci collation. The table size is 16 KB. The table is empty.

Tabel	Tindakan	Baris	Tipe	Penyortiran	Ukuran	Beban
migrations	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	3	InnoDB	utf8mb4_unicode_ci	16 KB	-
password_resets	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	utf8mb4_unicode_ci	16 KB	-
tools	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	utf8mb4_unicode_ci	16 KB	-
users	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	utf8mb4_unicode_ci	16 KB	-
4 tabel	Jumlah	3	InnoDB latin1_swedish_ci		64 KB	0 B

**Gambar 19.** Hasil Table yang sudah dibuat dengan Migration

# WebDev UMM Tutorial

## Template Integration

Template Integrasi di laravel cukup unik, tetapi setelah mengerti penggunaannya sangat simple. Di Laravel terdapat fitur blade template. Dengan fitur ini menjadikan view di laravel menjadi beberapa bagian.

Pada Laravel kita harus memasukkan folder .js dan .css nya kedalam folder public. Di Tutorial ini telah disediakan folder untuk melakukan templating

Name ^	Date modified	Type
Copy to public folder	01/05/2018 13:27	File folder
Copy to resources-views	01/05/2018 13:49	File folder
crud	01/05/2018 13:49	File folder

**Gambar 20. Folder yang Sudah ada untuk Tutorial ini**

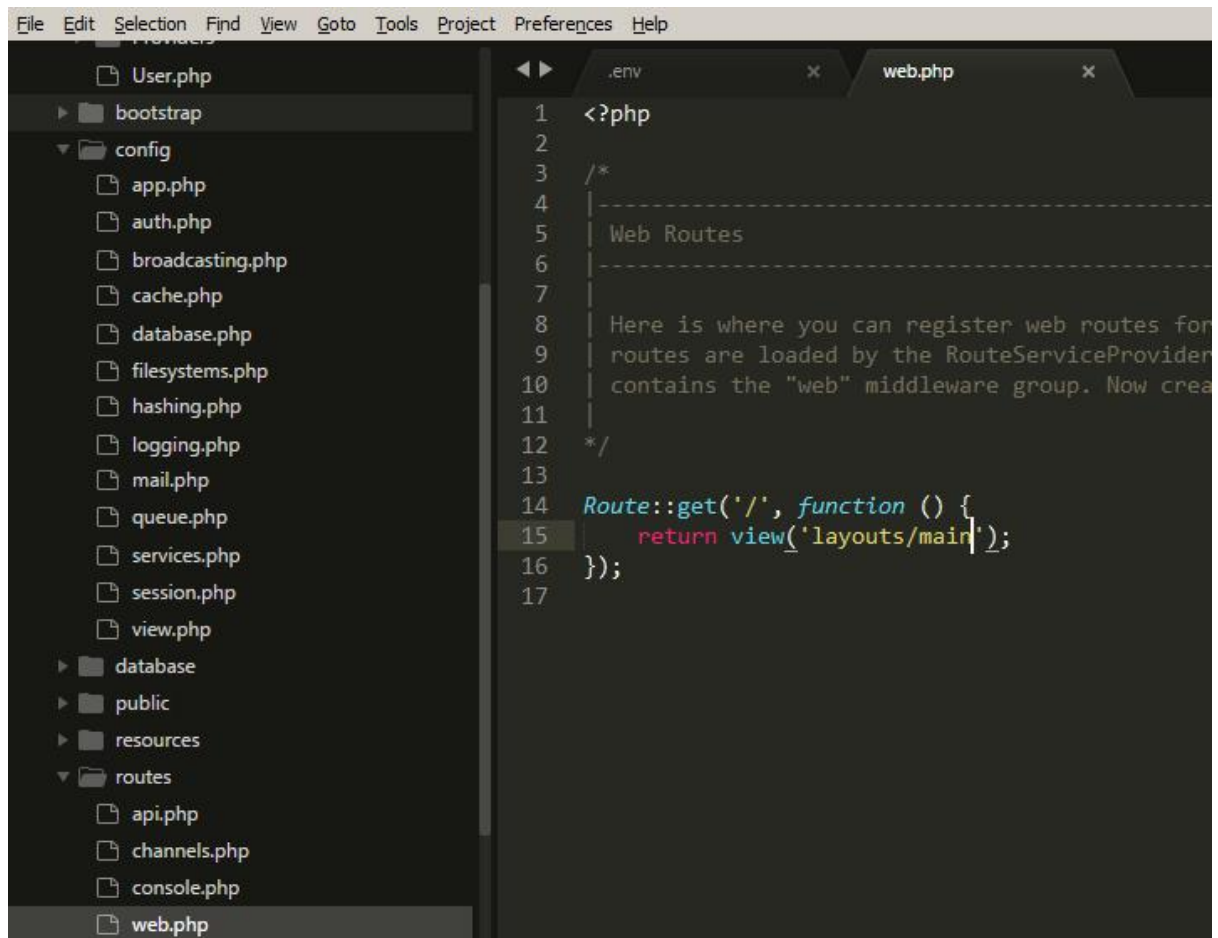
Silahkan untuk mencopy file yang sudah disiapkan ke Folder yang sudah ditentukan.

er ▾ Local Disk (D:) ▾ MyLara ▾ website ▾ public ▾				↻	Search public
Library ▾	Share with ▾	New folder			
Name ^	Date modified	Type	Size		
css	01/05/2018 11:02	File folder			
dist	01/05/2018 13:50	File folder			
js	01/05/2018 13:50	File folder			
vendor	01/05/2018 13:50	File folder			
.htaccess	01/05/2018 11:02	HTACCESS File	1 KB		
favicon.ico	01/05/2018 11:02	Icon	0 KB		
index.php	01/05/2018 11:02	PHP File	2 KB		
robots.txt	01/05/2018 11:02	Text Document	1 KB		

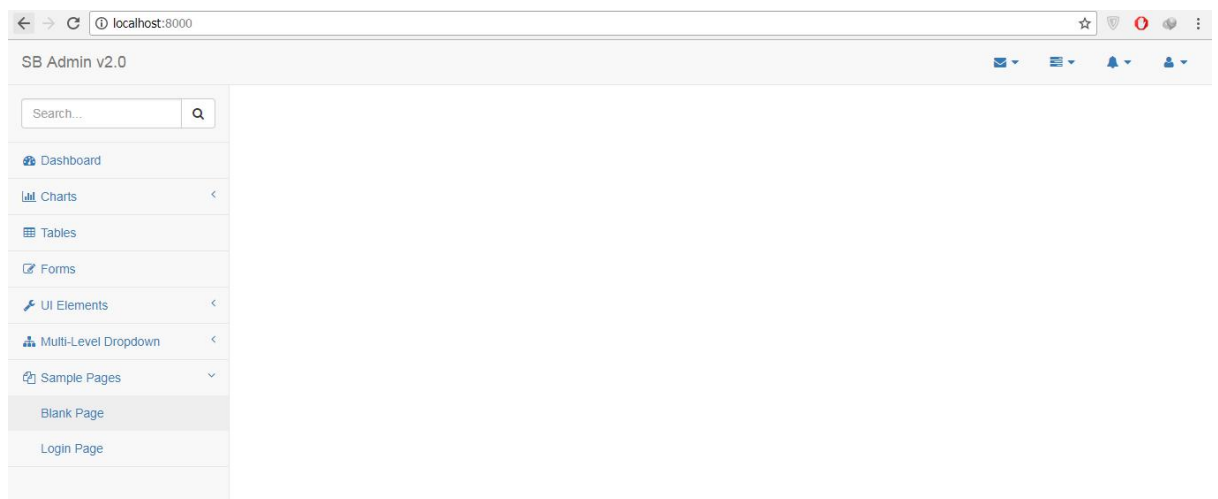
▾ Local Disk (D:) ▾ MyLara ▾ website ▾ resources ▾ views ▾				↻	Search views
Include in library ▾	Share with ▾	New folder			
Name ^	Date modified	Type	Size		
layouts	01/05/2018 13:51	File folder			
welcome.blade.php	01/05/2018 11:02	PHP File	3 KB		

**Gambar 21. Hasil Copy isi folder**

# WebDev UMM Tutorial



Gambar 22. Test dengan mengganti file routes/web.php

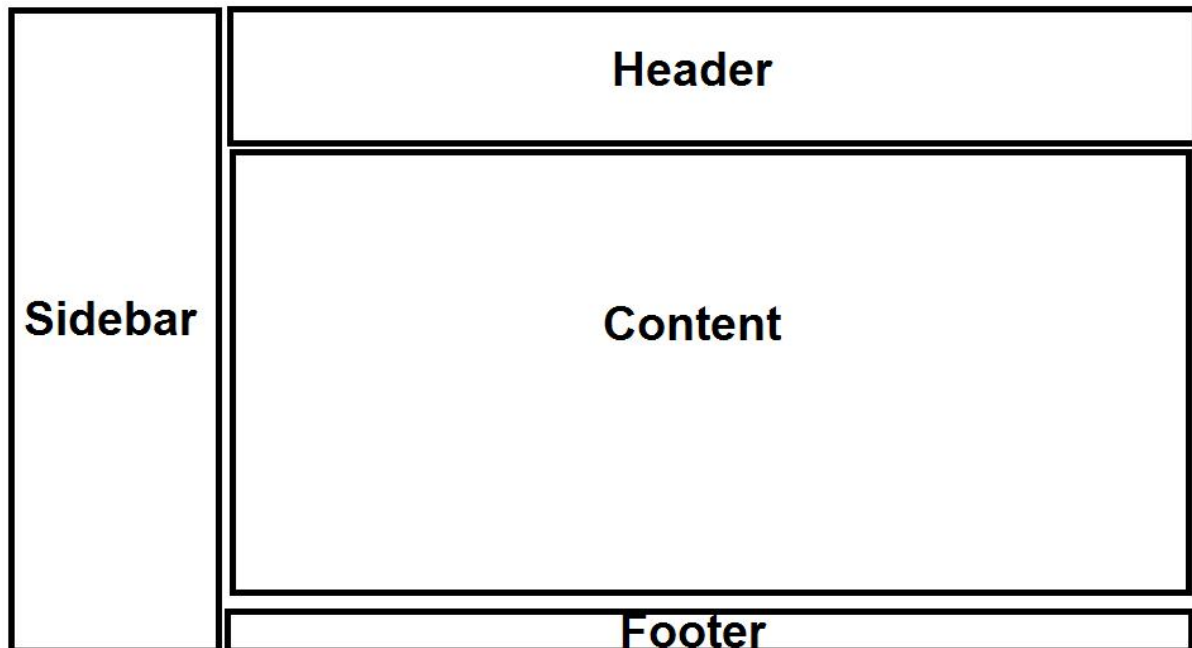


Gambar 23. Hasil Test Template

# WebDev UMM Tutorial

## Template Inheritance

Sama seperti PBO untuk penerapan template di Laravel menganut sistem inheritance.



Gambar 24. Struktur Template

Lihatlah pada Gambar 24. Pada contoh struktur template seperti ini terkadang kita tidak perlu untuk menampilkan keseluruhan Templatnya pada satu halaman. Kita dapat membuat sebuah layout yang berisi Header, sidebar, dan footer kemudian memisahkan Content karena Content biasanya adalah bagian yang berbeda pada halaman lain

Untuk kasus contoh ini buka file **resources/views/layouts/main.blade.php** kemudian edit bagian Content Here.

```
369 </ul>
370 </div>
371 <!-- /.sidebar-collapse -->
372 </div>
373 <!-- /.navbar-static-side -->
374 </nav>
375
376 <!-- Page Content -->
377 <div id="page-wrapper">
378   <div class="container-fluid">
379     <!-- Content Here -->
380     @yield('content')
381   </div>
382   <!-- /.container-fluid -->
383 </div>
384 <!-- /#page-wrapper -->
385
386 </div>
387 <!-- /#wrapper -->
388
389 <!-- jQuery -->
390 <script src="{asset('vendor/jquery/jquery.min.js')}"></script>
391
392 <!-- Bootstrap Core JavaScript -->
393 <script src="{asset('vendor/bootstrap/js/bootstrap.min.js')}"></script>
394
395 <!-- Metis Menu Plugin JavaScript -->
396 <script src="{asset('vendor/metisMenu/metisMenu.min.js')}"></script>
397
398 <!-- Custom Theme JavaScript -->
399 <script src="{asset('dist/js/sb-admin-2.js')}"></script>
400
401 <!-- DataTables JavaScript -->
```

Gambar 25. Edit main.blade.php

Dengan Penambahan ini maka setiap section content akan di masukkan kedalam bagian tersebut, Contoh pada Sesi CRUD.

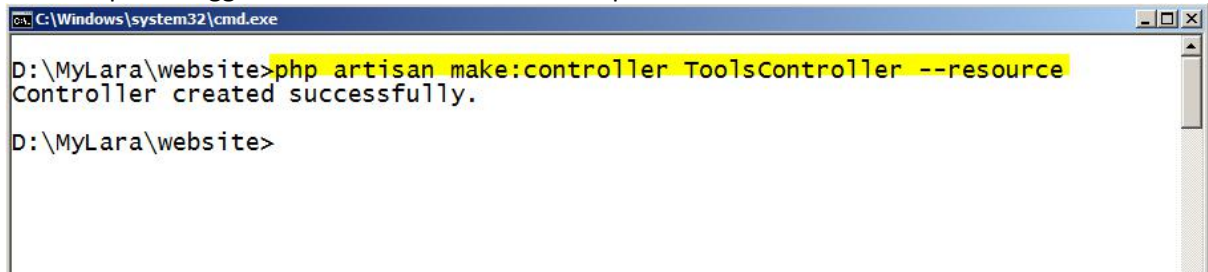
# WebDev UMM Tutorial

## CRUD ( Create Read Update Delete )

Pada tahap ini akan dilakukan untuk pembuatan CRUD menggunakan Laravel. Ada beberapa tahap yang harus diperhatikan

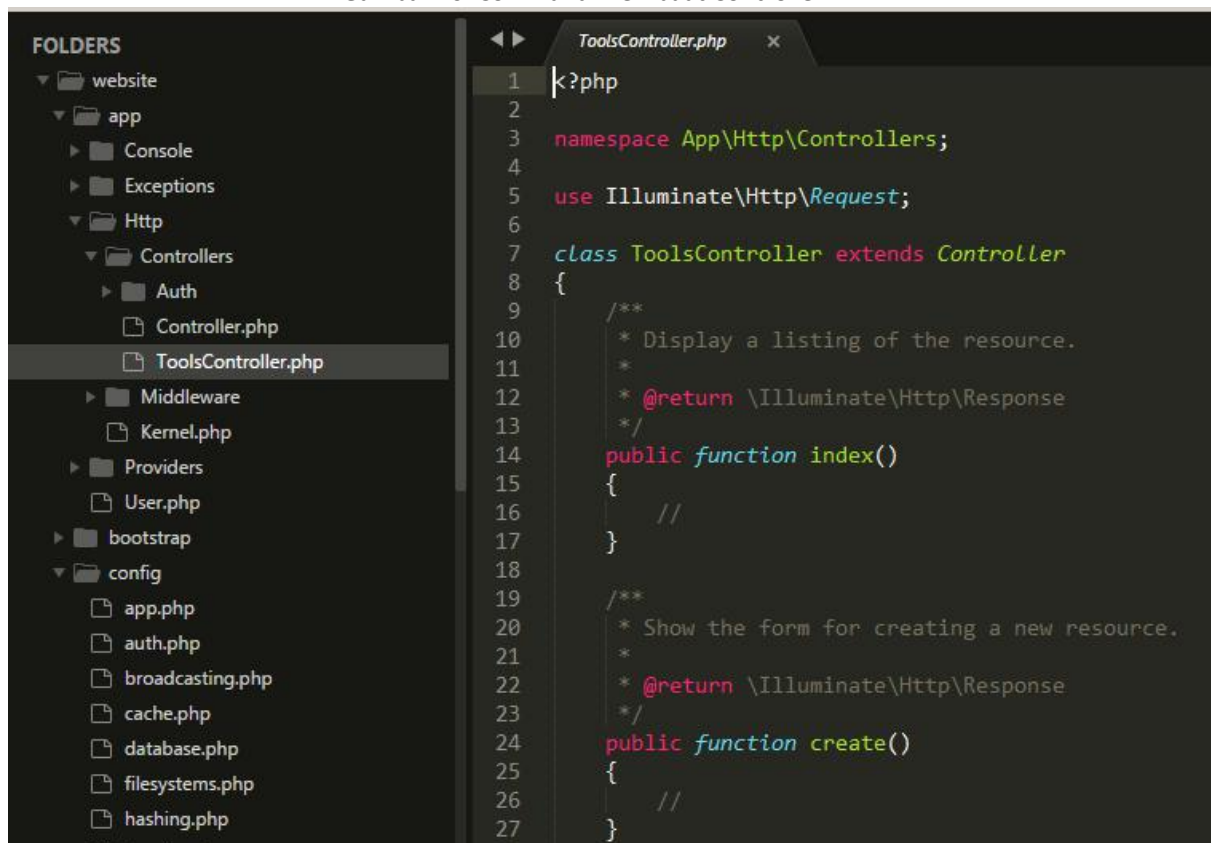
### 1. Pembuatan Controller

Pada kasus ini kita akan membuat controller untuk mengelola data table tools yang sudah kita buat pada Sesi **Migration**. Khusus untuk Controller yang digunakan untuk CRUD kita dapat menggunakan command --resource seperti contoh



```
C:\Windows\system32\cmd.exe
D:\MyLara\website>php artisan make:controller ToolsController --resource
Controller created successfully.
D:\MyLara\website>
```

Gambar 26. Command Membuat Controller



The image shows an IDE with a file explorer on the left and a code editor on the right. The file explorer shows the 'website' directory with subfolders 'app', 'bootstrap', and 'config'. The 'app' folder contains 'Console', 'Exceptions', 'Http', 'Middleware', 'Providers', and 'config'. The 'Http' folder contains 'Controllers', 'Auth', 'Kernel.php', 'User.php', and 'bootstrap'. The 'Controllers' folder contains 'Controller.php' and 'ToolsController.php'. The code editor shows the content of 'ToolsController.php'.

```
1 k?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class ToolsController extends Controller
8 {
9     /**
10      * Display a listing of the resource.
11      *
12      * @return \Illuminate\Http\Response
13      */
14     public function index()
15     {
16         //
17     }
18
19     /**
20      * Show the form for creating a new resource.
21      *
22      * @return \Illuminate\Http\Response
23      */
24     public function create()
25     {
26         //
27     }
28 }
```

Gambar 28. Hasil make:controller dari php artisan



# WebDev UMM Tutorial

## 2. Pembuatan Route

Bagian yang terpenting lagi kita perlu menambahkan route untuk melakukan akses ke controller yang sudah kita buat. Pada Gambar 28 maka hasil yang didapat untuk mengakses controller ToolsController adalah misal localhost:8000/tool



Gambar 28. Pembuatan Routes untuk ToolController

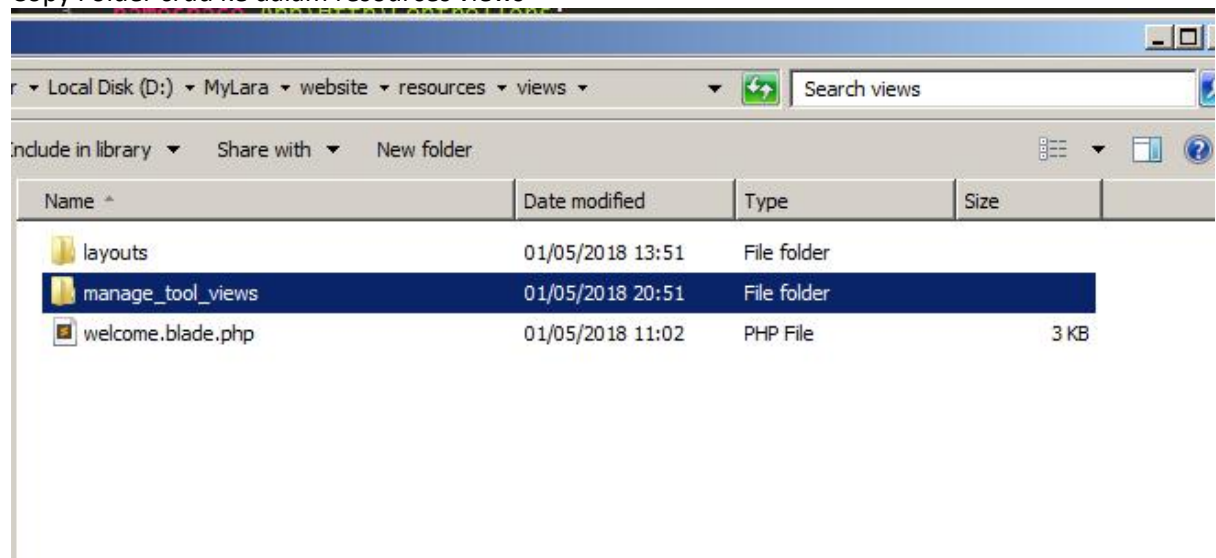
## 3. Pembuatan Model

```
D:\MyLara\website>php artisan make:model Tool
Model created successfully.
```

Gambar 29. Pembuatan Model Tool

## 4. Pembuatan Views

Copy Folder crud ke dalam resources views



Gambar 30. Hasil Copy File Views

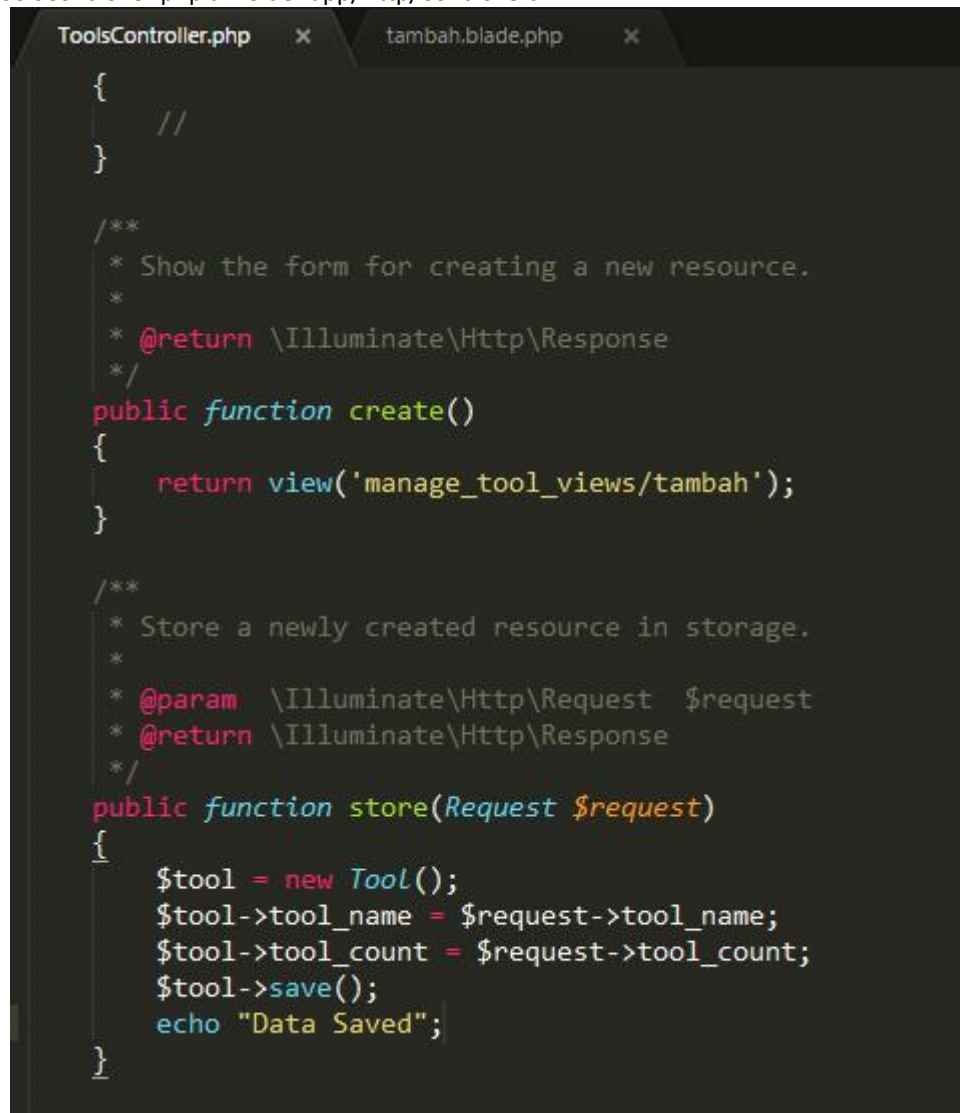
# WebDev UMM Tutorial

Sekarang Semua bahan sudah siap saatnya untuk melakukan pembuatan CRUD

## 1. Create

Ada beberapa File yang harus diedit yaitu

- a. ToolsController.php di Folder app/Http/Controllers



```
ToolsController.php x tambah.blade.php x
{
    //
}

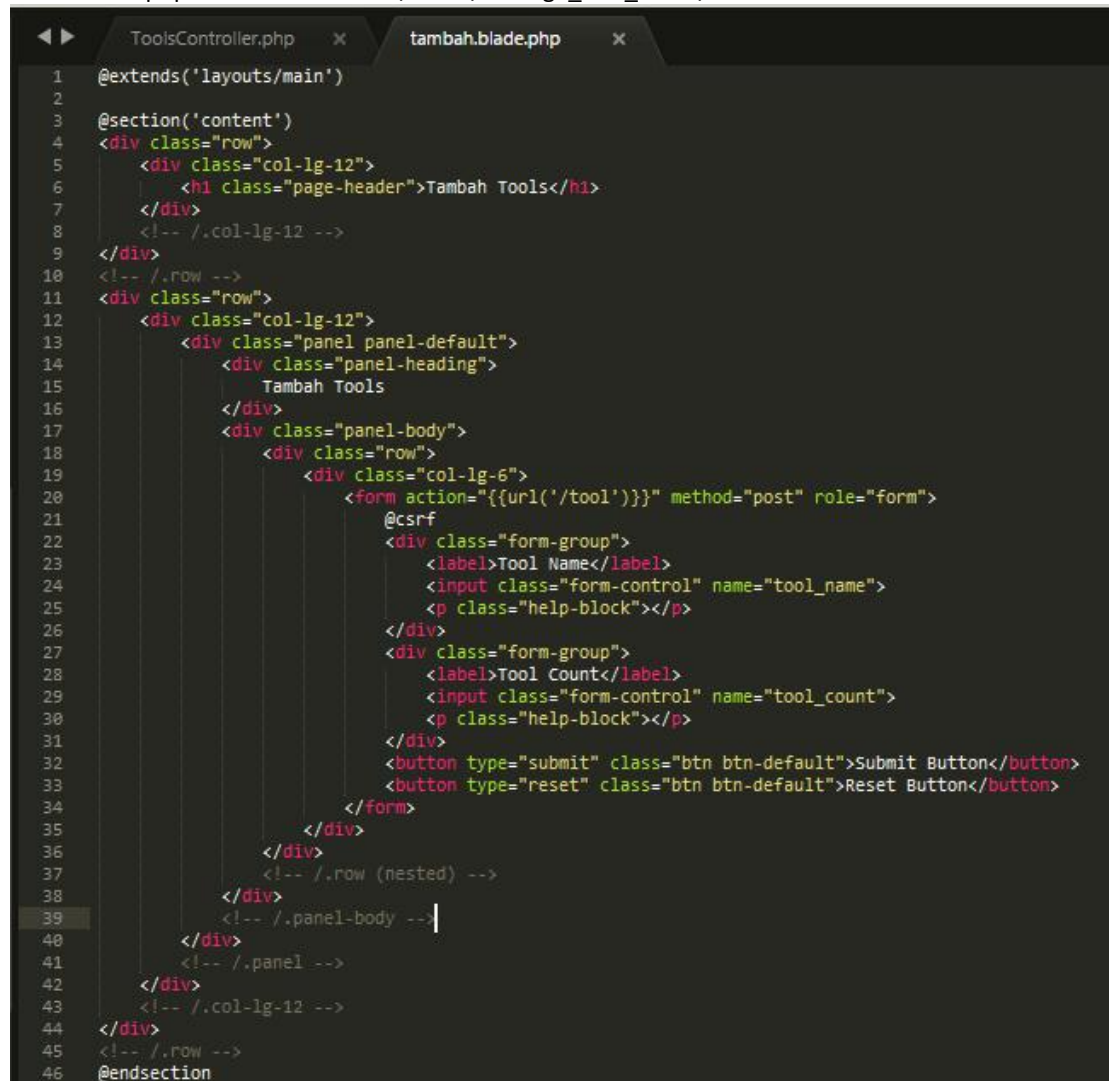
/**
 * Show the form for creating a new resource.
 *
 * @return \Illuminate\Http\Response
 */
public function create()
{
    return view('manage_tool_views/tambah');
}

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{
    $tool = new Tool();
    $tool->tool_name = $request->tool_name;
    $tool->tool_count = $request->tool_count;
    $tool->save();
    echo "Data Saved";
}
```

Gambar 31. ToolsController.php menambah fungsi tambah

# WebDev UMM Tutorial

b. tambah.blade.php di Folder resources/views/manage\_tool\_views/

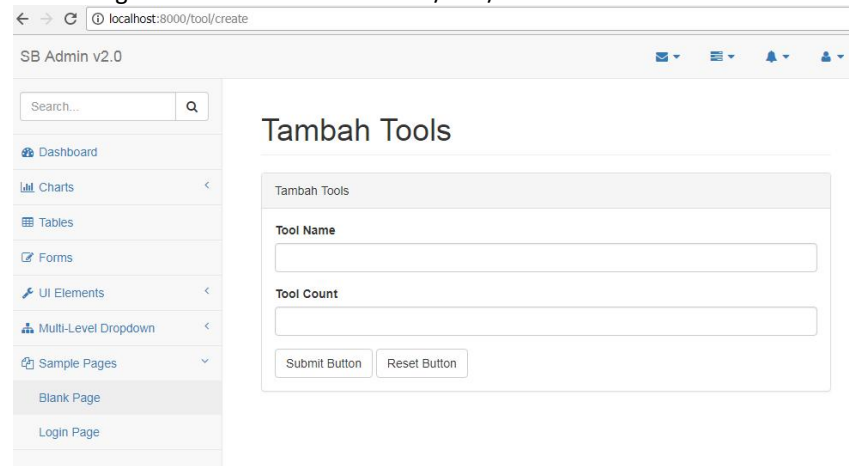


```
1 @extends('layouts/main')
2
3 @section('content')
4 <div class="row">
5     <div class="col-lg-12">
6         <h1 class="page-header">Tambah Tools</h1>
7     </div>
8     <!-- /.col-lg-12 -->
9 </div>
10 <!-- /.row -->
11 <div class="row">
12     <div class="col-lg-12">
13         <div class="panel panel-default">
14             <div class="panel-heading">
15                 Tambah Tools
16             </div>
17             <div class="panel-body">
18                 <div class="row">
19                     <div class="col-lg-6">
20                         <form action="{{url('/tool')}}" method="post" role="form">
21                             @csrf
22                             <div class="form-group">
23                                 <label>Tool Name</label>
24                                 <input class="form-control" name="tool_name">
25                                 <p class="help-block"></p>
26                             </div>
27                             <div class="form-group">
28                                 <label>Tool Count</label>
29                                 <input class="form-control" name="tool_count">
30                                 <p class="help-block"></p>
31                             </div>
32                             <button type="submit" class="btn btn-default">Submit Button</button>
33                             <button type="reset" class="btn btn-default">Reset Button</button>
34                         </form>
35                     </div>
36                 </div>
37             <!-- /.row (nested) -->
38         </div>
39     <!-- /.panel-body -->
40 </div>
41 <!-- /.panel -->
42 </div>
43 <!-- /.col-lg-12 -->
44 </div>
45 <!-- /.row -->
46 @endsection
```

Gambar 32. tambah.blade.php

Pada Gambar 32 barulah terlihat penggunaan extends layout pada Sesi **Template Integration** dengan ini akan mempermudah pembuatan view kita memasukkan section content ke dalam yield content ( Gambar 25 ). Sekarang untuk melakukan uji coba jalankan laravel **php artisan serve** jika belum dinyalakan.

Kemudian tes dengan membuka localhost:8000/tool/create maka akan muncul view seperti ini

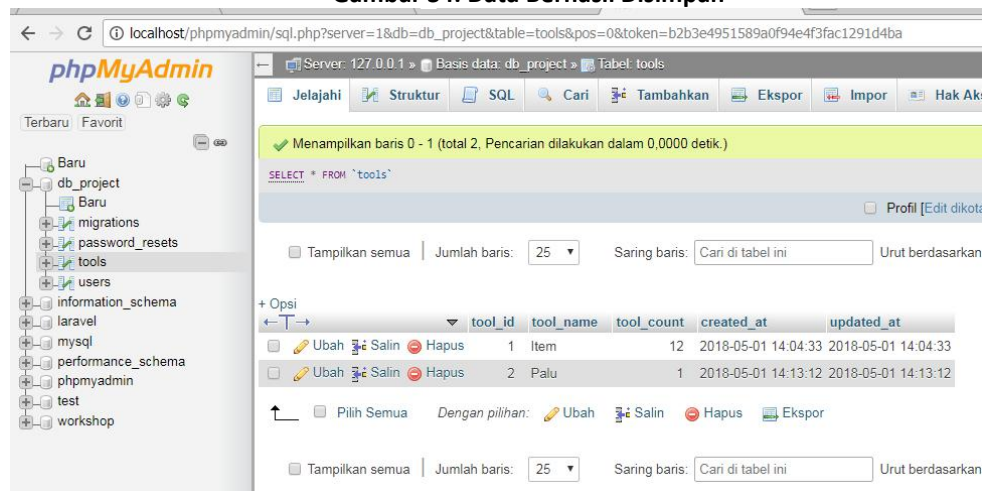


Gambar 33. localhost:8000/tool/create

# WebDev UMM Tutorial



**Gambar 34. Data Berhasil Disimpan**



**Gambar 35. Cek Database, Data Berhasil disimpan**

## 2. Read

Untuk read kita akan mengedit 2 File

a. ToolsController.php di app/Http/Controllers/ bagian fungsi index

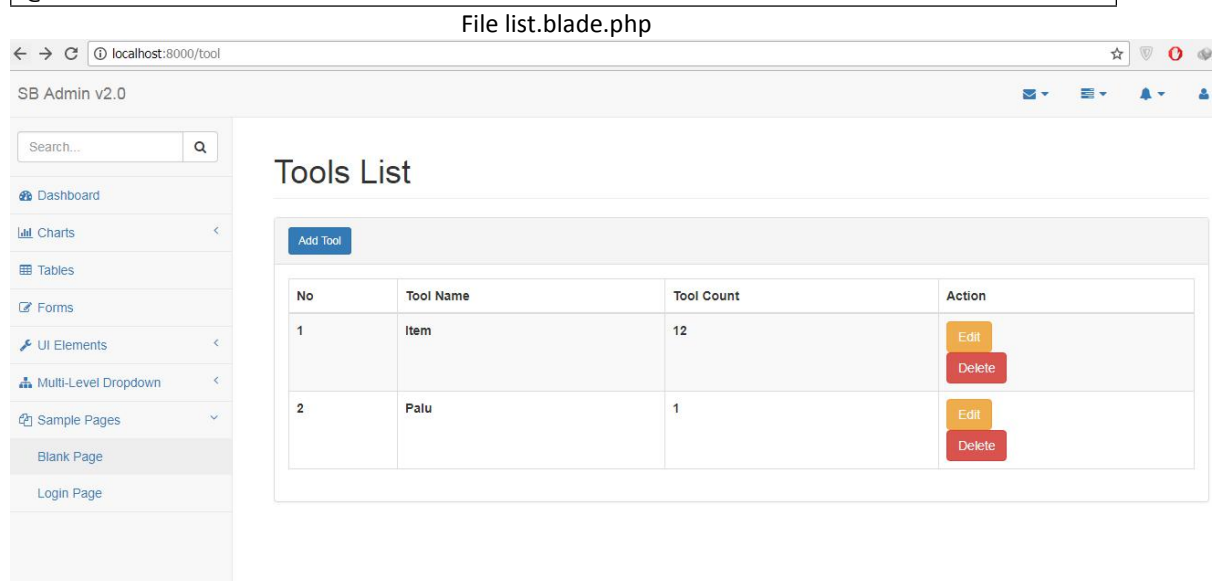
```
public function index()
{
    $data_tool = Tool::all();
    return view('manage_tool_views/list', compact('data_tool'));
}
```

b. list.blade.php di resources/views/manage\_tool\_views/

```
@extends('layouts/main')
@section('content')
<div class="row">
    <div class="col-lg-12">
        <h1 class="page-header">Tools List</h1>
    </div>
</div>
<!-- /.col-lg-12 -->
</div>
<!-- /.row -->
<div class="row">
    <div class="col-lg-12">
        <div class="panel panel-default">
            <div class="panel-heading">
                <a href="{{url('/tool/create')}}" class="btn btn-primary btn-sm">Add Tool</a>
            </div>
            <!-- /.panel-heading -->
            <div class="panel-body">
                <table width="100%" class="table table-striped table-bordered table-hover"
id="dataTables-example">
```

# WebDev UMM Tutorial

```
<thead>
  <tr>
    <th>No</th>
    <th>Tool Name</th>
    <th>Tool Count</th>
    <th>Action</th>
  </tr>
</thead>
<tbody>
  <?php $num=1;?>
  @foreach($data_tool as $key)
  <tr>
    <th>{{ $num++}}</th>
    <th>{{ $key->tool_name}}</th>
    <th>{{ $key->tool_count}}</th>
    <th>
      <a href="{{action('ToolsController@edit', $key->tool_id)}}" class="btn
btn-warning">Edit</a>
      <form action="{{action('ToolsController@destroy', $key->tool_id)}}"
method="post">
        @csrf
        <input name="_method" type="hidden" value="DELETE">
        <button class="btn btn-danger" type="submit">Delete</button>
      </form>
    </th>
  </tr>
  @endforeach
</tbody>
</table>
<!-- /.table-responsive -->
</div>
<!-- /.panel-body -->
</div>
<!-- /.panel -->
</div>
<!-- /.col-lg-12 -->
</div>
<!-- /.row -->
@endsection
```



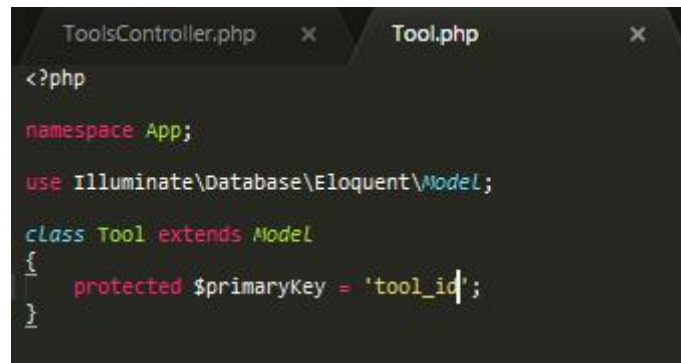
**Gambar 36. Hasil Test localhost:8000/tool**



# WebDev UMM Tutorial

## 3. Update

Pada Update File usahakan Model sudah benar bila id table / primary tidak default kita dapat setel di Tool.php seperti ini



```
<?php

namespace App;

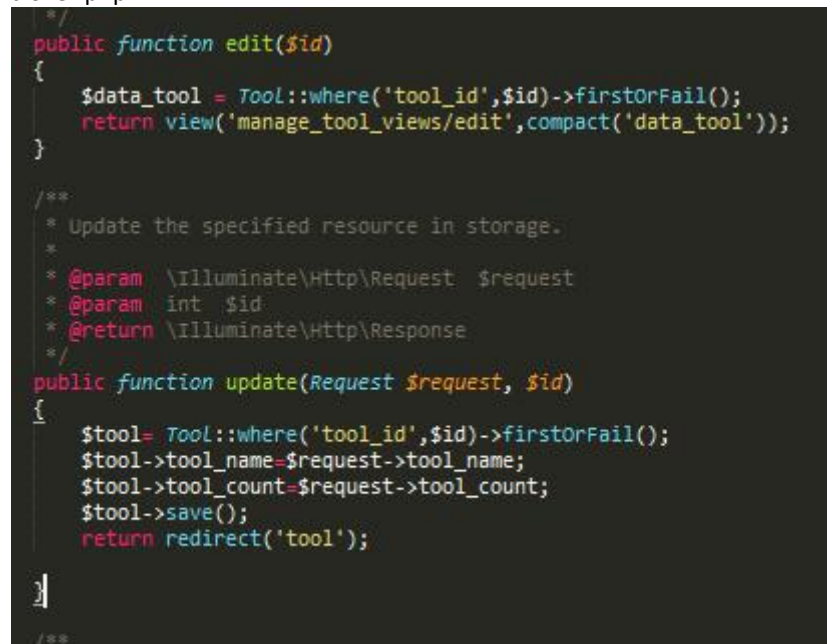
use Illuminate\Database\Eloquent\Model;

class Tool extends Model
{
    protected $primaryKey = 'tool_id';
}
```

Gambar 37. Set Primary Model

Edit File

a. ToolsController.php



```
/**
 * public function edit($id)
 * {
 *     $data_tool = Tool::where('tool_id',$id)->firstOrFail();
 *     return view('manage_tool_views/edit',compact('data_tool'));
 * }
 */
/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function update(Request $request, $id)
{
    $tool= Tool::where('tool_id',$id)->firstOrFail();
    $tool->tool_name=$request->tool_name;
    $tool->tool_count=$request->tool_count;
    $tool->save();
    return redirect('tool');
}
/**
```

Gambar 38. ToolsController.php untuk Update

# WebDev UMM Tutorial

## b. edit.blade.php

```
@extends('layouts/main')

@section('content')
<div class="row">
    <div class="col-lg-12">
        <h1 class="page-header">Edit Tools</h1>
    </div>
</div>
<!-- /.col-lg-12 -->
</div>
<!-- /.row -->
<div class="row">
    <div class="col-lg-12">
        <div class="panel panel-default">
            <div class="panel-heading">
                Edit Tools
            </div>
            <div class="panel-body">
                <div class="row">
                    <div class="col-lg-6">
                        <form action="{{action('ToolsController@update', $data_tool->tool_id)}}" method="post" role="form">
                            @method('PATCH')
                            @csrf
                            <div class="form-group">
                                <label>Tool Name</label>
                                <input class="form-control" name="tool_name" value="{{ $data_tool->tool_name }}">
                                <p class="help-block"></p>
                            </div>
                            <div class="form-group">
                                <label>Tool Count</label>
                                <input class="form-control" name="tool_count" value="{{ $data_tool->tool_count }}">
                                <p class="help-block"></p>
                            </div>
                            <button type="submit" class="btn btn-default">Submit Button</button>
                            <button type="reset" class="btn btn-default">Reset Button</button>
                        </form>
                    </div>
                </div>
            </div>
        </div>
    </div>
<!-- /.row (nested) -->
</div>
<!-- /.panel-body -->
</div>
<!-- /.panel -->
</div>
<!-- /.col-lg-12 -->
</div>
<!-- /.row -->
@endsection
```

Gambar 39. edit.blade.php

## 4. Delete

```
/**
 * Remove the specified resource from storage.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function destroy($id)
{
    $tool = Tool::find($id);
    $tool->delete();
    return redirect('tool')->with('success','Information has been deleted');
}
```

Gambar 40. fungsi untuk menghapus

Sekian untuk sesi Tutorial Laravel Kali ini. Jika mendapati error silahkan tanyakan di <http://t.me/webdevumm>