



# MODUL LARAVEL

XI RPL

TIARA KUSUMA DEWI

## DAFTAR ISI

MEMULAI PROJEK LARAVEL .....	2
MEMBUAT PAGE TAMPILAN BARU DI LARAVEL.....	4
Task 1: .....	10
MEMBUAT DATABASE MENGGUNAKAN LARAVEL.....	10
MENAMPILKAN TABLE POST MENGGUNAKAN LARAVEL .....	13
Task 2: .....	16
MENAMBAHKAN FITUR SEARCH DAN PAGINATION PADA POSTS LARAVEL.....	17
MENAMBAHKAN CATEGORIES PADA POST .....	19
Task 3 : .....	23
MEMBUAT FITUR LOGIN DAN REGISTER PADA LARAVEL .....	23
MEMBUAT FITUR INSERT DAN TAMPIL DATA POSTINGAN .....	38
Task 4: .....	47
MEMBUAT FITUR EDIT DAN DELETE DATA PADA POSTINGAN .....	47
Task 5: .....	52

## MEMULAI PROJEK LARAVEL

1. Download dan install aplikasi Composer pada link dibawah ini  
<https://getcomposer.org/download/>
2. Jalankan perintah dibawah untuk membuat project Laravel, atau bisa juga mengacu ke documentation Laravel pada <https://laravel.com/docs/>  
`composer create-project laravel/laravel laravelpost`
3. Jika project sudah jadi, maka akan muncul file dan folder dengan struktur seperti dibawah

app	29/08/2023 9:33	File folder	
bootstrap	29/08/2023 9:33	File folder	
config	29/08/2023 9:33	File folder	
database	29/08/2023 9:33	File folder	
public	30/08/2023 8:20	File folder	
resources	29/08/2023 9:33	File folder	
routes	29/08/2023 9:33	File folder	
storage	29/08/2023 9:33	File folder	
tests	29/08/2023 9:33	File folder	
vendor	25/10/2023 15:04	File folder	
.editorconfig	29/08/2023 9:33	Editor Config Sour...	1 KB
.env	01/09/2023 14:04	ENV File	1 KB
.env.example	29/08/2023 9:33	EXAMPLE File	1 KB
.gitattributes	29/08/2023 9:33	Git Attributes Sour...	1 KB
.gitignore	29/08/2023 9:33	Git Ignore Source ...	1 KB
.styleci.yml	29/08/2023 9:33	Yaml Source File	1 KB
artisan	29/08/2023 9:33	File	2 KB
composer.json	25/10/2023 15:04	JSON Source File	2 KB
composer.lock	25/10/2023 15:04	LOCK File	286 KB
package.json	29/08/2023 9:33	JSON Source File	1 KB
phpunit.xml	29/08/2023 9:33	XML Document	2 KB
README.md	29/08/2023 9:33	Markdown Source...	4 KB
server.php	29/08/2023 9:33	PHP Source File	1 KB
webpack.mix.js	29/08/2023 9:33	JavaScript File	1 KB

4. Jalankan aplikasi menggunakan script “php artisan serve”, lalu ctrl+klik pada IP yang tersedia.  
Atau jalankan url `laravelpost.test`



### **Documentation**

Laravel has wonderful, thorough documentation covering every aspect of the framework. Whether you are new to the framework or have previous experience with Laravel, we recommend reading all of the documentation from beginning to end.



### **Laracasts**

Laracasts offers thousands of video tutorials on Laravel, PHP, and JavaScript development. Check them out, see for yourself, and massively level up your development skills in the process.



### **Laravel News**

Laravel News is a community driven portal and newsletter aggregating all of the latest and most important news in the Laravel ecosystem, including new package releases and tutorials.



### **Vibrant Ecosystem**

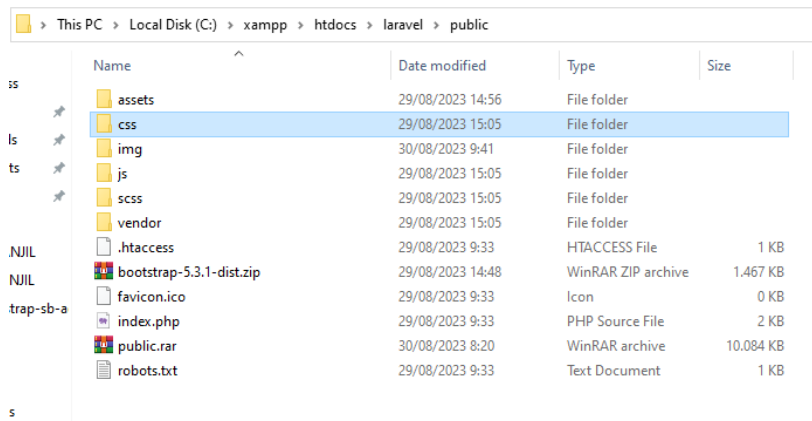
Laravel's robust library of first-party tools and libraries, such as [Forge](#), [Vapor](#), [Nova](#), and [Envoyer](#) help you take your projects to the next level. Pair them with powerful open source libraries like [Cashier](#), [Dusk](#), [Echo](#), [Horizon](#), [Sanctum](#), [Telescope](#), and more.

[Shop](#) [Sponsor](#)

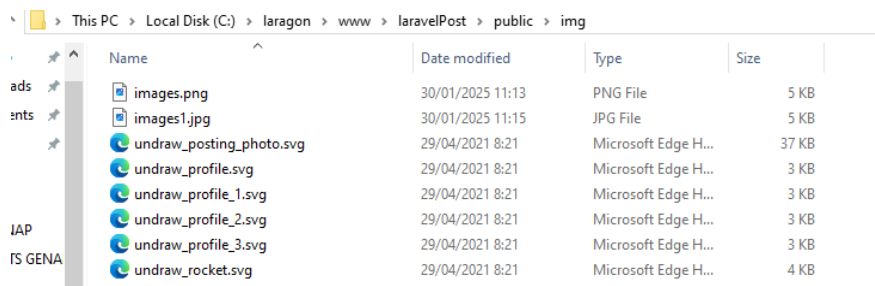
Laravel v6.50.0 (PHP v8.0.8)

## MEMBUAT PAGE TAMPILAN BARU DI LARAVEL

1. Buka aplikasi vscode dan buka folder pada project Laravel yang sudah dibuat.
2. Download folder assets, css, img, js yang telah disediakan dan letakkan pada folder “public”.



3. Letakkan gambar yang tersedia kedalam folder img



4. Buka folder routes/web.php untuk membuat rute link yang ingin dituju

```
Route::get('/about', function () {  
  
    return view('about', [  
  
        "title" => 'About',  
  
        "active" => 'about',  
  
        "name" => "Lulu Kumulu",  
  
        "email" => "lulukumulu@gmail.com",  
  
        "image" => "user2.png"  
  
    ]);  
  
});
```

5. Buka folder resource/view dan buat folder dengan nama layout, dan buat file dengan nama main.blade.php didalam folder tersebut

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="utf-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <meta name="description" content="">

    <meta name="author" content="">


    <title>Web Blog</title>

    <!-- Custom fonts for this template-->

    <link href="{{asset('vendor/fontawesome-free/css/all.min.css')}}" rel="stylesheet"
type="text/css">

    <link
href="{{asset('https://fonts.googleapis.com/css?family=Nunito:200,200i,300,300i,400,400i,600,6
00i,700,700i,800,800i,900,900i')}}"
rel="stylesheet">

    <!-- Custom styles for this template-->

    <link href="{{asset('css/sb-admin-2.min.css')}}" rel="stylesheet">


    <!-- Bootstrap icons -->

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.11.1/font/bootstrap-icons.css">
```

```

<!-- Bootstrap -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
T3c6Coli6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN"
crossorigin="anonymous">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
C6RzsynM9kWDrmNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0JlpcV8Qyq46cDfL"
crossorigin="anonymous"></script>

</head>

<body id="page-top">

<!-- Page Wrapper -->

<div id="wrapper">

<!-- Sidebar -->

<ul class="navbar-nav bg-gradient-primary sidebar sidebar-dark accordion"
id="accordionSidebar">

<!-- Sidebar - Brand -->

<a class="sidebar-brand d-flex align-items-center justify-content-center"
href="/dashboard">

<div class="sidebar-brand-icon rotate-n-15">

<i class="fas fa-laugh-wink"></i>

</div>

<div class="sidebar-brand-text mx-3">Web Blog</div>

</a>

<!-- Divider -->

```

```

<hr class="sidebar-divider my-0">

<!-- Nav Item - Dashboard -->

<li class="nav-item {{ Request::is('dashboard') ? 'active' : '' }}">

    <a class="nav-link" href="/dashboard">

        <i class="fas fa-fw fa-tachometer-alt"></i>

        <span>Dashboard</span></a>

    </li>

<li class="nav-item {{ Request::is('about') ? 'active' : '' }}">

    <a class="nav-link" href="/about">

        <i class="fas fa-fw fa-user"></i>

        <span>About</span></a>

    </li>

<li class="nav-item {{ Request::is('posts') ? 'active' : '' }}">

    <a class="nav-link" href="/posts">

        <i class="fas fa-fw fa-wrench"></i>

        <span>Blog</span></a>

    </li>

<li class="nav-item {{ Request::is('dashboard/post*') ? 'active' : '' }}">

    <a class="nav-link" href="/dashboard/post">

        <i class="fas fa-fw fa-table"></i>

        <span>My Posts</span></a>

    </li>

</ul>

```



```

        <!-- Content Wrapper -->

        <div id="content-wrapper" class="d-flex flex-column">

            <!-- Main Content -->

            <div id="content">

                <div class="container mt-4">

                    @yield('container')

                </div>

            </div>

        </div>

        <!-- Bootstrap core JavaScript-->

        <script src="{{asset('vendor/jquery/jquery.min.js')}}"></script>

        <script src="{{asset('vendor/bootstrap/js/bootstrap.bundle.min.js')}}"></script>

        <!-- Core plugin JavaScript-->

        <script src="{{asset('vendor/jquery-easing/jquery.easing.min.js')}}"></script>

        <!-- Custom scripts for all pages-->

        <script src="{{asset('js/sb-admin-2.min.js')}}"></script>

        <script src="https://unpkg.com/feather-icons"></script>

        <script>

            feather.replace();

        </script>

    </body>

</html>

```

6. Tambahkan file dengan nama about.blade.php pada resource/view dengan isi sebagai berikut

```
@extends('layout.main')
```

```
@section('container')
```

```
<span align="center" ><h2>Halaman {{$title}}</h2></span>
```

```
<form class="d-none d-sm-inline-block form-inline mr-auto ml-md-3 my-2 my-md-0 mw-100 navbar-search">
```

```
    <div class="card" style="width: 18rem;">
```

```
        
```

```
        <div class="card-body">
```

```
            <h5 class="card-title">{{$name}}</h5>
```

```
            <p class="card-text">Some quick example text to build on the card title and make up the bulk of the card's content.</p>
```

```
            <a href="#" class="btn btn-primary"><?= $email; ?></a>
```

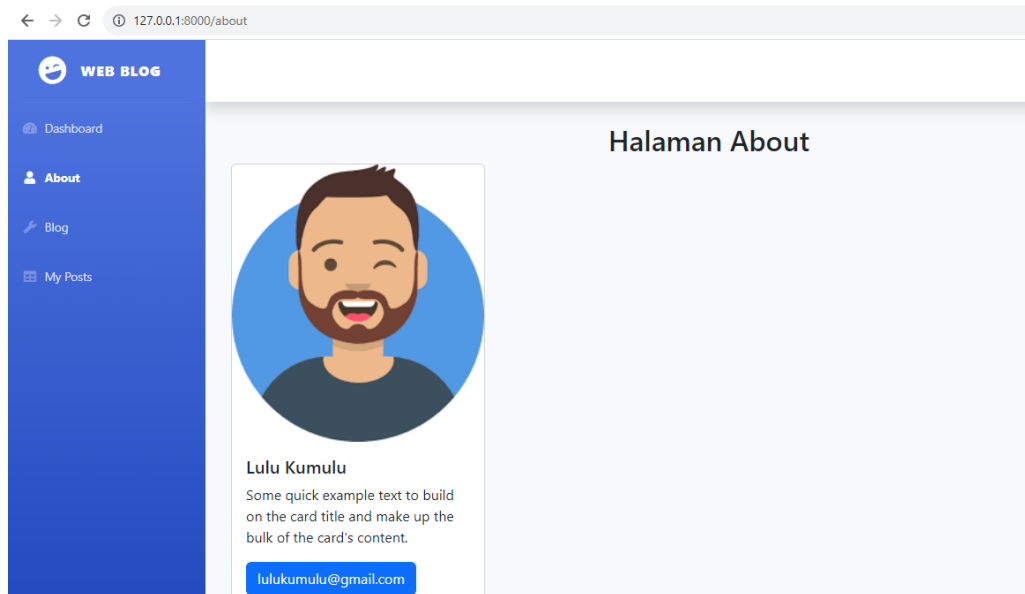
```
        </div>
```

```
    </div>
```

```
</form>
```

```
@endsection
```

7. Jalankan aplikasi menggunakan script “php artisan serve”, lalu ctrl+klik pada IP yang tersedia. Lalu tambahkan “/about” pada browser.



Task 1:

1. **BUATLAH 1 MENU DAN PAGE YANG BERISI TABLE BEBAS DAN TAMPILKAN!**

## MEMBUAT DATABASE MENGGUNAKAN LARAVEL

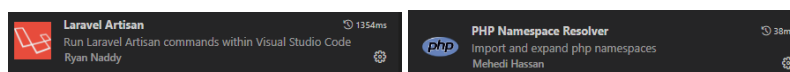
1. Untuk mengkoneksikan database dengan Laravel kita harus menuliskan nama database yang kita punya pada file .env

```

1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:M19ztZpHaQ9KDfpwVQXyHBmIJngOKbYOMkyJL/r5Dhg=
4 APP_DEBUG=true
5 APP_URL=http://lat-laravel.test/
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=blogging
15 DB_USERNAME=root
16 DB_PASSWORD=
17

```

2. Klik tab Extension pada vscode. Lalu cari dan download ekstensi Laravel Artisan dan PHP Namespace.



3. Buka folder database/migration. Didalam folder tersebut akan ditemukan beberapa file migration yang akan berguna untuk membuat struktur database dan tabel otomatis dari Laravel tanpa berhubungan langsung dengan DBMS.
4. Ketikkan perintah “php artisan migrate:fresh” pada terminal untuk membuat database dan tabel sesuai default.
5. Gunakan shortcut CTRL+SHIFT+P lalu ketikkan migration->create\_posts\_table->Yes->posts.
6. Lalu tunggu hingga proses selesai. Tambahkan perintah struktur tabel seperti dibawah ini.

```
public function up()
{
    Schema::create('posts', function (Blueprint $table) {
        $table->id();
        $table->foreignId('category_id');
        $table->foreignId('user_id');
        $table->string('title');
        $table->string('slug')->unique;
        $table->text('excerpt');
        $table->text('body');
        $table->timestamp('published_at')->nullable();
        $table->timestamps();
    });
}
```

7. Selain posts, buat juga migration table users dengan struktur seperti ini

```
public function up()
{
    Schema::create('users', function (Blueprint $table) {
        $table->id();
        $table->string('name');
        $table->string('username');
        $table->string('email')->unique();
        $table->timestamp('email_verified_at')->nullable();
        $table->string('password');
        $table->rememberToken();
        $table->timestamps();
    });
}
```

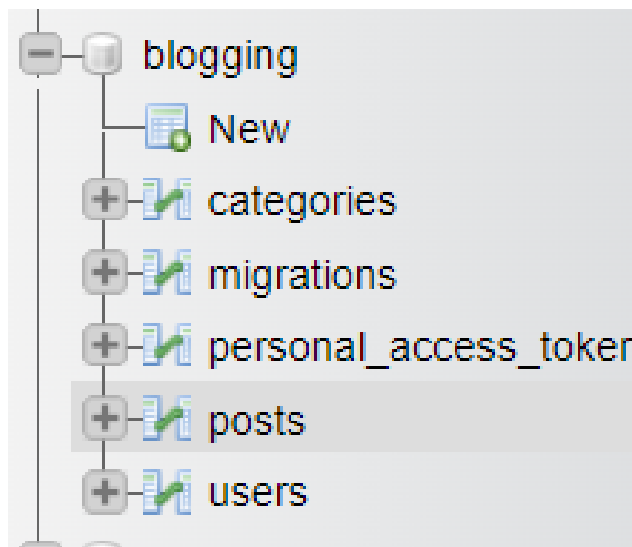
8. Dan juga migration categories dengan struktur dibawah ini.

```

public function up()
{
    Schema::create('categories', function (Blueprint $table) {
        $table->id();
        $table->string('name')->unique();
        $table->string('slug')->unique();
        $table->timestamps();
    });
}

```

9. Ketikkan perintah “php artisan migrate:fresh” pada terminal.



10. Silahkan isi tabel2 berikut dengan contoh isian seperti dibawah ini

- Table categories

id	name	slug	created_at	updated_at
1	Web Programming	web-programming	NULL	NULL
2	Android	android	NULL	NULL
3	Personal	personal	NULL	NULL
4	Daily Project	daily-project	NULL	NULL

- Table posts (user\_id dikosongkan saja terlebih dahulu)

id	category_id	user_id	title	slug	excerpt	body	publish
1	1	12	Judul Tulisan Pertama	judul-tulisan-pertama	Halooo ini adalah teks artikel tulisan pertama.	Halooo ini adalah teks artikel tulisan pertama. Lo...	NULL
2	2	11	Judul Tulisan Kedua	judul-tulisan-kedua	Halooo ini adalah teks artikel tulisan kedua.	Halooo ini adalah teks artikel tulisan kedua. LoLo...	NULL
3	1	10	Judul Tulisan Ketiga	judul-tulisan-ketiga	Halooo ini adalah teks artikel tulisan ketiga.	Halooo ini adalah teks artikel tulisan ketiga. LoL...	NULL
4	3	12	Judul Tulisan Keempat	judul-tulisan-keempat	Halooo ini adalah teks artikel tulisan keempat.	Halooo ini adalah teks artikel tulisan keempat. Lo...	NULL
5	4	11	test 123	test-123	Lorem ipsum dolor sit amet, consectetur adipiscing...	Lorem ipsum dolor sit amet, consectetur adipiscing...	NULL
8	3	11	post baru nih	post-baru-nih	Post baru nih. Lorem ipsum dolor sit amet, consectetur...	Post baru nih. Lorem ipsum dolor sit amet, consectetur...	NULL

## MENAMPILKAN TABLE POST MENGGUNAKAN LARAVEL

1. Buat file dengan nama posts.blade.php pada resource/view dengan isi sebagai berikut

```
@extends('layout.main')
```

```
@section('container')
```

```
<span align="center" ><h2>Halaman {{$title}}</h2></span>
```

```
@foreach($posts as $post)
```

```
<div class="card">
```

```
<div class="card-header">
```

```
{{$title}}
```

```
</div>
```

```
<div class="card-body">
```

```
<h5 class="card-title"><?= $post->title; ?></h5>
```

```
<p class="card-text"><?= $post->excerpt; ?></p>
```

```
<!-- <a href="/posts/{{$post['slug']}}" class="btn btn-primary">Read More</a> -->
```

```

        <a href="/posts/{{ $post->slug }}" class="btn btn-primary">Read More</a>

    </div>

</div>

<br>

@endforeach

@endsection

```

2. Buat file dengan nama `post.blade.php` dengan isi sebagai berikut

```

@extends('layout.main')

@section('container')

<div class="card">

    <div class="card-header">

        {{ $title }}

    </div>

    <div class="card-body">

        <h5 class="card-title"><?= $post->title; ?></h5>

        <!-- <?= $post["body"]; ?><br><br> -->

        {!! $post->body !!}<br>

        <a href=".." /posts" class="btn btn-primary">Back To Posts</a>

    </div>

</div>

<br>

@endsection

```

3. Tambahkan perintah rute berikut pada `web.php`

```

Route::get('/posts', [PostController::class, 'index']);

Route::get('/posts/{post:slug}', [PostController::class, 'show']);

```

4. Lalu tambahkan perintah dibawah ini pada bagian atas web.php

```
use App\Models\Post;
```

```
use App\Http\Controllers\PostController;
```

5. Klik CTRL + SHIFT + P, lalu ketik "CONTROLLER" -> enter->PostController untuk membuat controller. Lalu tambahkan script berikut.

```
public function index(){  
  
    return view('posts',[  
  
        "title" => "All Posts",  
  
        "active"=>'posts',  
  
        "posts" =>Post::all()  
  
    ]);  
  
}
```

```
public function show(Post $post){  
  
    return view('post',[  
  
        "title" =>"Single Post",  
  
        "active"=>'posts',  
  
        "post" => $post  
  
    ]);  
  
}
```

6. Lalu tambahkan perintah "use App\Models\Post;" dibagian atas class controllernya.
7. Klik CTRL+SHIFT+P, ketik MODEL->Post. Untuk membuat model dengan nama Post. Lalu tambahkan perintah "protected \$guarded=['id'];"
8. Buka terminal dan ketikkan "php artisan serve" untuk menjalankan web. Dan tampilan akan muncul sebagai berikut



All Posts

### post baru nih

Post baru nih. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean condimentum risus et rutrum elementum. Donec quis ipsum et quam facilisis vestibulum non ac eros. Donec tristique velit e...

[Read More](#)

All Posts

### test 123

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean condimentum risus et rutrum elementu...

[Read More](#)

Single Post

### post baru nih

By: Tiara Kusuma Dewi in [Personal](#)

Post baru nih. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean condimentum risus et rutrum elementum. Donec quis ipsum et quam facilisis vestibulum non ac eros. Donec tristique velit eget interdum vulputate. Curabitur vitae velit rhoncus, vestibulum metus at, tincidunt ante. Nunc dapibus nibh ac purus consequat tristique. In vel sapien fringilla orci semper finibus nec sodales est. Vivamus quis est finibus ligula faucibus pellentesque. Vestibulum dui elit, pulvinar vitae suscipit sit amet, rutrum sit amet est. Nam et laoreet libero. Aliquam arcu odio, semper vitae tortor in, pharetra egestas eros. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Suspendisse bibendum ante euismod risus pretium egestas. Nam suscipit purus sit amet odio feugiat, at rutrum arcu sollicitudin. Aliquam vitae metus nisl. Nullam a arcu tortor. Nunc at dui enim. Nulla facilisi. In nec posuere magna. Nulla consectetur est nibh, ac gravida nisl imperdiet euismod. Donec sit amet elementum felis. Suspendisse mollis vel metus eget euismod. Suspendisse ex quam, laoreet quis malesuada in, bibendum non mauris. Fusce a faucibus diam. Fusce bibendum pellentesque erat, a rutrum arcu interdum id. Proin vitae pellentesque nunc. Sed a lacus varius, fringilla ante quis, tincidunt enim. Sed iaculis, sem eu placerat volutpat, erat est mollis massa, eget bibendum nisl metus sed sapien.

[Back To Posts](#)

## Task 2:

1. SILAHKAN IMPLEMENTASIKAN TABLE COMMENTS DENGAN STRUKTUR DIBAWAH DENGAN ALUR YANG SAMA SEPERTI POSTS DAN POST DIATAS!

<input type="checkbox"/>	1	id	bigint(20)	UNSIGNED	No	None	AUTO_INCREMENT	<a href="#">Change</a>	<a href="#">Drop</a>	<a href="#">More</a>
<input type="checkbox"/>	2	category_id	bigint(20)		No	None		<a href="#">Change</a>	<a href="#">Drop</a>	<a href="#">More</a>
<input type="checkbox"/>	3	slug	varchar(255) utf8mb4_unicode_ci		No	None		<a href="#">Change</a>	<a href="#">Drop</a>	<a href="#">More</a>
<input type="checkbox"/>	4	username	varchar(255) utf8mb4_unicode_ci		No	None		<a href="#">Change</a>	<a href="#">Drop</a>	<a href="#">More</a>
<input type="checkbox"/>	5	tanggal_post	varchar(255) utf8mb4_unicode_ci		No	None		<a href="#">Change</a>	<a href="#">Drop</a>	<a href="#">More</a>
<input type="checkbox"/>	6	excerpt	varchar(255) utf8mb4_unicode_ci		No	None		<a href="#">Change</a>	<a href="#">Drop</a>	<a href="#">More</a>
<input type="checkbox"/>	7	body	varchar(255) utf8mb4_unicode_ci		No	None		<a href="#">Change</a>	<a href="#">Drop</a>	<a href="#">More</a>
<input type="checkbox"/>	8	created_at	timestamp		Yes	NULL		<a href="#">Change</a>	<a href="#">Drop</a>	<a href="#">More</a>
<input type="checkbox"/>	9	updated_at	timestamp		Yes	NULL		<a href="#">Change</a>	<a href="#">Drop</a>	<a href="#">More</a>

# MENAMBAHKAN FITUR SEARCH DAN PAGINATION PADA POSTS

## LARAVEL

1. Modifikasi file posts.blade.php menjadi seperti dibawah ini:

```
@extends('layout.main')

@section('container')

<span align="center" ><h2>Halaman {{$title}}</h2></span>

<div class="row justify-content-center mb-3">

    <div class="col-md-6">

        <form action="/posts" method="get">

            <div class="input-group mb-3">

                <input type="text" class="form-control" placeholder="Search.." name="search"
value="{{$request('search')}}">

                <button class="btn btn-primary" type="submit" id="button-addon2">Search</button>

            </div>

        </form>

    </div>

</div>

@foreach($posts as $post)

<div class="card">

    <div class="card-header">

        {{$title}}

    </div>

    <div class="card-body">

        <h5 class="card-title"><?= $post->title; ?></h5>

        <p class="card-text"><?= $post->excerpt; ?></p>

    </div>

</div>

</div>
```

```

<!-- <a href="/posts/{{ $post['slug'] }}" class="btn btn-primary">Read More</a> -->

<a href="/posts/{{ $post->slug }}" class="btn btn-primary">Read More</a>

</div>

</div>

<br>

@endforeach

```

```

<div class="d-flex justify-content-end">

    {{ $posts->links() }}

</div>

@endsection

```

## 2. Ubah index pada PostController menjadi

```

public function index(){

    // dd(request('search'));

    return view('posts',[

        "title" => "All Posts",

        "active"=>'posts',

        // "posts" =>Post::latest()->filter(request(['search']))->get(),

        "posts" =>Post::latest()->filter(request(['search']))->paginate(2)->withQueryString()

    ]);

}

```

## 3. Tambahkan function scopeFilter pada model Post.php

```

public function scopeFilter($query, array $filters)

```

```
{
    $query->when($filters['search'] ?? false, function ($query, $search){
        return $query->where('title','like','%'.$search.'%')
        ->orWhere('body','like','%'.$search.'%');
    });
}
```

#### 4. Tambahkan script pada Providers/AppProvider.php

```
use Illuminate\Pagination\Paginator;
```

```
public function boot()
```

```
{
    Paginator::useBootstrap();
}
```

## MENAMBAHKAN CATEGORIES PADA POST

### 1. Buat table bernama categories dengan struktur seperti dibawah ini:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>id</b> 🔑	bigint		UNSIGNED	No	None		AUTO_INCREMENT
2	<b>name</b> 🔑	varchar(255)	utf8mb4_unicode_ci		No	None		
3	<b>slug</b> 🔑	varchar(255)	utf8mb4_unicode_ci		No	None		
4	<b>description</b>	varchar(200)	utf8mb4_unicode_ci		No	None		
5	<b>created_at</b>	timestamp			Yes	NULL		
6	<b>updated_at</b>	timestamp			Yes	NULL		

### 2. Lalu isi tabel tersebut dengan data sebagai berikut, selain itu pastikan juga field category\_id pada tabel post sudah terisi

id	name	slug	description
1	Web Programming	web-programming	Berikut adalah kumpulan postingan tentang Web Prog...
2	Android	android	Berikut adalah kumpulan postingan tentang Android
3	Personal	personal	Berikut adalah kumpulan postingan tentang Personal
4	Daily Project	daily-project	Berikut adalah kumpulan postingan tentang Daily Pr...

3. Tambahkan route pada web dibawah ini

```
Route::get('/categories', [CategoryController::class, 'index']);
```

```
Route::get('categories/{category:slug}', function(Category $category){
    return view('category',[
        'title'=>$category->name,
        'posts'=>$category->posts,
        'category'=>$category->name
    ]);
});
```

4. Tambahkan perintah **“use App\Http\Controllers\CategoryController;”** Pada bagian atas web.php.

5. Buat model dengan nama Category.php dengan isi model seperti ini

```
use HasFactory;

protected $guarded = ['id'];

public function posts()
{
    return $this->hasMany(Post::class);
}
```

6. Buat file controller dengan CategoryController dengan isi file:

```
public function index()
{
    return view('categories', [
        'categories' => Category::all(),
        'posts' => Post::all()
    ]);
}
```

7. Tambahkan perintah **“use App\Models\Post;”** dan **“use App\Models\Category;”** pada file CategoryController.php.

8. Buat file dengan nama category.blade.php yang berisi

```
@extends('layout.main')

@section('container')

<h3> Post Category : {{$category}}</h3>

<div class="row row-cols-1 row-cols-md-2 g-4">

    <div class="col">

        @foreach($posts as $post)

            <div class="card">

                <!-- 

                <div class="card-body">

                    <h5 class="card-title">

                        <a href="/posts/{{$post->slug}}">{{$post->title}}</a>

                    </h5>

                    <p class="card-text">{{$post->excerpt}}</p>

                </div>

            </div><br>

        @endforeach

    </div></div>
```

```
</div>
```

```
</div>
```

```
@endsection
```

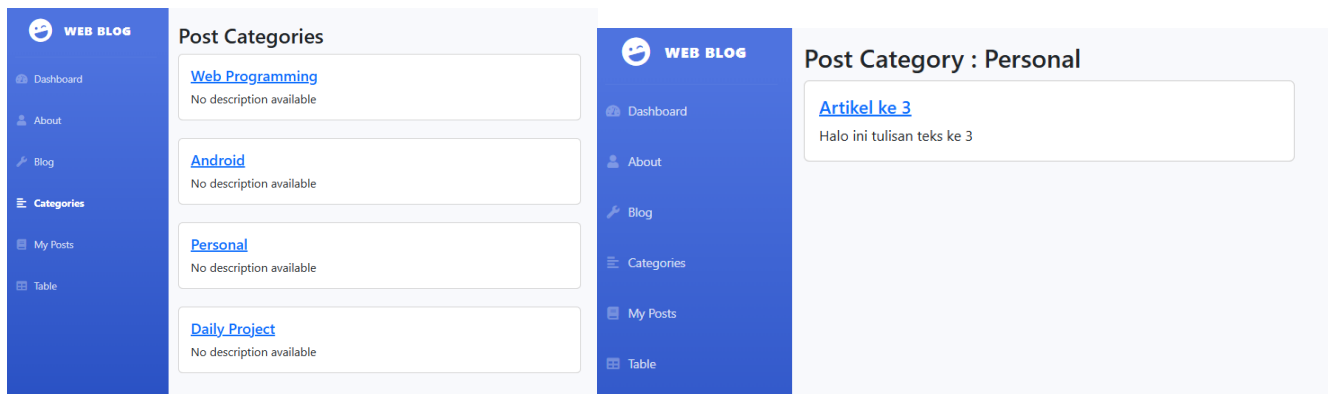
9. Buat file view dengan nama `categories.blade.php`. Lalu isi dengan file:

```
@extends('layout.main')
@section('container')
<h3>Post Categories</h3>
<div class="row row-cols-1 row-cols-md-2 g-4">
  <div class="col">
    @foreach($categories as $category)
      <div class="card">
        <div class="card-body">
          <h5 class="card-title">
            <a href="/categories/{{ $category->slug }}">{{ $category->name }}</a>
          </h5>
          <p class="card-text">
            {{ $category->description ?? 'No description available' }}
          </p>
        </div>
      </div><br>
    @endforeach
  </div>
</div>
@endsection
```

10. Buka file `main.blade.php` lalu tambahkan script:

```
<li class="nav-item {{ Request::is('categor*') ? 'active' : '' }}">
  <a class="nav-link" href="/categories">
    <i class="fas fa-fw fa-align-left"></i>
    <span>Categories</span></a>
</li>
```

SETELAH DIBUAT, MAKA AKAN MENAMPILKAN TAMPILAN KATEGORI SEPERTI DIBAWAH INI:



Task 3 :

1. TAMBAHKAN FITUR SEARCH, PAGINATION DAN CATEGORY PADA FIELD COMMENTS DAN TAMPIKAN!

## MEMBUAT FITUR LOGIN DAN REGISTER PADA LARAVEL

1. Tambahkan perintah dibawah pada web.php

```
Route::post('/login', [LoginController::class,'authenticate']);
```

```
Route::post('/register', [RegisterController::class,'store']);
```

```
Route::group(['middleware' => ['guest']],function() {
```

```
Route::get('/login', [LoginController::class,'index'])->name('login');
```

```
Route::get('/register', [RegisterController::class,'index']);
```

```
});
```

```
Route::get('/logout', [LoginController::class,'logout'])->name('logout');
```

```
Route::get('/dashboard',function () {
```

```
    return view('dashboard.index');
```

```
});
```

2. Tambahkan perintah dibawah diawal file web.php

```
use App\Http\Controllers\LoginController;
```

```
use App\Http\Controllers\RegisterController;
```



3. Buat controller dengan nama LoginController dan RegisterController
4. Isi file LoginController dengan file dibawah ini

```
public function index()

{

    return view('login.index',[

        'title'=>'Login'

    ]);

}

public function authenticate(Request $request)

{

    $credentials = $request->validate([

        'email'=>'required|email:dns',

        'password'=>'required'

    ]);

    if (Auth::attempt($credentials)) {

        $request->session()->regenerate();

        return redirect()->intended('/login');

    }

    return back()->with('loginError','Login Failed!');

}

public function logout(Request $request)

{

    Auth::logout();

    $request->session()->invalidate();

    $request->session()->regenerateToken();

}
```

```

        return redirect('/');
    }

```

5. Tambahkan perintah “use Illuminate\Support\Facades\Auth;” pada bagian atas LoginController.
6. Buat folder dan file dengan nama login/index.blade.php didalam folder view. Dengan isian file sebagai berikut

```

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="utf-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <meta name="description" content="">

    <meta name="author" content="">

    <title>Halaman Login</title>

    <!-- Custom fonts for this template-->

    <link href="vendor/fontawesome-free/css/all.min.css" rel="stylesheet" type="text/css">

    <link

href="https://fonts.googleapis.com/css?family=Nunito:200,200i,300,300i,400,400i,600,600i,700,700i,800,800i,900,900i"

rel="stylesheet">

    <!-- Custom styles for this template-->

    <link href="css/sb-admin-2.min.css" rel="stylesheet">

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
T3c6Coli6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN"
crossorigin="anonymous">

```

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
crossorigin="anonymous"></script>
```

```
</head>
```

```
<body class="bg-gradient-primary">
```

```
<div class="container">
```

```
<!-- Outer Row -->
```

```
<div class="row justify-content-center">
```

```
<div class="col-xl-10 col-lg-12 col-md-9">
```

```
@if(session()->has('success'))
```

```
<div class="alert alert-light alert-dismissible fade show" role="alert">
```

```
{{ session('success') }}
```

```
<button type="button" class="btn-close" data-bs-dismiss="alert" aria-
label="Close"></button>
```

```
</div>
```

```
@endif
```

```
@if(session()->has('loginError'))
```

```
<div class="alert alert-danger alert-dismissible fade show" role="alert">
```

```
{{ session('loginError') }}
```

```
<button type="button" class="btn-close" data-bs-dismiss="alert" aria-
label="Close"></button>
```

```
</div>
```

```
@endif
```

```

<div class="card o-hidden border-0 shadow-lg my-5">

    <div class="card-body p-0">

        <!-- Nested Row within Card Body -->

        <div class="row">

            <div class="col-lg-6 d-none d-lg-block bg-login-image"></div>

            <div class="col-lg-6">

                <div class="p-5">

                    <div class="text-center">

                        <h1 class="h4 text-gray-900 mb-4">Welcome Back!</h1>

                    </div>

                    <form class="user" action="/login" method="post">

                        @csrf

                        <div class="form-group">

                            <input name="email" type="email" class="form-control form-control-
user @error('email') is-invalid @enderror"

                                id="exampleInputEmail" aria-describedby="emailHelp"

                                placeholder="Email" autofocus required>

                                    @error('email')

                                <div class="invalid-feedback">

                                    {{ $message }}

                                </div>

                                    @enderror

                                </div>

                                <div class="form-group">

```

```

        <input name="password" type="password" class="form-control form-
control-user"

        id="exampleInputPassword" placeholder="Password" required>

    </div>

    <div class="form-group">

        <div class="custom-control custom-checkbox small">

            <input type="checkbox" class="custom-control-input"
id="customCheck">

            <label class="custom-control-label" for="customCheck">Remember
                Me</label>

        </div>

    </div>

    <button class="btn btn-primary btn-user btn-block">

        Login

    </button>

</form>

<hr>

<div class="text-center">

    <a class="small" href="/register">Create an Account!</a>

</div>

</div>

</div>

</div>

</div>

```

```

        </div>

    </div>

</div>

</div>

<!-- Bootstrap core JavaScript-->

<script src="vendor/jquery/jquery.min.js"></script>

<script src="vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

<!-- Core plugin JavaScript-->

<script src="vendor/jquery-easing/jquery.easing.min.js"></script>

<!-- Custom scripts for all pages-->

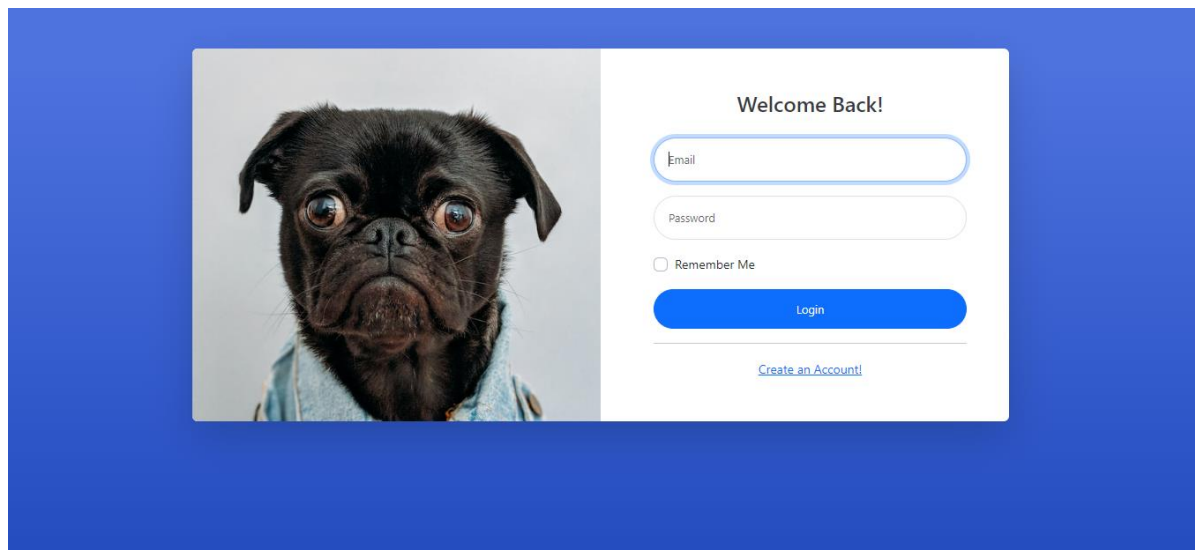
<script src="js/sb-admin-2.min.js"></script>

</body>

</html>

```

7. Coba running dan akan muncul tampilan login seperti dibawah ini



8. Isi file RegisterController dengan file dibawah ini

```

public function index()

{

```

```

return view('register.index',[

    'title' => 'Register',

    'active' => 'register'

]);

}

public function store(Request $request)
{

    $validatedData=$request->validate([

        'name'=>'required|max:255',

        'username'=>['required','min:3','max:255','unique:users'],

        'email'=>'required|email|unique:users|email:dns',

        'password'=>'required|min:5|max:255'

    ]);

    $validatedData['password']=Hash::make($validatedData['password']);

    User::create($validatedData);

    return redirect('/login')->with('success','Registration Successfull!! Please Login');

}

```

9. Tambahkan perintah “use App\Models\User;” dan “use Illuminate\Support\Facades\Hash;” pada awal RegisterController.

10. Buat folder dan file dengan nama register/index.blade.php didalam folder view dengan isi file sebagai berikut

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="">

```

```

<meta name="author" content="">
<title>Register</title>
<!-- Custom fonts for this template-->
<link href="vendor/fontawesome-free/css/all.min.css" rel="stylesheet" type="text/css">
<link
href="https://fonts.googleapis.com/css?family=Nunito:200,200i,300,300i,400,400i,600,600i,
700,700i,800,800i,900,900i"
rel="stylesheet">
<!-- Custom styles for this template-->
<link href="css/sb-admin-2.min.css" rel="stylesheet">
</head>
<body class="bg-gradient-primary">
<div class="container">
<div class="card o-hidden border-0 shadow-lg my-5">
<div class="card-body p-0">
<!-- Nested Row within Card Body -->
<div class="row">
<div class="col-lg-5 d-none d-lg-block bg-register-image"></div>
<div class="col-lg-7">
<div class="p-5">
<div class="text-center">
<h1 class="h4 text-gray-900 mb-4">Create an Account!</h1>
</div>
<form action="/register" method="POST" class="user">
@csrf
<div class="form-group">
<input type="text" name="name" class="form-control form-control-
user @error('name') is-invalid @enderror" id="exampleFirstName" placeholder="Name"
required value="{{ old('name') }}">
@error('name')
<div class="invalid-feedback">
{{ $message }}
</div>

```



```

        @enderror
    </div>
    <div class="form-group">
        <input type="text" name="username" class="form-control form-
control-user @error('username') is-invalid @enderror" id="exampleFirstName"
placeholder="Username" required value="{{ old('username') }}">
        @error('username')
        <div class="invalid-feedback">
            {{ $message }}
        </div>
        @enderror
    </div>
    <div class="form-group">
        <input type="email" name="email" class="form-control form-control-
user @error('email') is-invalid @enderror" id="exampleInputEmail" placeholder="Email
Address" required value="{{ old('email') }}">
        @error('email')
        <div class="invalid-feedback">
            {{ $message }}
        </div>
        @enderror
    </div>
    <div class="form-group">
        <input type="password" name="password" class="form-control form-
control-user @error('password') is-invalid @enderror"
id="exampleInputPassword" placeholder="Password" required
value="{{ old('password') }}">
        @error('password')
        <div class="invalid-feedback">
            {{ $message }}
        </div>
        @enderror
    </div>
    <button class="btn btn-primary btn-user btn-block">

```

```

        Register Account
    </button>
</form>
<hr>
<div class="text-center">
    <a class="small" href="/login">Already have an account? Login!</a>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<!-- Bootstrap core JavaScript-->
<script src="vendor/jquery/jquery.min.js"></script>
<script src="vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
<!-- Core plugin JavaScript-->
<script src="vendor/jquery-easing/jquery.easing.min.js"></script>
<!-- Custom scripts for all pages-->
<script src="js/sb-admin-2.min.js"></script>
</body>

</html>

```

11. Buka file user.php pada model dan modifikasi script sebagai berikut

```

class user extends Authenticatable
{
    use HasFactory;

    protected $table = 'users';

    protected $hidden = [

        'password',

        'remember_token'
    ]
}

```

```

];

// protected $fillable = [

//     'name',

//     'username',

//     'email',

//     'password'

// ];

protected $guarded=['id'];

protected $casts = [

    'email_verified_at' => 'datetime'

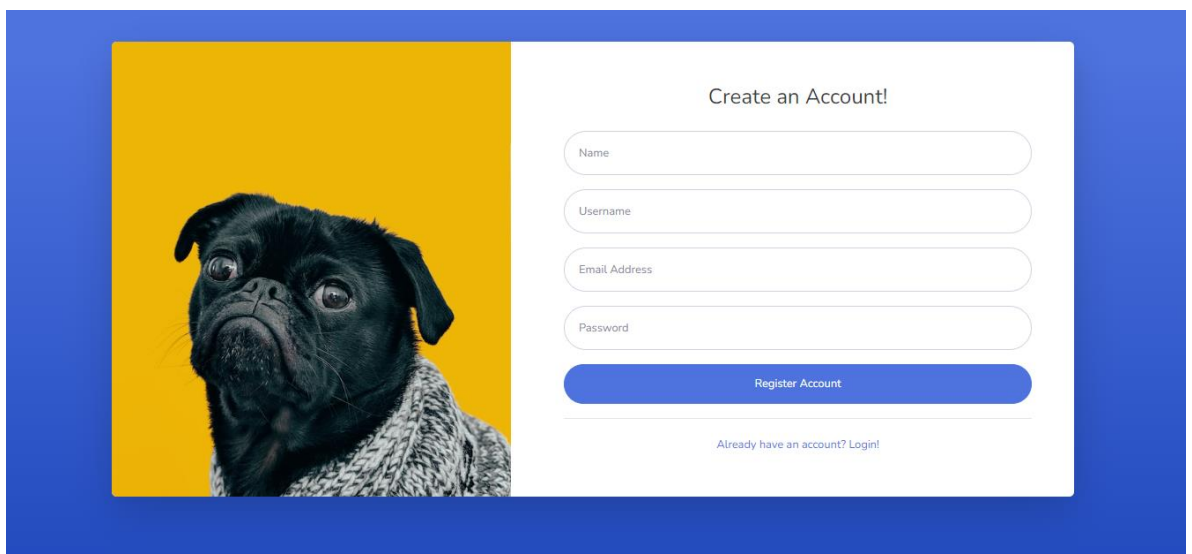
    // , 'password' => 'hashed'

];

}

```

12. Jika sudah, jalankan dan tampilannya akan sebagai berikut



13. Coba daftarkan 5 user dengan fitur create user yang telah dibuat!

14. Tambahkan perintah dibawah pada file layout/main.blade.php

```

<!-- Content Wrapper -->

<div id="content-wrapper" class="d-flex flex-column">

    <!-- Main Content -->

    <div id="content">

        <!-- Topbar -->

        <nav class="navbar navbar-expand navbar-light bg-white topbar mb-4 static-top
shadow">

            <ul class="navbar-nav ml-auto">

                @auth

                <li class="nav-item dropdown">

                    <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-
toggle="dropdown" aria-expanded="false">

                        Welcome Back, {{ auth()->user()->name}}

                    </a>

                    <ul class="dropdown-menu">

                        <li><a class="dropdown-item" href="/dashboard">My Dashboard</a></li>

                        <li><hr class="dropdown-divider"></li>

                        <li>

                            <form action="{{route('logout')}}" action="GET">

                                @csrf

                                <button type="submit" class="dropdown-item">

                                    <i class="bi bi-box-arrow-right"></i> Logout</a></li>

                            </form>

                        </li>

                    </ul>

                </li>

```

```

    @else

        <li>

            <a href="/login" class="nav-link">

                <i class="bi bi-box-arrow-in-right"></i> Login</a>

            </li>

        @endauth

    </ul>

</nav>

<div class="container mt-4">

    @yield('container')

</div>

</div>

</div>

```

15. Buat folder dan file dengan nama dashboard/index.blade.php didalam folder view dengan isi

```
@extends('layout.main')
```

```
@section('container')
```

```
<span align="center" ><h2>Halaman Dashboard</h2></span>
```

```

    <form class="d-none d-sm-inline-block form-inline mr-auto ml-md-3 my-2 my-md-0 mw-100
    navbar-search">

```

```
</form>
```

```
@endsection
```

16. Buka models Users.php dan modifikasi file menjadi dibawah ini

```
class user extends Authenticatable
```

```
{
```

```

use HasFactory;

protected $table = 'users';

protected $hidden = [

    'password',

    'remember_token'

];

// protected $fillable = [

//     'name',

//     'username',

//     'email',

//     'password'

// ];

protected $guarded=['id'];

protected $casts = [

    'email_verified_at' => 'datetime'

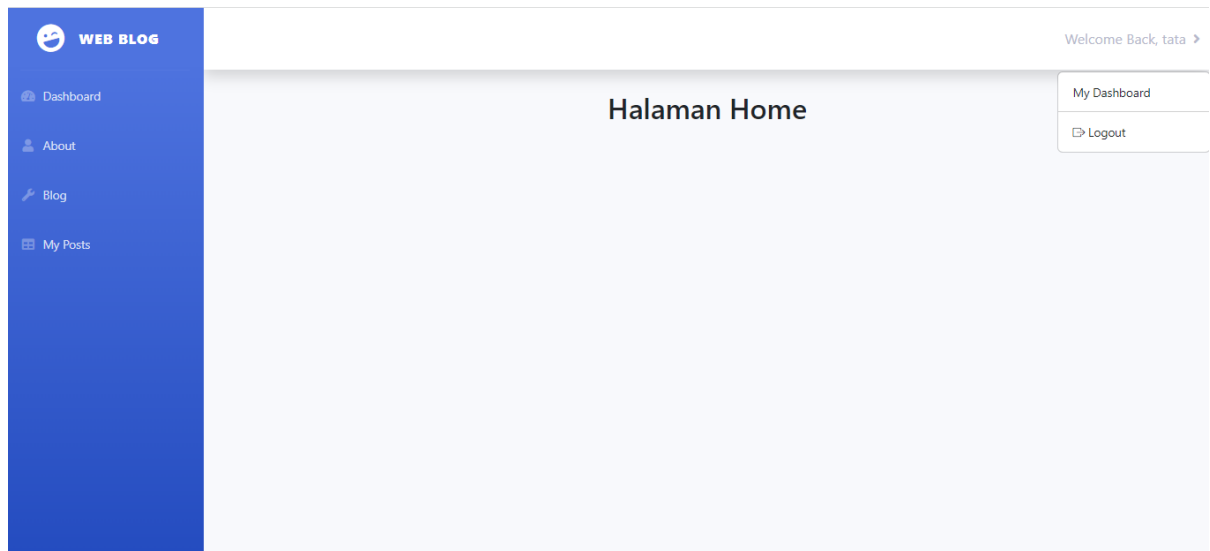
    // ,'password' => 'hashed'

];

}

```

17. Jika sudah, maka akan muncul tampilan seperti dibawah Ketika sudah berhasil login



18. Buka file web.php lalu modifikasi semua route menjadi didalam perintah

“Route::group(['middleware' => ['auth']],function () { //seluruh route });”

```
Route::group(['middleware' => ['auth']],function () {

    Route::get('/', function () {
        return view('about',[
            'title'=>"Home",
            'active'=>'home'
        ]);
    });

    Route::get('/posts', [PostController::class,'index']);
    Route::get('/posts/{post:slug}',
    [PostController::class, 'show']);

    Route::get('categories', function(){
        return view('categories',[
            'title'=>'Post Categories',
            'posts'=>Category::all()
        ]);
    });

    Route::get('categories/{category:slug}', function(Category $category){
        return view('category',[
            'title'=>$category->name,
            'posts'=>$category->posts,
            'category'=>$category->name
        ]);
    });

    Route::get('/logout', [LoginController::class,'logout']->name('logout'));
    Route::get('/dashboard',function () {
        return view('dashboard.index');
    });

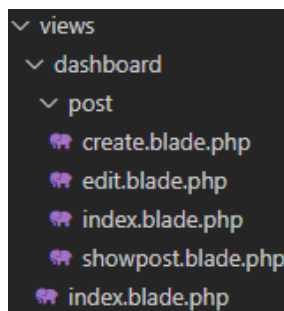
});
```

## MEMBUAT FITUR INSERT DAN TAMPIL DATA POSTINGAN

1. Tambahkan field user\_id pada tabel posts dibawah category\_id, lalu isi dengan id yang tersedia pada table users.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>id</b> 🔑	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	<b>category_id</b>	bigint(20)		UNSIGNED	No	None		
3	<b>user_id</b>	varchar(10)	utf8mb4_unicode_ci		No	None		
4	<b>title</b>	varchar(255)	utf8mb4_unicode_ci		No	None		
5	<b>slug</b>	varchar(255)	utf8mb4_unicode_ci		No	None		
6	<b>excerpt</b>	text	utf8mb4_unicode_ci		No	None		
7	<b>body</b>	text	utf8mb4_unicode_ci		No	None		
8	<b>published_at</b>	timestamp			Yes	NULL		
9	<b>created_at</b>	timestamp			Yes	NULL		
10	<b>updated_at</b>	timestamp			Yes	NULL		

2. Buat folder post didalam folder view/dashboard



3. Buat file dengan nama showpost.blade.php dengan isi sebagai berikut

```
@extends('layout.main')

@section('container')

<div class="card">

    <div class="card-header">

    </div>

    <div class="card-body">

        <h5 class="card-title"> {{ $post->title; }}</h5>

        <a href="/dashboard/post" class="btn btn-success"><span data-feather="arrow-left"></span> Back to all posts</a>

        <a href="/dashboard/post/{{ $post->slug }}/edit" class="btn btn-warning"><span data-feather="edit"></span> Edit</a>

    </div>

</div>
```



```

<form action="/dashboard/post/{{ $post->slug }}" method="post" class="d-inline">

    @method('delete')

    @csrf

    <button class="btn btn-danger border-0" onclick="return confirm('Are you sure?')"><span
data-feather="x-circle"></span> Delete</button>

</form>

<!-- <?=$post["body"]; ?><br><br> -->

<br>

{!! $post->body !!}

<br>

</div>

</div>

<br>

@endsection

```

4. Buat file dengan nama create.blade.php dengan isi sebagai berikut

```

@extends('layout.main')

@section('container')

<span align="center" ><h2>Create New Post</h2></span>

<div class="col-lg-8">

<form method="post" action="/dashboard/post">

    @csrf

<div class="mb-3">

    <label for="title" class="form-label">Title</label>

    <input type="text" class="form-control @error('title') is-invalid @enderror" id="title"
name="title" required autofocus value="{{ old('title') }}">


```

```

        @error('title')

        <div class="invalid-feedback">

            {{$message}}

        </div>

    @enderror

</div>

<div class="mb-3">

    <label for="slug" class="form-label">Slug</label>

    <input type="text" class="form-control @error('slug') is-invalid @enderror" id="slug"
    name="slug" required value="{{ old('slug') }}">

    @error('slug')

    <div class="invalid-feedback">

        {{$message}}

    </div>

    @enderror

</div>

<div class="mb-3">

    <label for="category" class="form-label">Category</label>

    <select name="category_id" id="category_id" class="form-select">

        @foreach($categories as $category)

            <option value="{{ $category->id }}">{{ $category->name }}</option>

        @endforeach

    </select>

</div>

<div class="mb-3">

```

```

<label for="body" class="form-label">Body</label>

<textarea name="body" id="body" class="form-control" cols="10" rows="5" value="{{
old('body')}}"></textarea>

@error('body')

<p class="text-danger">{{$message}}</p>

@enderror

</div>

<button type="submit" class="btn btn-primary">Create Post</button>

</form>

</div>

<script>

const title = document.querySelector('#title');

const slug = document.querySelector('#slug');

title.addEventListener('change',function(){

    fetch('/dashboard/post/checkSlug?title=' + title.value)

    .then(response=>response.json())

    .then(data=>slug.value=data.slug)

});

</script>

@endsection

```

5. Buat file dengan nama index.blade.php dengan isi sebagai berikut

```

@extends('layout.main')

@section('container')

<h2>My Posts</h2>

@if(session()->has('success'))

```

```

<div class="alert alert-success" role="alert">

    {{session('success')}}

</div>

@endif

<div class="table-responsive col-lg-8">

<a href="/dashboard/post/create" class="btn btn-primary mb-3">Create new post</a>

<table class="table table-striped table-hover">

<thead>

<tr>

<th scope="col">#</th>

<th scope="col">Title</th>

<th scope="col">Category</th>

<th scope="col">Action</th>

</tr>

</thead>

<tbody>

@foreach ($posts as $post)

<tr>

<th scope="row">{{ $loop->iteration }}</th>

<td>{{ $post->title }}</td>

<td>{{ $post->category->name }}</td>

<td>

<a href="/dashboard/post/{{ $post->slug }}" class="badge bg-info"><span data-
feather="eye"></span></a>


```

```

        <a href="/dashboard/post/{{ $post->slug }}/edit" class="badge bg-warning"><span data-
feather="edit"></span></a>

        <form action="/dashboard/post/{{ $post->slug }}" method="post" class="d-inline">

            @method('delete')

            @csrf

            <button class="badge bg-danger border-0" onclick="return confirm('Are you sure?')"><span
data-feather="x-circle"></span></button>

        </form>

    </td>

</tr>

@endforeach

</tbody>

</table>

</div>

@endsection

```

6. Buka file web.php, lalu tambahkan route didalam group auth sebagai berikut

```
Route::get('/dashboard/post/checkSlug', [TabPostController::class, 'checkSlug']);
```

```
Route::resource('/dashboard/post', TabPostController::class);
```

7. Tambahkan perintah pada bagian atas web.php dengan

```
use App\Http\Controllers\TabPostController;
```

8. Buat controller dengan nama TabPostController.php dengan tipe “Resource” tanpa harus membuat model. Lalu isi file pada bagian public class index nya sebagai berikut

```

public function index()

{

    return view('dashboard.post.index',[

        'posts'=> Post::where('user_id', auth()->user()->id)->get()
    ])
}

```

```

    ));

}

```

9. Tambahkan public class create nya menjadi seperti ini

```

public function create()

{

    return view('dashboard.post.create',[

        'categories'=>Category::all()

    ]);

}

```

10. Tambahkan public class store nya menjadi seperti ini

```

public function store(Request $request)

{

    $validatedData=$request->validate([

        'title'=>'required|max:255',

        'slug'=>'required|unique:posts',

        'category_id'=>'required',

        'body'=>'required'

    ]);

    $validatedData['user_id']=auth()->user()->id;

    $validatedData['excerpt']=Str::limit(strip_tags($request->body), 200);

    Post::create($validatedData);

    return redirect('/dashboard/post')->with('success','New Post has been added!');

}

```

11. Tambahkan public function view nya menjadi seperti ini

```

public function show(Post $post)

```

```

{

    return view('dashboard.post.showpost',[

        'post'=>$post

    ]);

}

```

12. Tambahkan public function checkSlug pada bagian bawah dengan script seperti ini

```

public function checkSlug(Request $request)

{

    $slug = SlugService::createSlug(Post::class,'slug',$request->title);

    return response()->json(['slug'=>$slug]);

}

```

13. Tambahkan perintah dibawah pada bagian atas TabPostController

```

use App\Models\Post;

use App\Models\Category;

use \Cviebrock\EloquentSluggable\Services\SlugService;

use Illuminate\Support\Str;

```

14. Buka file model Post.php dan tambahkan script dibawah ini

```

public function getRouteKeyName(){

    return 'slug';

}

public function sluggable(): array{

    return[

        'slug'=>[

            'source'=>'title'

        ]

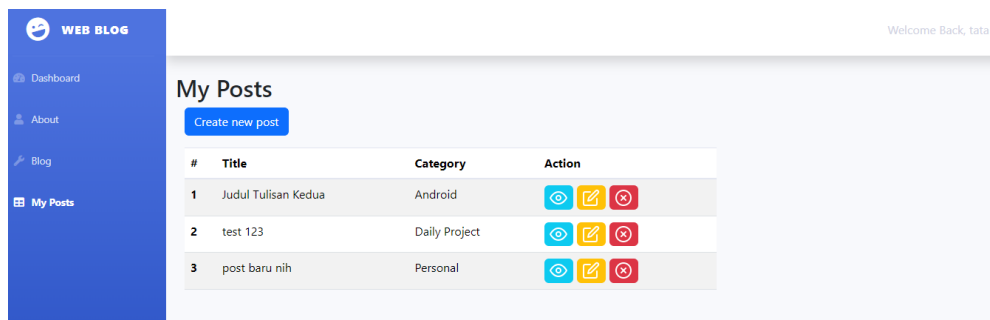
    ]
}

```

```
};

}
```

15. Tambahkan script “use Cviebrock\EloquentSluggable\Sluggable;” pada bagian atas Post.php
16. Jalankan perintah “composer require cviebrock/eloquent-sluggable” hingga berhasil
17. Tambahkan perintah “, Sluggable” disamping HasFactory pada model post.php.
18. Jika sudah, coba jalankan kode program sehingga muncul tampilan data postingan sebagai berikut



#### Task 4:

1. Buatlah fitur tampil dan insert pada tabel Comments

## MEMBUAT FITUR EDIT DAN DELETE DATA PADA POSTINGAN

1. Pada folder dashboard/post buat file dengan nama edit.blade.php dengan isi sebagai berikut

```
@extends('layout.main')
```

```
@section('container')
```

```
<span align="center" ><h2>Edit Post</h2></span>
```

```
<div class="col-lg-8">
```

```
<form method="post" action="/dashboard/post/{{ $post->slug }}">
```

```
    @method('put')
```

```
    @csrf
```

```
<div class="mb-3">
```



```

<label for="title" class="form-label">Title</label>

<input type="text" class="form-control @error('title') is-invalid @enderror" id="title"
name="title" required autofocus value="{{ old('title', $post->title)}}">

    @error('title')

    <div class="invalid-feedback">

        {{$message}}

    </div>

    @enderror

</div>

<div class="mb-3">

    <label for="slug" class="form-label">Slug</label>

    <input type="text" class="form-control @error('slug') is-invalid @enderror" id="slug"
name="slug" required value="{{ old('slug', $post->slug)}}">

    @error('slug')

    <div class="invalid-feedback">

        {{$message}}

    </div>

    @enderror

</div>

<div class="mb-3">

    <label for="category" class="form-label">Category</label>

    <select name="category_id" id="category_id" class="form-select">

        @foreach($categories as $category)

            @if(old('category_id', $post->category_id)==$category->id)

                <option value="{{ $category->id }}" selected>{{$category->name}}</option>

```

```

    @else

    <option value="{{ $category->id }}">{{ $category->name }}</option>

    @endif

    @endforeach

</select>

</div>

<div class="mb-3">

    <label for="body" class="form-label">Body</label>

    <textarea name="body" id="body" class="form-control" cols="10" rows="5">{{ old('body',
    $post->body) }}</textarea>

    @error('body')

    <p class="text-danger">{{ $message }}</p>

    @enderror

</div>

<button type="submit" class="btn btn-primary">Update Post</button>

</form>

</div>

<script>

const title = document.querySelector('#title');

const slug = document.querySelector('#slug');

title.addEventListener('change',function(){

    fetch('/dashboard/post/checkSlug?title=' + title.value)

```

```

        .then(response=>response.json())

        .then(data=>slug.value=data.slug)

    });

</script>

```

@endsection

2. Buka file TabPostController.php lalu tambahkan script pada public function edit

```

public function edit(Post $post)

{

    return view('dashboard.post.edit',[

        'post'=>$post,

        'categories'=>Category::all()

    ]);

}

```

3. Tambahkan script pada public function update

```

public function update(Request $request, Post $post)

{

    $rules=[

        'title'=>'required|max:255',

        'category_id'=>'required',

        'body'=>'required'

    ];

    if($request->slug != $post->slug){

        $rules['slug'] = 'required|unique:posts';
    }
}

```

```

    }

    $validatedData = $request->validate($rules);

    $validatedData['user_id']=auth()->user()->id;

    $validatedData['excerpt']=Str::limit(strip_tags($request->body), 200);

    Post::where('id',$post->id)

    ->update($validatedData);

    return redirect('/dashboard/post')->with('success','Post has been updated!');
}

```

#### 4. Tambahkan script pada public function destroy


```

public function destroy(Post $post)
{
    Post::destroy($post->id);

    return redirect('/dashboard/post')->with('success','Post has been deleted!');
}

```

#### 5. Jika sudah, silahkan jalankan kode program dan coba fitur edit dan hapusnya

 WEB BLOG

Dashboard

About

Blog

**My Posts**

Welcome Back, tata

## Edit Post

Title

Judul Tulisan Kedua

Slug

judul-tulisan-kedua

Category

Android

Body

Halooo ini adalah teks artikel tulisan kedua. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt

Update Post

## Task 5:

1. Tambahkan fitur update dan delete pada tabel Comments