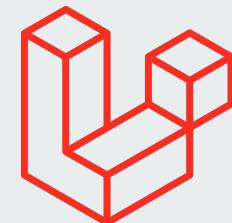
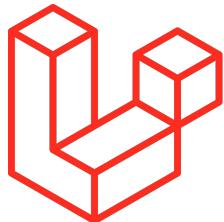

Dokumentasi Laravel

M Ikhsan Adriansyah



Tech Stack



tailwindcss



sweetalert2

NGINX



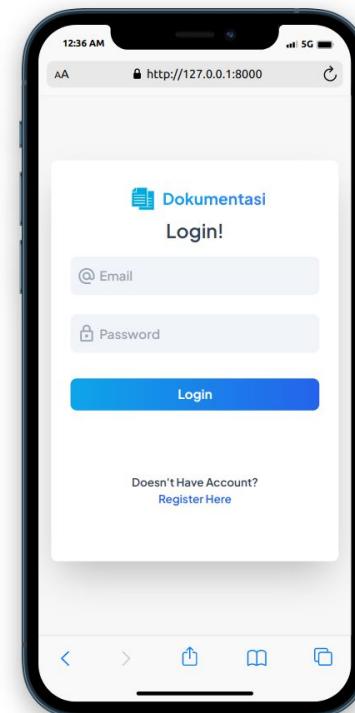
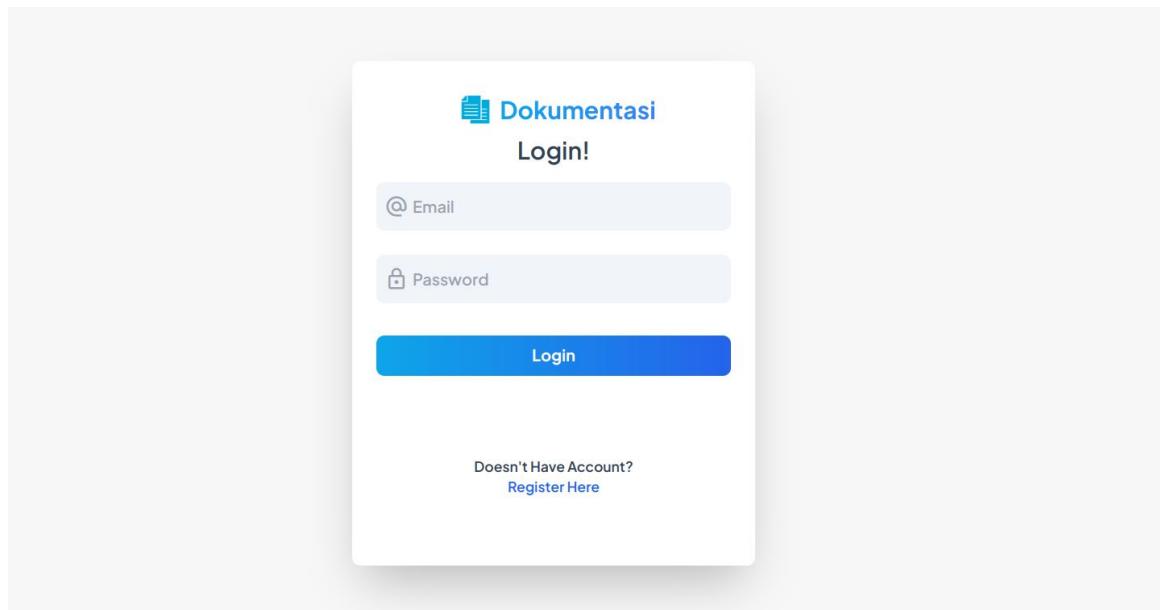
Chart.js

jQuery
write less, do more.



CKEditor 4

User Auth UI



Ui Dan UX

Dokumentasi Search CTRL K

Category Topic Article About

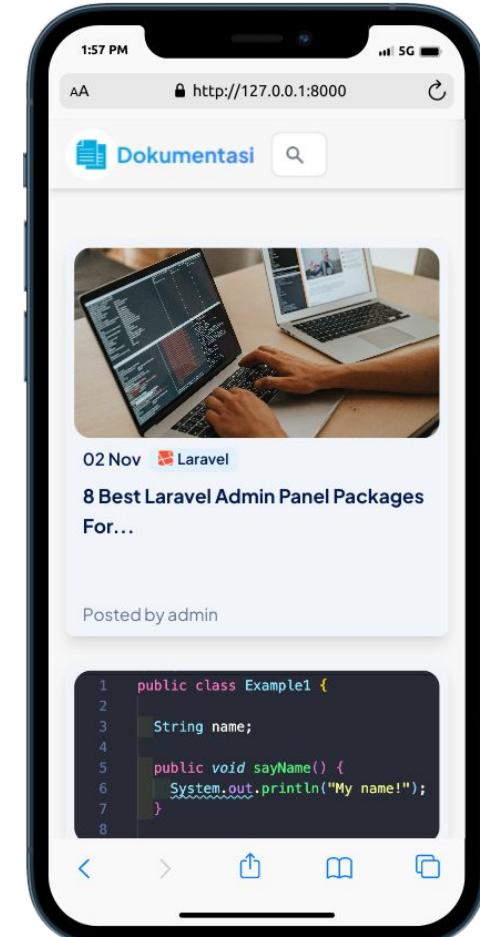
Dashboard Kanban Pro
Inbox 3 Users Products Sign In Sign Up Upgrade to Pro Documentation Components Help

 02 Nov Laravel 8 Best Laravel Admin Panel Packages For... Posted by admin

 02 Nov Java Some Java Object Oriented Programming Co... Posted by Raya O'Brien

 02 Nov OpenCV Python Building a Real-Time Object Recognition... Posted by Ali Wafa

 02 Nov MQTT IOT Real Time Data Transfer for IoT with MQT... Posted by admin



Ui Dan UX

The screenshot shows a web-based documentation platform. At the top, there's a navigation bar with links for Category, Topic, Article, About, and a search bar. Below the header, a sidebar on the left contains links for Dashboard, Kanban, Inbox, Users, Products, Sign In, Sign Up, Upgrade to Pro, Documentation, Components, and Help. The main content area features a post by Raya O'Brien published on Nov 02, 2022. The title of the post is "Some Java Object Oriented Programming Concepts". Below the title is a code editor window showing the following Java code:

```
Example1.java > ...
public class Example1 {
    String name;
    public void sayName() {
        System.out.println("My name!");
    }
}
```

At the bottom of the post, there's a note: "Object Oriented Programming (OOP) might sound intimidating at first, especially if meaningless 15 letter words are just being thrown at you, but the goal of this post is to clear up some of the more common OOP concepts." Another note at the very bottom states: "Objects are created with a Class and that class has variables and methods that say what that object can".



Dasboard UI

Dokumentasi

Category Topic Article About

Statistik Kanban Pro

Tambah Artikel Admin Settings

Products Image Uploader

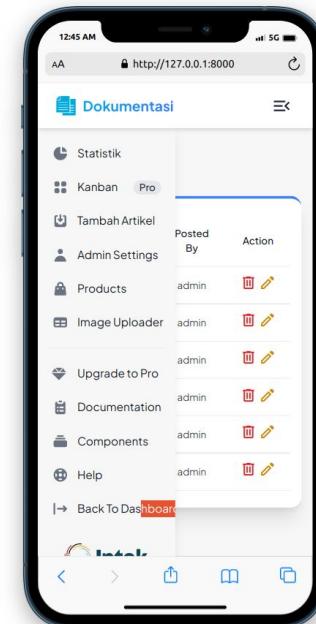
Upgrade to Pro Documentation

Components Help

Back To Dashboard

Dashboard

ID	Title	Content	Category	Date	Posted By	Action
1	dsadsads	<p>dsadsadsadsa...		2022-10-27 13:10:09	admin	
2	dsadsads	<p>dsadsadsadsa...		2022-10-27 13:10:15	admin	
3	dsadsads	<p>dsadsadsadsa...		2022-10-27 13:11:12	admin	
4	dsadsads	<p>dsadsadsadsa...		2022-10-27 13:11:19	admin	
5	SQLSTATE[23000]...	<p>dassasdgears...		2022-10-27 13:30:52	admin	
6	How To Install...	<hl><span style...		2022-10-28 00:21:12	admin	



Dashboard Ui

 Dokumentasi

Category Topic Article About ↗

Statistik Kanban Pro
Tambah Artikel Admin Settings
Products Image Uploader
Upgrade to Pro Documentation Components Help
Back To Dashboard

Dashboard

Add New Article

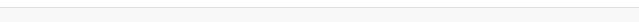
Masukan Title Article

Select Category

Upload Thumbnail

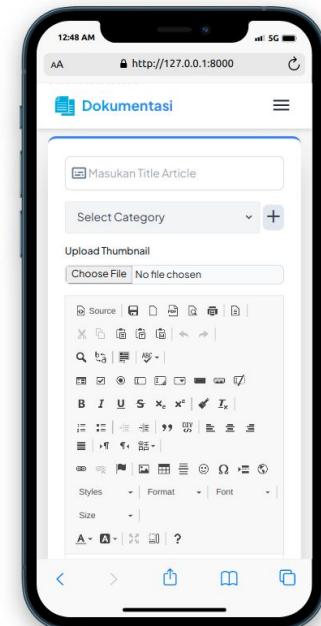
Choose File No file chosen

Source 

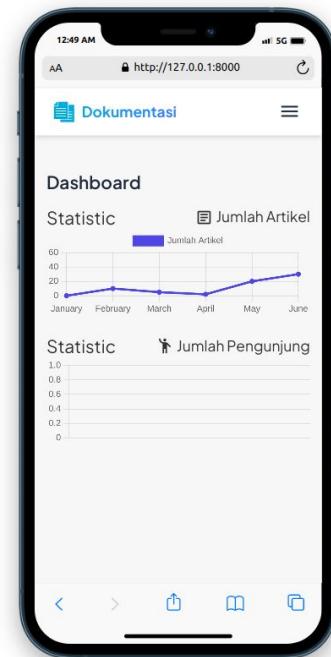
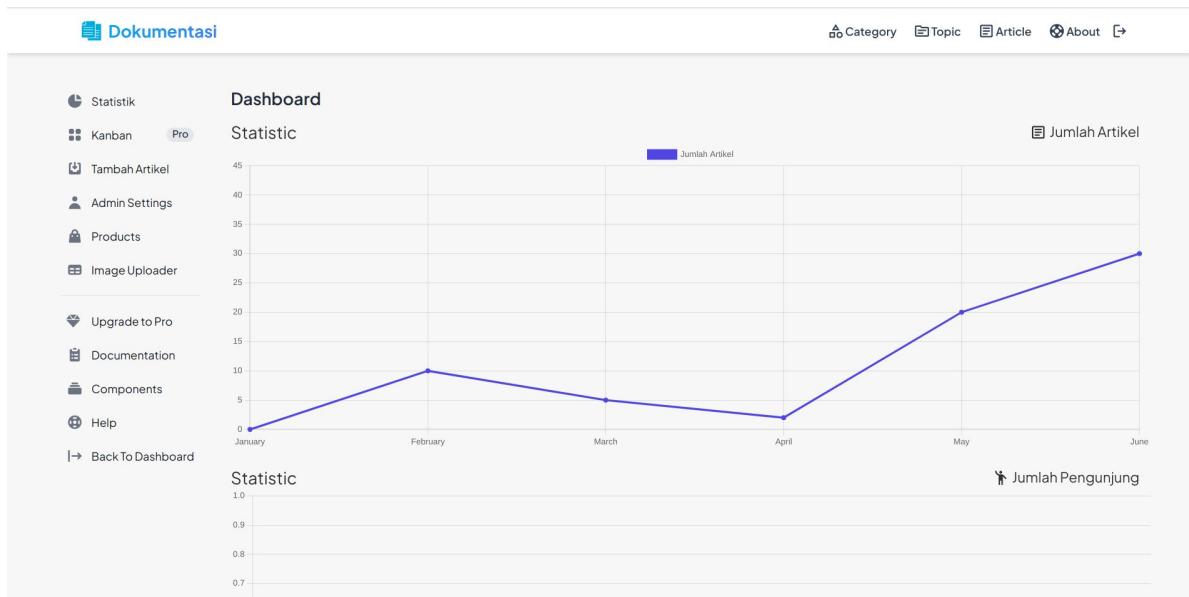
B I U S           

Styles Format Font Size Align List Table Symbol

Create



Dashboard UI



Dashboard UI

 Dokumentasi

Category Topic Article About

Dashboard

Image Uploader

ID	Title	URL	Created_at	Size	Action
1	title image	url	created_at	size	

Upload Your Image Here

Title
Example : Image For Article #7!

Attach Document

Drag and drop files here or select a file from your computer

File type: doc,pdf,types of images

Upload

Statistik

Kanban

Tambah Artikel

Admin Settings

Products

Image Uploader

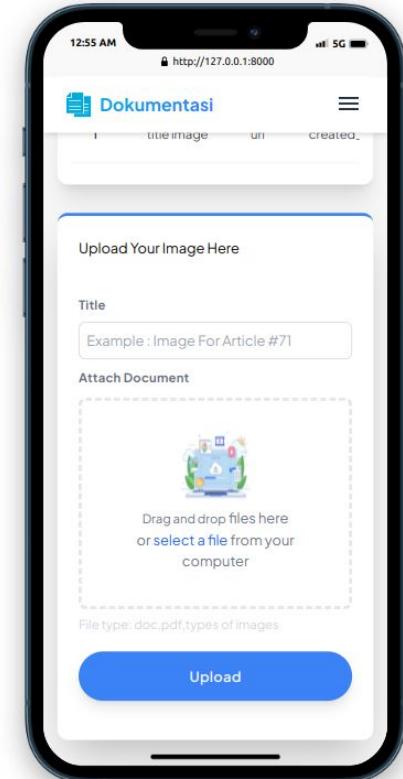
Upgrade to Pro

Documentation

Components

Help

Back To Dashboard

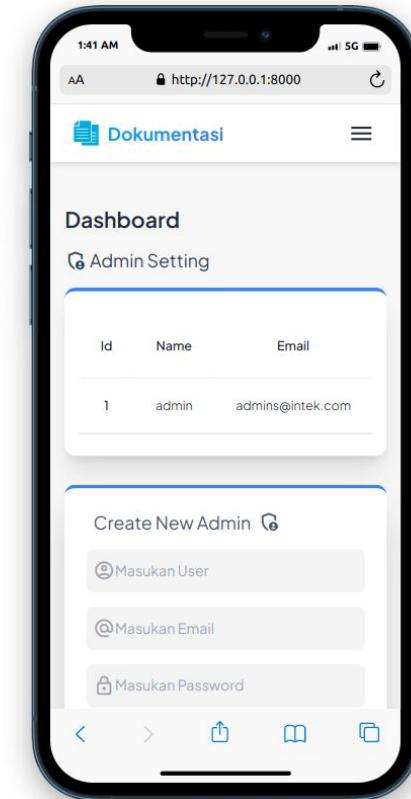


Dashboard UI

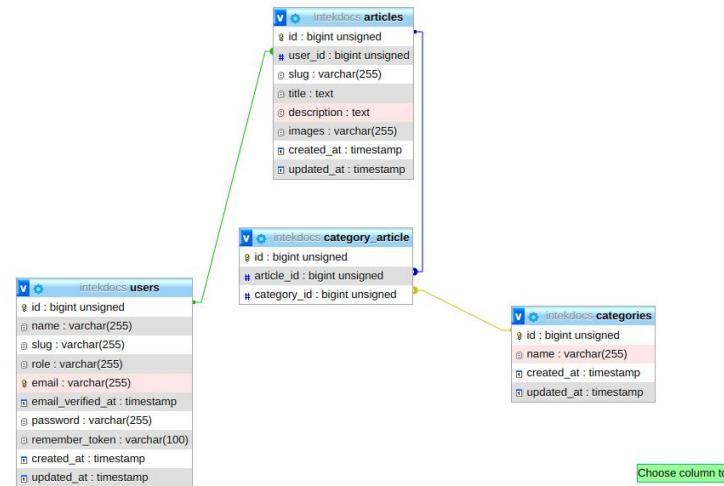
The screenshot shows a desktop application interface for 'Dokumentasi'. On the left is a sidebar with various navigation items: Statistik, Kanban (Pro), Tambah Artikel, Admin Settings, Products, Image Uploader, Upgrade to Pro, Documentation, Components, Help, and Back To Dashboard. The main content area has a header with links for Category, Topic, Article, About, and a search bar. Below the header is a section titled 'Dashboard' with a sub-section 'Admin Setting'. It displays a table with one row:

ID	Name	Email	Date Of Join	Article Posted	Action
1	admin	admins@intek.com	2022-10-27 13:08:32	6	

Below this is a form titled 'Create New Admin' with fields for User, Email, Password, and Konfirmasi Password, followed by a 'Create' button.



Database Design (Masih Development)



Routes

Route Berfungsi untuk
mengatur rute dari url akan
diarahkan ke page atau logic
backend mana ,
melalui method pada
controller

```
Web.php

1 <?php
2
3 use Illuminate\Support\Facades\Route;
4 use App\Http\Controllers\IndexController;
5 use App\Http\Controllers\RedirectHandlesController;
6 use App\Http\Controllers\AdminAuthController;
7 use App\Http\Controllers\ArticleController;
8 use App\Http\Controllers\UserAuthController;
9 /*
10 | -----
11 | Web Routes
12 | -----
13 |
14 | Here is where you can register web routes for your application. These
15 | routes are loaded by the RouteServiceProvider within a group which
16 | contains the "web" middleware group. Now create something great!
17 |
18 */
19
20 Route::get('/', [IndexController::class, 'index']);
21 Route::post('/', [IndexController::class, 'index']);
22 Route::get('loginadmin', [AdminAuthController::class, 'index']); //login admin
23 Route::get('/article/{slug}', [IndexController::class, 'show']);
24 Route::get('dashboard', [AdminAuthController::class, 'dashboard']);
25 Route::get('login', [UserAuthController::class, 'index'])->name('userlogin'); //login user
26 Route::post('login', [UserAuthController::class, 'auth'])->name('userloginpost'); //login user
27 Route::get('register', [UserAuthController::class, 'registerindex'])->name('registerindex');
28 Route::post('register', [UserAuthController::class, 'register'])->name('registerpost');
29 Route::post('loginadmin', [AdminAuthController::class, 'auth'])->name('authpost'); //login admin
30 Route::get('logout', [AdminAuthController::class, 'signout'])->name('signout');
31 Route::get('/article/{id}/delete', [ArticleController::class, 'destroy']);
32 Route::get('/article/{id}/edit', [ArticleController::class, 'edit']);
33 Route::put('/article/{id}/update', [ArticleController::class, 'update'])->name('update');
34 Route::get('dashboard/tambahartikel', [ArticleController::class, 'index'])->name('addarticle');
35 Route::post('dashboard/tambahartikel', [ArticleController::class, 'create'])->name('addarticlepost');
36 Route::get('dashboard/statistik', [AdminAuthController::class, 'statistic'])->name('statisticindex');
37 Route::get('dashboard/admin', [AdminAuthController::class, 'setting'])->name('adminsetting');
38 Route::post('dashboard/admin', [AdminAuthController::class, 'createnewadmin'])->name('createadmin');
39 Route::get('dashboard/admin/{user:slug}/details', [AdminAuthController::class, 'admindetails'])->name('admindetails');
40 Route::get('dashboard/imageuploader', [AdminAuthController::class, 'imageuploaderview'])->name('imageuploader');
41 Route::post('dashboard/imageuploader', [AdminAuthController::class, 'imageuploader'])->name('imageuploaderpost');
42 Route::get('404', [RedirectHandlesController::class, 'index']);
```

Javascript File

Untuk Confirm Password Validation Pada Register dengan cara membandingan value antara 2 input dan mengeluarkan output tertentu apabila password dan confirm password tidak sama

```
$document).ready(function () {  
    $("#submits").attr("disabled",true);  
    $("#passwords").on('keyup',function(){  
        let valuepassword = $(this).val()  
        $("#confirmpasswords").on('keyup',function(){  
            let confirmvalue = $(this).val()  
  
            if(valuepassword != confirmvalue){  
                $("#submits").attr("disabled",true);  
                $("#message").show()  
                $("#message").text("Password Tidak Cocok!")  
            } else if ( valuepassword == "" && confirmvalue == "" ) {  
                $("#message").hide()  
                $("#submits").attr("disabled",true);  
            } else {  
                $("#submits").removeAttr("disabled");  
                $("#message").empty()  
            }  
  
       });  
    });  
  
});
```

Javascript File

Untuk Hamburger Menu Pada Mobile

Apabila Ada perubahan pada checkbox
tambahkan atau hapus class

```
$('document').ready(function () {
    $('#hamburgercheckbox').change(function(){
        if($('#this').is(':checked')){
            let hamburgericon = $('#hamburgericon')
            if(hamburgericon.find('menu')){
                $('#hamburgericon').empty();
                $('#hamburgericon').html("menu_open");
                $(".mobile-dashboard-nav").addClass("slide");
            }
        } else {
            $(".mobile-dashboard-nav").removeClass("slide");
            $('#hamburgericon').empty();
            $('#hamburgericon').html("menu");
        }
    });
});
```

Javascript File

Upload Image Logic

```
$(document).ready(function () {
    let fileInput = document.querySelector('#input');
    let dropArea = document.querySelector('#dropImageHere');
    let thumbnail = document.querySelector('#dropImageThumbnail');
    let borderDrop = document.querySelector("#borderdropImage");
    let file
    $("#dropImageHere").on('dragover',function(event){
        event.preventDefault();
        $("#borderdropImage").removeClass("border-dashed");
    })
    $("#dropImageHere").on('dragleave',function(event){
        event.preventDefault();
        $("#borderdropImage").addClass(".border-dashed");
    })
    fileInput.addEventListener("change",(event) => {
        event.preventDefault();
        if(!borderDrop.classList.contains("border-dashed")){
            borderDrop.classList.add("border-solid")
            borderDrop.classList.remove("border-dashed");
        } else {
            borderDrop.classList.remove("border-dashed");
            borderDrop.classList.add("border-solid")
        }
        file = event.dataTransfer.files[0];
        showFile();
    })
}

dropArea.addEventListener("drop",(event) => {
    event.preventDefault();
    if(!dropArea.classList.contains("border-dashed")){
        borderDrop.classList.add("border-solid")
        borderDrop.classList.remove("border-dashed");
    } else {
        borderDrop.classList.remove("border-dashed");
        borderDrop.classList.add("border-solid")
    }
    file = event.dataTransfer.files[0];
    showFile();
})

function showFile(){
    let fileType = file.type;
    let validExtensions = ["image/jpeg","image/jpg","image/png"];
    if(validExtensions.includes(fileType)){
        let fileReader = new FileReader();
        fileReader.onload = ()=> {
            let fileURL = fileReader.result;
            let imgTag = ``;
            thumbnail.innerHTML = imgTag;
        }
        fileReader.readAsDataURL(file);
    } else {
        file = ""
        Swal.fire({
            icon: 'error',
            title: 'Oops...',
            text: 'File Bukan Gambar',
        })
        $("#borderdropImage").addClass("border-dashed");
    }
}
});
```

Javascript File

Copy To ClipBoard Logic

Mencari Semua Id dari class tablerows

dan melakukan “for of” untuk mendapat id satu per satu lalu dicocokan

```
const tablerows = document.getElementsByClassName('table-rows');
for (const tbrow of tablerows) {
    $('.copy-button').on('click', function(e) {
        let singletbrow = tbrow.querySelector('#image-id')
        let tbrows = tbrow.querySelector('#image-id').innerText
        if($(this).attr("id") == tbrows){

            navigator.clipboard.writeText(singletbrow.parentNode.querySelector('#image-url').getAttribute('title'))
                tippy('.copy-button', {
                    content: 'Copied!',
                    trigger:'click'
                });

        }
    });
}
```

Javascript File

Toggle Dark Javascript

jika button setLight di click

dan setLight Mempunyai text dark_mode

kosongkan class dan tambah class baru

atau sebaliknya

```
$document).ready(function () {
  $('#setlight').on('click',function(){
    if($('#setlight').text().includes('dark_mode')){
      $('#setlight').empty();
      $('#setlight').text('light_mode');
      $('.toggle').removeClass('text-indigo-500')
      $('.toggle').addClass('text-red-500')
    } else {
      $('#setlight').empty()
      $('#setlight').text('dark_mode');
      $('.toggle').removeClass('text-red-500')
      $('.toggle').addClass('text-indigo-500')
    }
  })
})
```

Javascript File

Add New Category Di

Dashboard Tambah Article

The screenshot shows a web application interface. On the left is a sidebar with various icons and links: Dokumentasi, Statistik, Add Category, Tambah Artikel, Admin Settings, Products, Image Uploader, Upgrade to Pro, Documentation, Components, Help, and Back To Dashboard. The main area is titled "Dashboard" and contains a sub-section "Add New Article". It features a text input field labeled "Masukan Title Article", a dropdown menu labeled "Select Category", and a file upload section with a "Browse..." button and a message "No file selected. SVG, PNG, JPG or GIF (MAX. 800x400px)". Below these is a rich text editor toolbar with various styling options like bold, italic, underline, and alignment.

A terminal window is shown with a dark background and three colored window control buttons (red, yellow, green) at the top. The terminal displays a block of Javascript code:

```
$('#addnewcategory').on('click', ()=>{  
  let clone = $('#inputcategory').clone(true);  
  $('.category-input').after(clone)  
  let categoryevery = $('.category-every')  
  if(categoryevery.length > 2){  
    $("#addnewcategory").hide()  
  }  
};
```

Javascript File

Search Ajax Jquery

Jika Keyboard diketik , get value kirimkan
melalui ajax request menuju controller
dan dikembalikan lagi dalam
bentuk json data kemudian di append
ke html

```
$document).ready(function () {
    $("#default-search").on('keyup',function(){
        let searchQuest = $(this).val();
        var url = window.location.pathname
        $.ajax({
            method: 'get',
            url: url,
            dataType: 'json',
            data: {
                "searchQuest" : searchQuest,
                "_token" : $('#token').val()
            },
            success: (data) => {
                if(searchQuest == ""){
                    $('#result').html("No Recent Searches")
                } else {
                    $('#results').html(data)
                }
            }
        })
    })
})
```

Models Dan Migrations

Untuk User

Model / Eloquent pada laravel digunakan untuk **mewakili pemanggilan tabel**. Dengan demikian, model merupakan bagian yang utama dimana kita akan berinteraksi ke tabel yang bersangkutan melalui model yang dibuat.

Migration Adalah Konsep Isi Table Yang Akan dibuat

Untuk Membuat Model Sekaligus Migrations

```
php artisan make:model NamaModel -m
```

-m artinya migrations

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('user_profile')->nullable();
            $table->string('slug')->nullable();
            $table->string('role')->nullable();
            $table->string('email')->unique();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->rememberToken();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('users');
    }
};
```

```
<?php

namespace App\Models;

// use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Laravel\Sanctum\HasApiTokens;

class User extends Authenticatable
{
    use HasApiTokens, HasFactory, Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    protected $fillable = [
        'name',
        'slug',
        'user_profile',
        'role',
        'email',
        'password',
    ];

    /**
     * The attributes that should be hidden for serialization.
     *
     * @var array<int, string>
     */
    protected $hidden = [
        'password',
        'remember_token',
    ];

    /**
     * The attributes that should be cast.
     *
     * @var array<string, string>
     */
    protected $casts = [
        'email_verified_at' => 'datetime',
    ];

    public function article(){
        return $this->hasMany(Article::class);
    }

    public function getRouteKeyName()
    {
        return 'slug';
    }
}
```

Model Dan Migrations Categories

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use App\Models\Article;
use App\Models\ArticleCategory;
use Illuminate\Database\Eloquent\Model;

class Category extends Model
{
    use HasFactory;

    protected $fillable = [
        "name",
        "icon"
    ];

    public function articles(){
        return
$this->belongsToMany(Article::class,'category_article')->withPivot('category_article');
    }

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

```
...
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('categories', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('icon');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('categories');
    }
};
```

Model Dan Migrations

Untuk Artikel



```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('articles', function (Blueprint $table) {
            $table->id();
            $table->foreignId('user_id')->constrained();
            $table->string('slug');
            $table->text('title');
            $table->text('description');
            $table->string('images');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('articles');
    }
};
```



```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use App\Models\Category;
use Illuminate\Support\Str;

class Article extends Model
{
    use HasFactory;

    protected $fillable = [
        "user_id",
        "title",
        "slug",
        "category_id",
        "description",
        "images",
    ];

    public function user(){
        return $this->belongsTo(User::class);
    }

    public function categories(){
        return $this->belongsToMany(Category::class,'category_article');
    }

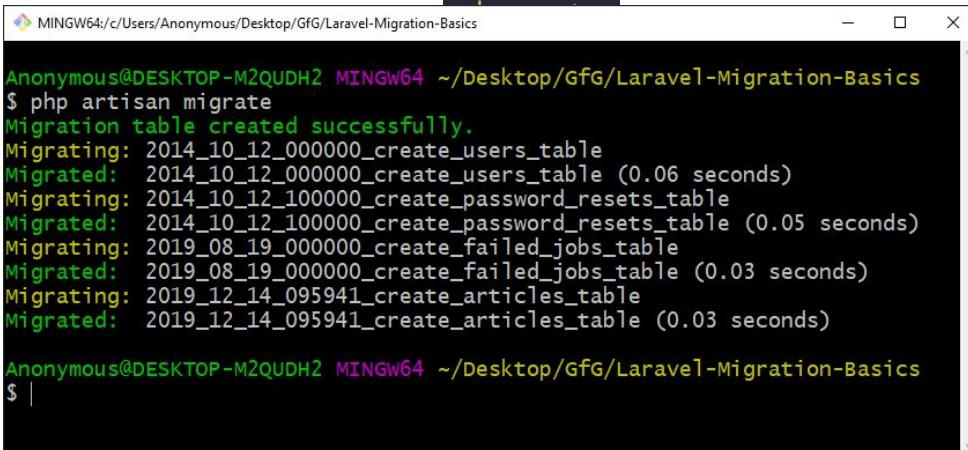
    public function getShortDescriptionAttribute()
    {
        return Str::limit(
            nl2br(strip_tags($this->description)),
            90
        );
    }
}

* Reverse the migrations.
*
* @return void
*/
public function down()
{
    Schema::dropIfExists('users');
}
```

Lakukan Migrate

Migrate bertujuan untuk memasukan rancangan table kita ke dalam database asli

Ketikan ini pada terminal `php artisan`



```
Anonymous@DESKTOP-M2QUDH2 MINGW64 ~/Desktop/GfG/Laravel-Migration-Basics
$ php artisan migrate
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table (0.06 seconds)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table (0.05 seconds)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated: 2019_08_19_000000_create_failed_jobs_table (0.03 seconds)
Migrating: 2019_12_14_095941_create_articles_table
Migrated: 2019_12_14_095941_create_articles_table (0.03 seconds)

Anonymous@DESKTOP-M2QUDH2 MINGW64 ~/Desktop/GfG/Laravel-Migration-Basics
$ |
```

Model & Migration Article_Category

Article_Category adalah sebuah pivot table
yang berguna menghubungkan antar 2 table
pada relasi many to many

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('category_article', function (Blueprint $table) {
            $table->bigIncrements('id');
            $table->unsignedBigInteger('article_id');
            $table->unsignedBigInteger('category_id');

            $table->foreign('article_id')->references('id')->onDelete('cascade');

            $table->foreign('category_id')->references('id')->on('categories')->onDelete('cascade');
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('category_article');
    }
};
```

Melakukan first seeder

Seperti artinya, Seeder (benih) ini adalah sebuah **fungsi untuk menanamkan Benih berupa Data pada Database kita**. Jadi saat nanti kita melakukan migrations , Database kita sudah terisi Data dari hasil Seeder kita

cara melakukan seeder

```
php artisan make: seeder NamaSeeder
```

buka folder /database/seeders/FirstSeeder.php

masukan script berikut ke function run() pada file FirstSeeder.php

```
$this->call([  
    NamaSeeder::class,  
]);
```

Value FirstSeeder

```
Params::create([
    "Isi Table Pertama" => "isi",
    "Isi Table Kedua" => "isi"
]);
```

Params adalah nama table yang akan diakses

create adalah method untuk table yang akan kita akses , selain create terdapat method lain seperti update,delete dll

Untuk Mengakses isi table dapat menggunakan string nama struktur table kemudian dilanjut dengan **=>**

Dan di isi dengan valuenya bisa juga menggunakan **\$request->** ataupun langsung menggunakan string dsb

Memasukan value Ke FirstSeeder

-buka file namaSeeder.php

-use terlebih dahulu model yang akan kita masukan value

contoh:`use App\Models\User;`

-Kemudian masuk pada function Run()

-masukan script berikut

```
User::create([
    "name" => "admin",
    "Slug" => Str::slug('admin'),
    "role" => "moderator",
    "email" => "admins@intek.com",
    "password" => bcrypt("admin123")
]);
```

Lakukan Migrate Fresh beserta dengan seed

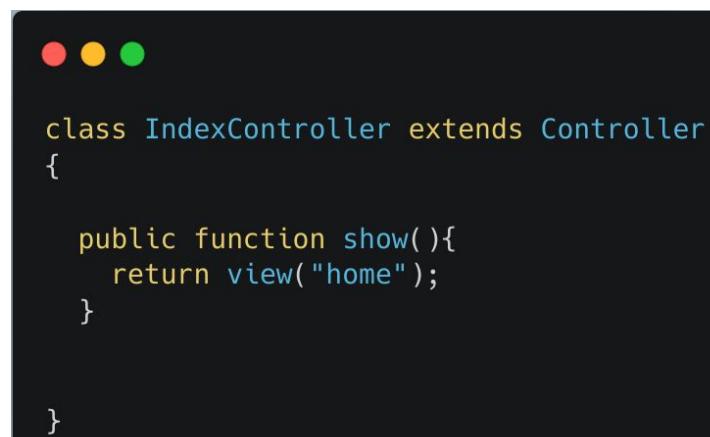
Migrate Fresh bertujuan untuk merancang kembali rancangan table kita ke dalam database asli serta memasukan seeder `php artisan migrate:fresh -seed`

Cek Kembali PhpMyAdmin , maka value dan struktur table akan terbentuk

Membuat Controller

Controller merupakan jembatan atau penghubung antara view dan model. jadi secara mudah nya, controller bisa kita pahami sebagai pengatur view dan model. controller sendiri biasanya berperan sebagai pengolah data.

Controller memiliki banyak method tergantung
berapa banyak method yang akan kita buat
method method tersebut terbungkus kedalam class
NamaController yang merupakan Inheritances dari
class controllers



```
class IndexController extends Controller {  
  public function show(){  
    return view("home");  
  }  
}
```

contoh controller

