

Laporan Praktikum Web 11

Forum Rumpi!



Disusun oleh:

105221025

Reva Ananda

Program Studi Ilmu Komputer

Universitas Pertamina

2023

Daftar Isi	
Bab 1 Pendahuluan	2
1.1 Latar Belakang	2
1.2 Tujuan	3
1.3 Batasan masalah	3
Bab 2 Metode Pelaksanaan	3
2.1 Tools dan Aplikasi Pendukung	3
2.2 Bahasa Pemograman	3
Bab 3 Hasil	3
3.1 Laravel MVC (Model-View-Controller)	3
3.1.1 Database Schema MYSQL	3
3.1.2 Models	5
3.1.3 Controllers	8
3.1.4 Views	10
3.1.5 Routes	17
3.2 Kode CSS	17
3.2.1 Stylesheet layout.css	18
3.1.2 Stylesheet login.css	21
3.2.3 Stylesheet home.css	22
3.2.4 Stylesheet createPost.css	23
3.2.5 Stylesheet Comment.css	25
3.2.6 Stylesheet profil.css	26
3.3 Kode Javascript	27
3.3.1 Script Modal Sidebar Menu Pengaturan modal.js	27
3.3.2 Script Fitur Tema darkMode.js	28
3.3.3 Script Fitur Menambahkan Gambar Dan Video postingan serta tampilan Preview addFile.js	30
Bab 4 Penutup	32
4.1 Kesimpulan	32
4.2 Referensi	34
4.3 Link github kalian	34

Bab 1 Pendahuluan

1.1 Latar Belakang

Dengan kemudahan akses untuk berkomunikasi dalam era digital, forum online menjadi salah satu wadah yang sangat populer bagi orang-orang untuk berbagi informasi, pendapat, dan pengalaman. Forum online memungkinkan pengguna untuk terlibat dalam diskusi, memperluas jaringan sosial, dan mendapatkan wawasan baru. Dengan melihat potensi ini, saya merasa tertantang untuk membuat proyek web forum yang sederhana namun bermanfaat.

1.2 Tujuan

Tujuan dari dibuatnya proyek web forum ini selain untuk melaksanakan tanggung jawab, juga dapat menambah pengalaman saya sebagai pengembang. Selain itu, proyek ini juga akan menjadi tambahan berharga dalam portofolio saya.

1.3 Batasan masalah

Web Forum yang akan saya buat sangat terbatas dalam hal fungsionalitas maupun fitur mengingat masih sedikitnya pengalaman saya sebagai pengembang web dan keterbatasan sumber daya. Perpesanan pada forum ini akan berbasis text dan mungkin juga gambar dan video sebagai pelengkap.

Bab 2 Metode Pelaksanaan

2.1 Tools dan Aplikasi Pendukung

Pada pembuatan forum ini saya menggunakan VS Code sebagai text editor, dikarenakan sifat versatile dan telah familiar dengan environmentnya, saya juga menginstall beberapa extension dan add-on untuk memudahkan proses pengembangan. Saya juga menggunakan framework Laravel versi 9, Laravel adalah framework PHP modern yang mempermudah pengembangan aplikasi web. yang dikenal dengan sintaksnya yang elegan dan readability yang tinggi, sedangkan untuk database saya menggunakan MySQL untuk menyimpan data berupa data user, pesan pos, komentar dan lainnya. Untuk aset berupa file gambar dan video saya akan menyimpannya secara pada folder project dan akan menyimpan data path file tersebut dengan kolom url dalam table mysql.

2.2 Bahasa Pemograman

Dalam pengembangan proyek web forum menggunakan Laravel, Bahasa Pemrograman utama yang digunakan adalah PHP. Eloquent ORM dan Blade Template Engine, dua komponen kunci dalam Laravel, memberikan kemudahan dalam manipulasi data dan pembuatan tampilan. Eloquent membuat operasi database menjadi lebih intuitif, sedangkan Blade menyederhanakan penulisan tampilan. Penggunaan CSS untuk mengatur tampilan dan JavaScript untuk fungsionalitas dinamis memperkaya pengalaman pengguna. Integrasi ketiganya, bersama dengan dukungan kuat dari Laravel, memungkinkan pengembang membangun aplikasi web responsif, menarik, dan efisien dengan cepat.

Bab 3 Hasil

3.1 Laravel MVC (Model-View-Controller)

MVC adalah singkatan dari Model-View-Controller, sebuah pola desain arsitektur perangkat lunak yang memisahkan aplikasi menjadi tiga komponen utama: Model, View, dan Controller. Setiap komponen memiliki tanggung jawab tertentu dalam pengembangan perangkat lunak.

3.1.1 Database Schema MYSQL

Pada Laravel, disediakan migrasi untuk memanipulasi database mysql. Migrasi pada Laravel adalah suatu konsep yang memungkinkan untuk mengelola skema database secara terstruktur dan terdokumentasi menggunakan kode PHP daripada menggunakan SQL langsung. Migrasi memungkinkan membuat, mengubah, dan menghapus tabel atau kolom dalam database Anda menggunakan sintaksis kelas dan metode yang didefinisikan oleh Laravel.

Untuk membuat migrasi baru bisa menggunakan command `php artisan make:migration [namaMigrasi]`, Untuk menjalankan migrasi maka bisa menggunakan command `php artisan migrate` dan `php artisan migration:rollback` untuk menarik migrasi terakhir.

Contoh migrasi yang digunakan untuk membuat table Posts beserta kolom, tipe data dan relasinya:

```
<?php

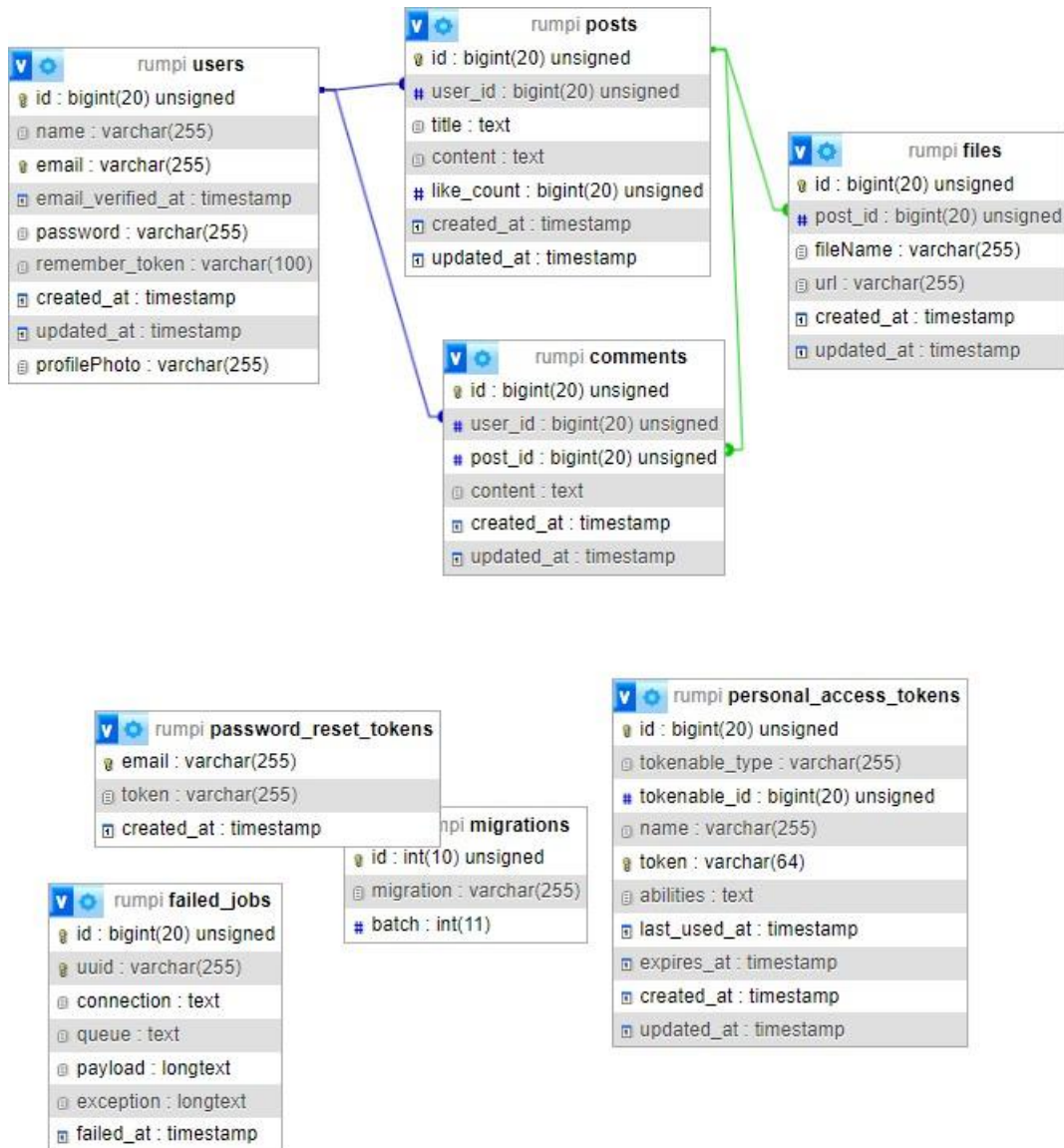
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /
    * Run the migrations.
    */
    public function up()
    {
        Schema::create('posts', function (Blueprint $table) {
            $table->id();
            $table->foreignId('user_id')->constrained('users')->onDelete('cascade');
            $table->text('content');
            $table->unsignedBigInteger('like_count')->default(0);
            $table->timestamps();

            $table->foreign('user_id')->references('id')->on('users')->onDelete('cascade');
        });
    }

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

Tampilan designer dari skema database



3.1.2 Models

Model pada MVC adalah komponen yang bertanggung jawab untuk mengelola data dan logika bisnis aplikasi. Model merepresentasikan struktur data, serta menyediakan cara untuk berinteraksi dengan dan memanipulasi data dalam database.

Berikut setiap model setiap table yang menggambarkan skema dari database .

3.1.2.1 Model User (User.php)

```
<?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',
        'email',
        'password',
        'photoProfile',
    ];

    /
    * 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',
        'password' => 'hashed',
    ];
}

```

3.1.2.2 Model Posts (Posts.php)

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Post extends Model
{
    use HasFactory;

    protected $table = 'posts';
    protected $fillable = [
        'user_id',
    ];
}

```

```

        'title',
        'content',
        'like_count'
    ];

    //Post dimiliki 1 user
    public function user()
    {
        return $this->belongsTo(User::class);
    }

    // 1 post mempunyai banyak komentar
    public function comments()
    {
        return $this->hasMany(Comment::class);
    }

    // 1 post memiliki banyak file
    public function files()
    {
        return $this->hasMany(File::class, 'id', 'post_id');
    }
}

```

3.1.2.3 Model Comments (Comment.php)

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Comment extends Model
{
    // use HasFactory;
    protected $table = 'comments';
    protected $fillable = [
        'user_id',
        'post_id',
        'content'
    ];

    // dimiliki oleh hanya 1 user
    public function user()
    {
        return $this->belongsTo(User::class);
    }

    // dimiliki oleh 1 post
    public function post()
    {

```

```

        return $this->belongsTo(Post::class);
    }
}

```

3.1.2.4 Model File (File.php)

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class File extends Model
{
    use HasFactory;

    protected $table = 'files';
    protected $fillable = [
        'post_id',
        'fileName',
        'url'
    ];

    // dimiliki oleh 1 post
    public function post()
    {
        return $this->belongsTo(Post::class);
    }
}

```

3.1.3 Controllers

Controller pada MVC adalah komponen yang bertanggung jawab untuk menerima input dari pengguna, memproses logika bisnis, dan mengirimkan output kepada pengguna. Controller berfungsi sebagai perantara antara Model (data) dan View (tampilan).

3.1.3.1 Post Controller (PostController.php)

Controller Post memiliki fungsi-fungsi CRUD pada Post, dimana terdapat fungsi untuk melihat, membuat, mengedit dan menghapus pos.

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Storage;

use App\Models\User;

```



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

class PostController extends Controller
{
    public function index()
    {
        /* User::where */

        // Ambil semua data post
        $postDatas = Post::orderBy('created_at', 'desc')->with('user.profile', 'files')->paginate(5);

        //ambil data user
        $userId = User::get('id')->where('id', auth()->user()->id)->first();
        /* dd($userId); */
        /* dd($postDatas); */
        foreach ($postDatas as $post) {
            // Akses user untuk setiap post
            dd($post->user);
        } /*

        return view('home', compact('postDatas', 'userId'));
    }

    public function showComment($id)
    {
        $post = Post::find($id);
        return view('post', compact('post'));
    }

    public function create()
    {
        return view('createPostView');
    }

    public function store(Request $request)
    {
        /* dd($request->all()); */
        $userId = $request->user()->id;

        $title = $request->title;
        $content = $request->postContent;
        $files = $request->file('files');

        $post = Post::create([
            'user_id' => $userId,
            'title' => $title,
            'content' => $content,
        ]);
    }
}
```

```

if ($files) {
    foreach ($files as $file) {
        $fileName = $file->getClientOriginalName();
        $file->move(public_path('post_images'), $fileName);

        File::create([
            'post_id' => $post->id,
            'fileName' => $fileName,
            'url' => 'post_images/' . $fileName,
        ]);
    }
}

return redirect()->route('home')->with('success', 'Pos Berhasil Dibuat');
}

public function edit($postId)
{
    $post = Post::with('files')->find($postId);
    return view('editPostView', compact('post'));
}

public function update(Request $request, $postId)
{
    $title = $request->title;
    $content = $request->content;
    $selectedFiles = $request->input('selected_files');

    //update post
    Post::where('id', $postId)->update([
        'title' => $title,
        'content' => $content,
    ]);

    if (is_array($selectedFiles)) {
        foreach ($selectedFiles as $selectedFile) {
            // Pisahkan ID dan URL dari nilai checkbox yang terpilih
            list($fileId, $fileUrl) = explode(':', $selectedFile);

            // Hapus dari penyimpanan (storage)
            Storage::delete($fileUrl);

            // Hapus di table
            File::find($fileId)->delete();
        }
    }

    return redirect()->back()->with('success', 'Pos Berhasil Diubah');
}

public function destroy($postId)

```

```
{

$post = Post::find($postId);
$post->delete();

if ($post) {

    $post = Post::find($postId);

    // cari file post
    $files = File::where('post_id', $postId)->get();

    // cek ada file gk
    if ($files->count() > 0) {

        foreach ($files as $file) {

            Storage::delete($file->url);

            $file->delete();
        }
    }
}

$idLogin = auth()->user()->id;

return redirect()->route('profile.show', ['idLogin'=>$idLogin])->with('success', 'Post
Berhasil Dihapus');
}
```

3.1.3.2 Comment Controller (CommentController.php)

```
<?php

namespace App\Http\Controllers;

use App\Models\Post;

use Illuminate\Http\Request;

class CommentController extends Controller
{
    public function showComment($id)
    {
        $post = Post::find($id);
        $postComments = $post->comments()->with('user')->orderBy('created_at', 'desc')->paginate(5);
        // dd($postComments);
        return view('commentView', compact('postComments', 'post'));
    }
}
```

3.1.3.3 Profil Controller (ProfileController.php)

Controller mempunyai fungsi-fungsi untuk menampilkan dan mengedit profile dari user.

```
<?php

namespace App\Http\Controllers;

use App\Models\User;
use App\Models\Post;
use Illuminate\Http\Request;

class ProfileController extends Controller
{
    public function showProfile()
    { /* dd($userID = auth()->user()->id); */

        $userID = auth()->user()->id;

        $dataUser = User::with('profile', 'Posts')->find($userID);
        /* dd($dataUser); */

        //Data Post user

        /* $postsUser = Post::find('user_id', $userID);
        dd($postsUser); */

        return view('profileView', compact('dataUser'));
```

```
}  
}
```

3.1.4 Views

Views pada structure MVC bertanggung jawab untuk menangani presentasi data dan berinteraksi dengan pengguna. Dalam Laravel, Views menggunakan Blade, sebuah template engine yang memudahkan dalam pembuatan tampilan yang dinamis.

3.1.4.1 View Home (home.blade.php)

Merupakan tampilan beranda yang berisi pos-pos terbaru.

```
<div>
    <!-- People find pleasure in different ways. I find it in keeping my mind clear. - Marcus
    Aurelius -->
</div>
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Forum Rumpi!</title>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
    awesome/6.4.0/css/all.min.css">

    <link rel="stylesheet" href="{{ asset('css/layout.css') }}">
    <link rel="stylesheet" href="{{ asset('css/home.css') }}">
</head>

<body>
    <header>
        <h1>Forum Rumpi!</h1>
        <nav class="nav">
            <ul>

                <li>Kalo gk bisa login dulu
                /rumpi.com/login

                <li><a href="{{ route('profile.show', ['idLogin' => $userId->id])
                }}">Profil</a></li>

            </ul>
        </nav>
        <button type="button" class="modal-button" onclick="openModal('modal1')"> <i
        class="fa-solid fa-gear"></i>
    </button>
    </header>

    <!--Isi kontene dinamis dari db-->

    <main>

    <!-- Modal Setting -->
```

```

<div id="modal1" class="modal">
  <div class="modal-content">
    <button type="button" class="close-button"
onclick="closeModal('modal1')"><</button>
    <br>
    <center>

        <br>
        <div class="profil-box">{{--
            session_start();
            if (isset($_SESSION['username'])) {
                $username = $_SESSION['username'];
                echo "<strong> $username</strong>";
            } else {
                // Handle the case when the username is
not set in the session

                $username = "Guest";
            }

            --}}</div>

        <h3>Pengaturan</h3>
        <button onclick="darkMode()">Mode Gelap</button>

        <br>

    </center>
  </div>
</div>

<script src={{ asset('js/modal.js') }}></script>
<script>
    applyDarkModePreference();
</script>

<script src={{ asset('js/darkMode.js') }}></script>

<br>
<div id="createPost">
  <a href="{{ route('post.create') }}" style="text-decoration: none;">
    <i class="fas fa-circle fa-lg" style="color: green; display: inline-block;
position: relative;">
      <i class="fas fa-plus"
        style="color: white; position: absolute; top: 50%; left: 50%;
transform: translate(-50%, -50%);"></i>
    </i>
    Buat Postingan
  </a>
</div>

```

```

</a>

</div>
<br>

@if (session('success'))
    <div class="alert alert-success">
        {{ session('success') }}
    </div>
@endif

{{--    @foreach ($postDatas as $post)
        {{ $post->content }}
        {{ $post->created_at }}
        {{ $post->user->username }}
            {{ $post->user->profile->profile_image }}
        {{ $post->post_image }}
        {{ $post->likes }}
        {{ $post->comments->count() }}
        {{ $post->comments->first()->content }}
        {{ $post->comments->first()->user->username }}
        {{ $post->comments->first()->user->profile->profile_image }}
        {{ $post->comments->first()->created_at }}
        {{ $post->comments->first()->updated_at }}
        {{ $post->comments->first()->likes->count() }}
        {{ $post->comments->first()->likes->first()->user->username }}
        {{ $post->comments->first()->likes->first()->user->profile->profile_image }}
        {{ $post->comments->first()->likes->first()->created_at }}
        {{ $post->comments->first()->likes->first()->updated_at }}
        {{ $post->comments->first()->likes->first()->like }}
        {{ $post->likes->first()->user->username }}
        {{ $post->likes->first()->user->profile->profile_image }}
        {{ $post->likes->first()->created_at }}
        {{ $post->likes->first()->up }}
    @endforeach --}}

@foreach ($postDatas as $post)
    <section class="post-section">
        <div class="post">
            <div class="post-header">
                <div class="user-info">

                    {{--    {{dd($post->user->profile->url)}} --}}
                    @if ($post->user->profile && $post->user->profile->url)
                        

                    @else
                        
                    @endif
                </div>
            </div>
        </div>
    </section>
@endforeach

```



```

        <h3> {{ $post->user->name }} </h3>
    </div>

</div>
</div>

<div class="post-content">
    {{-- @dd($post->files) --}}
    @if ($post->files->isEmpty())
        @foreach ($post->files as $file)
            @if ($file->url != null)
                
            @endif
        @endforeach
    @endif

    <h3>{{ $post->title }}</h3>
    <p>{{ $post->updated_at->format('d-m-Y') }}</p>
    <p>{{ $post->content }}</p>
</div>

<div class="post-footer">
    <button type="button" style="display: inline-block;">
        <i class="fas fa-thumbs-up fa-lg" style="color: red;"></i> {{ $post->like_count }}
    </button>

    <form action="{{ route('comment.show', ['idPost' => $post->id]) }}"
method="get"
style="display: inline-block;">
        <button type="submit" style="background: none; border: none; cursor:
pointer;">
            <i class="fas fa-comment fa-lg" style="color: blue;"></i>
            Komentari
        </button>
    </form>

</div>
</section>
@endforeach

<div class="pagination-wrapper">
    {{ $postDatas->links() }}
</div>

</main>

```

```
<footer>
  <p>&copy; 2023 Forum Rumpi! </p>
</footer>
</body>

</html>
```

3.1.4.2 View Comment Post(comment.balde.php)

Berisi tampilan komentar-komentar yang ada pada po tertentu

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Komentar</title>
  <link rel="stylesheet" href="{{ asset('css/layout.css') }}">

  <link rel="stylesheet" href="{{ asset('css/comment.css') }}">

  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.0/css/all.min.css">
</head>

<body>
  <header>
    <h1>Forum Rumpi!</h1>
    <nav class="nav">
      <ul>
        <li><a href="{{ route('home') }}">Home</a></li>

        <li><a href="profile.php">Profil</a></li>

      </ul>
    </nav>
    <button type="button" class="modal-button" onclick="openModal('modal1')"> <i
class="fa-solid fa-gear"></i>
    </button>
  </header>
  <!-- nanti isi comment nha dinamis dari db-->
  <main>

    <div id="modal1" class="modal">
      <div class="modal-content">
        <button type="button" class="close-button"
onclick="closeModal('modal1')"></button>
        <br>
        <center>
          <h3>Pengaturan</h3>
          <button onclick="darkMode()">Mode Gelap</button>
        </center>
      </div>
    </div>
    <script src="{{ asset('js/modal.js') }}"></script>
  </script>
```

```

        applyDarkModePreference();
    </script>

    <script src="{{ asset('js/darkmode.js') }}"></script>

    {{--      @dd($post) --}}

    <section class="comments-section">
        <div class="post-content">

            @if ($post->files->isEmpty())
                @foreach ($post->files as $file)
                    @if ($file->url != null)
                        
                    @endif
                @endforeach
            @endif

            <h3>{{ $post->title }}</h3>
            <p>{{ $post->updated_at->format('d-m-Y') }}</p>
            <p>{{ $post->content }}</p>
        </div>
    </section>

    <section class="comments-section">

        <h2>Komentar</h2>

        @foreach ($postComments as $comment)
            <div class="comment">
                <div class="comment-header">
                    @if ($comment->user->profilePhoto == null)
                        
                    @else
                        
                    @endif
                <div class="comment-info">
                    <div class="user-info">
                        <h3>{{ $comment->user->username }}</h3>
                    </div>
                </div>
            </div>
            <div class="comment-content">
                <p>{{ $comment->updated_at->format('d-m-Y') }}</p>
                <p>{{ $comment->content }}</p>
            </div>
        </div>
    </section>

```

```

        @endforeach

        <h2>Tambahkan Komentar</h2>
        <div class="comment-form">
            <form action="comment.php" method="post" id="comment-form">
                <label for="comment-content">Komentar Anda:</label> <br>
                <textarea id="comment-content" name="comment_content" rows="5"
required></textarea><br>
                <button type="submit">Kirim Komentar</button>
            </form>
        </div>

    </section>

</section>

<div>
    <a> Kembali ke postingan</a>
</div>

</main>

<footer>
    <p>&copy; 2023 Forum Rumpi!</p>
</footer>

</body>

</html>

```

Berisi tampilan profile dari user, pada halaman ini juga ada tampilan daftar pos yang di unggah oleh pengguna tersebut dan dapat menghapus, da mengeditnya:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Buat Postingan Baru</title>

  <link rel="stylesheet" href="{{ asset('css/profile.css') }}">
  <link rel="stylesheet" href="{{ asset('css/layout.css') }}">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.0/css/all.min.css">
</head>

<body>
  <header>
    <h1>Forum Rumpi!</h1>
    <nav class="nav">
      <ul>
        <li><a href={{ route('home') }}>Home</a></li>
      </ul>
    </nav>

    <div style="display: flex; justify-content: space-between;">
      <button id="backBtn" onclick="window.history.back()">
        <i class="fas fa-arrow-left"></i>
      </button>

      <button id="setBtn" type="button" class="modal-button"
onclick="openModal('modal1')">
        <i class="fa-solid fa-gear"></i>
      </button>
    </div>
  </header>

  <main>

    <!-- Modal RejMsg -->
    <div id="modal1" class="modal">
      <div class="modal-content">
        <button type="button" class="close-button"
onclick="closeModal('modal1')"></button>
        <br>
        <center>
          <h3>Pengaturan</h3>
          <button onclick="darkMode()">Mode Gelap</button>
        </center>
      </div>
    </div>
```

```

</div>

<script src="js/modal.js"></script>
<script>
    applyDarkModePreference();
</script>

<script src="js/darkmode.js"></script>

<br>
<div id="createPost">
    <a href="{{ route('post.create') }}" style="text-decoration: none;">
        <i class="fas fa-circle fa-lg" style="color: green; display: inline-block; position: relative;">
            <i class="fas fa-plus"
                style="color: white; position: absolute; top: 50%; left: 50%; transform: translate(-50%, -50%);"></i>
            </i>
            Buat Postingan
        </a>
    </div>
<section class="profil-section">
    <div class="profil-left">
        <div class="profil-image">

            @if ($dataUser->profile->url == null)
                
            @else
                
            @endif

        </div>
        <div class="post-info">
            <h3><span>{{ $dataUser->name }}</span></h3>
            <p id="biography"> {{ $dataUser->profile->biografi }}</p>
        </div>
    </div>
    <div class="profil-right">
        {{--      {{dd($dataUser->profile->url)}} --}}
        <div class="proflie-data">
            <h3>Informasi Pribadi</h3>
            <ul>
                <li>
                    <strong>Email</strong><br>
                    <span>{{ $dataUser->email }}</span>
                    <hr>
                </li>
                <li>
                    <strong>Jenis Kelamin</strong><br>
                    <span>{{ $dataUser->profile->jenis_kelamin }}</span>
                    <hr>
                </li>
            </ul>
        </div>
    </div>
</div>

```



```

@foreach ($dataUser->posts as $post)
    <section class="post-section">
        <div class="button-container">

            <form method="get" action="{{ route('post.edit', ['idPost' => $post->id])
        }}">

                @csrf
                <button style="background: blue; color: white;" type="submit">
                    Edit
                </button>
            </form>

            <form method="post" action="{{ route('post.delete', ['idPost' => $post-
        >id]) }}">

                @csrf
                <button style="background: red; color: white;"
                    onclick="return confirm(' Yakin ingin Menghapus Pos? Pos Akan
        dihapus permanen')">Hapus</button>
            </form>
        </div>

        <div class="post-content">
            {{-- @dd($post->files) --}}
            @if ($post->files->isNotEmpty())
                @foreach ($post->files as $file)
                    @if ($file->url != null)
                        
                    @endif
                @endforeach
            @endif

            <h3>{{ $post->title }}</h3>
            <p>{{ $post->updated_at->format('d-m-Y') }}</p>
            <p>{{ $post->content }}</p>
        </div>

        <div class="post-footer">

            <i class="fas fa-thumbs-up fa-lg" style="color: red;"></i> {{ $post-
        >like_count }}

            <form action="{{ route('comment.show', ['idPost' => $post->id]) }}"
        method="get"

                style="display: inline-block;"
                <button type="submit" style="background: none; border: none; cursor:
        pointer;">

                    <i class="fas fa-comment fa-lg" style="color: blue;"></i>

        Komentar

                </button>
            </form>

```

```

        </div>
    </section>
@endforeach

    {{-- {{dd($dataUser->posts)}} --}}

    <br>
</main>

<footer>
    <p>&copy; 2023 Forum Rumpi!</p>
</footer>
</body>

</html>

```

3.1.4.4 View edit Post Post(editPostView.balde.php)

Berisi tampilan halaman edit dari post yang dipilih pada tampilah profile sebelumnya.

```

<div>
    <!-- People find pleasure in different ways. I find it in keeping my mind clear. - Marcus
Aurelius -->
</div>
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Forum Rumpi!</title>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.0/css/all.min.css">

    <link rel="stylesheet" href="{{ asset('css/layout.css') }}">
    <link rel="stylesheet" href="{{ asset('css/home.css') }}">
</head>

<body>
    <header>
        <h1>Forum Rumpi!</h1>
        <nav class="nav">
            <ul>

```

```

        <li><a href={{ route('home') }}>Home</a></li>

    </ul>
</nav>
<div style="display: flex; justify-content: space-between;">
    <button id="backBtn" onclick="window.history.back()">
        <i class="fas fa-arrow-left"></i>
    </button>

    <button id="setBtn" type="button" class="modal-button"
onclick="openModal('modal1')">
        <i class="fa-solid fa-gear"></i>
    </button>
</div>
<style>
    body,
    h1,
    h2,
    h3,
    p,
    ul,
    li {
        margin: 0;
        padding: 0;
    }

    body {
        font-family: 'Arial', sans-serif;
        font-size: 16px;
    }

    .post-section {
        max-width: 600px;
        margin: 0 auto;
        padding: 20px;
    }

    form {
        margin-bottom: 20px;
    }

    label {
        display: block;
        margin-bottom: 5px;
        font-weight: bold;
    }

    input {
        width: 100%;
        padding: 8px;
        margin-bottom: 10px;
        box-sizing: border-box;
        max-width: 200;
    }

```



```

        <br>
        <div class="profil-box">{{--
            session_start();
            if (isset($_SESSION['username'])) {
                $username = $_SESSION['username'];
                echo "<strong> $username</strong>";
            } else {
                // Handle the case when the username is
not set in the session

                $username = "Guest";
            }

            --}}</div>

        <h3>Pengaturan</h3>
        <button onclick="darkMode()">Mode Gelap</button>

        <br>

    </center>
</div>
</div>

<script src={{ asset('js/modal.js') }}></script>
<script>
    applyDarkModePreference();
</script>

<script src={{ asset('js/darkMode.js') }}></script>

<br>

@if (session('success'))
    <div class="alert alert-success">
        {{ session('success') }}
    </div>
@endif

<br>
<section class="post-section">
    {{-- {{ dd($post) }} --}}
    <div>
        <center>
            <h2> Edit Post </h2>

```

```

</center>

<form method="post" action="{{ route('post.update', ['idPost' => $post->id])
}}"
    enctype="multipart/form-data">
    @csrf

    <label> Judul</label>
    <input type="text" value="{{ $post->title }}" name="title" required> <br>

    <label> Konten</label>
    <textarea rows="4" name="content" required>{{ $post->content
}}</textarea>

    <br>

    <label>Hapus File</label> <br>

    @foreach ($post->files as $file)
        <div>
            <input type="checkbox" name="selected_files[]"
                value="{{ $file->id }}:{{ $file->url }}">
            {{ $file->fileName }} <br>
            
        </div>
    @endforeach

    <br>

    <button type="submit" class="btn btn-primary"
        onclick="return confirm(' Yakin ingin menyimpan
perubahan?')">Update</button>

    </form>

</div>

</section>

</main>

```

```
<footer>
  <p>&copy; 2023 Forum Rumpi! </p>
</footer>
</body>
</html>
```

3.1.1 Routes

Routes pada MVS mekanisme untuk menentukan cara aplikasi web menanggapi permintaan HTTP. Routes mendefinisikan hubungan antara URL dan tindakan (action) yang harus diambil oleh aplikasi. Laravel menyediakan beberapa jenis routes yang dapat diatur dalam file routes/web.php untuk routes web dan routes/api.php untuk routes API.

3.1.1.1 Routes Web (web.php)

```
<?php

use App\Http\Controllers\CommentController;
use Illuminate\Support\Facades\Route;

use App\Http\Controllers\AuthController;
use App\Http\Controllers\PostController;
use App\Http\Controllers\ProfileController;

/*
|-----
| Web Routes
|-----
|
| Here is where you can register web routes for your application. These
| routes are loaded by the RouteServiceProvider and all of them will
| be assigned to the "web" middleware group. Make something great!
|
*/
Route::get('/rumpi.com/login', [AuthController::class, 'login'])->name('login');
Route::post('/rumpi.com/login', [AuthController::class, 'authenticating']);
Route::get('/rumpi.com/logout', [AuthController::class, 'logout']);

Route::get('/rumpi.com/home', [PostController::class, 'index'])->name('home');
Route::get('/rumpi.com/create_post', [PostController::class, 'create'])->
>name('post.create');
Route::post('/rumpi.com/store', [PostController::class, 'store'])->name('post.store');
Route::get('/rumpi.com/edit_pos/{idPost}', [PostController::class, 'edit'])->
>name('post.edit');
Route::post('/rumpi.com/update/{idPost}', [PostController::class, 'update'])->
>name('post.update');
Route::post('/rumpi.com/delete/{idPost}', [PostController::class, 'destroy'])->
```

3.2 Kode CSS

Pada CSS layout saya menambahkan kode untuk mode gelap dimana warna latar dan teks untuk body, header dan footer akan menyesuaikan. Saya memilih warna latar hitam untuk latar header dan footer serta latar container setiap halaman dan warna putih untuk teksnya. Untuk warna latar mode gelap bagian body saya menggunakan warna abu-abu terang. File-file CSS harus berada pada folder public agar dapat terdeteksi dan tampil dalam balde, penggunaan CSS membutuhkan prefix asset

```
<link rel="stylesheet" href="{{ asset('css/comment.css') }}">
```


3.2.1 Stylesheet layout.css

```
1  body {
2    display: flex;
3    flex-direction: column;
4    min-height: 100vh;
5    margin: 0;
6    padding: 0;
7    transition: background-color 0.5s, color 0.5s;
8  }
9
10
11  .dark-mode h1 {
12
13    color: white;
14  }
15
16  #modal1 {
17    display: none;
18    position: fixed;
19    z-index: 1;
20    padding-top: 20px;
21    padding-bottom: 20px;
22    right: 0;
23    top: 0;
24    width: 300px;
25    height: 100%;
26    overflow: auto;
27    background-color: #333;
28    border: 1px solid white;
29    right: 5px;
30    bottom: 5px;
31  }
32
33  .modal-content {
34    background-color: #fefefe;
35    margin: auto;
36    padding: 10px;
37    border: 1px solid #888;
38    width: 80%;
39    height: 90%;
40  }
41
42  .close-button {
43    color: #aaaaaa;
44    float: right;
45    font-size: 20px;
46    font-weight: bold;
47  }
48
49  .modal-button {
50    background-color: silver !important;
51    color: #333;
52    float: right;
53    border: none;
54    font-size: 20px;
55    font-weight: bold;
56  }
57
58  .dark-mode .modal {
59    background-color: black;
60    color: white;
61  }
62
63  .dark-mode .modal-content {
64    background-color: black;
65    border: 1px solid white;
66    color: white;
67  }
68
69
```

```
69
70
71 .dark-mode .modal-button {
72     background-color: #333 !important;
73     color: white;
74 }
75
76 main {
77     flex: 1;
78     background-color: silver;
79 }
80
81 .dark-mode main {
82     background-color: #333;
83 }
84
85 header {
86     background-color: #333;
87     color: #fff;
88     text-align: center;
89     padding: 10px;
90 }
91
92 .dark-mode header {
93     background-color: black;
94 }
95
96 nav ul {
97     list-style-type: none;
98     margin: 0;
99     padding: 0;
100 }
101
102 nav li {
103     display: inline;
104     margin-right: 20px;
105 }
106
107 nav a {
108     text-decoration: none;
109     color: #fff;
110 }
111
112 footer {
113     background-color: #333;
114     color: #fff;
115     text-align: center;
116     padding: 10px;
117 }
118
119 .dark-mode footer {
120     background-color: black;
121 }
122
123
```

3.1.2 Stylesheet login.css

```
1  .login-box {
2      width: 300px;
3      margin: 100px auto;
4      padding: 20px;
5      background: #fff;
6
7      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
8  }
9
10 .login-box h2 {
11     text-align: center;
12     color: #333;
13 }
14
15 .login-box form {
16     display: flex;
17     flex-direction: column;
18 }
19
20 .login-box label {
21
22     color: #555;
23 }
24
25 .login-box input {
26     padding: 8px;
27     margin-bottom: 15px;
28     border: 1px solid #ccc;
29     border-radius: 4px;
30 }
31
32 .login-box input[type="submit"] {
33     background-color: gray;
34     color: #fff;
35     cursor: pointer;
36 }
37
38 .login-box input[type="submit"]:hover {
39     background: #2187c4;
40 }
41
42
43 /* Dark Mode */
44 .dark-mode .login-box {
45     background-color: black;
46     color: white;
47 }
48
49 .dark-mode .login-box h2 {
50     color: white;
51 }
52
53 .dark-mode .login-box label {
54     color: white;
55 }
56
```

3.2.3 Stylesheet home.css

```
1  /* Light Mode */
2  #createPost {
3      text-align: center;
4      margin: auto;
5      padding: 10px;
6      border: solid 1px black;
7      border-radius: 10px;
8      max-width: 150px;
9      background-color: white;
10 }
11
12 #imageProfile {
13     width: 45px;
14     height: 45px;
15     border-radius: 50%;
16 }
17
18 #imagePost {
19     max-width: 400px;
20     max-height: 300px;
21     border-radius: 10px;
22     display: block;
23     margin: 0 auto;
24 }
25
26 .post-section {
27     background-color: white;
28     padding: 20px;
29     border-radius: 5px;
30     width: 500px;
31     margin: 0 auto;
32     display: flex;
33     margin-bottom: 20px;
34     flex-direction: column;
35 }
36
37 .comments-section {
38     background-color: black;
39     padding: 20px;
40     border-radius: 5px;
41 }
42
43 .user-info {
44     display: flex;
45     gap: 10px;
46 }
47
48 .post-header {
49     display: flex;
50 }
51
```

```
51
52 .comment {
53     border: 1px solid #ccc;
54     padding: 10px;
55     margin: 10px 0;
56     border-radius: 5px;
57     max-width: 100%;
58 }
59
60 .close {
61     color: #aaaaaa;
62     float: right;
63     font-size: 28px;
64     font-weight: bold;
65 }
66
67 /* Dark Mode */
68 .dark-mode #createPost {
69     background-color: black;
70     color: white;
71 }
72
73 .dark-mode #imageProfile {
74     border: 2px solid white;
75 }
76
77 .dark-mode #imagePost {
78     border: 2px solid white;
79 }
80
81 .dark-mode .post-section {
82     background-color: black;
83     color: white;
84     border: 1px solid white;
85 }
86
87 .dark-mode .comments-section {
88     background-color: white;
89     color: black;
90     border: 1px solid black;
91 }
92
93 .dark-mode .comment {
94     border: 1px solid white;
95 }
96
```

3.2.4 Stylesheet createPost.css

```
1  /* Light mode styles */
2  .create-post-container {
3    max-width: 800px;
4    margin: 20px auto;
5    padding: 20px;
6    background-color: #fff;
7    border-radius: 5px;
8    box-shadow: 0 0 5px rgba(0, 0, 0, 0.2);
9  }
10
11 .create-post-container .grid-container {
12   display: grid;
13   grid-template-columns: 1fr 1fr;
14   grid-template-rows: 1fr 1fr;
15   grid-gap: 20px;
16 }
17
18 .create-post-container .grid-item {
19   width: 100%;
20 }
21
22 .create-post-container textarea {
23   width: 100%;
24   margin-bottom: 10px;
25   padding: 10px;
26   border: 1px solid #ccc;
27   border-radius: 5px;
28 }
29
30 .file-inputs {
31   margin-bottom: 10px;
32 }
33
34 .preview-container {
35   display: flex;
36   flex-wrap: wrap;
37 }
38
39 .preview-item {
40   margin-right: 10px;
41   margin-bottom: 10px;
42 }
43
44 .preview-container {
45   text-align: center;
46 }
47
48 .preview-item img,
49 .preview-item video {
50   max-width: 100px;
51   max-height: 100px;
52   border-radius: 5px;
53 }
54
55 /* Dark mode styles */
56 .dark-mode .create-post-container {
57   background-color: #333;
58   color: #fff;
59   box-shadow: 0 0 5px rgba(255, 255, 255, 0.2);
60 }
61
62 .dark-mode .create-post-container textarea {
63   border: 1px solid #999;
64   background-color: #444;
65   color: #fff;
66 }
67
68 .dark-mode .preview-item img,
69 .dark-mode .preview-item video {
70   border: 1px solid #666;
71 }
72
```

3.2.5 Stylesheet Comment.css

```
1  /* Light Mode */
2  .comments-section {
3      display: grid;
4      grid-template-columns: 1fr;
5      grid-gap: 20px;
6      background-color: #fff;
7      padding: 20px;
8      border-radius: 5px;
9      max-width: 500px;
10     margin: 10px auto;
11 }
12
13 .comment {
14     border: 1px solid black;
15     padding: 20px;
16     border-radius: 5px;
17 }
18
19 .comment-header {
20     display: flex;
21     align-items: center;
22 }
23
24 #imageProfile {
25     width: 40px;
26     height: 40px;
27     margin-right: 10px;
28     border-radius: 50%;
29 }
30
31 .comment-content {
32     margin-top: 10px;
33 }
34
35 /* Dark Mode */
36 .dark-mode .comments-section {
37     background-color: black;
38     color: white;
39 }
40
41 .dark-mode .comment {
42     border: 1px solid white;
43 }
44
45 .dark-mode #imageProfile {
46     border: 2px solid white;
47 }
48
```


3.2.6 Stylesheet profil.css

```
1  #createPost {
2    text-align: center;
3    margin: auto;
4    padding: 10px;
5    border: solid 1px black;
6    border-radius: 10px;
7    max-width: 150px;
8    background-color: white;
9  }
10
11  .profil-section {
12    display: grid;
13    grid-template-columns: 1fr 1fr;
14    /* bagi kolom jadi dua */
15    grid-gap: 30px;
16    background-color: #fff;
17    padding: 20px;
18    border-radius: 5px;
19    max-width: 800px;
20    margin: 20px auto;
21    box-shadow: 0 0 5px rgba(0, 0, 0, 0.2);
22  }
23
24
25  #img-profil {
26    border-radius: 50%;
27    border: 5px solid transparent;
28    width: 150px;
29    height: 150px;
30    margin: 0 auto;
31    display: block;
32  }
33
34  #biography {
35    margin: 10px;
36    padding: 10px;
37    border: solid 1px black;
38    border-radius: 10px;
39    max-width: 400px;
40    background-color: white;
41    margin: 0 auto;
42    text-align: center;
43  }
44
45  .proprofile-data {
46    display: grid;
47    grid-template-columns: 1fr;
48    grid-gap: 10px;
49  }
50
51  button {
52    background-color: #333;
53    color: #fff;
54    padding: 10px 20px;
55    border: none;
56    border-radius: 5px;
57    cursor: pointer;
58    margin-top: 10px;
59  }
60
61  button:hover {
62    background-color: #555;
63  }
64
65  .button-position {
66    text-align: center;
67    margin: auto;
68    padding: 10px;
69    max-width: 150px;
70  }
71
72
73
74  #createPost {
75    text-align: center;
76    margin: auto;
77    padding: 10px;
78    border: solid 1px black;
79    border-radius: 10px;
80    max-width: 150px;
81    background-color: white;
82  }
83
84  .profil-section {
85    display: grid;
86    grid-template-columns: 1fr 1fr;
87    /* bagi kolom jadi dua */
88    grid-gap: 30px;
89    background-color: #fff;
90    padding: 20px;
91    border-radius: 5px;
92    max-width: 800px;
93    margin: 20px auto;
94    box-shadow: 0 0 5px rgba(0, 0, 0, 0.2);
95  }
96
97
```

```
116 }
117
118 .proprofile-data {
119   display: grid;
120   grid-template-columns: 1fr;
121   grid-gap: 10px;
122 }
123
124 button {
125   background-color: #333;
126   color: #fff;
127   padding: 10px 20px;
128   border: none;
129   border-radius: 5px;
130   cursor: pointer;
131   margin-top: 10px;
132 }
133
134 button:hover {
135   background-color: #555;
136 }
137
138 .button-position {
139   text-align: center;
140   margin: auto;
141   padding: 10px;
142   max-width: 150px;
143 }
144
145
146
147 /* Dark Mode */
148 .dark-mode #createPost {
149   background-color: black;
150   color: white;
151 }
152
153 .dark-mode .profil-section {
154   background-color: black;
155   color: white;
156   box-shadow: 0 0 5px rgba(255, 255, 255, 0.2);
157 }
158
159 .dark-mode #img-profil {
160   border-color: white;
161 }
162
163 .dark-mode #biography {
164   background-color: #333;
165   color: white;
166   border: solid 1px white;
167 }
168
169 .dark-mode .proprofile-data {
170   color: white;
171 }
172
173 .dark-mode button {
174   background-color: #333;
175   color: #fff;
176 }
177
178 .dark-mode button:hover {
179   background-color: #777;
180 }
181
182 .dark-mode #postsBtn:hover {
183   background-color: #777;
184 }
185
186 .dark-mode #postsBtn {
187   background-color: #555;
188   color: #fff;
189 }
190
```

3.3 Kode Javascript

Kode script javascript digunakan untuk mengimplementasikan fitur menu pengaturan menggunakan sidebar pada setiap halaman web dibuat menggunakan fungsi-fungsi untuk mendapatkan responsif, menampilkan dan menutup modal dan isinya didalam tag elemen div modalId.

Button dengan kelas modal-button pada gambar dibawah dengan menggunakan onclick pada tag nya akan memicu fungsi openModal() yang akan menampilkan modal berupa sidebar menubpnegaturan.

```
1 <button type="button" class="modal-button" onclick="openModal('modal1')"> <i class="fa-solid fa-gear"></i>
2 </button>
3
4
5 <!-- Modal RejMsg -->
6 <div id="modal1" class="modal">
7   <div class="modal-content">
8     <button type="button" class="close-button" onclick="closeModal('modal1')"></button>
9     <br>
10    <center>
11      <h3>Pengaturan</h3>
12      <button onclick="darkMode()">Mode Gelap</button>
13    </center>
14  </div>
15 </div>
16
17
18 <script src="js/modal.js"></script>
```

Lalu didalam modal sidebar tersebut ada button untuk menutup modal dimana button akan memicu fungsi closeModal()..

Selanjutnya terdapat tombol dan untuk mengubah tema latar menjadi mode gelap atau terang, Tombol ini akan memicu fungsi darkMode() yang berisi fungsi-fungsi untuk mengganti warna elemen-elemen css dan menyimpan tema yang sedang digunakan agar tema tetap digunakan dihalaman lainnya.

```
1 <center>
2     <h3>Pengaturan</h3>
3     <button onclick="darkMode()">Mode Gelap</button>
4 </center>
```

3.3.1 Script Modal Sidebar Menu Pengaturan modal.js

Fungsi opeModal(modalId) digunakan untuk membuka modal dengan Id yang diberikan sebagai parameter (modalId). Pertama, fungsi mengambil elemen modal dari dokumen HTML dengan menggunakan document.getElementById(modalId). Selanjutnya, properti style.display dari elemen tersebut diatur menjadi "block", sehingga membuat modal menjadi terlihat atau tampil di layar.

```

1
2 // Fungsi untuk membuka modal
3 function openModal(modalId) {
4     var modal = document.getElementById(modalId);
5     modal.style.display = "block";
6 }
7

```

Fungsi `closeModal()` digunakan untuk menutup modal dengan ID yang diberikan sebagai parameter (`modalId`). Sama seperti sebelumnya, fungsi mengambil elemen modal berdasarkan ID, dan kemudian properti `style.display` diatur menjadi `"none"`, sehingga modal menjadi tidak terlihat atau disembunyikan.

```

7
8 // Fungsi untuk menutup modal
9 function closeModal(modalId) {
10     var modal = document.getElementById(modalId);
11     modal.style.display = "none";
12 }
13

```

Event Handler `window.onclick` akan dipanggil ketika pengguna mengklik di luar area modal. Ini digunakan untuk menutup modal jika latar belakang di luar modal (area di luar modal) diklik. Pengecekan dilakukan dengan memeriksa apakah elemen yang diklik memiliki kelas (`className`) `'modal'`. Jika ya, maka properti `style.display` dari elemen tersebut diatur menjadi `'none'`, sehingga modal ditutup.

```

13
14 // Menutup modal ketika latar belakangnya diklik
15 window.onclick = function(event) {
16     if (event.target.className === 'modal') {
17         event.target.style.display = 'none';
18     }
19 }

```

3.3.2 Script Fitur Tema `darkMode.js`

Fungsi ini dipanggil ketika tombol "Mode Gelap" pada modal diklik.

`document.body.classList.toggle("dark-mode")` digunakan untuk beralih antara menambah dan menghapus kelas `"dark-mode"` pada elemen `body`.

Status dark mode (aktif atau tidak) disimpan di `localStorage` menggunakan `localStorage.setItem()`.

```

1  function darkMode() {
2      var element = document.body;
3      element.classList.toggle("dark-mode");
4
5      // chck apakah dark mode aktif
6      var isDarkMode = element.classList.contains("dark-mode");
7      localStorage.setItem("darkMode", isDarkMode);
8  }
9

```

Fungsi ini dipanggil saat halaman dimuat untuk menerapkan preferensi dark mode yang disimpan. Melalui `localStorage.getItem()`, status dark mode yang disimpan diambil kembali. Berdasarkan status tersebut, kelas "dark-mode" ditambahkan atau dihapus pada elemen body untuk mengatur tampilan dark mode.

```

10
11 function applyDarkModePreference() {
12     var isDarkMode = localStorage.getItem("darkMode") === "true";
13     var element = document.body;
14
15     // simpan informasi darkmode
16     if (isDarkMode) {
17         element.classList.add("dark-mode");
18     } else {
19         element.classList.remove("dark-mode");
20     }
21 }
22
23 // panggil fungsi dark mode yang talh disimpan
24 applyDarkModePreference();

```

Dengan cara ini, preferensi dark mode pengguna akan disimpan secara lokal dan diterapkan setiap kali halaman dimuat kembali.

3.3.3 Script Fitur Menambahkan Gambar Dan Video postingan serta tampilan Preview addFile.js

```
1 document.getElementById('addFileInput').addEventListener('click', function () {
2     const fileInputsContainer = document.getElementById('fileInputs');
3
4     // Periksa jumlah file input yang sudah ada
5     const existingFileInputs = fileInputsContainer.querySelectorAll('.form-group').length;
6
7     if (existingFileInputs < 4) {
8         // Jika kurang dari 4, tambahkan file input baru
9         const fileInput = document.createElement('div');
10        fileInput.innerHTML = `
11            <div class="form-group">
12                <input type="file" name="files[]" class="form-control-file" accept="image/*,video/*" required>
13                <button type="button" class="removeFileInput btn btn-danger">Hapus File</button>
14            </div>
15        `;
16        fileInputsContainer.appendChild(fileInput);
17
18        // Event listener untuk menghapus file input yang ditambahkan
19        fileInput.querySelector('.removeFileInput').addEventListener('click', function () {
20            fileInput.remove();
21            updatePreview(); // Perbarui preview ketika file dihapus
22        });
23
24        // Event listener untuk memperbarui preview saat file dipilih
25        fileInput.querySelector('input[name="files[]"]').addEventListener('change', function (event) {
26            updatePreview();
27        });
28    } else {
29        alert('Anda telah mencapai batas maksimal (4) file.');
```

Fungsi tersebut berfungsi untuk mengelola penambahan dan penghapusan file input pada suatu formulir secara dinamis. Pertama-tama, code ini menetapkan event listener pada elemen dengan ID 'addFileInput', yang akan memicu suatu fungsi ketika elemen tersebut di-klik.

Dalam fungsi tersebut, script mengambil elemen dengan ID 'fileInputs', yang merupakan kontainer untuk file input. Selanjutnya, ia memeriksa jumlah file input yang sudah ada di dalam kontainer tersebut dengan menggunakan `querySelectorAll` dan menghitung panjang nodelist yang ditemukan.

Jika jumlah file input yang sudah ada kurang dari 4, maka script akan membuat elemen file input baru dalam bentuk div dengan class 'form-group'. Elemen ini berisi input file untuk memilih file dan tombol untuk menghapus file input yang baru ditambahkan. Kemudian, file input baru ini ditambahkan ke dalam kontainer 'fileInputs'. Event listener juga ditambahkan pada tombol 'removeFileInput' untuk menghapus file input yang terkait.


```

31
32 // Fungsi untuk memperbarui preview file
33 function updatePreview() {
34     const previewContainer = document.getElementById('previewContainer');
35     previewContainer.innerHTML = '';
36
37     const allFileInputs = document.querySelectorAll('input[name="files[]"]');
38     allFileInputs.forEach(input => {
39         Array.from(input.files).forEach(file => {
40             const previewItem = document.createElement('div');
41             previewItem.classList.add('preview-item');
42
43             if (file.type.startsWith('image/')) {
44                 const img = document.createElement('img');
45                 img.src = URL.createObjectURL(file);
46                 previewItem.appendChild(img);
47             } else if (file.type.startsWith('video/')) {
48                 const video = document.createElement('video');
49                 video.src = URL.createObjectURL(file);
50                 video.setAttribute('controls', 'controls');
51                 previewItem.appendChild(video);
52             }
53
54             previewContainer.appendChild(previewItem);
55         });
56     });
57 }
58 });
59

```

Fungsi `updatePreview()` berfungsi agar setiap kali file input baru ditambahkan akan memperbarui tampilan pratinjau (preview) setiap kali file dipilih atau dihapus. Fungsi `updatePreview` akan membuat elemen pratinjau baru berdasarkan file-file yang dipilih. Jika file adalah gambar, maka elemen pratinjau berupa tag ``, dan jika file adalah video, maka elemen pratinjau berupa tag `<video>`.

```

31
32 // Fungsi untuk memperbarui preview file
33 function updatePreview() {
34     const previewContainer = document.getElementById('previewContainer');
35     previewContainer.innerHTML = '';
36
37     const allFileInputs = document.querySelectorAll('input[name="files[]"]');
38     allFileInputs.forEach(input => {
39         Array.from(input.files).forEach(file => {
40             const previewItem = document.createElement('div');
41             previewItem.classList.add('preview-item');
42
43             if (file.type.startsWith('image/')) {
44                 const img = document.createElement('img');
45                 img.src = URL.createObjectURL(file);
46                 previewItem.appendChild(img);
47             } else if (file.type.startsWith('video/')) {
48                 const video = document.createElement('video');
49                 video.src = URL.createObjectURL(file);
50                 video.setAttribute('controls', 'controls');
51                 previewItem.appendChild(video);
52             }
53
54             previewContainer.appendChild(previewItem);
55         });
56     });
57 }
58 });
59
60 document.addEventListener('click', function (event) {
61     if (event.target && event.target.className === 'removeFileInput btn btn-danger') {
62         event.target.parentElement.remove();
63         updatePreview(); // Perbarui preview ketika file dihapus
64     }
65 });
66

```

```

59
60 document.addEventListener('click', function (event) {
61     if (event.target && event.target.className === 'removefileInput btn btn-danger') {
62         event.target.parentElement.remove();
63         updatePreview(); // Perbarui preview ketika file dihapus
64     }
65 });
66

```

Selain itu, terdapat penanganan event click global pada seluruh dokumen (document) dengan event listener yang memeriksa apakah elemen yang diklik memiliki class 'removefileInput' dan class 'btn btn-danger'. Jika iya, itu berarti tombol untuk menghapus file input diklik, dan file input yang terkait akan dihapus dari DOM. Setelah itu, fungsi updatePreview dipanggil untuk memperbarui tampilan pratinjau setelah file dihapus. Dengan demikian, code ini memastikan bahwa formulir dapat menangani penambahan dan penghapusan file input dengan baik.

File-file Javascript sama seperti CSS harus berada pada folder public dan dipanggil pada blade menggunakan prefix asset

```
<script src="{{ asset('js/modal.js') }}"></script>
```

Bab 4 Penutup

4.1 Kesimpulan

Dalam praktikum pemrograman web pada tanggal 15 Desember 2023, kami sebagai mahasiswa Ilmu Komputer telah mempelajari dasar-dasar dari penggunaan framework Laravel untuk pengembangan web, di bimbing oleh para asisten praktikum selama sesi pemberian materi. Kami juga telah mengimplementasikan pemahaman tersebut dengan mengerjakan tugas posttest dan THT yang diberikan setelah sesi praktikum.

PHP (Hypertext Preprocessor) adalah bahasa pemrograman server-side yang digunakan untuk pengembangan aplikasi web dinamis. PHP secara khusus dirancang untuk mengolah data di sisi server sebelum mengirimkan hasilnya ke klien (browser). Beberapa fitur utama PHP melibatkan manipulasi string, pengolahan formulir, pengelolaan file, dan berinteraksi dengan database.

Laravel adalah sebuah framework pengembangan web yang ditulis dalam bahasa pemrograman PHP. Dikembangkan oleh Taylor Otwell, Laravel menjadi salah satu framework PHP paling populer dan digunakan secara luas dalam pengembangan aplikasi web modern. Berikut adalah beberapa fitur utama dari Laravel:

1. Eloquent ORM:
Eloquent adalah sistem ORM (Object-Relational Mapping) yang memungkinkan pengembang berinteraksi dengan basis data menggunakan sintaks bahasa pemrograman PHP daripada SQL tradisional. Ini membuat manipulasi data lebih intuitif dan ekspresif.
2. Blade Templating Engine:
Blade adalah engine template yang kuat dan sederhana. Blade menyediakan sintaks yang bersih dan mudah dimengerti untuk membuat tampilan aplikasi. Blade juga mendukung fitur seperti inheritance, section, dan komponen.

3. **Artisan Console:**
Artisan adalah CLI (Command Line Interface) yang terintegrasi dengan Laravel. Dengan Artisan, pengembang dapat melakukan berbagai tugas seperti pembuatan model, pengendali, migrasi database, dan lainnya dengan mudah.
4. **Routing:**
Laravel menyediakan sistem routing yang ekspresif dan fleksibel. Routes dapat dengan mudah didefinisikan untuk menangani permintaan HTTP dan menyusun aplikasi ke dalam pengendali dan metode yang terorganisir.
5. **Middleware:**
Middleware memungkinkan pengembang mengatur serangkaian tindakan yang harus dijalankan sebelum atau setelah permintaan HTTP mencapai pengendali. Ini berguna untuk menangani otentikasi, logging, dan lainnya.
6. **Migrations dan Seeders:**
Laravel menyediakan migrasi untuk mengelola skema basis data dan seeding untuk mengisi basis data dengan data uji atau awal. Ini membantu dalam pengelolaan struktur basis data secara efisien.
7. **Dependency Injection dan Container:**
Laravel menggunakan kontainer IOC (Inversion of Control) yang kuat dan menyediakan mekanisme dependency injection. Ini membantu dalam manajemen dan penyusunan objek aplikasi dengan lebih baik.
8. **Eloquent Relationships:**
Eloquent menyederhanakan hubungan antar-model, termasuk one-to-one, one-to-many, many-to-many, dan sebagainya. Ini memungkinkan pengembang dengan mudah mengambil dan menyimpan data yang berkaitan.
9. **Sistem Cache dan Session:**
Laravel menyediakan sistem caching untuk meningkatkan kinerja aplikasi dan sistem manajemen sesi untuk menyimpan data pengguna di antara permintaan HTTP.
10. **Eloquent Factory:**
Eloquent Factory digunakan untuk membuat data uji dengan mudah dan cepat. Ini membantu dalam pengujian dan pengembangan aplikasi.
11. **Task Scheduling dan Job Queue:**
Laravel menyediakan fitur task scheduling untuk menjadwalkan tugas yang akan dijalankan secara otomatis. Job Queue membantu dalam menangani tugas latar belakang yang memakan waktu.
12. **Authentication dan Authorization:**
Laravel menyediakan sistem otentikasi yang terintegrasi dengan mudah. Pengembang dapat dengan cepat mengimplementasikan otentikasi dan otorisasi pengguna.
13. **Testing:**
Laravel mendukung pengujian dengan fitur pengujian bawaan. Dengan dukungan PHPUnit, pengembang dapat melakukan pengujian unit dan fungsional aplikasi.
14. **Environment Configuration:**
Laravel menyediakan file konfigurasi yang mudah diatur untuk berbagai lingkungan, termasuk konfigurasi basis data, cache, mail, dan lainnya.
15. **Laravel Mix:**
Laravel Mix adalah alat untuk kompilasi dan manajemen sumber daya web seperti CSS dan JavaScript. Ini memanfaatkan Webpack untuk mengoptimalkan aset dan memberikan pengembang kemudahan dalam pengelolaan.

Dengan berbagai fitur ini, Laravel memungkinkan pengembang untuk membangun aplikasi web dengan cepat, efisien, dan dengan kode yang bersih dan terstruktur. Itu juga terus berkembang dengan pembaruan dan ekosistem komunitas yang kuat.

4.2 Referensi

<https://laravel.com/docs/10.x/readme>

https://www.youtube.com/watch?v=wRxxUSAkGZ4&list=PLnrs9DcLyeJTG-_mjD68Gn0sC5hbzaU2T

4.3 Link github kalian

<https://github.com/Ananadareva/Praktikum-Web/tree/Modul-10/THT10>