# Laporan Praktikum Web 12

# Forum Rumpi!


**Disusun oleh:**

105221025 Reva Ananda

**Program Studi Ilmu Komputer**

**Universitas Pertamina**

**2023**

Daftar Isi



# 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 tanggug 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 mengigat masih sedikitnya pengalaman saya sebagai pengembang web dan keterbatsan sumber daya. Perpesanan pada forum ini akan berbasis text dan mungkin juga gambar dan video sebagai pelengkap.

# Bab 2 Metode Pelaksanan

## 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 10, 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 menyoimpan 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

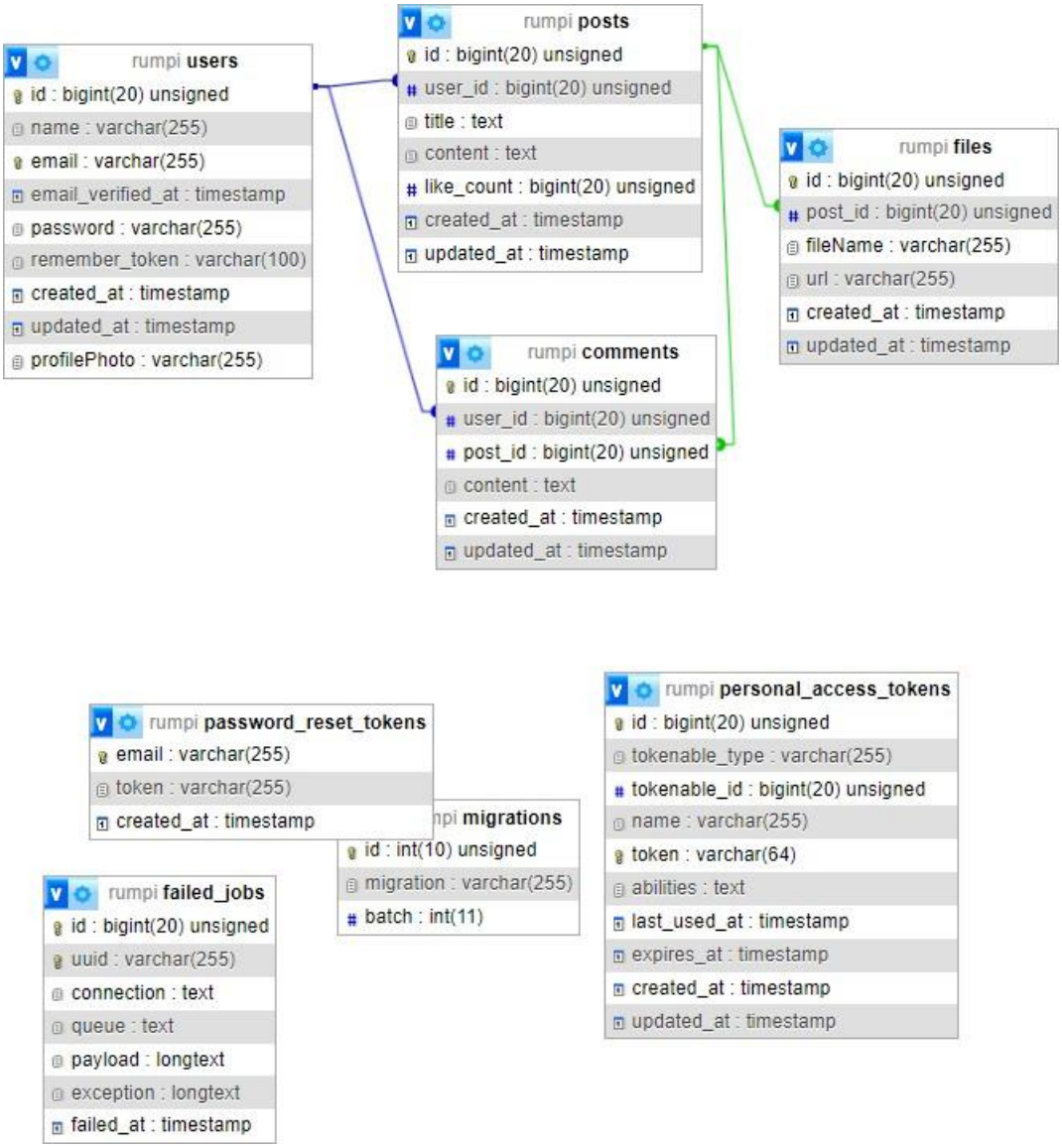
Pada Laravel, disediakan migrasi untuk memanipulasi database mysql.
Migrasi pada Laravel adalah suatu konsep yang memungkinkan untuk mengelola skema database secara terstruktur dan terdokumentasi menggunakan kode PHP daripada menggunakan SQL langsung. Migrasi memungkinkan membuat, mengubah, dan menghapus 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 menjalan migrasi maka bisa menggunakan command php artisan migrate dan php artisan migration:rollback untuk menarik migrasi terkahir.

Contoh migaris yang digunakan untuk membuat table Posts neserta kolom, tipe data dan relasinya:

```php
<?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 mengambarkan skema dari database .

### 3.1.2.1 Model User (User.php)

```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

Controllerpada 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 Illuminate\Support\Facades\Auth;
```

```php
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);



    $userId = Auth::check() ? Auth::user()->id : 0;

    $usernameLogin = Auth::check() ? Auth::user()->name : 0;









    //ambil data user

    /*    dd($userId); */
    /*    dd($postDatas);
     foreach ($postDatas as $post) {
        // Akses user untuk setiap post
        dd($post->user);
    } */

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

  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
<?php

namespace App\Http\Controllers;

use App\Models\Post;

use Illuminate\Http\Request;
use App\Models\User;
use Illuminate\Support\Facades\Auth;


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);



    $userId = Auth::check() ? Auth::user()->id : 0;


        return view('commentView', compact('postComments', 'post', 'userId'));
    }
}
```

### 3.1.3.3 Profil Controller (ProfileController.php)

Controller mempunyai fungsi-fungsi untuk menampilkan dan mengedit profile dari user. Selain itu saya juga menggunakan API untuk mendapatkan list negara diseluruh dunia. Untuk menggunakan API saya terlebih daahulu menginstal  Guzzle untuk membuat permintaan HTTP GET, POST dll.

```php
<?php

namespace App\Http\Controllers;

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

use GuzzleHttp\Client;
use Illuminate\Support\Facades\Storage;

class ProfileController extends Controller
{
```

```php
public function showProfile()
{ /*   dd($userID = auth()->user()->id); */

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

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


    $jenisKelaminAwal = Profile::where('user_id', $userID)->value('jenis_kelamin');

    /*   dd($jenisKelaminAwal); */
    // Narik Api Countries dan city
    $client = new Client();

    try {

        // Buat GET request ke APi
        $response = $client->get('https://countriesnow.space/api/v0.1/countries/population/cities');

        // Decode data Json
        $data = json_decode($response->getBody(), true);

        //Ubah struktur jadi array negara dan kota-kotanya
        /*   $negaraDanKotanya = [];

        foreach ($data['data'] as $cityData) {
            $cityName = $cityData['city'];
            $countryName = $cityData['country'];

            //Jika negara tidak ada pada data
            if (!isset($negaraDanKotanya[$countryName])) {
                $negaraDanKotanya[$countryName] = [];
            }

            // Tambahkan kota ke negara masing-masing
            $negaraDanKotanya[$countryName][] = $cityName;
        }

        return view('profileView', compact('negaraDanKotanya', 'dataUser', 'jenisKelaminAwal')); */

        //  array untuk nyimpen nama-nama negara
        $negaraList = [];

        foreach ($data['data'] as $cityData) {
            $countryName = $cityData['country'];

            // Jika negara belum ada dalam list, tambahkan ke list
            if (!in_array($countryName, $negaraList)) {
                $negaraList[] = $countryName;
            }
        }

        /* dd($negaraList); */

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

```
        } catch (\Exception $e) {

            return response()->json(['error' => $e->getMessage()], 500);
        }


        /*  return view('profileView', compact('dataUser')); */
    }


    public function updateProfile(Request $request, $idLogin)
    {


        // cek gambar diaupdate apa tidak
        if ($request->hasFile('profileUpdate')) {
            $url = 'post_images/' . $request->profileUpdate->getClientOriginalName();


            Storage::putFileAs('public', $request->profileUpdate, $url);

            Profile::where('user_id', $idLogin)->update([
                'jenis_kelamin' => $request->jenis_kelamin,
                'negara' => $request->negara,
                'tanggal_lahir' => $request->tanggl_lahir,
                'biografi' => $request->biografi,
                'url' => $url,
            ]);
        } else {
            Profile::where('user_id', $idLogin)->update([
                'jenis_kelamin' => $request->jenis_kelamin,
                'negara' => $request->negara,
                'tanggal_lahir' => $request->tanggl_lahir,
                'biografi' => $request->biografi,
            ]);
        }









        return redirect()->route('profile.show', $idLogin)->with('success', 'Profile Berhasil Diupdate');
    }
}
```

### 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 tampila 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>


            @if ($userId == 0)
            <li>   <a href="{{ route('login') }}">Pengunjung</a> </li>
            @else
               <li><a href="{{ route('profile.show', ['idLogin' => $userId]) }}">Profil</a></li>
            @endif



         </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">

                @if ($userId == 0)
                    <a href="{{ route('login') }}">Pengunjung</a>
                @else
                  <a href="{{ route('profile.show', ['idLogin' => $userId]) }}">{{$usernameLogin}}</a>
                @endif


            </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">




    @if ($userId == 0)
        <a href="{{ route('login') }}">Login untuk <br>membuat Pos</a>
    @else
        <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>
    @endif









</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 && $post->user->profile->url)
                <img id="imageProfile" src="{{ asset($post->user->profile->url) }}">
            @else
                <img id="imageProfile" src="{{ asset('profile_images/default_profile.webp') }}">
            @endif



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

    </div>
</div>

<div class="post-content">


    <div class="flex-content-center">
        <div class="w3-content w3-display-container" style="">

            @php
                $imageCount = $post->files->where('url', '!=', null)->count(); /* itung gambar */
            @endphp

            @foreach ($post->files->where('url', '!=', null)->take(1) as $file)
                {{-- Ambil gambar index pertama --}}
                <img class="mySlides" src="{{ asset($file->url) }}"
                    style="width:400px; height:400px; object-fit: contain; border:solid 2px;"> <br>
                <a href="{{ route('comment.show', ['idPost' => $post->id]) }}">Lihat
                    {{ $imageCount }} lainnya</a>
            @endforeach




        </div>
    </div>









    <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> Komentar
                    </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 pos 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>
        <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>
    <!-- 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">

        <div class="flex-content-center">
            <div class="w3-content w3-display-container" style="">
                @foreach ($post->files as $file)
                    @if ($file->url != null)
                        <img class="mySlides" src="{{ asset($file->url) }}"
                            style="width:400px; height:400px; object-fit: contain; border:solid 2px;">
                    @endif
                @endforeach

                <button class="w3-button w3-black w3-display-left" onclick="plusDivs(-1)">&#10094;</button>
                <button class="w3-button w3-black w3-display-right" onclick="plusDivs(1)">&#10095;</button>
            </div>
        </div>



        <script>
            var slideIndex = 1;
            showDivs(slideIndex);

            function plusDivs(n) {
                showDivs(slideIndex += n);
            }

            function showDivs(n) {
                var i;
                var x = document.getElementsByClassName("mySlides");
                if (n > x.length) {
                    slideIndex = 1
                }
                if (n < 1) {
                    slideIndex = x.length
                }
                for (i = 0; i < x.length; i++) {
```

```
                x[i].style.display = "none";
            }
            x[slideIndex - 1].style.display = "block";
        }
    </script>


    <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)
          <img id="imageProfile" src="{{ asset('profile_images/default_profile.webp') }}">
        @else
          <img id="imageProfile" src="{{ $comment->user->profilePhoto }}">
        @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>
  @endforeach


  <h2>Tambahkan Komentar</h2>
  <div class="comment-form">
    <form action="comment.php" method="post" id="comment-form">




      @if ($userId == 0)
        <a href="{{ route('login') }}">Login Untuk Menambahkan Komentar</a>
      @else
        <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>
      @endif
```

```
            </form>
        </div>

    </section>

    </section>

  </main>

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

</body>

</html>
```

#### 3.1.4.3 View Profile (profileView.balde.php)

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>


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

<section class="profil-section">


    <div class="profil-left">
        <form action="{{ route('profile.update', ['idLogin' => Auth::user()->id]) }}" method="post"
            enctype="multipart/form-data">
            @csrf
            <div class="profil-image">

                @error('profileUpdate')
                    <div class="alert alert-danger">{{ $message }}</div>
                @enderror

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

                {{-- Pakai Old Agar saat nggak diedit tetap ada value yang lama (tida k null) --}}
                @if ($dataUser->profile->url == null)
                    <img id="imageProfile" src="{{ asset('profile_images/default_profile.webp') }}"
                        alt="profile">
                @else
                    <img id="imageProfile" src="{{ asset($dataUser->profile->url) }}" alt="profile">
                @endif

                <label for="profileUpdate" style="position: relative; display: inline-block; cursor: pointer;">
                    <input accept="image/png, image/jpeg, image/jpeg, image/svg" type="file"
                        name="profileUpdate" id="profileUpdate" style="display: none;">
                    <i class="fas fa-camera fa-lg"
                        style="color: black; position: absolute; top: 0; left: 0;"></i>
                </label>


                <br>
                Update gambar profil

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

```
            <div class="post-info">
                <h3><span>{{ $dataUser->name }}</span></h3>

                <textarea cols="50" rows="5" id="biography" required name="biografi">{{ old('biografi', $dataUser->profile->biografi) }}</textarea>
                @error('biografi')
                    <div class="alert alert-danger">{{ $message }}</div>
                @enderror
            </div>

        </div>

        <div class="profil-right">
            <div class="proflie-data">
                <h3>Informasi Pribadi</h3>

                <ul>
                    <li>
                        <strong>Email</strong><br>
                        <!-- Use old() to populate the input with the previous value -->
                        <input required type="text" name="email" value="{{ old('email', $dataUser->email) }}">
                        <hr>
                    </li>
                    <li>
                        <strong>Jenis Kelamin</strong><br>
                        <select name="jenis_kelamin" required>
                            <option value="L"
                                {{ old('jenis_kelamin', $jenisKelaminAwal) == 'L' ? 'selected' : '' }}>L</option>
                            <option value="P"
                                {{ old('jenis_kelamin', $jenisKelaminAwal) == 'P' ? 'selected' : '' }}>P</option>
                            <option value="T"
                                {{ old('jenis_kelamin', $jenisKelaminAwal) == 'T' ? 'selected' : '' }}>T</option>
                        </select>
                        <hr>
                    </li>
                    <li>
                        <strong>Tanggal Lahir</strong><br>
                        <!-- Use old() to populate the input with the previous value -->
                        <input required type="date" name="tanggl_lahir"
                            value="{{ old('tanggl_lahir', $dataUser->profile->tanggal_lahir) }}">
                        <hr>
                    </li>
                    <li>
                        <strong>Negara</strong><br>
                        <select name="negara" required>
                            @foreach ($negaraList as $negara)
                                <option value="{{ $negara }}"
                                    {{ old('negara', $dataUser->profile->negara) == $negara ? 'selected' : '' }}>
                                    {{ $negara }}
                                </option>
                            @endforeach
                        </select>
                        <hr>
                    </li>
                </ul>
```

```
            <button style="background: blue; color: white;" type="submit" onclick="confirmAlert()">
                Edit Profile
            </button>

            <script>
                function confirmAlert() {
                    alert("Anda Yakin ingin Menyimpan Perubahan?");
                }
            </script>
        </div>
    </div>
    </form>
</section>



<div class="h2-position">

    <h2> Daftar Pos </h2>

</div>


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




@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>
        <br>



        <div class="post-content">
```

```
            {{--   @dd($post->files) --}}
            <div class="flex-content-center">

                @php
                    $imageCount = $post->files->where('url', '!=', null)->count(); /* itung gambar */
                @endphp

                @foreach ($post->files->where('url', '!=', null)->take(1) as $file)
                    {{-- Ambil gambar index pertama --}}
                    <img class="mySlides" src="{{ asset($file->url) }}"
                        style="width:400px; height:400px; object-fit: contain; border:solid 2px;">
                    <br> <br>
                    <a href="{{ route('comment.show', ['idPost' => $post->id]) }}">Lihat
                        {{ $imageCount }} lainnya</a>
                @endforeach

            </div>


            <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,
```

```css
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;
}

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

label[for="selected_files[]"] {
    margin-top: 15px;
    display: block;
}

div {
    margin-bottom: 10px;
}

img {
    max-width: 100px;
    max-height: 100px;
```

```
            margin-right: 10px;
            margin-bottom: 5px;
        }

        .btn-primary {
            background-color: #333;
            color: #fff;
            padding: 10px 15px;
            border: none;
            cursor: pointer;
        }

        .btn-primary:hover {
            background-color: #0056b3;
        }
    </style>
</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>

@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 }}">
                    {{ $file->fileName }} <br>
                    <img src="{{ asset($file->url) }}" alt="File Image"
```

```
                        style="max-width: 100px; max-height: 100px;">
                    </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
<?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/home', [PostController::class, 'index'])->name('home');
Route::get('/rumpi.com/{idPost}/comment}', [CommentController::class, 'showComment'])->name('comment.show');



Route::get('/rumpi.com//login', [AuthController::class, 'login'])->name('login');
Route::post('/rumpi.com//login', [AuthController::class, 'authenticating']);


Route::group(['prefix' => 'rumpi.com', 'middleware' => ['auth']], function () {


    Route::get('/logout', [AuthController::class, 'logout']);


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

    Route::get('/rumpi.com/{idLogin}', [ProfileController::class, 'showProfile'])->name('profile.show');
    Route::post('/rumpi.com/{idLogin}', [ProfileController::class, 'updateProfile'])->name('profile.update');
});
```

## 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 unutk 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

```
body {
    display: flex;
    flex-direction: column;
    min-height: 100vh;
    margin: 0;
    padding: 0;
    transition: background-color 0.5s, color 0.5s;
}


.dark-mode h1 {

    color: white;
}

#modal1 {
    display: none;
    position: fixed;
    z-index: 1;
    padding-top: 20px;
    padding-bottom: 20px;
    right: 0;
    top: 0;
    width: 300px;
    height: 100%;
    overflow: auto;
    background-color: #333;
    border: 1px solid white;
    right: 5px;
    bottom: 5px;
}

.modal-content {
    background-color: #fefefe;
    margin: auto;
    padding: 10px;
    border: 1px solid #888;
    width: 80%;
    height: 90%;
}

.close-button {
    color: #aaaaaa;
    float: right;
    font-size: 20px;
    font-weight: bold;
}

.modal-button {
    background-color: #333 !important;
    color: silver;
    float: right;
```

```
    border: none;
    font-size: 20px;
    font-weight: bold;
}

.dark-mode .modal {
    background-color: black;
    color: white;
}

.dark-mode .modal-content {
    background-color: black;
    border: 1px solid white;
    color: white;
}



.dark-mode .modal-button {
    background-color: black !important;
    color: white;
}

main {
    flex: 1;
    background-color: silver;
}

.dark-mode main {
    background-color: #333;
}

header {
    background-color: #333;
    color: #fff;
    text-align: center;
    padding: 10px;
}

.dark-mode header {
    background-color: black;
}

nav ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
}

nav li {
    display: inline;
    margin-right: 20px;
}

nav a {
```

```
    text-decoration: none;
    color: #fff;
}

footer {
    background-color: #333;
    color: #fff;
    text-align: center;
    padding: 10px;
}

.dark-mode footer {
    background-color: black;
}


/* pagination.css */


.pagination {
    display: flex;
    justify-content: center;
    display: flex;
    list-style: none;
    padding: 0;
    margin: 20px 0;
}

.pagination li {
    margin-right: 10px;
}

.pagination a,
.pagination span {
    padding: 8px 12px;
    display: inline-block;
    text-decoration: none;
    color: #3490dc;
    background-color: #fff;
    border: 1px solid #d2d6dc;
    border-radius: 4px;
    transition: background-color 0.3s;
}

.pagination .active a {
    background-color: #3490dc;
    color: #fff;
}

.pagination a:hover {
    background-color: #f0f4f8;
}

.alert {
    padding: 15px;
```

```
    margin-bottom: 20px;
    border: 1px solid transparent;
    border-radius: 4px;
    text-align: center;
}

.alert-success {
    color: #3c763d;
    background-color: #dff0d8;
    border-color: #d6e9c6;
}
```

### 3.1.2 Stylesheet login.css

```
main {
    padding: 1rem;
}

.login-box {
    max-width: 300px;
    margin: 0 auto;
    background: #333;
    color: #fff;
    padding: 1.5rem;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

.login-box h2 {
    text-align: center;
    color: #fff;
}

label {
    display: block;
    margin-bottom: 0.5rem;
}

input {
    width: 100%;
    padding: 0.3rem;
    margin-bottom: 1rem;
    border: 1px solid #ccc;
    border-radius: 3px;
}

input[type="submit"] {
    background-color: #555;
    color: #fff;
    border: none;
    padding: 0.5rem 1rem;
    cursor: pointer;
}
```

### 3.2.3 Stylesheet home.css

```
/* Light Mode */
#createPost {
    text-align: center;
    margin: auto;
    padding: 10px;
    border: solid 1px black;
    border-radius: 10px;
    max-width: 150px;
    background-color: white;
}

#imageProfile {
    width: 45px;
    height: 45px;
    border-radius: 50%;
}

#imagePost {
    max-width: 400px;
    max-height: 300px;
    border-radius: 10px;
    display: block;
    margin: 0 auto;
}

.post-section {
    background-color: white;
    padding: 20px;
    border-radius: 5px;
    width: 500px;
    margin: 0 auto;
    display: flex;
    margin-bottom: 20px;
    flex-direction: column;
}

.comments-section {
    background-color: black;
    padding: 20px;
    border-radius: 5px;
}

.user-info {
    display: flex;
    gap: 10px;
}

.post-header {
    display: flex;
}
```

```css
.comment {
    border: 1px solid #ccc;
    padding: 10px;
    margin: 10px 0;
    border-radius: 5px;
    max-width: 100%;
}

.close {
    color: #aaaaaa;
    float: right;
    font-size: 28px;
    font-weight: bold;
}

/* Dark Mode */
.dark-mode #createPost {
    background-color: black;
    color: white;
}

.dark-mode #imageProfile {
    border: 2px solid white;
}

.dark-mode #imagePost {
    border: 2px solid white;
}

.dark-mode .post-section {
    background-color: black;
    color: white;
    border: 1px solid white;
}

.dark-mode .comments-section {
    background-color: white;
    color: black;
    border: 1px solid black;
}

.dark-mode .comment {
    border: 1px solid white;
}

.flex-content-center{
    display: flex;
    justify-content: center;
    align-items: center;
}
```

### 3.2.4 Stylesheet createPost.css

```
/* Light mode styles */
.create-post-container {
    max-width: 800px;
    margin: 20px auto;
    padding: 20px;
    background-color: #fff;
    border-radius: 5px;
    box-shadow: 0 0 5px rgba(0, 0, 0, 0.2);
}

.create-post-container .grid-container {
    display: grid;
    grid-template-columns: 1fr 1fr;
    grid-template-rows:  1fr;
    grid-gap: 20px;
}

.create-post-container .grid-item {
    width: 100%;
}

.create-post-container textarea {
    width: 100%;
    margin-bottom: 10px;
    padding: 10px;
    border: 1px solid #ccc;
    border-radius: 5px;
}

.file-inputs {
    margin-bottom: 10px;
}

.preview-container {
    display: flex;
    flex-wrap: wrap;
}

.preview-item {
    margin-right: 10px;
    margin-bottom: 10px;
}

.preview-container {
    text-align: center;
}

.preview-item img,
.preview-item video {
    max-width: 100px;
    max-height: 100px;
    border-radius: 5px;
}
```

```css
/* Dark mode styles */
.dark-mode .create-post-container {
    background-color: black;
    color: #fff;
    box-shadow: 0 0 5px rgba(255, 255, 255, 0.2);
}

.dark-mode .create-post-container textarea {
    border: 1px solid #999;
    background-color: black;
    color: #fff;
}

.dark-mode .preview-item img,
.dark-mode .preview-item video {
    border: 1px solid #666;
}

.form-group {
    margin-bottom: 10px;
}

.removeFileInput {
    margin-left: 10px;
}
```

### 3.2.3 Stylesheet Comment.css

```
/* Light Mode */
.comments-section {
    display: grid;
    grid-template-columns: 1fr;
    grid-gap: 20px;
    background-color: #fff;
    padding: 20px;
    border-radius: 5px;
    max-width: 500px;
    margin: 10px auto;
}

.comment-form {
    display: flex;
    flex-direction: column;
}

.comment-form label {
    margin-bottom: 5px;
    font-weight: bold;
}

.comment-form textarea {
    width: 95%;
    padding: 10px;
    margin-bottom: 10px;

    border: 1px solid #ccc;
    border-radius: 5px;
}

.comment-form button {
    background-color: #333;
    color: #fff;
    padding: 10px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
}


.comment {
    border: 1px solid black;
    padding: 20px;
    border-radius: 5px;
}

.comment-header {
    display: flex;
    align-items: center;
}

#imageProfile {
```

```
    width: 40px;
    height: 40px;
    margin-right: 10px;
    border-radius: 50%;
}

.comment-content {
    margin-top: 10px;
}

/* Dark Mode */
.dark-mode .comments-section {
    background-color: black;
    color: white;
}

.dark-mode .comment {
    border: 1px solid white;
}

.dark-mode #imageProfile {
    border: 2px solid white;
}
```

### 3.2.3 Stylesheet profil.css

```css
#createPost {
    text-align: center;
    margin: auto;
    padding: 10px;
    border: solid 1px black;
    border-radius: 10px;
    max-width: 150px;
    background-color: white;
}


.post-section {
    background-color: white;
    padding: 20px;
    border-radius: 5px;
    width: 500px;
    margin: 0 auto;
    display: flex;
    margin-bottom: 20px;
    flex-direction: column;
}

.button-container {
    display: flex;
    justify-content: flex-end;
    margin-bottom: 10px;
}

.button-container button {
    margin-left: 10px;
}


.profil-section {
    display: grid;
    grid-template-columns: 1fr 1fr;
    grid-gap: 30px;
    background-color: #fff;
    padding: 20px;
    border-radius: 5px;
    max-width: 800px;
    margin: 20px auto;
    box-shadow: 0 0 5px rgba(0, 0, 0, 0.2);
}

.profil-left {
    max-width: calc(50% - 15px);
}

.profil-right {
    max-width: calc(50% - 15px);
}
```

```css
#biography {
    margin: 10px;
    padding: 10px;
    border: solid 1px black;
    border-radius: 10px;
    max-width: 400px;
    background-color: white;
    margin: 0 auto;
    text-align: center;
}

.proproflie-data {
    display: grid;
    grid-template-columns: 1fr;
    grid-gap: 10px;
}

.button-edit-lihatPost {
    text-align: center;
    padding: 10px;
    background-color: gray;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    color: #fff;
}

.h2-position {
    text-align: center;
    margin: 5px auto;

    background-color: #333;
    color: #fff;
    cursor: pointer;
}



#createPost {
    text-align: center;
    margin: auto;
    padding: 10px;
    border: solid 1px black;
    border-radius: 10px;
    max-width: 150px;
    background-color: white;
}
```

```css
#imageProfile {
    border-radius: 50%;
    border: 5px solid transparent;
    width: 150px;
    height: 150px;
    margin: 0 auto;
    display: block;
}

#biography {
    margin: 10px;
    padding: 10px;
    border: solid 1px black;
    border-radius: 10px;
    max-width: 400px;
    background-color: white;
    margin: 0 auto;
    text-align: center;
}

.proproflie-data {
    display: grid;
    grid-template-columns: 1fr;
    grid-gap: 10px;
}




/* Dark Mode */
.dark-mode #createPost {
    background-color: black;
    color: white;
}

.dark-mode .profil-section {
    background-color: black;
    color: white;
    box-shadow: 0 0 5px rgba(255, 255, 255, 0.2);
}

.dark-mode #img-profil {
    border-color: white;
}

.dark-mode #biography {
    background-color: #333;
    color: white;
    border: solid 1px white;
}

.dark-mode .proprofile-data {
    color: white;
}
```

```
.flex-content-center{
    display: flex;
    justify-content: center;
    align-items: center;
}
```

I

## 3.2 Kode Javascript

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


        <!-- 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>


        <script src="js/modal.js"></script>
```

Kode script javascript digunakan untuk mengimplementasikan fitur menu pengaturan menggunakan sidebar pada setiap halaman web dibuat menggunakan funsgi-fungsi unutk 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.

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

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

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

### 3.2.1 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.

```
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 }
```

### 3.2.2 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().

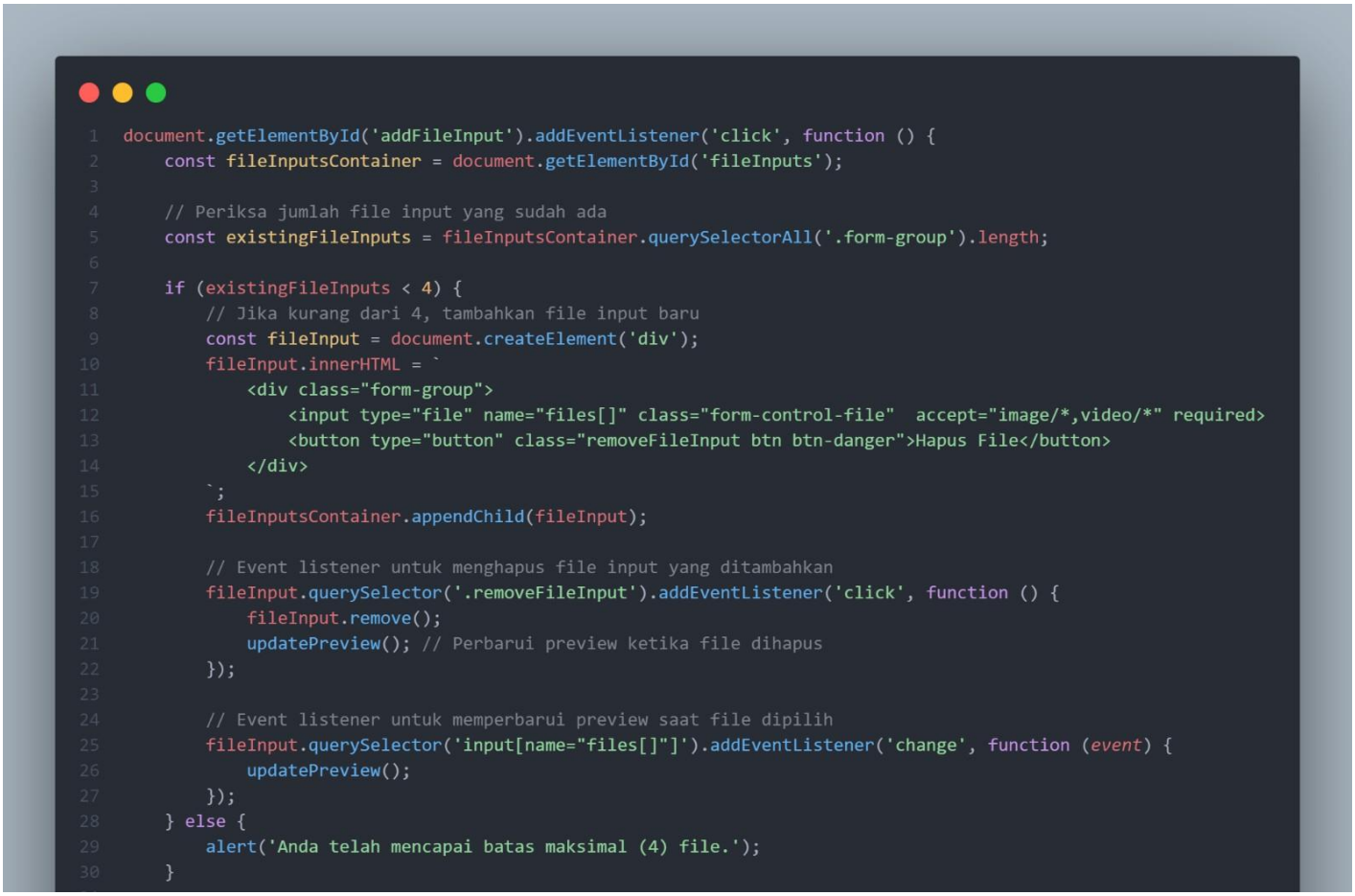
```
function darkMode() {
    var element = document.body;
    element.classList.toggle("dark-mode");

    // chck apakah dark mode aktif
    var isDarkMode = element.classList.contains("dark-mode");
    localStorage.setItem("darkMode", isDarkMode);
}
```

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.

```

function applyDarkModePreference() {
    var isDarkMode = localStorage.getItem("darkMode") === "true";
    var element = document.body;

    // simpan informasi darkmode
    if (isDarkMode) {
        element.classList.add("dark-mode");
    } else {
        element.classList.remove("dark-mode");
    }
}

// panggil funsgi dark mode yang talh disimpan
applyDarkModePreference();
```

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

### 3.2.3 Script Fitur Menambahkan Gambar Dan Video postingan serta tampilan PreviewaddFile.js



```
document.getElementById('addFileInput').addEventListener('click', function () {
    const fileInputsContainer = document.getElementById('fileInputs');

    // Periksa jumlah file input yang sudah ada
    const existingFileInputs = fileInputsContainer.querySelectorAll('.form-group').length;

    if (existingFileInputs < 4) {
        // Jika kurang dari 4, tambahkan file input baru
        const fileInput = document.createElement('div');
        fileInput.innerHTML = `
            <div class="form-group">
                <input type="file" name="files[]" class="form-control-file"  accept="image/*,video/*" required>
                <button type="button" class="removeFileInput btn btn-danger">Hapus File</button>
            </div>
        `;
        fileInputsContainer.appendChild(fileInput);

        // Event listener untuk menghapus file input yang ditambahkan
        fileInput.querySelector('.removeFileInput').addEventListener('click', function () {
            fileInput.remove();
            updatePreview(); // Perbarui preview ketika file dihapus
        });

        // Event listener untuk memperbarui preview saat file dipilih
        fileInput.querySelector('input[name="files[]"]').addEventListener('change', function (event) {
            updatePreview();
        });
    } else {
        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.

```
    // Fungsi untuk memperbarui preview file
    function updatePreview() {
        const previewContainer = document.getElementById('previewContainer');
        previewContainer.innerHTML = '';

        const allFileInputs = document.querySelectorAll('input[name="files[]"]');
        allFileInputs.forEach(input => {
            Array.from(input.files).forEach(file => {
                const previewItem = document.createElement('div');
                previewItem.classList.add('preview-item');

                if (file.type.startsWith('image/')) {
                    const img = document.createElement('img');
                    img.src = URL.createObjectURL(file);
                    previewItem.appendChild(img);
                } else if (file.type.startsWith('video/')) {
                    const video = document.createElement('video');
                    video.src = URL.createObjectURL(file);
                    video.setAttribute('controls', 'controls');
                    previewItem.appendChild(video);
                }

                previewContainer.appendChild(previewItem);
            });
        });
    }
});
```

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 <img>, dan jika file adalah video, maka elemen pratinjau berupa tag <video>.

```
    // Fungsi untuk memperbarui preview file
    function updatePreview() {
        const previewContainer = document.getElementById('previewContainer');
        previewContainer.innerHTML = '';

        const allFileInputs = document.querySelectorAll('input[name="files[]"]');
        allFileInputs.forEach(input => {
            Array.from(input.files).forEach(file => {
                const previewItem = document.createElement('div');
                previewItem.classList.add('preview-item');

                if (file.type.startsWith('image/')) {
                    const img = document.createElement('img');
                    img.src = URL.createObjectURL(file);
                    previewItem.appendChild(img);
                } else if (file.type.startsWith('video/')) {
                    const video = document.createElement('video');
                    video.src = URL.createObjectURL(file);
                    video.setAttribute('controls', 'controls');
                    previewItem.appendChild(video);
                }

                previewContainer.appendChild(previewItem);
            });
        });
    }
});

document.addEventListener('click', function (event) {
    if (event.target && event.target.className === 'removeFileInput btn btn-danger') {
        event.target.parentElement.remove();
        updatePreview(); // Perbarui preview ketika file dihapus
    }
});
```

```
document.addEventListener('click', function (event) {
    if (event.target && event.target.className === 'removeFileInput btn btn-danger') {
        event.target.parentElement.remove();
        updatePreview(); // Perbarui preview ketika file dihapus
    }
});
```

### 3.2.4 Script Fitur Preview Gambar Profile saat Edit . imageProfile.js

```
document.getElementById('profileUpdate').addEventListener('change', function(event) {
    var input = event.target;
    var reader = new FileReader();

    reader.onload = function() {
        var imageProfile = document.getElementById('imageProfile');
        imageProfile.src = reader.result;
    };

    if (input.files && input.files[0]) {
        reader.readAsDataURL(input.files[0]);
    }
});
```

skrip tersebut memungkinkan antarmuka pengguna untuk memberikan umpan balik visual secara langsung ketika pengguna memilih file baru, mengupdate tampilan gambar profil tanpa memerlukan penyegaran halaman. Ini merupakan teknik yang umum digunakan dalam pengembangan antarmuka pengguna yang responsif dan interaktif.

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 API

## 4.1 Routes API

Routes API pada Laravel adalah mekanisme untuk menentukan endpoint atau titik akhir yang dapat diakses oleh aplikasi klien atau perangkat lunak eksternal untuk berinteraksi dengan sistem Laravel. Dalam konteks pengembangan web, API (Application Programming Interface) digunakan untuk memungkinkan komunikasi dan pertukaran data antara berbagai sistem.

```
Route::get('/countries/population/cities', [ProfileController::class,
'getCountriesPopulation']);
```

## 4.2 Instalasi Guzzle pada Composer

Pada Laravel kita dapat menggunakan Guzzle HTTP Client untuk melakukan permintaan HTTP di Laravel., untuk proses instalasi bisa memasukkan command berikut pada terminal folder project Laravel:
composer require guzzlehttp/guzzle

## 4.3 Request HTTP pada Controller

```
  $client = new Client();

        try {

            // Buat GET request ke APi
            $response = $client-
>get('https://countriesnow.space/api/v0.1/countries/population/cities');

            // Decode data Json
            $data = json_decode($response->getBody(), true);

            //  array untuk nyimpen nama-nama negara
            $negaraList = [];

            foreach ($data['data'] as $cityData) {
                $countryName = $cityData['country'];

                // Jika negara belum ada dalam list, tambahkan ke list
                if (!in_array($countryName, $negaraList)) {
                    $negaraList[] = $countryName;
                }
            }

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

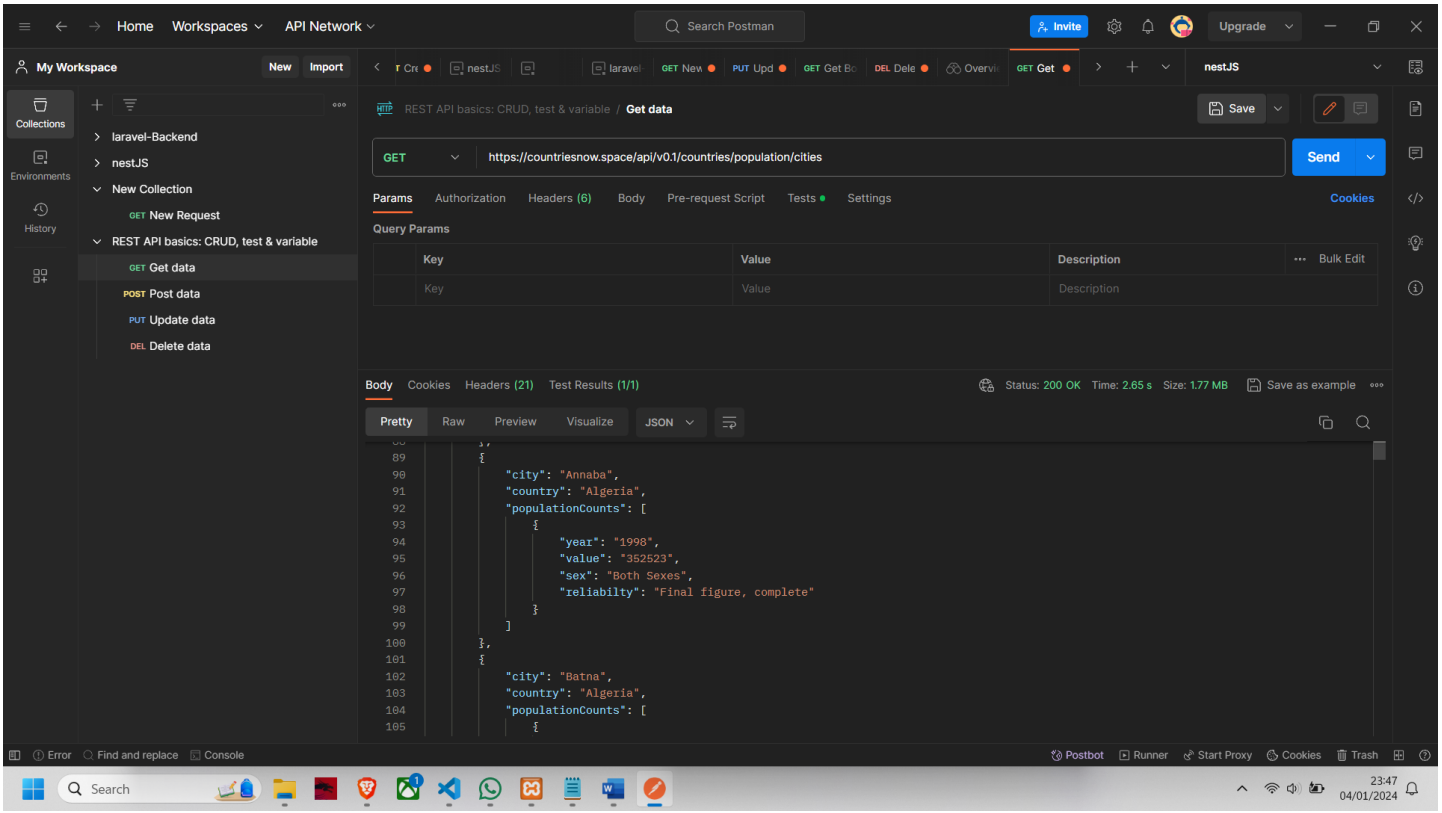
Fungsi menggunakan Guzzle HTTP Client (diasumsikan) untuk membuat GET request ke endpoint API 'https://countriesnow.space/api/v0.1/countries/population/cities'.
Hasil respons dari API di-decode sebagai array PHP menggunakan json_decode.
Penyusunan List Negara:

Fungsi kemudian menciptakan sebuah array kosong bernama $negaraList.
Menggunakan loop foreach, untuk setiap data kota yang diterima dari API, nama negara diekstrak dan ditambahkan ke dalam $negaraList jika negara tersebut belum ada dalam list.
Pengembalian Data ke View:

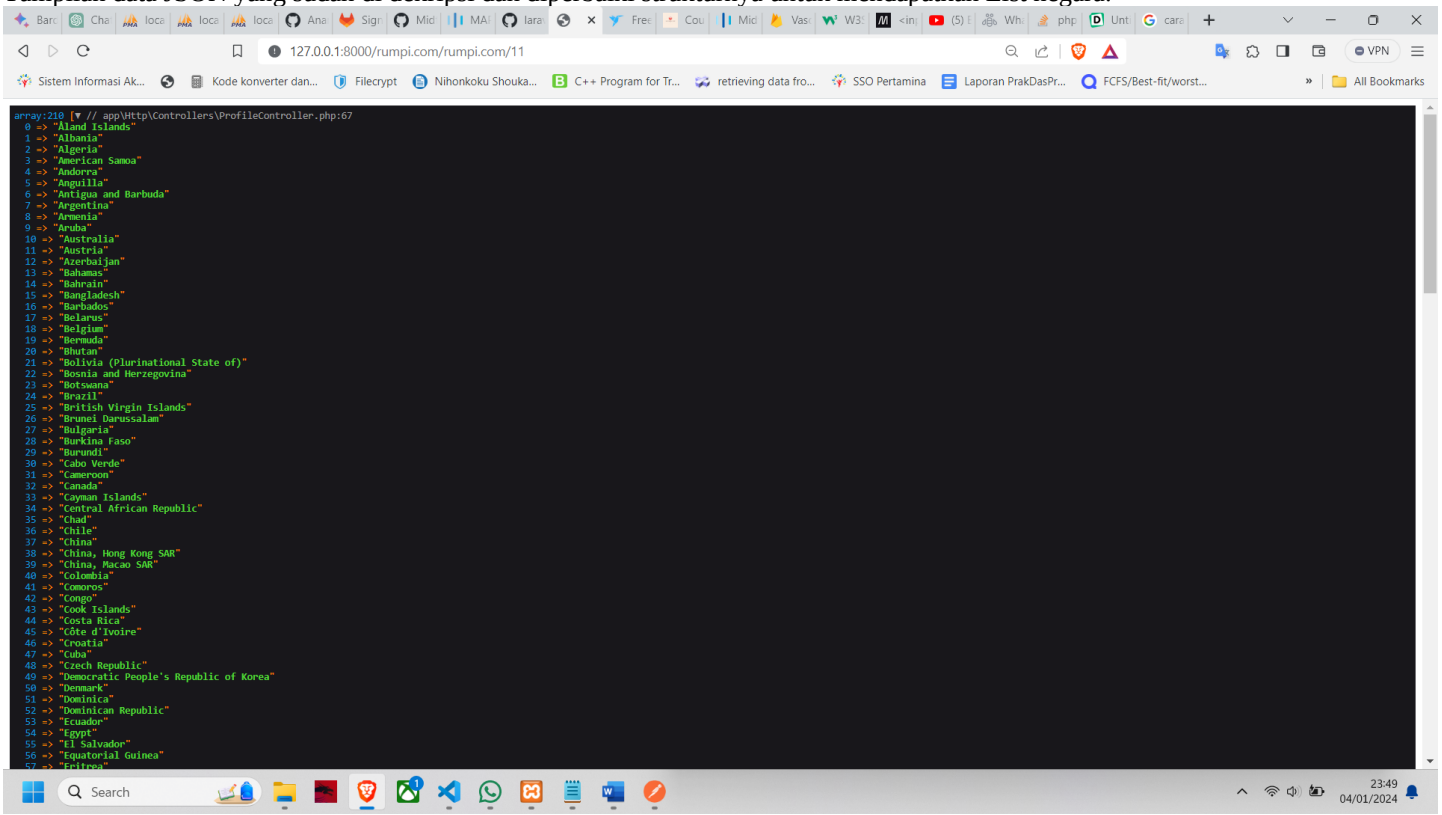
Setelah mendapatkan daftar negara, data pengguna (user), dan jenis kelamin, fungsi mengembalikan view 'profileView' dengan variabel yang menyimpan informasi tersebut.
Penanganan Error:

Dalam blok catch, ada penanganan kesalahan yang memberikan respons JSON jika terjadi kesalahan.

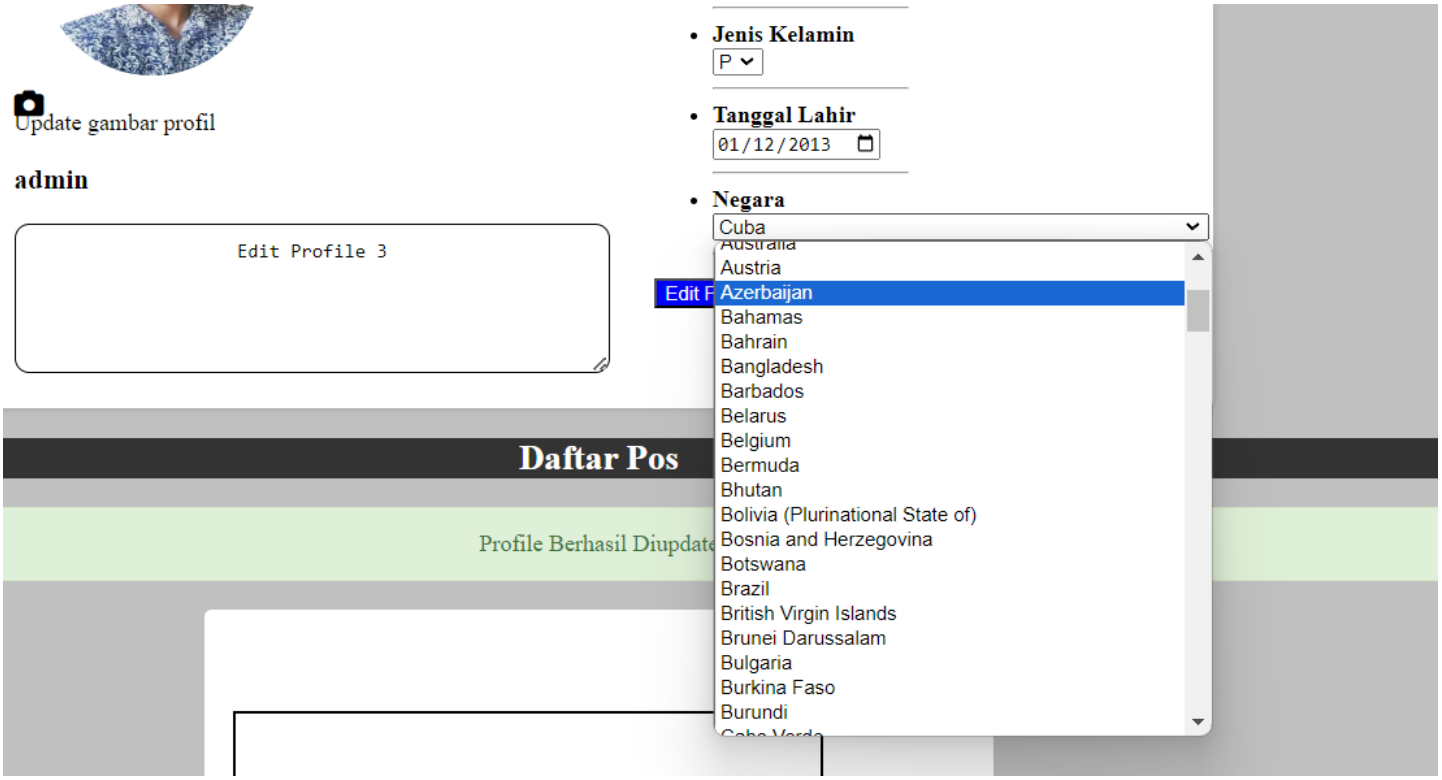
Tampilan data JSON pada HTTP https://countriesnow.space/api/v0.1/countries/population/cities




Tampilan data JSON yang sudah di dekripsi dan diperbaiki strukturnya untuk mendapatkan List negara:




Tampilan data API yang sudahterstruktur pada View Profile




# Bab 5 Penutup

## 5.1 Kesimpulan

Dalam praktikum pemrograman web pada tanggal 29 Desember 2023, kami sebagai mahasiswa Ilmu Komputer telah mempelajari mengenai API serta cara pemakaiannya terutama pada framework Laravel untuk pengembangan web,di bimbing oleh para asisten praktikum selama sesi pemberian materi. Kami juga telah mengimplementasikan pemahaman tersebut dengan mengerjakan tugas postest dan THT yang diberikan setelah sesi praktikum.

API, atau Application Programming Interface, adalah sekumpulan aturan dan protokol yang memungkinkan berbagai perangkat lunak untuk berinteraksi satu sama lain. API menentukan cara bagaimana komponen perangkat lunak harus berkomunikasi, memberikan cara bagi pengembang untuk mengakses dan menggunakan fungsionalitas atau data dari suatu aplikasi atau layanan tanpa harus memahami detail internalnya.

API dapat memiliki berbagai jenis, namun dua jenis umum adalah:

1. Web APIs (API Web): API yang dapat diakses melalui protokol web seperti HTTP. Web API sering digunakan untuk memungkinkan aplikasi atau layanan eksternal berkomunikasi dengan aplikasi atau layanan inti melalui internet. Beberapa contoh umum termasuk RESTful APIs dan GraphQL APIs.

2. Library APIs: API yang disediakan oleh perpustakaan atau framework perangkat lunak tertentu. Dalam hal ini, API memberikan cara bagi pengembang untuk berinteraksi dengan fungsi dan objek yang telah didefinisikan di dalam perpustakaan tersebut.

3. Beberapa poin penting terkait API meliputi:

    - Endpoint: Sebuah titik akhir (URL atau URI) di mana API dapat diakses dan berinteraksi. Endpoint ini mungkin menyediakan akses ke berbagai fungsi atau data.
    - Request: Pengiriman pesan atau permintaan dari klien ke server melalui API. Permintaan ini dapat berupa operasi untuk mendapatkan, menyimpan, atau memperbarui data.
    - Response: Jawaban yang diberikan oleh server setelah menerima permintaan. Response ini biasanya

berisi data yang diminta atau mengkonfirmasi bahwa operasi telah berhasil atau gagal.

4. Format Data: Format data yang digunakan untuk mengirim dan menerima informasi antara klien dan server. Umumnya, format seperti JSON (JavaScript Object Notation) atau XML digunakan.

Guzzle Laravel

Guzzle adalah sebuah library HTTP client yang populer untuk PHP, dan dalam ekosistem Laravel, Guzzle sering digunakan untuk membuat dan mengelola permintaan HTTP ke berbagai API eksternal. Dalam konteks pengembangan web dengan Laravel, terkadang diperlukan komunikasi dengan API pihak ketiga untuk mendapatkan atau mengirimkan data. Guzzle menyediakan antarmuka yang kaya dan mudah digunakan untuk menangani operasi HTTP dengan efisien.

Dalam implementasi Laravel, langkah pertama yang perlu dilakukan adalah menginstal Guzzle menggunakan Composer. Setelah diinstal, kita dapat menggunakannya dalam proyek Laravel dengan mengimpor kelas `Client` dari namespace `GuzzleHttp`. Selanjutnya, kita membuat instance dari `Client` ini untuk membuat permintaan HTTP.

Penggunaan Guzzle dalam Laravel melibatkan berbagai metode HTTP seperti GET, POST, PUT, atau DELETE, tergantung pada jenis operasi yang ingin dilakukan terhadap API. Permintaan dapat disertakan dengan parameter seperti query string, data formulir, atau payload JSON, sesuai dengan kebutuhan spesifik API yang diakses.

Setelah permintaan dikirim, kita dapat mengelola respons yang diterima dari API, termasuk informasi seperti kode status, header, dan isi respons. Guzzle menyediakan metode yang memudahkan ekstraksi dan pengelolaan data ini.

Dalam penanganan kesalahan, sangat penting untuk memasukkan mekanisme manajemen exception. Kesalahan selama komunikasi dengan API, seperti ketidaktersediaan server atau respons yang tidak sesuai, dapat diatasi dan ditangani dengan baik dalam blok `try-catch`.

Dengan menggunakan Guzzle, pengembang Laravel dapat dengan mudah dan efisien mengintegrasikan layanan eksternal dan berkomunikasi dengan berbagai API, memperluas fungsionalitas proyek mereka tanpa harus menulis kembali fungsi HTTP client secara manual.

## 5.2 Referensi

https://laravel.com/docs/10.x/http-client

https://medium.com/@fatoni_ach/akses-api-menggunakan-guzzle-di-laravel-a6e2c64d4ca1

https://www.ibm.com/topics/api
https://countriesnow.space/api/v0.1/countries/population/cities

## 5.3 Link github

https://github.com/Ananadareva/Praktikum-Web/tree/Modul-12