

Multimedia Web (Laravel)

A. Rich Text Editor

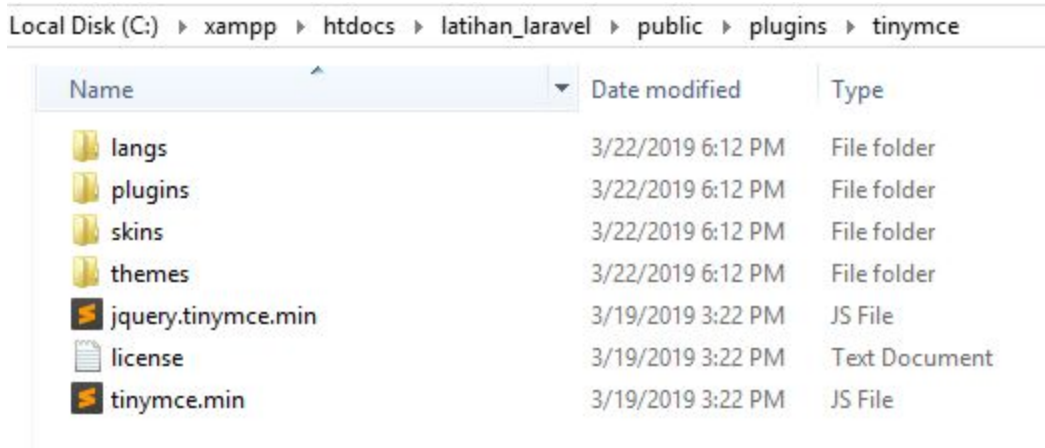
Membuat sebuah editor pada halaman website merupakan fitur yang menarik terlebih lagi untuk kebutuhan blogging maupun kebutuhan lainnya, seperti wordpress yang memiliki editor powerful dimana editor tersebut memiliki banyak fitur untuk memudahkan penggunaannya. Sebenarnya kita bisa memanfaatkan library yang sudah ada, salah satunya yaitu tinymce.

Tinymce adalah semacam library yang di bangun dari dengan javascript. Untuk memudahkan kita dalam membuat form posting/text editor pada form. Jadi dengan ada nya text editor ini, kita bisa membuat form posting seperti pada blogspot atau wordpress. Beberapa fiturnya antara lain bisa membuat text tebal, text heading dan lain-lainnya dengan sangat mudah. Berikut adalah tahapan yang dilakukan untuk mengimplementasikan rich text editor ini :

1. Download Tinymce pada official website : <https://www.tiny.cloud/>
2. Kemudian buat project Laravel, dengan perintah :

`composer create-project laravel/laravel multimedia --prefer-dist`

3. Extract file yang sudah di download sebelumnya dan tempatkan kedalam folder public yang ada di dalam directory project Laravel. Berikut adalah isi dari folder tinymce.



Name	Date modified	Type
langs	3/22/2019 6:12 PM	File folder
plugins	3/22/2019 6:12 PM	File folder
skins	3/22/2019 6:12 PM	File folder
themes	3/22/2019 6:12 PM	File folder
jquery.tinymce.min	3/19/2019 3:22 PM	JS File
license	3/19/2019 3:22 PM	Text Document
tinymce.min	3/19/2019 3:22 PM	JS File

4. Generate controller post dengan menjalankan command berikut ini :

`php artisan make:controller PostController --resource`

5. Tambahkan kode di bawah ini, untuk mengarahkan controller ke view post-insert.

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
class PostController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        return view('post_insert');
    }
}
```

6. Edit file routes/web.php, tambahkan code dibawah ini :

```
Route::get('post', 'PostController@index');
```

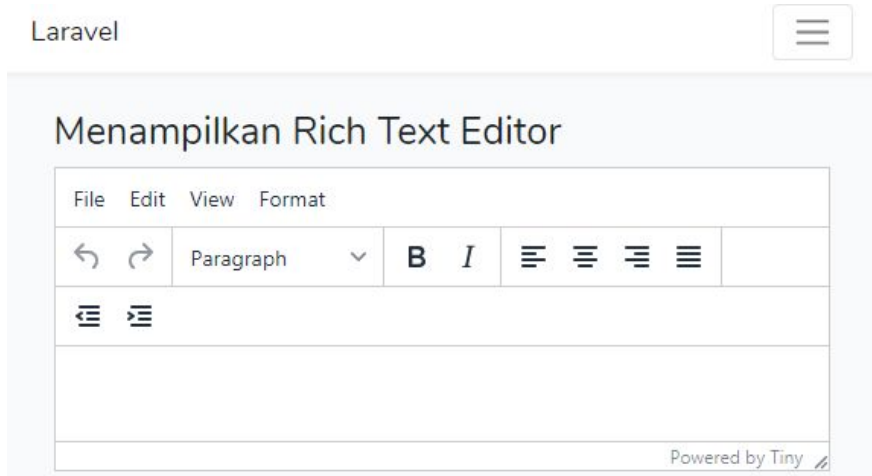
7. Kemudian buat view untuk menampilkan text editor tersebut. Buat file resources/views/post_insert.blade.php

```
@extends('layouts.app')

@section('content')
<div class="container">
    <div class="row">
        <div class="col-md-12"><h3>Menampilkan Rich Text Editor</h3></div>
        <div class="col-md-12">
            <form>
                <textarea name="editor" id="ckview"></textarea>
            </form>
        </div>
    </div>
</div>
@endsection

<script src="{{url('plugins/tinymce/jquery.tinymce.min.js')}}"></script>
<script src="{{url('plugins/tinymce/tinymce.min.js')}}"></script>
<script>tinymce.init({ selector:'#ckview' });</script>
```

8. Kemudian jalankan dengan perintah : `php artisan serve`, jika berhasil maka akan muncul tampilan seperti berikut ini.



B. Membuat Halaman Upload Image

Melanjutkan materi sebelumnya, setelah menampilkan text editor, akan dilanjutkan dengan menambahkan fitur upload image dan menampilkan image yang telah di upload sebelumnya. Berikut adalah tahapan implementasinya :

1. Buat database baru dengan nama db_multimedia.
2. Setelah membuat database, kemudian buka file .env yang ada pada direktori dalam project laravelnya. pada file tersebut lakukan pengaturan koneksi database laravel.
3. Terapkan pengaturan berikut, kedalam file .env. (sesuaikan dengan settingan masing-masing laptop)

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=db_multimedia
DB_USERNAME=root
DB_PASSWORD=
```

4. Kemudian buat folder di dalam folder storage, yaitu folder image_files.
5. Edit kembali file .env, tambahkan direktori folder yang digunakan untuk menyimpan gambar yang di upload. Tambahkan kode berikut :

```
# path settings, all relative to storage path
PATH_IMAGE=image_files/
```

6. Buat folder baru `app/Models`, kemudian buat model `Images.php` di dalam folder tersebut. Tambahkan kode berikut :

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Model;
class Images extends Model
{
    protected $table = 'image';
    protected $fillable = ['image_title', 'image_desc', 'image_src'];
}
```

7. Jika sudah, selanjutnya buat table `image` dengan field dan dengan detail tabel sebagai berikut :

Name	Type	Length
image_id	int	11
image_title	varchar	100
image_desc	text	0
image_src	varchar	255
created_at	timestamp	0
updated_at	timestamp	0

8. Kemudian buat controller dengan nama `ImageController`. Edit controller tersebut menjadi seperti berikut :

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

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{
    $file = $request->file('image_src');
    $ext = $file->getClientOriginalExtension();

    $dateTime = date('Ymd_his');
    $newName = 'image_'. $dateTime . '.' . $ext;

    $file->move(storage_path(env('PATH_IMAGE')), $newName);

    $images = new images;
    $images->image_title = $request->image_title;
    $images->image_desc = $request->image_desc;
    $images->image_src = $newName;

    $images->save();

    return redirect('image')->with('success', 'Image have been saved');
}
```

9. Edit file routes/web.php, tambahkan code dibawah ini

```
Route::get('image', 'ImageController@index')->name('image');
Route::get('image/insert', 'ImageController@create')->name('image_insert');
Route::post('image/save', 'ImageController@store')->name('image_save');
Route::get('image/view/{fileImage}', 'ImageController@viewImage')->name('image_view');
```

Baris 1 : digunakan untuk menampilkan halaman image gallery

Baris 2 : digunakan untuk menampilkan form insert image

Baris 3 : digunakan untuk menyimpan image

Baris 4 : digunakan untuk menampilkan image

10. Buat view pada resource/views/image_insert.blade.php. Tambahkan kode berikut :

```
@extends('layouts.app')

@section('content')
<div class="container">
  <div class="row">
    <div class="col-md-12"><h3>Insert Image</h3></div>
    <div class="col-md-12">
      <form action="{{ route('image_save') }}" method="POST" enctype="multipart/form-data">
        @csrf
        <div class="form-group">
          <label>Image Title</label>
          <input type="text" name="image_title" class="form-control">
        </div>
        <div class="form-group">
          <label>Image Desc</label>
          <textarea name="image_desc" id="ckview"></textarea>
        </div>
        <div class="form-group">
          <label>Upload</label>
          <input type="file" name="image_src" class="form-control">
        </div>
        <div class="form-group">
          <button type="submit" class="btn btn-primary">Save Image</button>
        </div>
      </form>
    </div>
  </div>
</div>
@endsection

<script src="{{url('plugins/tinymce/jquery.tinymce.min.js')}}"></script>
<script src="{{url('plugins/tinymce/tinymce.min.js')}}"></script>
<script>tinymce.init({ selector: '#ckview' });</script>
```

11. Jika dijalankan, akan muncul form insert seperti berikut :

Laravel Image Gallery Video Gallery henny ▾

Insert Image

Image Title

Image Desc

File Edit View Format

↶ ↷ Paragraph **B** *I*

≡ ≡ ≡ ≡ ≡ ≡

Powered by Tiny

Upload

Choose File No file chosen

Save Image

C. Membuat Halaman Image Gallery

Selanjutnya, yang harus dibuat adalah halaman image gallery, dimana ketika Gambar berhasil di upload, akan di arahkan ke image gallery untuk menampilkan image tersebut. Berikut tahapan yang harus dilakukan untuk dapat menampilkan image.

1. Jalankan perintah : `composer require intervention/image`.
2. Edit file `config/app.php`. Pada bagian providers, tambahkan :

`Intervention\Image\ImageServiceProvider::class,`

Kemudian pada bagian alias, tambahkan:

`'Image' => Intervention\Image\Facades\Image::class,`

3. Edit file `controllers/ImageController`, tambahkan kode berikut :

```
<?php
namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Images;
use Image;

class ImageController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $images = Images::all();
        return view(
            'image_gallery', [
                'images' => $images
            ]
        );
    }
}
```

4. Masih pada file yang sama, tambahkan 1 fungsi dibawahnya, seperti berikut ini :

```
public function viewImage($fileImage)
{
    $filePath = storage_path(env('PATH_IMAGE') . $fileImage);
    return Image::make($filePath)->response();
}
```

Fungsi tersebut bertujuan untuk mengubah path penyimpanan image menjadi dapat ditampilkan.

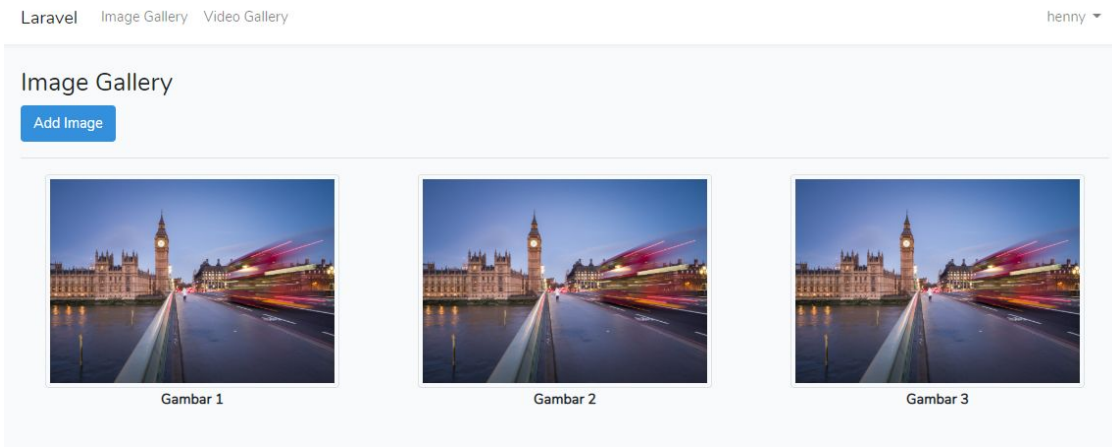
5. Kemudian buatlah file `resources/views/image_gallery.blade.php`, dan tambahkan kode berikut :

```
@extends('layouts.app')

@section('content')
<div class="container">
  <div class="row">
    <div class="col-md-12"><h3>Image Gallery</h3>
    <a href="{{ route('image_insert') }}" class="btn btn-primary">Add Image</a>
    <hr>
  </div>

  @if(!empty($images))
    @foreach($images as $val)
      <div class="col-md-4" align="center">
        
        <b>{{ $val->image_title }}</b>
      </div>
    @endforeach
  @endif
</div>
</div>
@endsection
```

6. Jika berhasil, maka halaman tersebut akan menampilkan tampilan seperti berikut :



D. Membuat Halaman Upload Video

Setelah menampilkan dan membuat upload image, akan dilanjutkan dengan menambahkan fitur upload video dan memutar video yang telah di upload sebelumnya. Berikut adalah tahapan implementasinya :

1. Kemudian buat folder di dalam folder storage, yaitu folder `video_files`.

2. Edit kembali file .env, tambahkan direktori folder yang digunakan untuk menyimpan video yang di upload. Tambahkan kode berikut :

```
# path settings, all relative to storage path
PATH_IMAGE=image_files/
PATH_VIDEO=video_files/
```

3. Buat folder baru app/Models, kemudian buat model Video.php di dalam folder tersebut. Tambahkan kode berikut :

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Model;
class Video extends Model
{
    protected $table = 'video';
    protected $fillable = ['video_title', 'video_desc', 'video_src'];
}
```

4. Jika sudah, selanjutnya buat table video dengan field dan detail tabel sebagai berikut :

Name	Type	Length
video_id	int	11
video_title	varchar	100
video_desc	text	0
video_src	varchar	255
created_at	timestamp	0
updated_at	timestamp	0

5. Kemudian buat controller dengan nama VideoController. Edit controller tersebut menjadi seperti berikut :

```
public function create()
{
    return view('video_insert');
}

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{
    $file = $request->file('video_src');
    $ext = $file->getClientOriginalExtension();

    $dateTime = date('Ymd_his');
    $newName = 'video_'. $dateTime . '.' . $ext;

    $file->move(storage_path(env('PATH_VIDEO')), $newName);

    $videos = new video;
    $videos->video_title = $request->video_title;
    $videos->video_desc = $request->video_desc;
    $videos->video_src = $newName;

    $videos->save();

    return redirect('video')->with('success', 'video have been saved');
}
```

6. Edit file routes/web.php, tambahkan code dibawah ini

```
Route::get('video', 'VideoController@index')->name('video');
Route::get('video/insert', 'VideoController@create')->name('video_insert');
Route::post('video/save', 'VideoController@store')->name('video_save');
Route::get('video/view/{fileImage}', 'VideoController@viewVideo')->name('video_view');
```

Baris 1 : digunakan untuk menampilkan halaman image gallery

Baris 2 : digunakan untuk menampilkan form insert image

Baris 3 : digunakan untuk menyimpan image

Baris 4 : digunakan untuk menampilkan image

7. Buat view pada resource/views/video_insert.blade.php. Tambahkan kode berikut :

```
@extends('layouts.app')

@section('content')
<div class="container">
  <div class="row">
    <div class="col-md-12"><h3>Insert Video</h3></div>
    <div class="col-md-12">
      <form action="{{ route('video_save') }}" method="POST" enctype="multipart/form-data">
        @csrf
        <div class="form-group">
          <label>Video Title</label>
          <input type="text" name="video_title" class="form-control">
        </div>
        <div class="form-group">
          <label>Video Desc</label>
          <textarea name="video_desc" id="ckview"></textarea>
        </div>
        <div class="form-group">
          <label>Upload</label>
          <input type="file" name="video_src" class="form-control">
        </div>
        <div class="form-group">
          <button type="submit" class="btn btn-primary">Save Video</button>
        </div>
      </form>
    </div>
  </div>
</div>
@endsection

<script src="{{ url('plugins/tinymce/jquery.tinymce.min.js') }}"></script>
<script src="{{ url('plugins/tinymce/tinymce.min.js') }}"></script>
<script>tinymce.init({ selector: '#ckview' });</script>
```

8. Jika dijalankan, akan muncul form insert seperti berikut :

Laravel Image Gallery Video Gallery henny ▾

Insert Video

Video Title

Video Desc

File Edit View Format

↶ ↷ Paragraph ▾ **B** *I*

≡ ≡ ≡ ≡ ≡ ≡

≡ ≡

Powered by Tiny

Upload

Choose File No file chosen

Save Video

E. Membuat Halaman Video Gallery

Selanjutnya, yang harus dibuat adalah halaman video gallery, dimana ketika video berhasil di upload, akan di arahkan ke video gallery untuk menampilkan atau memutar video tersebut. Berikut tahapan yang harus dilakukan untuk dapat menampilkan image.

1. Edit file controllers/VideoController, tambahkan kode berikut :

```
/**
 * Display a listing of the resource.
 *
 * @return \Illuminate\Http\Response
 */
public function index()
{
    $video = Video::all();
    return view(
        'video_gallery', [
            'videos' => $video
        ]
    );
}
```

2. Masih pada file yang sama, tambahkan 1 fungsi dibawahnya, seperti berikut ini :

```
public function viewVideo($fileVideo)
{
    return response()->file(storage_path(env('PATH_VIDEO')).$fileVideo), [
        'Content-Type' => 'video/mp4',
        'Content-Disposition' => 'inline; filename="Lesson-file"'
    ]);

    return false;
}
```

Fungsi tersebut bertujuan untuk mengubah path penyimpanan video menjadi dapat ditampilkan.

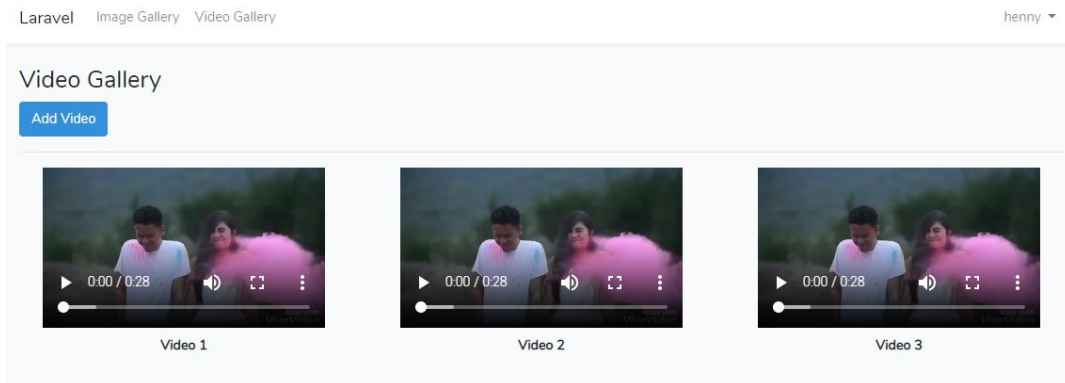
3. Kemudian buatlah file resources/views/video_gallery.blade.php, dan tambahkan kode berikut :

```
@extends('layouts.app')

@section('content')
<div class="container">
  <div class="row">
    <div class="col-md-12"><h3>Video Gallery</h3>
    <a href="{{ route('video_insert') }}" class="btn btn-primary">Add Video</a>
    <hr>
  </div>

  @if(empty($videos))
    @foreach($videos as $val)
      <div class="col-md-4" align="center">
        <video width="300" controls>
          <source src="{{ url('video/view/'.$val->video_src) }}" type="video/mp4">
        </video><br>
        <b>{{ $val->video_title }}</b>
      </div>
    @endforeach
  @endif
</div>
@endsection
```

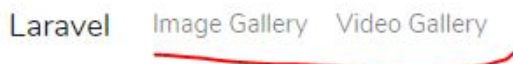
4. Jika berhasil, maka halaman tersebut akan menampilkan tampilan seperti berikut, selain itu, video yang terdapat pada halaman tersebut dapat diputar .



Catatan : video yang dapat di upload harus berukuran dibawah 2mb, jika ingin meningkatkan batas maksimal upload, dapat melakukan konfigurasi pada php.ini. ubah upload_max_filesize sesuai batas upload yang di inginkan.

F. Membuat Menu

Untuk membuat menu seperti Gambar berikut :



Tambahkan kode berikut ini, pada layout/app.blade.php :

```
<ul class="navbar-nav mr-auto">
  @if (Auth::check())
  <li class="nav-item">
    <a class="nav-link" href="{{ route('image') }}" role="button">Image Gallery</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="{{ route('video') }}" role="button">Video Gallery</a>
  </li>
  @endif
</ul>
```

G. Latihan : ???