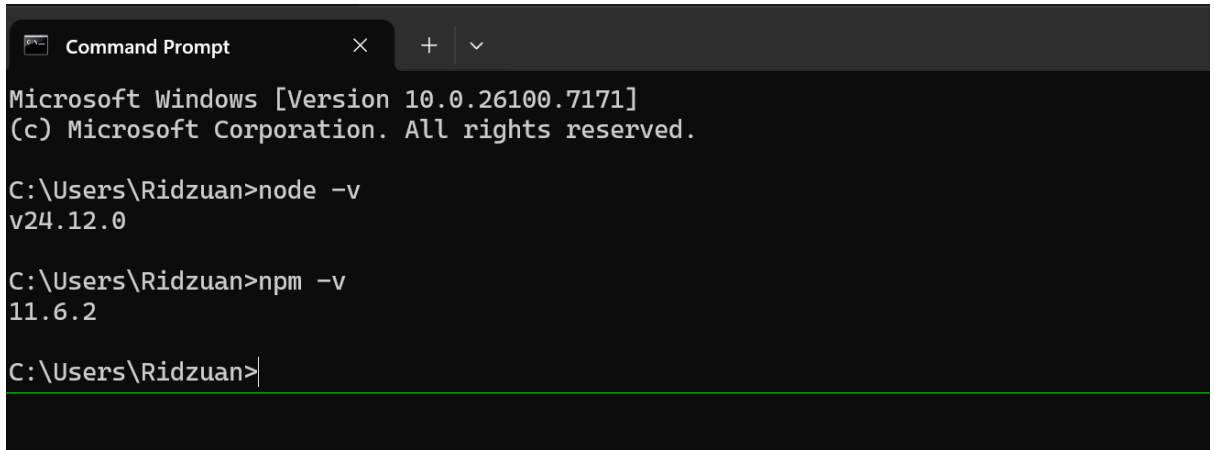


TP MOD 14 WEBPRO

Muhammad Ilham Ridzuan – 103022300033

1. Download node.js



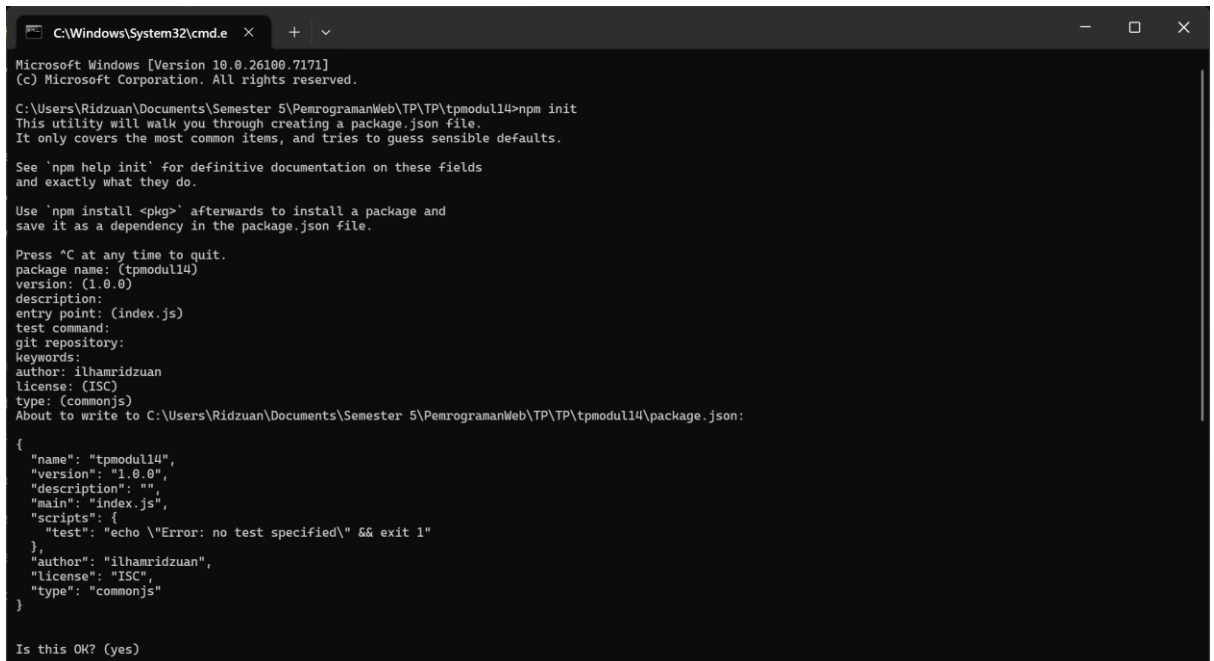
```
Command Prompt
Microsoft Windows [Version 10.0.26100.7171]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Ridzuan>node -v
v24.12.0

C:\Users\Ridzuan>npm -v
11.6.2

C:\Users\Ridzuan>
```

2.



```
C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.26100.7171]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Ridzuan\Documents\Semester 5\PemrogramanWeb\TP\TP\tpmodul14>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See 'npm help init' for definitive documentation on these fields
and exactly what they do.

Use 'npm install <pkg>' afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (tpmodul14)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author: ilhamridzuan
license: (ISC)
type: (commonjs)
About to write to C:\Users\Ridzuan\Documents\Semester 5\PemrogramanWeb\TP\TP\tpmodul14\package.json:
{
  "name": "tpmodul14",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "ilhamridzuan",
  "license": "ISC",
  "type": "commonjs"
}

Is this OK? (yes)
```

3. Install express & dependency tambahan

```
C:\Windows\System32\cmd.exe X + v X
git repository:
keywords:
author: ilhamridzuan
license: (ISC)
type: (commonjs)
About to write to C:\Users\Ridzuan\Documents\Semester 5\PenrogramanWeb\TP\TP\tpmodul14\package.json:
{
  "name": "tpmodul14",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "author": "ilhamridzuan",
  "license": "ISC",
  "type": "commonjs"
}

Is this OK? (yes)

npm notice
npm notice New minor version of npm available! 11.6.2 -> 11.7.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.7.0
npm notice To update run: npm install -g npm@11.7.0
npm notice

C:\Users\Ridzuan\Documents\Semester 5\PenrogramanWeb\TP\TP\tpmodul14>npm install express mysql2 dotenv

added 76 packages, and audited 77 packages in 5s

22 packages are looking for funding
  run 'npm fund' for details

found 0 vulnerabilities

C:\Users\Ridzuan\Documents\Semester 5\PenrogramanWeb\TP\TP\tpmodul14>
```

4. Project API FILM

The screenshot shows the VS Code interface for the 'tpmodul14' project. The Explorer panel on the left displays the project structure, including folders like 'src' and 'node_modules', and files like 'config', 'db.js', 'controllers', 'repositories', 'services', 'app.js', 'package-lock.json', and 'package.json'. The main editor area shows the 'src/repositories.js' file, which contains the following code:

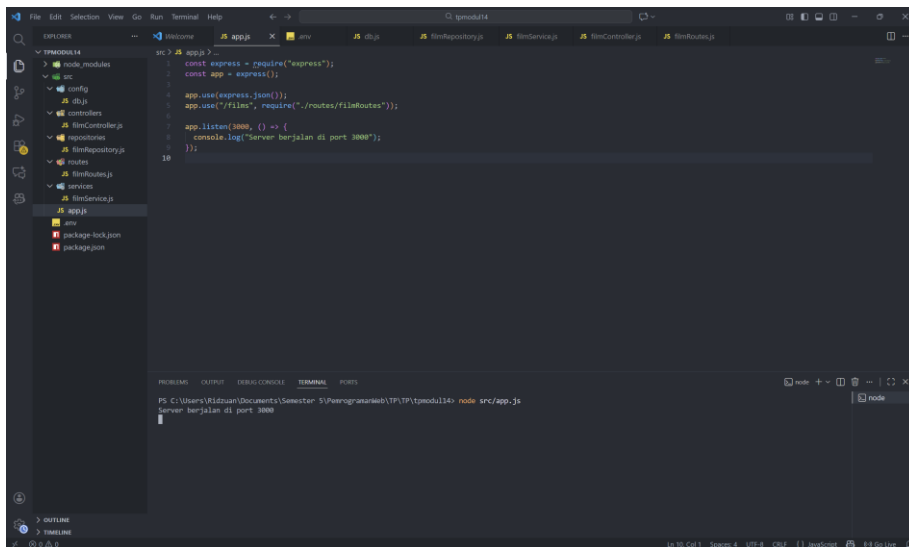
```
1 const db = require("../config/db");
2
3 const getallfiles = async () => {
4   const [rows] = await db.query("SELECT * FROM files");
5   return rows;
6 };
7
8 const getfileById = async (id) => {
9   const [rows] = await db.query(
10     "SELECT * FROM files WHERE id = ?",
11     [id]
12   );
13   return rows[0];
14 };
15
16 const createfile = async (file) => {
17   const { nama, deskripsi, tanggal_rilis, rating } = file;
18   const [result] = await db.query(
19     "INSERT INTO files (nama, deskripsi, tanggal_rilis, rating) VALUES (?, ?, ?, ?)",
20     [nama, deskripsi, tanggal_rilis, rating]
21   );
22   return result.insertId;
23 };
24
25 const updatefile = async (id, file) => {
26   const { nama, deskripsi, tanggal_rilis, rating } = file;
27   await db.query(
28     "UPDATE files SET nama=?, deskripsi=?, tanggal_rilis=?, rating=? WHERE id=?",
29     [nama, deskripsi, tanggal_rilis, rating, id]
30   );
31 };
32
33 const deletefile = async (id) => {
34   await db.query("DELETE FROM files WHERE id=?", [id]);
35 };
36
37 module.exports = {
38   getallfiles,
39   getfileById,
40   createfile,
41   updatefile,
42   deletefile
43 };
44
```

The bottom panel shows the terminal output, which displays the command 'PS C:\Users\Ridzuan\Documents\Semester 5\PenrogramanWeb\TP\TP\tpmodul14> node src/app.js' and the response 'Server berjalan di port 3000'.

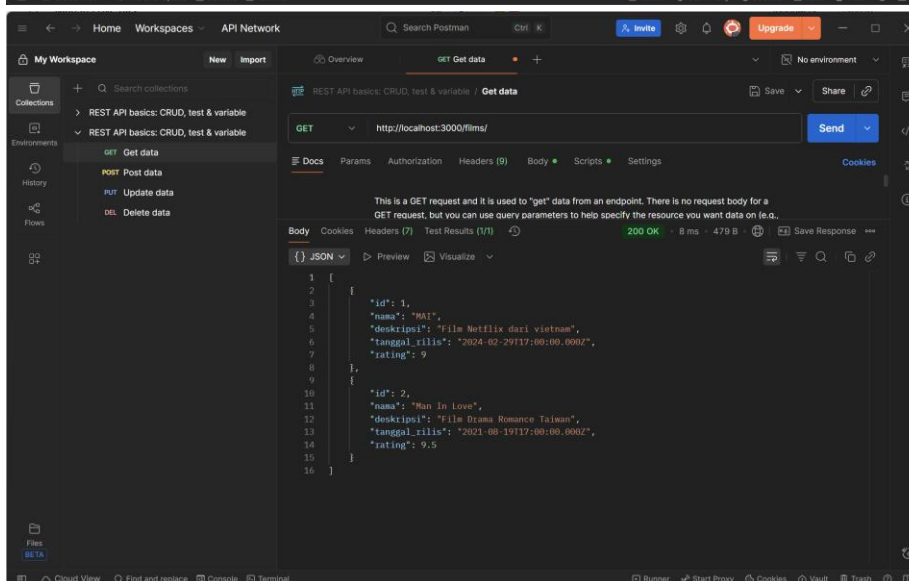
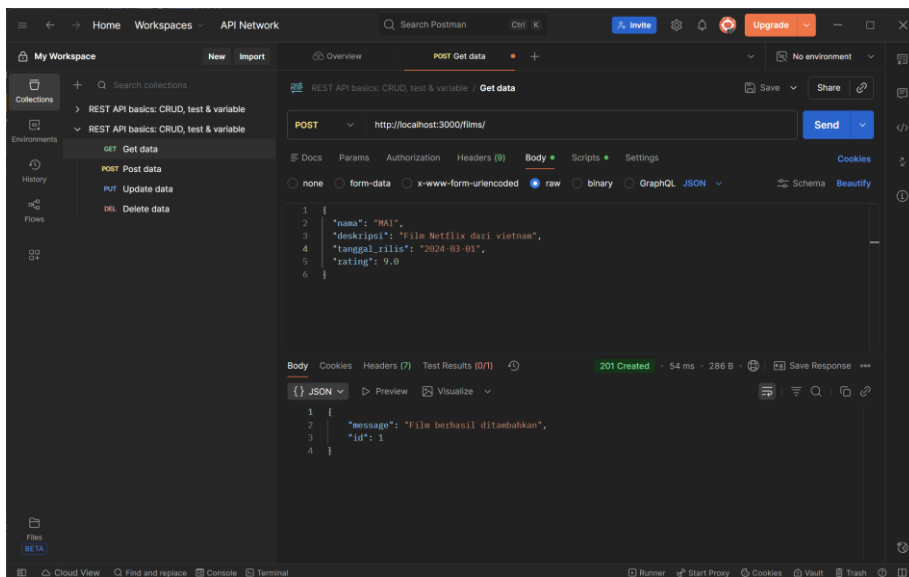
```
File Edit Selection View Go Run Terminal Help
src > services > JS filmService.js > ...
1 const filmRepository = require("../repositories/filmRepository");
2
3 const getAllFiles = async () => {
4   return await filmRepository.getAllFiles();
5 };
6
7 const getFilmById = async (id) => {
8   if (!id) {
9     throw new Error("ID film wajib diisi!");
10   }
11   return await filmRepository.getFilmById(id);
12 };
13
14 const createFile = async (file) => {
15   if (!file.name) {
16     throw new Error("Nama file wajib diisi!");
17   }
18   return await filmRepository.createFile(file);
19 };
20
21 const updateFile = async (id, file) => {
22   if (!id) {
23     throw new Error("ID film tidak valid!");
24   }
25   await filmRepository.updateFile(id, file);
26 };
27
28 const deleteFile = async (id) => {
29   if (!id) {
30     throw new Error("ID film tidak valid!");
31   }
32   await filmRepository.deleteFile(id);
33 };
34
35 module.exports = {
36   getAllFiles,
37   getFilmById,
38   createFile,
39   updateFile,
40   deleteFile
41 };
42
```

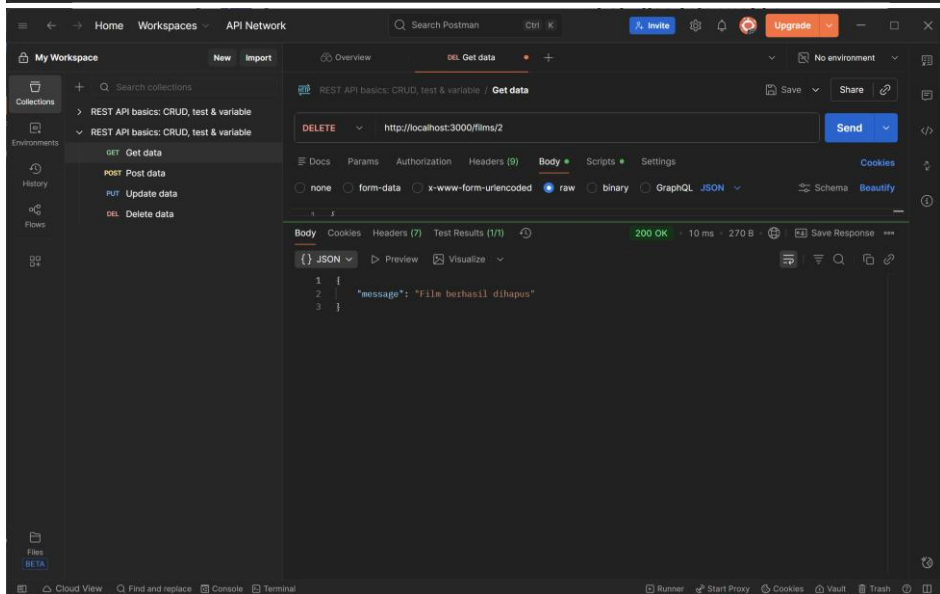
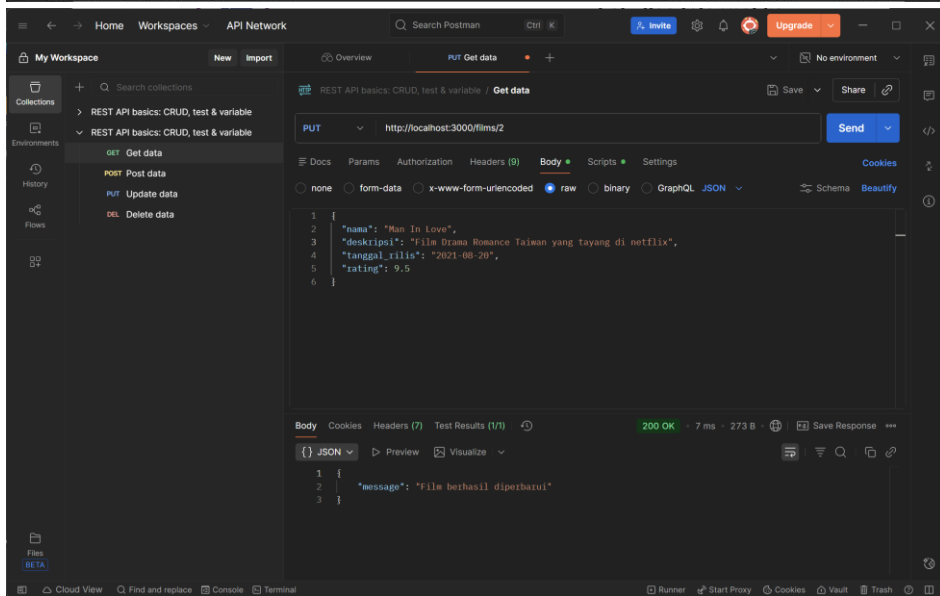
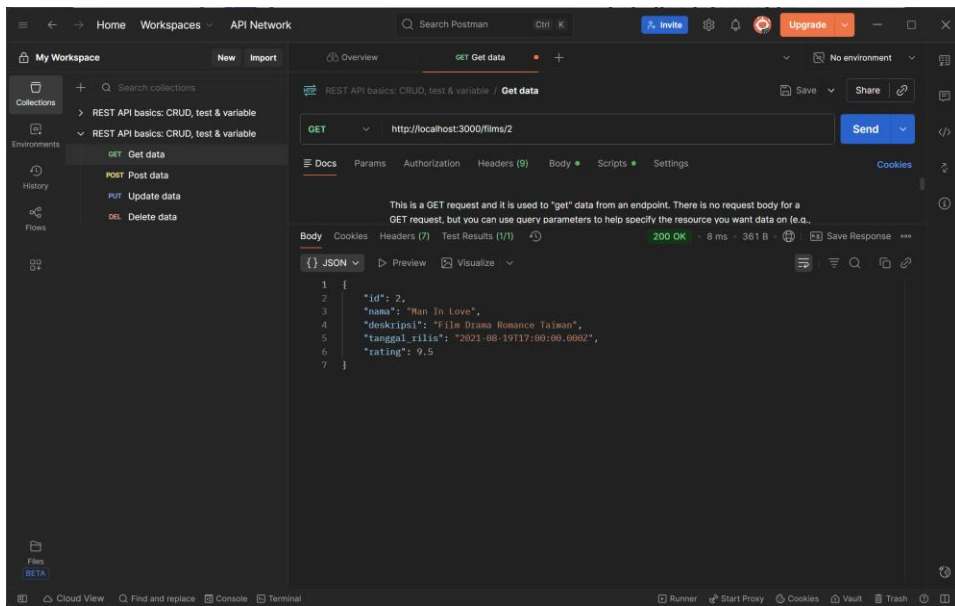
```
File Edit Selection View Go Run Terminal Help
src > controllers > JS filmController.js > ...
1 const filmService = require("../services/filmService");
2
3 const getAllFiles = async (req, res) => {
4   try {
5     const files = await filmService.getAllFiles();
6     res.status(200).json(files);
7   } catch (error) {
8     res.status(500).json({ message: error.message });
9   }
10 };
11
12 const getFilmById = async (req, res) => {
13   try {
14     const film = await filmService.getFilmById(req.params.id);
15     res.status(200).json(film);
16   } catch (error) {
17     res.status(400).json({ message: error.message });
18   }
19 };
20
21 const createFile = async (req, res) => {
22   try {
23     const id = await filmService.createFile(req.body);
24     res.status(200).json({ message: "Film berhasil ditambahkan", id });
25   } catch (error) {
26     res.status(400).json({ message: error.message });
27   }
28 };
29
30 const updateFile = async (req, res) => {
31   try {
32     await filmService.updateFile(req.params.id, req.body);
33     res.status(200).json({ message: "Film berhasil diperbarui" });
34   } catch (error) {
35     res.status(400).json({ message: error.message });
36   }
37 };
38
39 const deleteFile = async (req, res) => {
40   try {
41     await filmService.deleteFile(req.params.id);
42     res.status(200).json({ message: "Film berhasil dihapus" });
43   } catch (error) {
44     res.status(400).json({ message: error.message });
45   }
46 };
47
48 module.exports = {
49   getAllFiles,
50   getFilmById,
51   createFile,
52   updateFile,
53   deleteFile
54 };
55
```

```
File Edit Selection View Go Run Terminal Help
src > routes > JS filmRoutes.js > ...
1 const express = require("express");
2 const router = express.Router();
3 const filmController = require("../controllers/filmController");
4
5 router.get("/", filmController.getAllFiles);
6 router.get("/:id", filmController.getFilmById);
7 router.post("/", filmController.createFile);
8 router.put("/:id", filmController.updateFile);
9 router.delete("/:id", filmController.deleteFile);
10
11
12 module.exports = router;
```

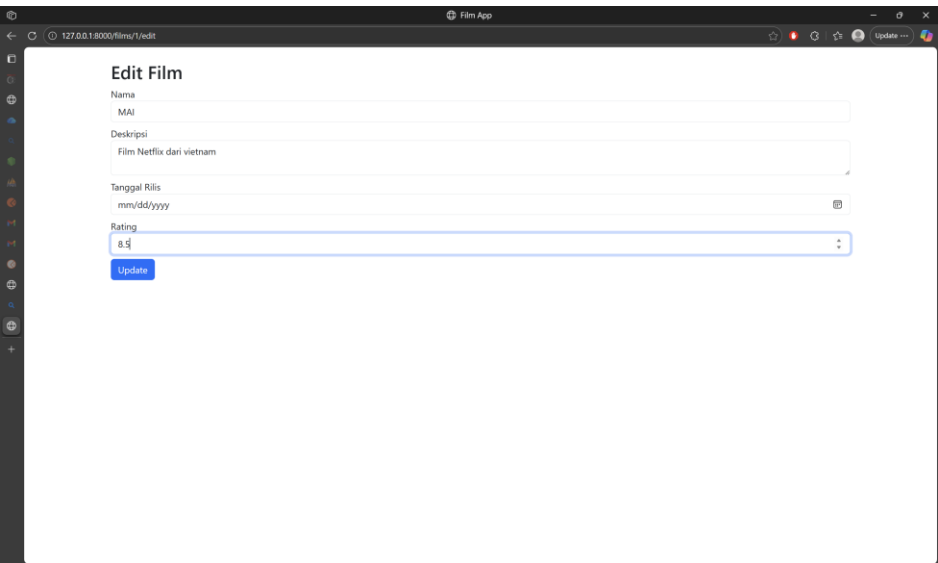
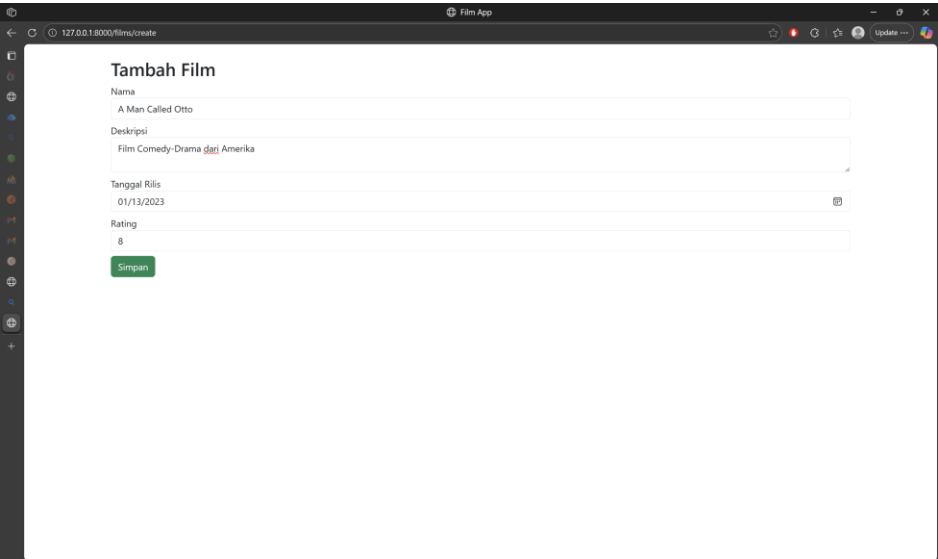
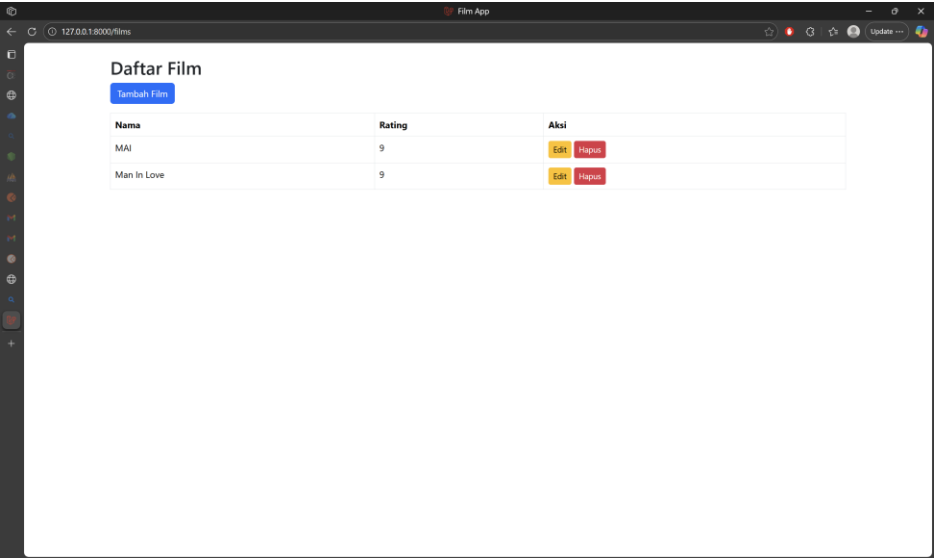


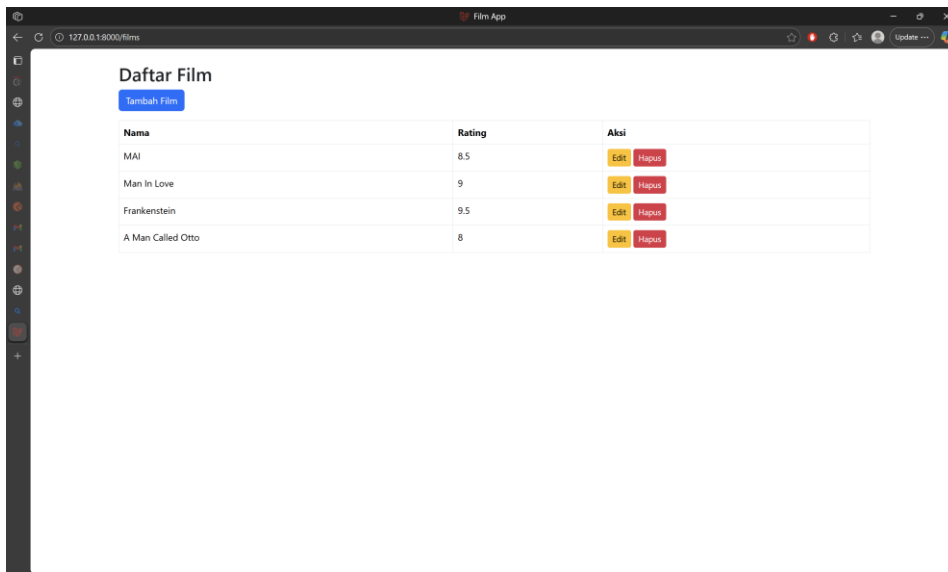
TESTING API



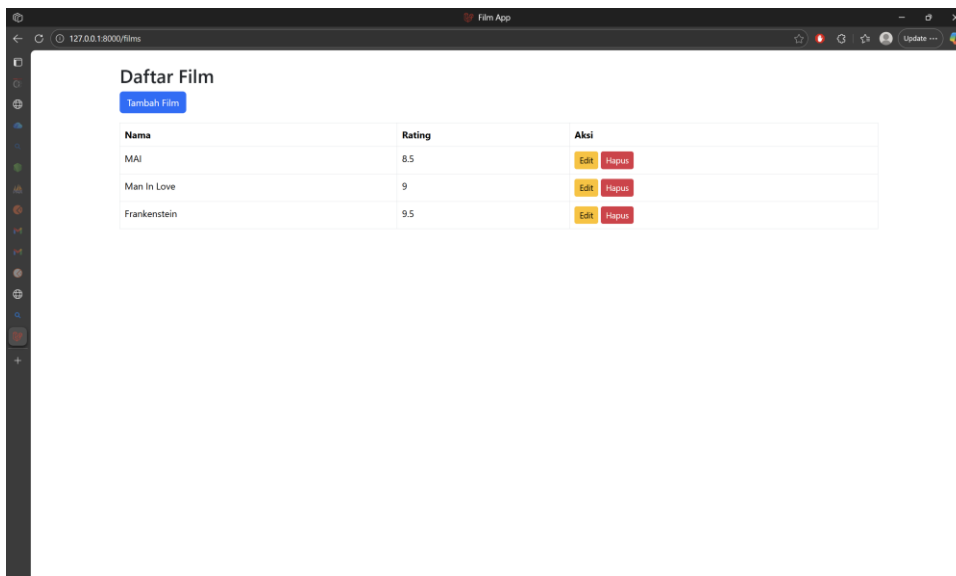


Project Laravel





Setelah hapus film ke-4



Code Laravel :

FilmController

```
<?php

namespace App\Http\Controllers;

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

class FilmController extends Controller
{
    private $apiUrl = 'http://localhost:3000/films';

    public function index()
    {
```

```

        $response = Http::get($this->apiUrl);
        $films = $response->json();
        return view('index', compact('films'));
    }

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

    public function store(Request $request)
    {
        Http::post($this->apiUrl, $request->all());
        return redirect('/films');
    }

    public function edit($id)
    {
        $film = Http::get($this->apiUrl.'/'.$id)->json();
        return view('edit', compact('film'));
    }

    public function update(Request $request, $id)
    {
        Http::put($this->apiUrl.'/'.$id, $request->all());
        return redirect('/films');
    }

    public function destroy($id)
    {
        Http::delete($this->apiUrl.'/'.$id);
        return redirect('/films');
    }
}

```

Routes

```

<?php

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

Route::get('/films', [FilmController::class, 'index']);
Route::get('/films/create', [FilmController::class, 'create']);
Route::post('/films', [FilmController::class, 'store']);
Route::get('/films/{id}/edit', [FilmController::class, 'edit']);
Route::put('/films/{id}', [FilmController::class, 'update']);
Route::delete('/films/{id}', [FilmController::class, 'destroy']);

```


App.blade.php

```
<!DOCTYPE html>
<html>
<head>
    <title>Film App</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>
<div class="container mt-4">
    @yield('content')
</div>
</body>
</html>
```

index.blade.php

```
@extends('app')

@section('content')
<h2>Daftar Film</h2>

<a href="/films/create" class="btn btn-primary mb-3">Tambah Film</a>

<table class="table table-bordered">
    <tr>
        <th>Nama</th>
        <th>Rating</th>
        <th>Aksi</th>
    </tr>
    @foreach ($films as $film)
    <tr>
        <td>{{ $film['nama'] }}</td>
        <td>{{ $film['rating'] }}</td>
        <td>
            <a href="/films/{{ $film['id'] }}/edit" class="btn btn-warning
btn-sm">Edit</a>
            <form action="/films/{{ $film['id'] }}" method="POST"
style="display:inline;">
                @csrf
                @method('DELETE')
                <button class="btn btn-danger btn-sm">Hapus</button>
            </form>
        </td>
    </tr>
    @endforeach
```

```
</table>
@endsection
```

create.blade.php

```
@extends('app')

@section('content')
<h2>Tambah Film</h2>

<form method="POST" action="/films">
@csrf
<div class="mb-2">
    <label>Nama</label>
    <input type="text" name="nama" class="form-control">
</div>

<div class="mb-2">
    <label>Deskripsi</label>
    <textarea name="deskripsi" class="form-control"></textarea>
</div>

<div class="mb-2">
    <label>Tanggal Rilis</label>
    <input type="date" name="tanggal_rilis" class="form-control">
</div>

<div class="mb-2">
    <label>Rating</label>
    <input type="number" step="0.1" name="rating" class="form-control">
</div>

<button class="btn btn-success">Simpan</button>
</form>
@endsection
```

edit.blade.php

```
@extends('app')

@section('content')
<h2>Edit Film</h2>

<form method="POST" action="/films/{{ $film['id'] }}">
@csrf
@method('PUT')
```

```
<div class="mb-2">
  <label>Nama</label>
  <input type="text" name="nama" value="{{ $film['nama'] }}" class="form-
control">
</div>

<div class="mb-2">
  <label>Deskripsi</label>
  <textarea name="deskripsi" class="form-control">{{ $film['deskripsi']
}}</textarea>
</div>

<div class="mb-2">
  <label>Tanggal Rilis</label>
  <input type="date" name="tanggal_rilis" value="{{ $film['tanggal_rilis']
}}" class="form-control">
</div>

<div class="mb-2">
  <label>Rating</label>
  <input type="number" step="0.1" name="rating" value="{{ $film['rating']
}}" class="form-control">
</div>

<button class="btn btn-primary">Update</button>
</form>
@endsection
```