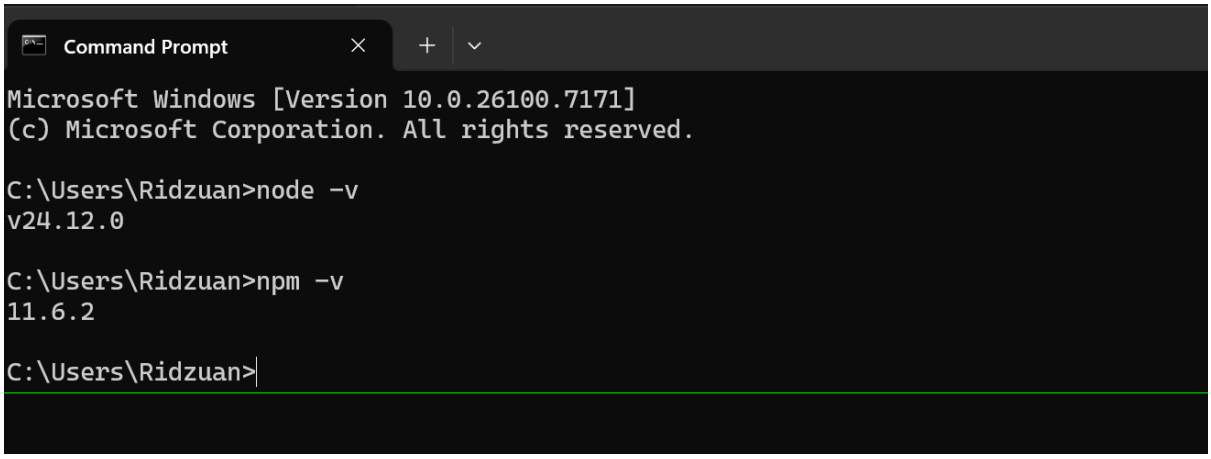
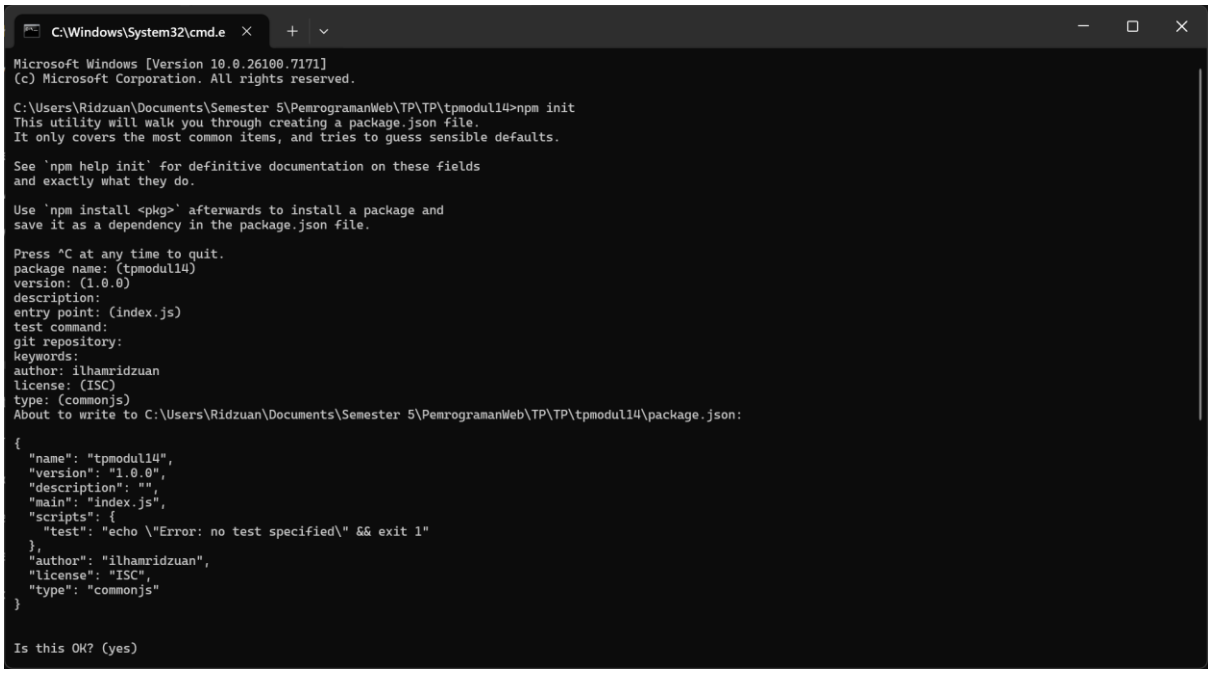
TP MOD 14 WEBPRO

Muhammad Ilham Ridzuan – 103022300033

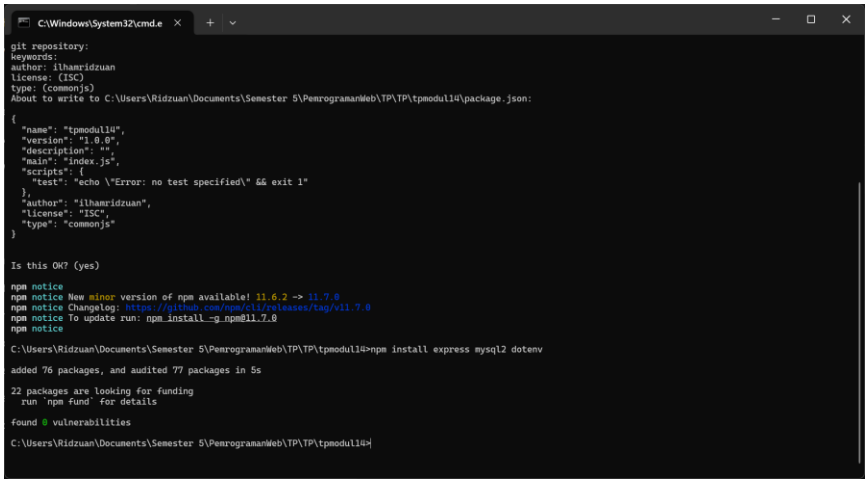
1. Download node.js



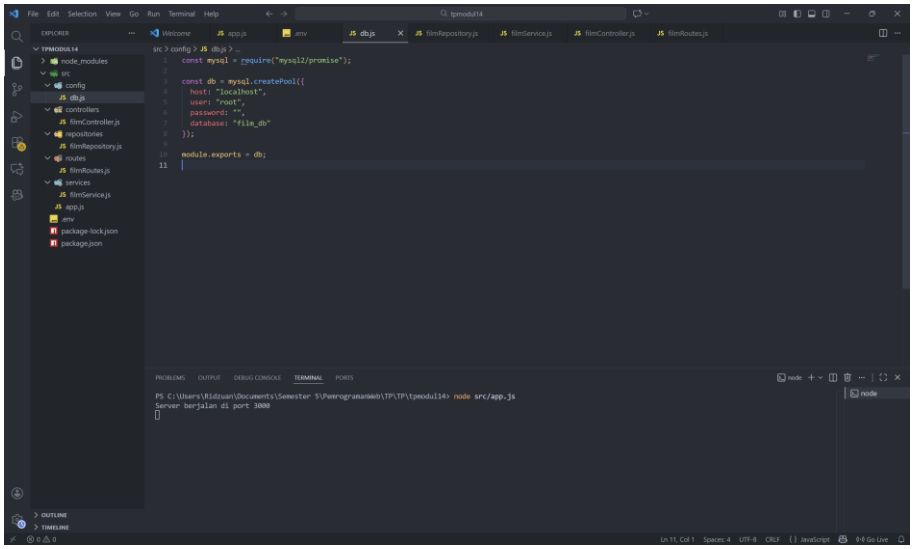
2.

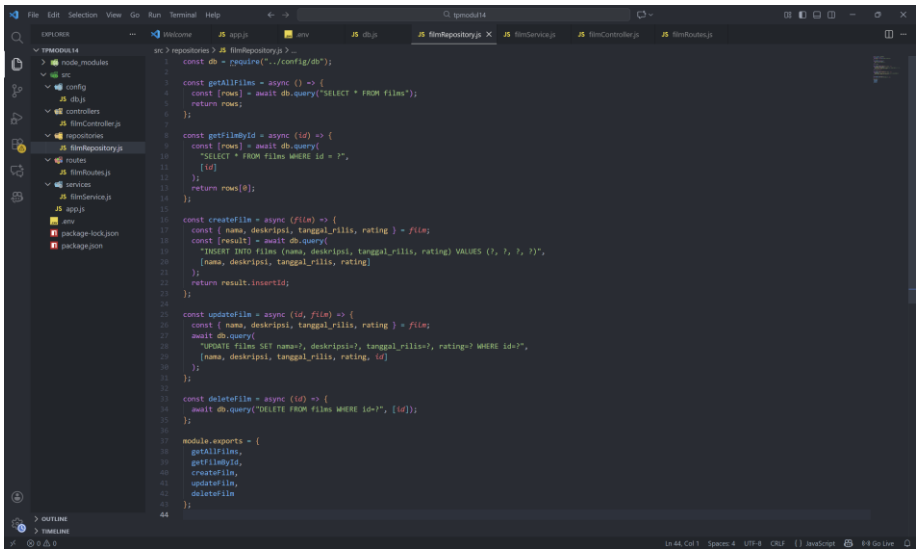


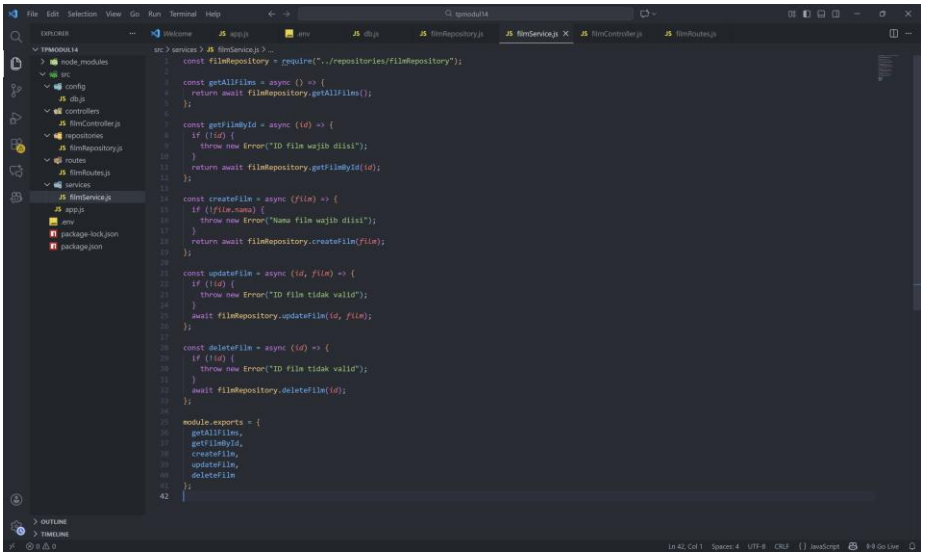
3. Install express & dependency tambahan



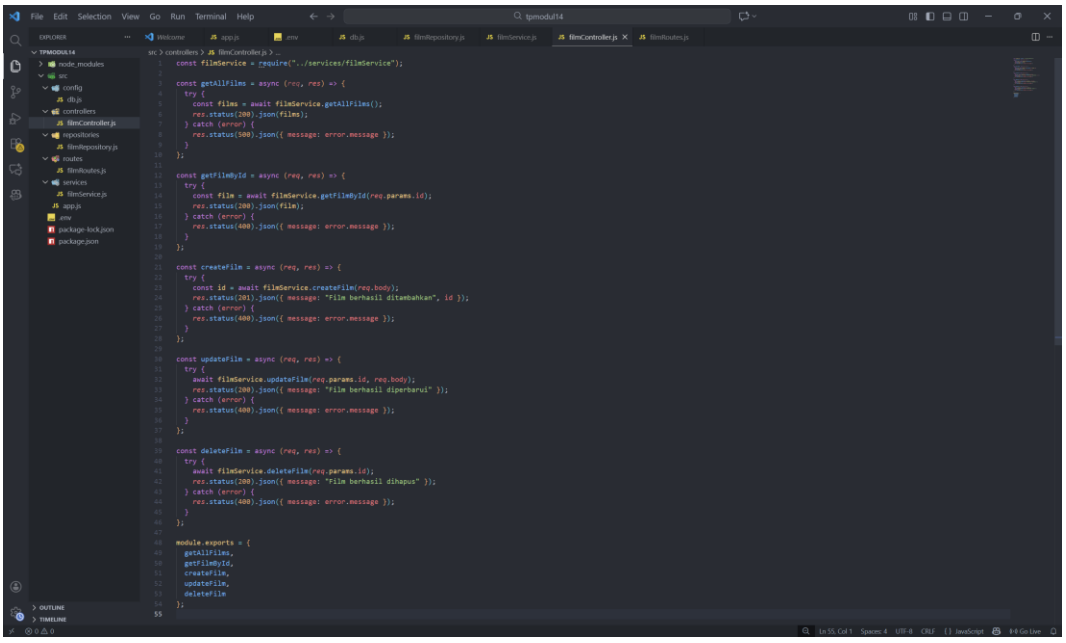
4. Project API FILM



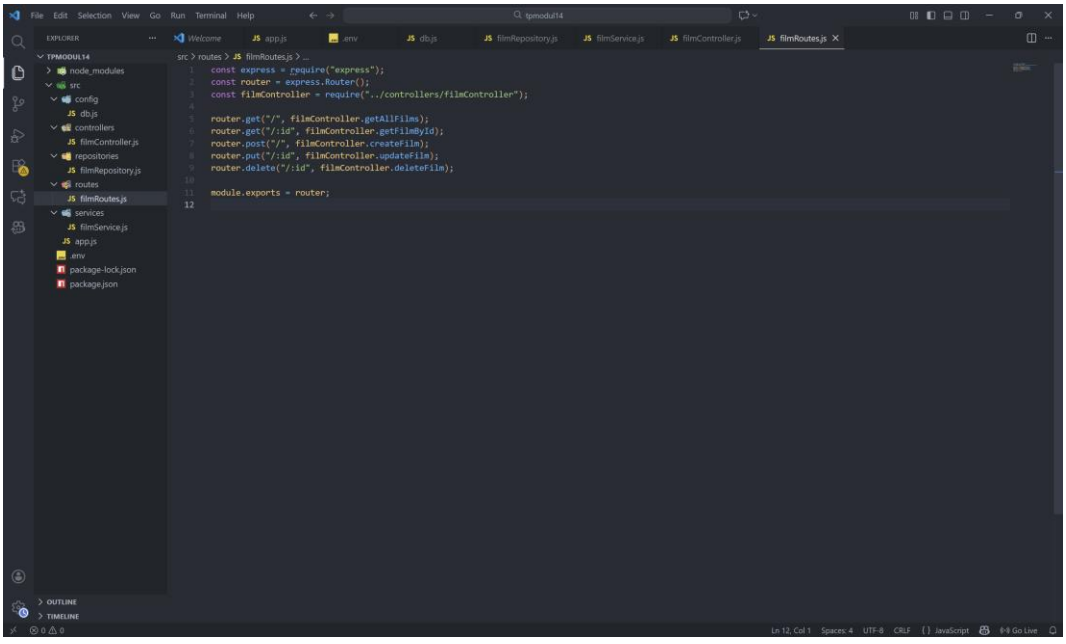




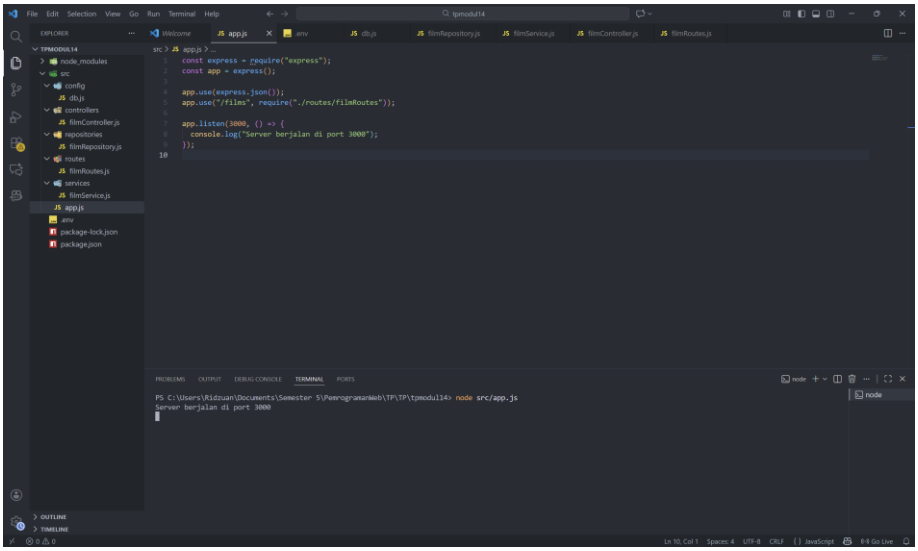
```
const filmRepository = require("../repositories/filmRepository");

const getAllFilms = async () => {
  return await filmRepository.getAllFilms();
};

const getFilmById = async (id) => {
  if (!id) {
    throw new Error("ID film wajib diisi");
  }
  return await filmRepository.getFilmById(id);
};

const createFilm = async (film) => {
  if (!film.nama) {
    throw new Error("Nama film wajib diisi");
  }
  return await filmRepository.createFilm(film);
};

const updateFilm = async (id, film) => {
  if (!id) {
    throw new Error("ID film tidak valid");
  }
  await filmRepository.updateFilm(id, film);
};

const deleteFilm = async (id) => {
  if (!id) {
    throw new Error("ID film tidak valid");
  }
  await filmRepository.deleteFilm(id);
};

module.exports = {
  getAllFilms,
  getFilmById,
  createFilm,
  updateFilm,
  deleteFilm
};
```



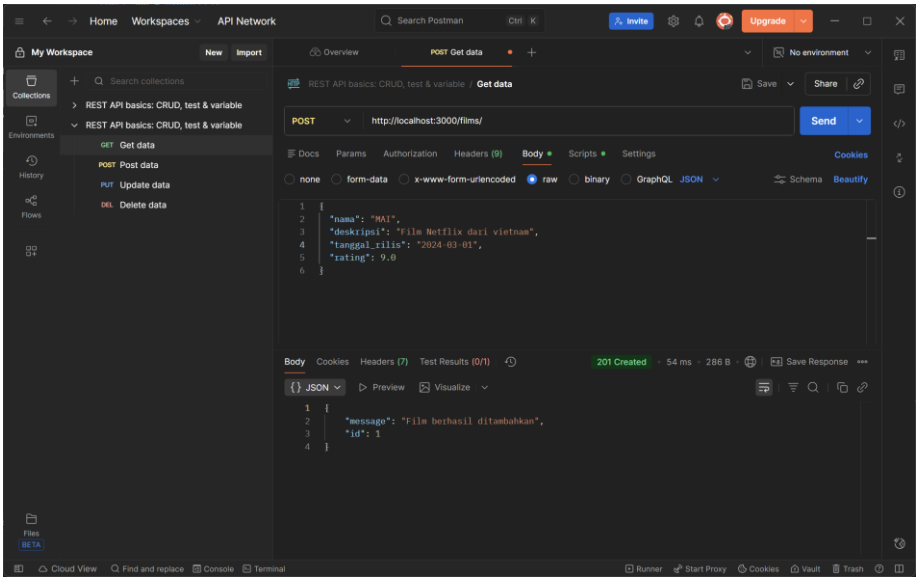
```
const filmService = require("../services/filmService");

const getAllFilms = async (req, res) => {
  try {
    const films = await filmService.getAllFilms();
    res.status(200).json(films);
  } catch (error) {
    res.status(500).json({ message: error.message });
  }
};

const getFilmById = async (req, res) => {
  try {
    const film = await filmService.getFilmById(req.params.id);
    res.status(200).json(film);
  } catch (error) {
    res.status(400).json({ message: error.message });
  }
};

const createFilm = async (req, res) => {
  try {
    const id = await filmService.createFilm(req.body);
    res.status(201).json({ message: "Film berhasil ditambahkan", id });
  } catch (error) {
    res.status(400).json({ message: error.message });
  }
};

const updateFilm = async (req, res) => {
  try {
    await filmService.updateFilm(req.params.id, req.body);
    res.status(200).json({ message: "Film berhasil diperbarui" });
  } catch (error) {
    res.status(400).json({ message: error.message });
  }
};

const deleteFilm = async (req, res) => {
  try {
    await filmService.deleteFilm(req.params.id);
    res.status(200).json({ message: "Film berhasil dihapus" });
  } catch (error) {
    res.status(400).json({ message: error.message });
  }
};

module.exports = {
  getAllFilms,
  getFilmById,
  createFilm,
  updateFilm,
  deleteFilm
};
```

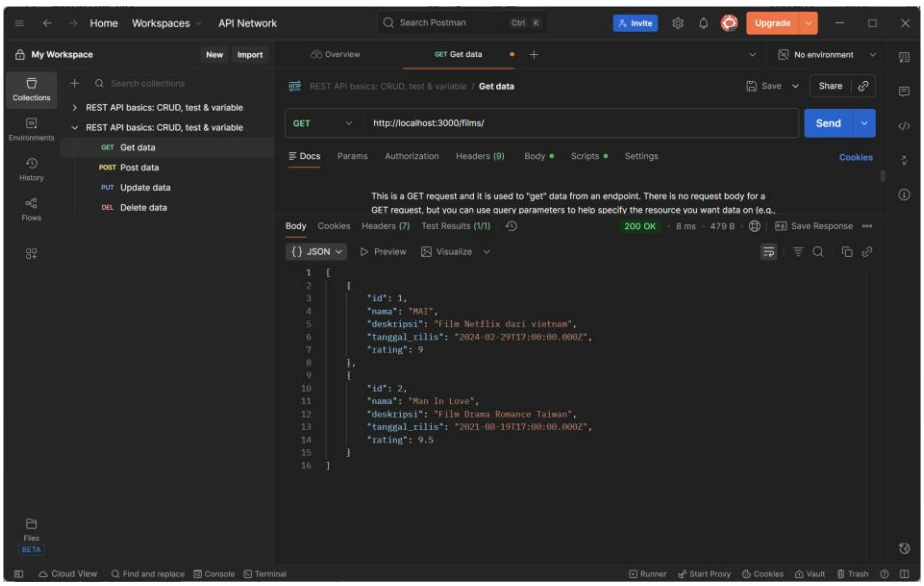


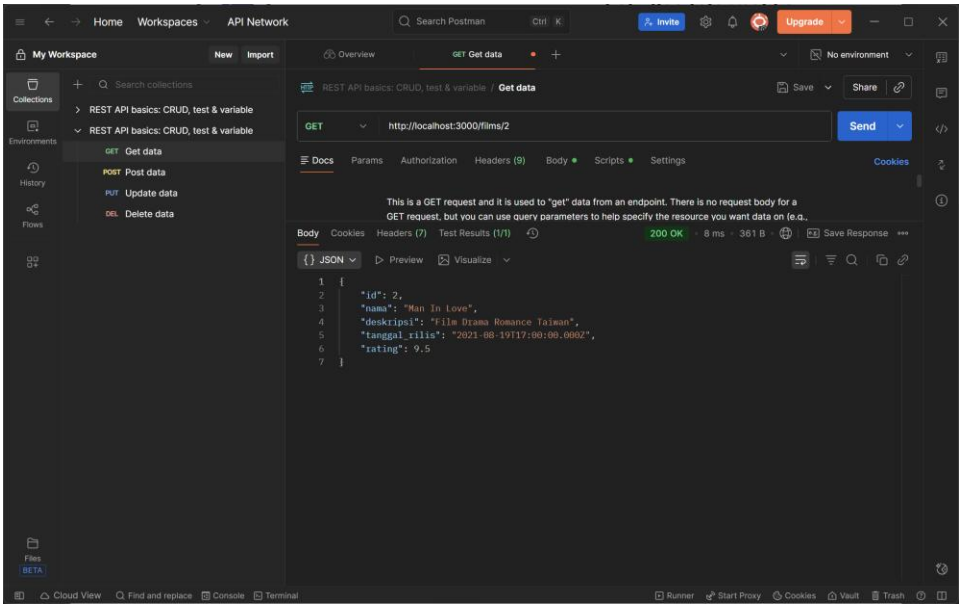
```
const express = require("express");
const router = express.Router();
const filmController = require("../controllers/filmController");

router.get("/", filmController.getAllFilms);
router.get("/:id", filmController.getFilmById);
router.post("/", filmController.createFilm);
router.put("/:id", filmController.updateFilm);
router.delete("/:id", filmController.deleteFilm);

module.exports = router;
```

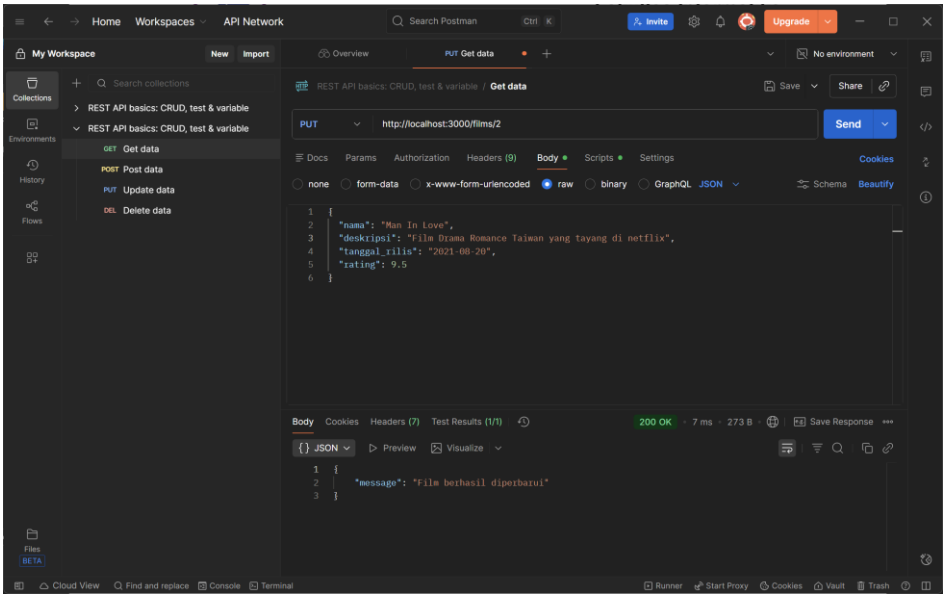

const express = require("express");
const app = express();
app.use(express.json());
app.use("/films", require("./routes/filmRoutes"));
app.listen(3000, () => {
console.log("Server berjalan di port 3000");
});
PS C:\Users\Ridzuan\Documents\Semester 5\Pemrogramanweb\TP\TP\tpmodul14> node src/app.js
Server berjalan di port 3000

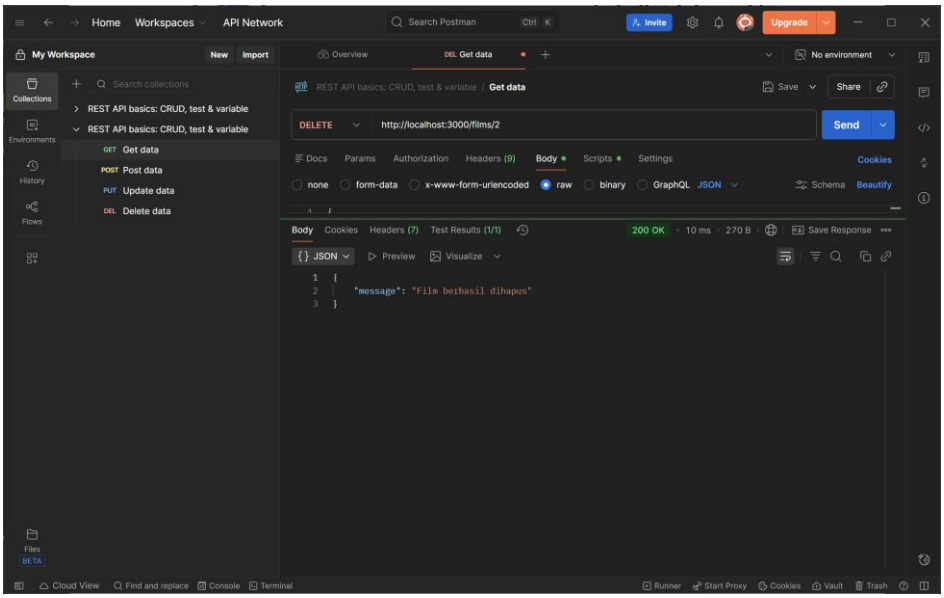
TESTING API


Home
Workspaces
API Network
Search Postman
Invite
Upgrade
My Workspace
New
Import
Collections
Environments
History
Flows
REST API basics: CRUD, test & variable
GET Get data
POST Post data
PUT Update data
DEL Delete data
POST Get data
No environment
Save
Share
POST
http://localhost:3000/films/
Send
Docs
Params
Authorization
Headers (9)
Body
Scripts
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Schema
Beautify
{
"nama": "MAI",
"deskripsi": "Film Netflix dari vietnam",
"tanggal_rilis": "2024-03-01",
"rating": 9.0
}
Body
Cookies
Headers (7)
Test Results (0/1)
201 Created
54 ms
286 B
Save Response
JSON
Preview
Visualize
{
"message": "Film berhasil ditambahkan",
"id": 1
}
Files
BETA
Cloud View
Find and replace
Console
Terminal
Runner
Start Proxy
Cookies
Vault
Trash

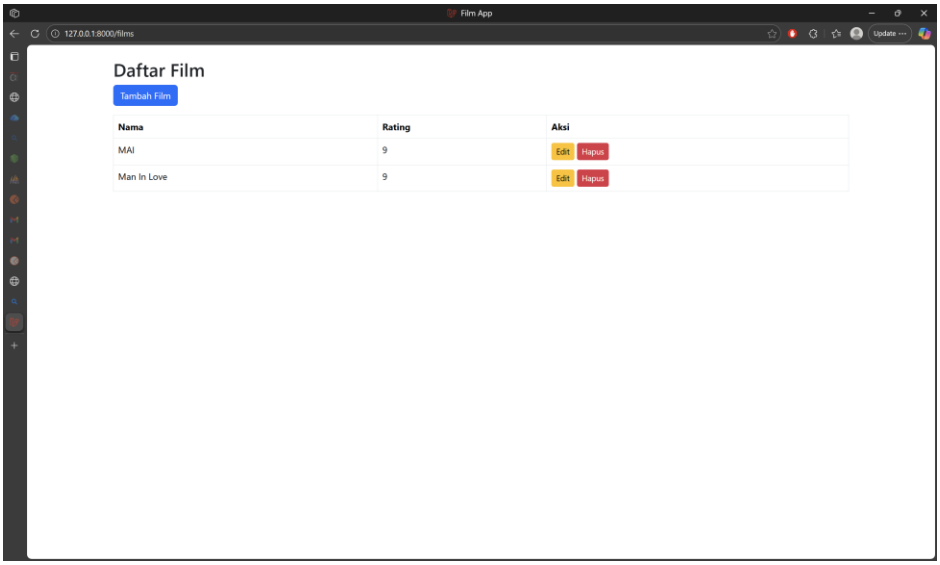

Home
Workspaces
API Network
Search Postman
Invite
Upgrade
My Workspace
New
Import
Collections
Environments
History
Flows
REST API basics: CRUD, test & variable
GET Get data
POST Post data
PUT Update data
DEL Delete data
GET Get data
No environment
Save
Share
GET
http://localhost:3000/films/
Send
Docs
Params
Authorization
Headers (9)
Body
Scripts
Settings
Cookies
This is a GET request and it is used to "get" data from an endpoint. There is no request body for a GET request, but you can use query parameters to help specify the resource you want data on (e.g.,
Body
Cookies
Headers (7)
Test Results (1/1)
200 OK
8 ms
479 B
Save Response
JSON
Preview
Visualize
[
{
"id": 1,
"nama": "MAI",
"deskripsi": "Film Netflix dari vietnam",
"tanggal_rilis": "2024-02-29T17:00:00.000Z",
"rating": 9
},
{
"id": 2,
"nama": "Man In Love",
"deskripsi": "Film Drama Romance Taiwan",
"tanggal_rilis": "2021-08-19T17:00:00.000Z",
"rating": 9.5
}
]
Files
BETA
Cloud View
Find and replace
Console
Terminal
Runner
Start Proxy
Cookies
Vault
Trash

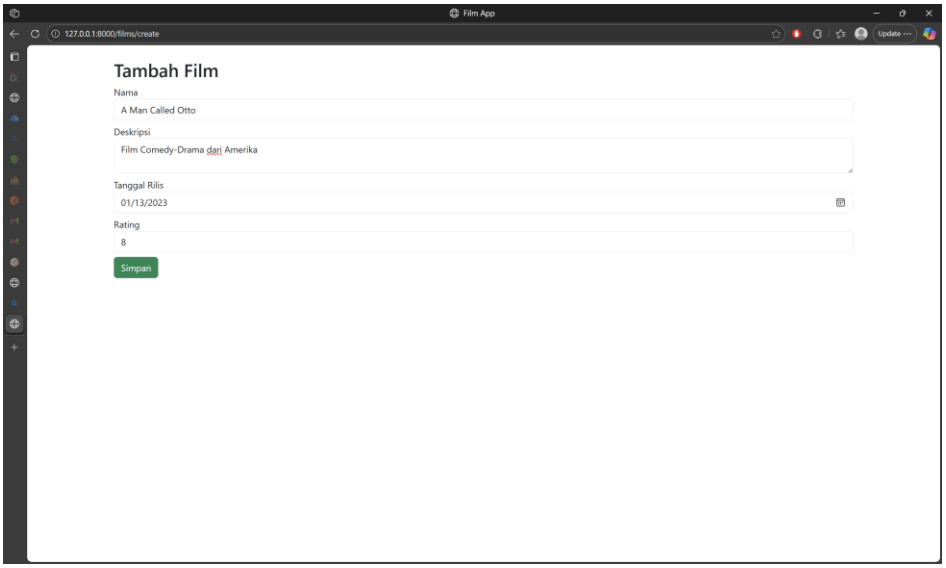

Home Workspaces API Network
Search Postman Ctrl K
Invite Upgrade
My Workspace New Import
Collections Environments History Flows
REST API basics: CRUD, test & variable
GET Get data
POST Post data
PUT Update data
DEL Delete data
Overview GET Get data
No environment
REST API basics: CRUD, test & variable / Get data
Save Share
GET http://localhost:3000/films/2
Send
Docs Params Authorization Headers (9) Body Scripts Settings Cookies
This is a GET request and it is used to "get" data from an endpoint. There is no request body for a GET request, but you can use query parameters to help specify the resource you want data on (e.g.,
Body Cookies Headers (7) Test Results (1/1)
200 OK 8 ms 361 B Save Response
{ } JSON Preview Visualize
{
"id": 2,
"nama": "Man In Love",
"deskripsi": "Film Drama Romance Taiwan",
"tanggal_rilis": "2021-08-19T17:00:00.000Z",
"rating": 9.5
}
Files BETA
Cloud View Find and replace Console Terminal
Runner Start Proxy Cookies Vault Trash

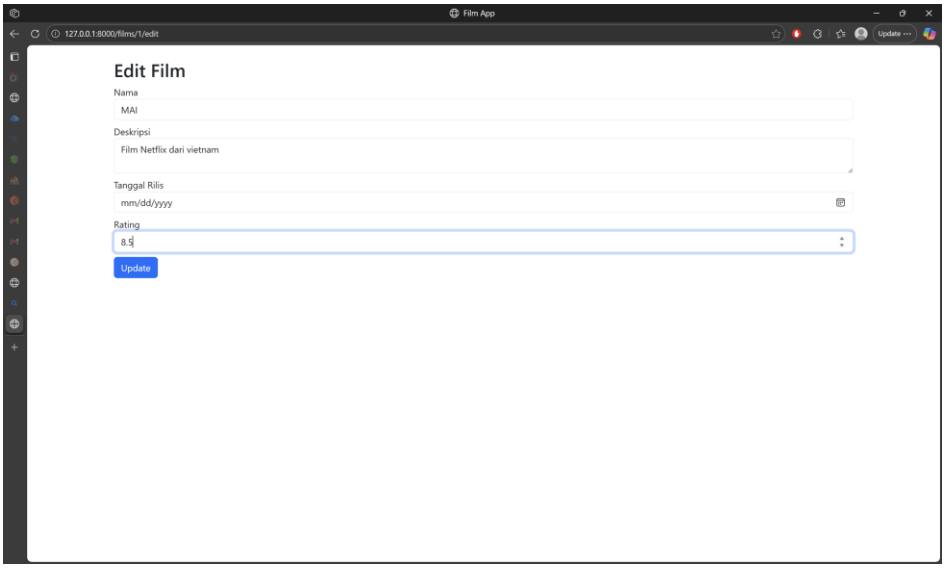

Home Workspaces API Network
Search Postman Ctrl K
Invite Upgrade
My Workspace New Import
Collections Environments History Flows
REST API basics: CRUD, test & variable
GET Get data
POST Post data
PUT Update data
DEL Delete data
Overview PUT Get data
No environment
REST API basics: CRUD, test & variable / Get data
Save Share
PUT http://localhost:3000/films/2
Send
Docs Params Authorization Headers (9) Body Scripts Settings Cookies
none form-data x-www-form-urlencoded raw binary GraphQL JSON
Schema Beautify
{
"nama": "Man In Love",
"deskripsi": "Film Drama Romance Taiwan yang tayang di netflix",
"tanggal_rilis": "2021-08-20",
"rating": 9.5
}
Body Cookies Headers (7) Test Results (1/1)
200 OK 7 ms 273 B Save Response
{ } JSON Preview Visualize
{
"message": "Film berhasil diperbarui"
}
Files BETA
Cloud View Find and replace Console Terminal
Runner Start Proxy Cookies Vault Trash

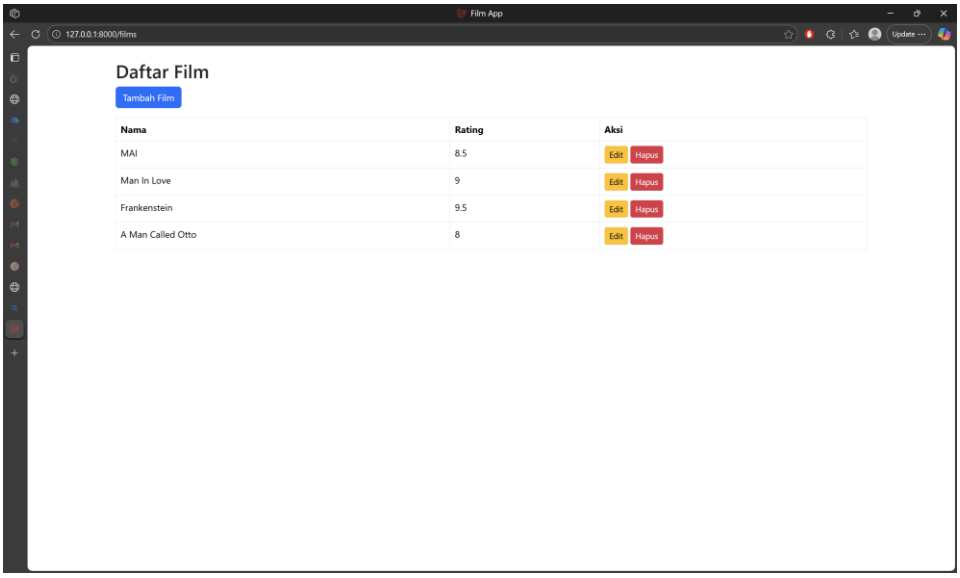

Home Workspaces API Network
Search Postman Ctrl K
Invite Upgrade
My Workspace New Import
Collections Environments History Flows
REST API basics: CRUD, test & variable
GET Get data
POST Post data
PUT Update data
DEL Delete data
Overview DEL Get data
No environment
REST API basics: CRUD, test & variable / Get data
Save Share
DELETE http://localhost:3000/films/2
Send
Docs Params Authorization Headers (9) Body Scripts Settings Cookies
none form-data x-www-form-urlencoded raw binary GraphQL JSON
Schema Beautify
Body Cookies Headers (7) Test Results (1/1)
200 OK 10 ms 270 B Save Response
{ } JSON Preview Visualize
{
"message": "Film berhasil dihapus"
}
Files BETA
Cloud View Find and replace Console Terminal
Runner Start Proxy Cookies Vault Trash

Project Laravel

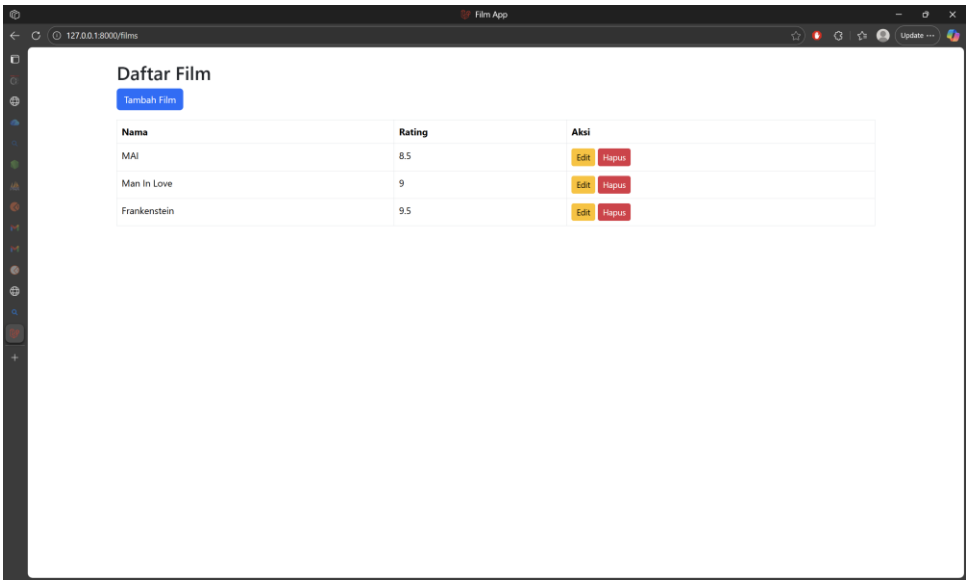








Setelah hapus film ke-4



Code Laravel :

FilmController

```php
<?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
```