

**LAPORAN PRAKTIKUM**  
**JURNAL MODUL 9**



**Nama :**

**Aulia Jasifa Br Ginting      2311104060**

**S1SE-07-02**

**Dosen :**

**Yudha Islami Sulistya, S.Kom., M.Kom**

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

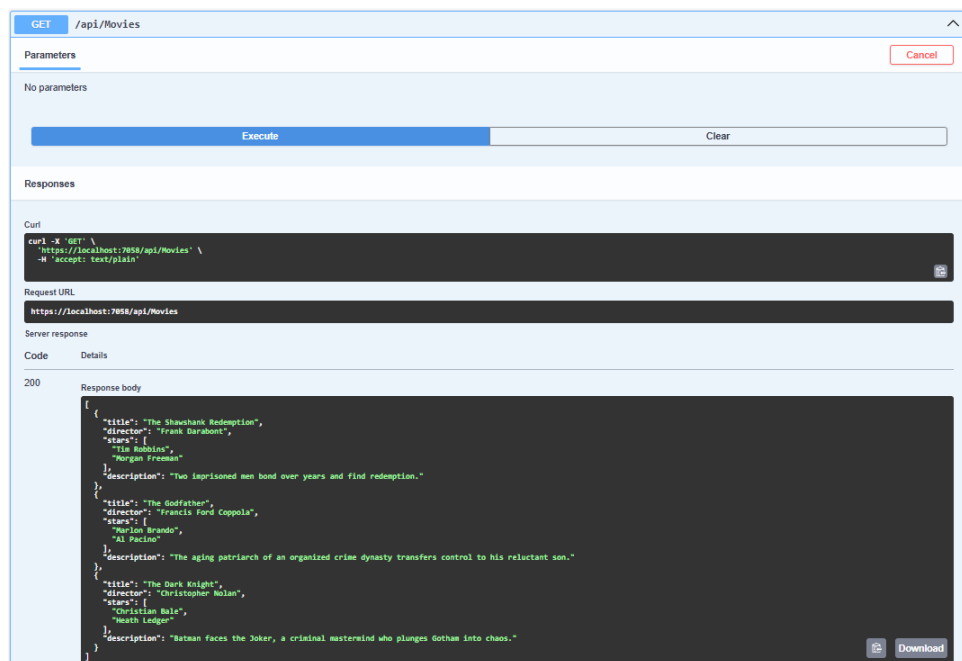
**2025**

## I. Link GitHub

[https://github.com/auliajsf06/KPL\\_Aulia-Jasifa-Br-Ginting\\_2311104060\\_SE-07-02](https://github.com/auliajsf06/KPL_Aulia-Jasifa-Br-Ginting_2311104060_SE-07-02)

## II. MENDEMONSTRASI WEB API

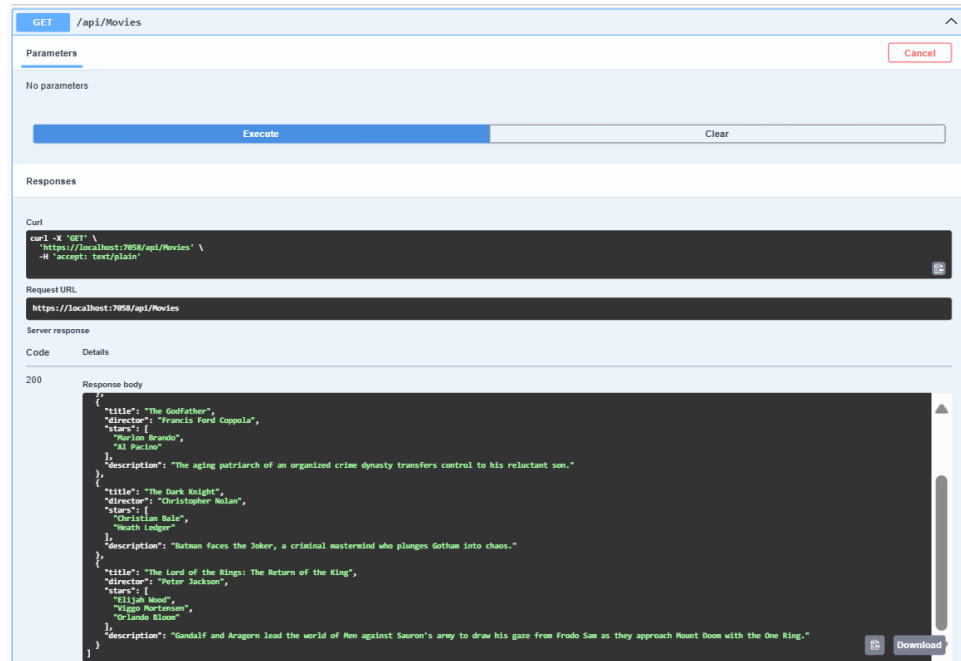
A. Mencoba “GET /api/Movies” saat baru dijalankan yang mengeluarkan list film dari TOP 3 IMDB seperti pada tampilan berikut pada saat dicoba dengan menekan tombol “Try it out” dan tombol “Execute”



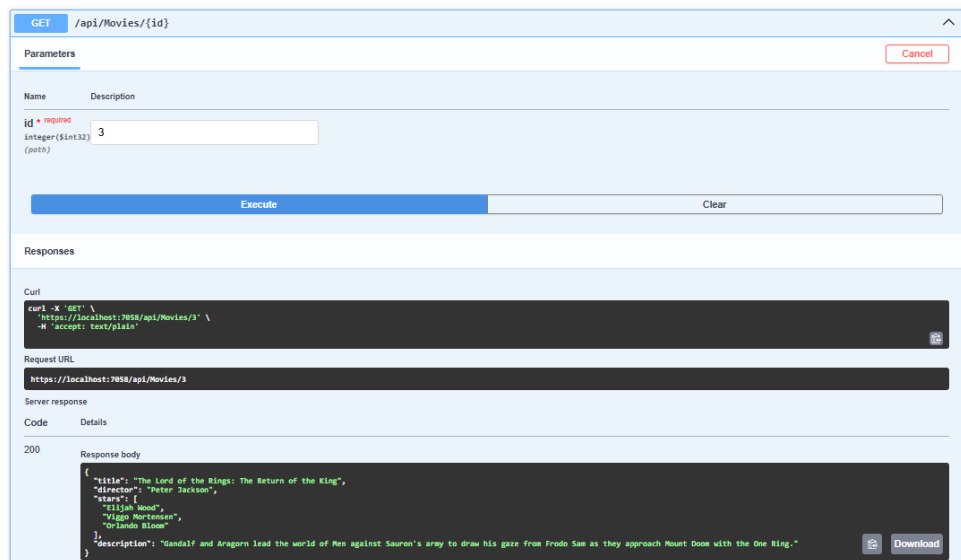
B. Menambahkan Movie baru yaitu urutan ke-4 pada TOP IMDB list dengan memanggil API pada bagian “POST /api/Movies”



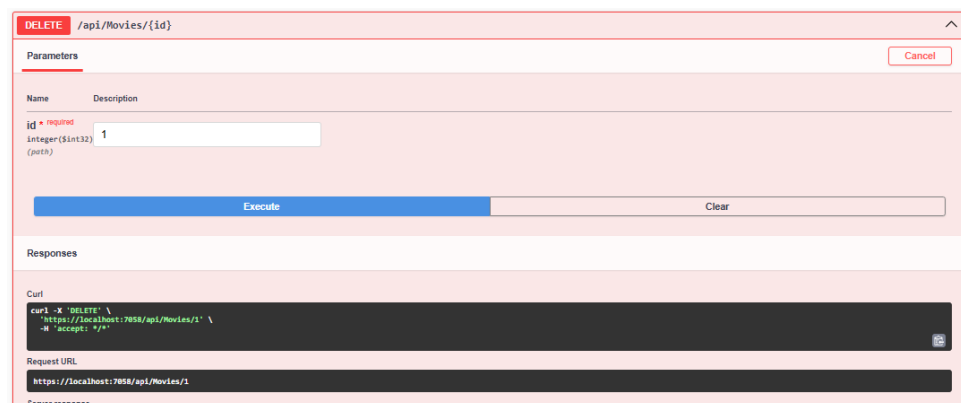
C. Cek list/array dari semua Movie lagi dengan “GET /api/Movies”, pastikan Movie yang baru ditambahkan sebelumnya sudah ada:



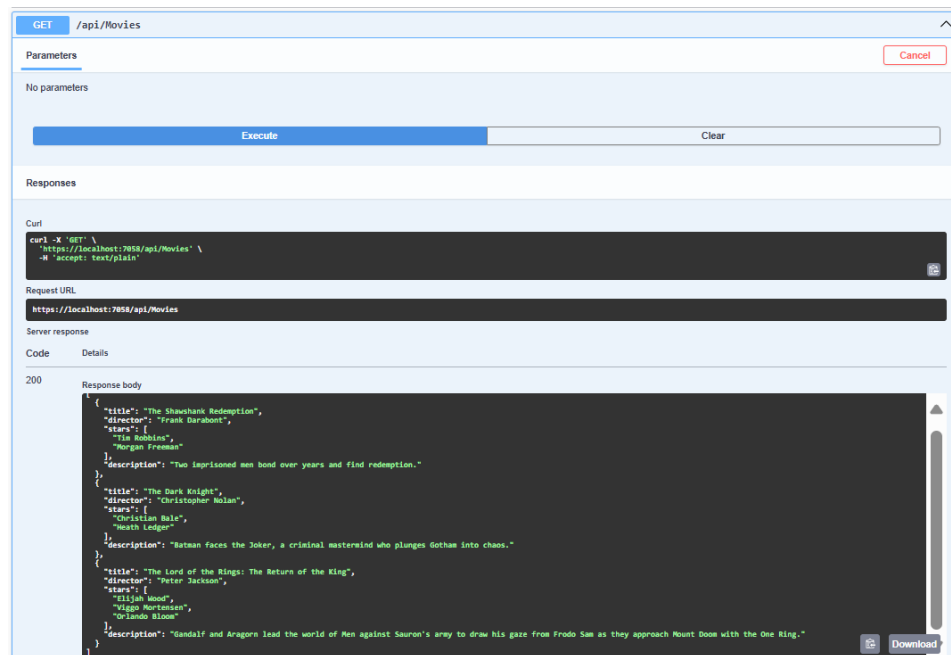
D. Mencoba meminta Movie dengan index 3, “GET /api/Movies/3” yang seharusnya mengembalikan Movie yang baru saja ditambah:



E. Menghapus objek Movie dengan index ke-1 dengan “DELETE /api/Movies/1”



- F. Cek list/array dari semua Movie sekali lagi dengan “GET /api/Movies”, film dengan ranking kedua “Godfather” sudah tidak ada di list:



## G. Penjelasan

Pada praktikum Modul 9 ini, mahasiswa diminta untuk membuat sebuah aplikasi Web API menggunakan ASP.NET Core di Visual Studio 2022. Project yang dibuat harus diberi nama sesuai format yaitu modul9\_NIM. Dalam proses pembuatan, mahasiswa diminta untuk memilih template ASP.NET Core Web API dan memastikan fitur OpenAPI (Swagger) diaktifkan agar dokumentasi API dapat diakses melalui browser. Setelah project selesai dibuat, mahasiswa diminta untuk mengimplementasikan sebuah API dengan model data Movie yang memiliki atribut Title, Director, Stars (dalam bentuk list), dan Description.

API yang dikembangkan memiliki endpoint dasar di /api/Movies dan mendukung empat jenis request: GET untuk mengambil seluruh data film, GET/{id} untuk mengambil data berdasarkan indeks, POST untuk menambahkan film baru, dan DELETE/{id} untuk menghapus data berdasarkan indeks. Data film tidak disimpan dalam database, melainkan dalam sebuah list static, dan diinisialisasi dengan tiga film teratas dari daftar IMDb.

Setelah API berhasil diimplementasikan, mahasiswa diminta untuk menguji fungsionalitas API melalui halaman Swagger. Beberapa skenario pengujian yang wajib dilakukan meliputi: menampilkan daftar film awal menggunakan GET, menambahkan film keempat melalui POST, mengecek kembali daftar film menggunakan GET, mengambil film berdasarkan index 3, menghapus film index 1, dan memverifikasi bahwa film kedua sudah terhapus.

### III. SYNTAX

#### A. Class “Movies.cs”

```
namespace JURNAL9_2311104060.Models
{
    public class Movie
    {
        public string Title { get; set; }
        public string Director { get; set; }
        public List<string> Stars { get; set; }
        public string Description { get; set; }

        public Movie() { }
    }
}
```

#### B. Class “ControllerMovies.cs”

```
using Microsoft.AspNetCore.Mvc;
using JURNAL9_2311104060.Models;

namespace JURNAL9_2311104060.Controllers
{
    [ApiController]
    [Route("api/[controller]")]
    public class MoviesController : ControllerBase
    {
        private static List<Movie> movies = new List<Movie>
        {
            new Movie {
                Title = "The Shawshank Redemption",
                Director = "Frank Darabont",
                Stars = new List<string> { "Tim Robbins", "Morgan Freeman" },
                Description = "Two imprisoned men bond over years and find redemption."
            }
        }
    }
}
```

```

    },
    new Movie {
        Title = "The Godfather",
        Director = "Francis Ford Coppola",
        Stars = new List<string> { "Marlon Brando", "Al Pacino" },
        Description = "The aging patriarch of an organized crime dynasty transfers
control to his reluctant son."
    },
    new Movie {
        Title = "The Dark Knight",
        Director = "Christopher Nolan",
        Stars = new List<string> { "Christian Bale", "Heath Ledger" },
        Description = "Batman faces the Joker, a criminal mastermind who plunges
Gotham into chaos."
    }
};

```

```

[HttpGet]
public ActionResult<List<Movie>> GetMovies()
{
    return Ok(movies);
}

```

```

[HttpGet("{id}")]
public ActionResult<Movie> GetMovie(int id)
{
    if (id < 0 || id >= movies.Count) return NotFound();
    return Ok(movies[id]);
}

```

```

[HttpPost]
public ActionResult AddMovie([FromBody] Movie movie)

```

```
{  
    movies.Add(movie);  
    return Ok();  
}  
  
[HttpDelete("{id}")]  
public ActionResult DeleteMovie(int id)  
{  
    if (id < 0 || id >= movies.Count) return NotFound();  
    movies.RemoveAt(id);  
    return Ok();  
}  
}
```