TUGAS JURNAL
KONSTRUKSI PERANGKAT LUNAK

MODUL VIII

Runtime_Configuration_dan_Internationalization

Disusun Oleh :

Ahmad Junaidi / 2211104002

SE-06-01

Asisten Praktikum:

Naufal El Kamil Aditya Pratama Rahman

Imelda

Dosen Pengampu :

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

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

TUGAS JURNAL

1. **Link Github Repository:**
   https://github.com/Ahmadjunaidi101105/KPL_AHMAD-JUNAIDI_2211104002_SE0601/tree/main/09_API_Design_dan_Construction_Using_Swagger
2. **MEMBUAT PROJECT CONSOLE/TANPA GUI**

3. **IMPLEMENTASI WEB API**
   - File Program.cs

```
var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddControllers();
// Learn more about configuring Swagger/OpenAPI at https://aka.ms/aspnetcore/swashbuckle
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();

var app = builder.Build();

// Configure the HTTP request pipeline.
if (app.Environment.IsDevelopment())
{
    app.UseSwagger();
    app.UseSwaggerUI();
}

app.UseHttpsRedirection();

app.UseAuthorization();

app.MapControllers();

app.Run();
```

   - File MoviesController.cs

```
2211104002 ▾ modul9_NIM.Controllers.MoviesController ▾ Movies
using Microsoft.AspNetCore.Mvc;
using modul9_2211104002;
using modul9_NIM.Models;
using System.Collections.Generic;

namespace modul9_NIM.Controllers
{
    [ApiController]
    [Route("api/[controller]")]
    0 references
    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 a number of years, finding solace and eventual redemption through acts of common decency."
            },
            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 of his clandestine empire to his reluctant son."
            },
            new Movie
            {
                Title = "The Dark Knight",
                Director = "Christopher Nolan",
                Stars = new List<string> { "Christian Bale", "Heath Ledger" },
                Description = "When the menace known as the Joker wreaks havoc and chaos on the people of Gotham..."
            }
        };

        [HttpGet]
        0 references
        public ActionResult<List<Movie>> GetAll()
        {
            return Ok(Movies);
        }
```

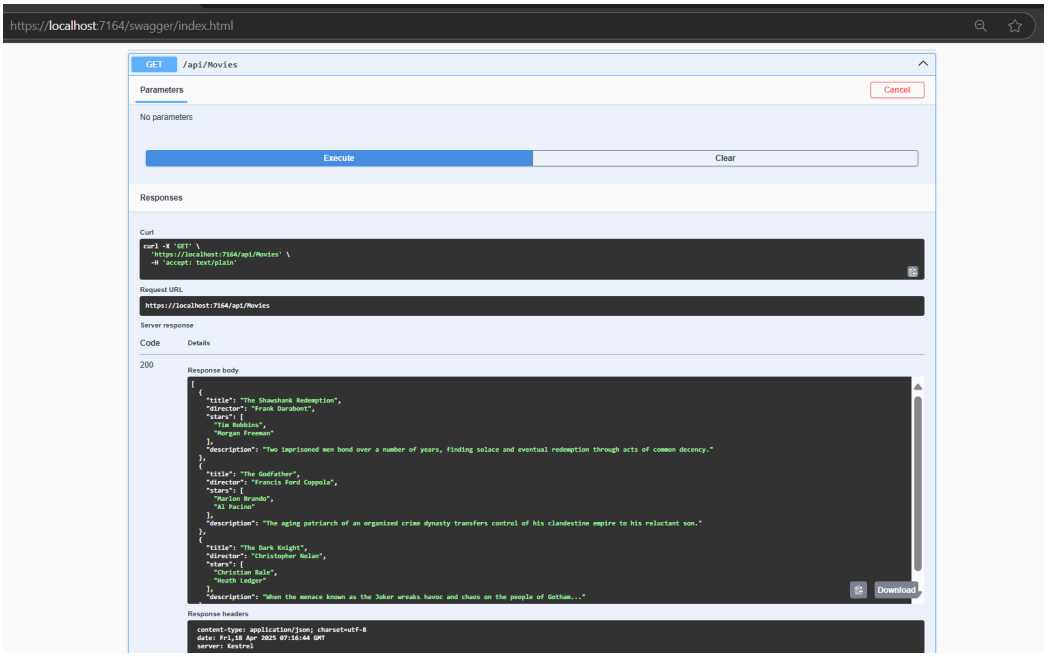
```
        {
            return Ok(Movies);
        }

        [HttpGet("{id}")]
        0 references
        public ActionResult<Movie> Get(int id)
        {
            if (id < 0 || id >= Movies.Count)
                return NotFound("Movie not found");

            return Ok(Movies[id]);
        }

        [HttpPost]
        0 references
        public ActionResult Add(Movie movie)
        {
            Movies.Add(movie);
            return Ok(new { message = "Movie added successfully" });
        }

        [HttpDelete("{id}")]
        0 references
        public ActionResult Delete(int id)
        {
            if (id < 0 || id >= Movies.Count)
                return NotFound("Movie not found");

            Movies.RemoveAt(id);
            return Ok(new { message = "Movie deleted successfully" });
        }
    }
}
```

- Movie.cs

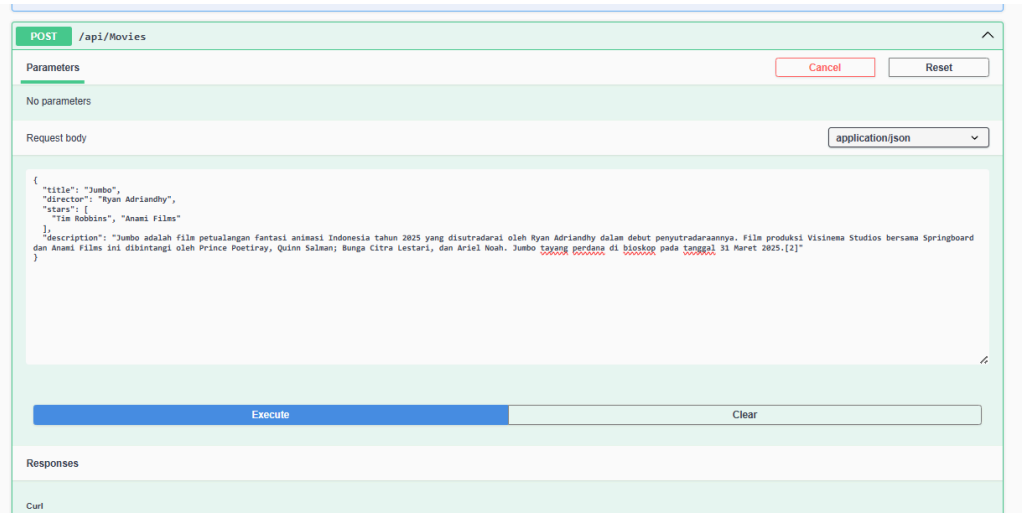
```
dul9_2211104002                                          modul9
1   using System.Collections.Generic;
2
3   namespace modul9_NIM.Models
4   {
        8 references
5       public class Movie
6       {
            3 references
7           public string Title { get; set; }
            3 references
8           public string Director { get; set; }
            3 references
9           public List<string> Stars { get; set; }
            3 references
10          public string Description { get; set; }
11      }
12  }
13
```

4. MENDEMONSTRASI WEB API

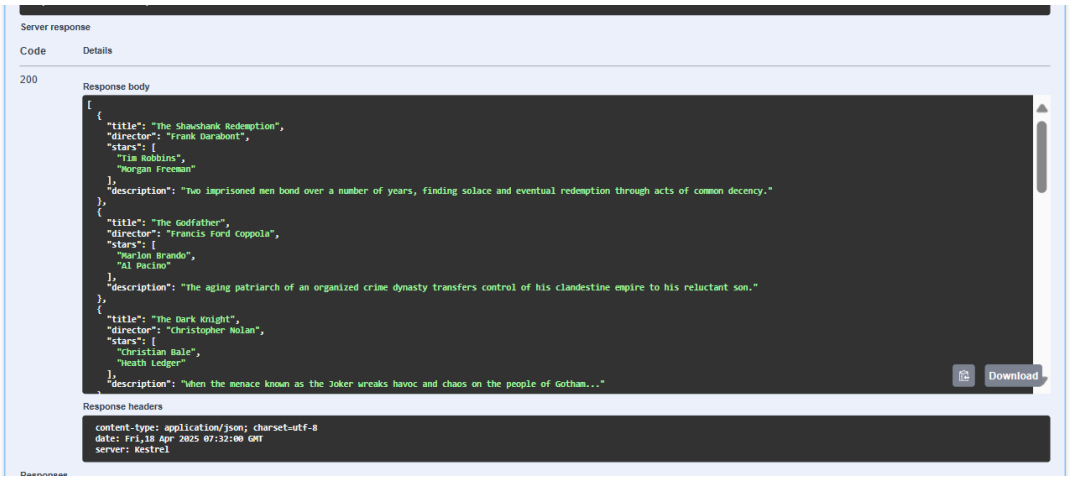
- Mencoba “GET /api/Movies” saat baru dijalankan yang mengeluarkan list film dari TOP 3 IMDB seperti pada tampilan berikut pada saat dicoba dengan menekan tombol “Try it out” dan tombol “Execute”



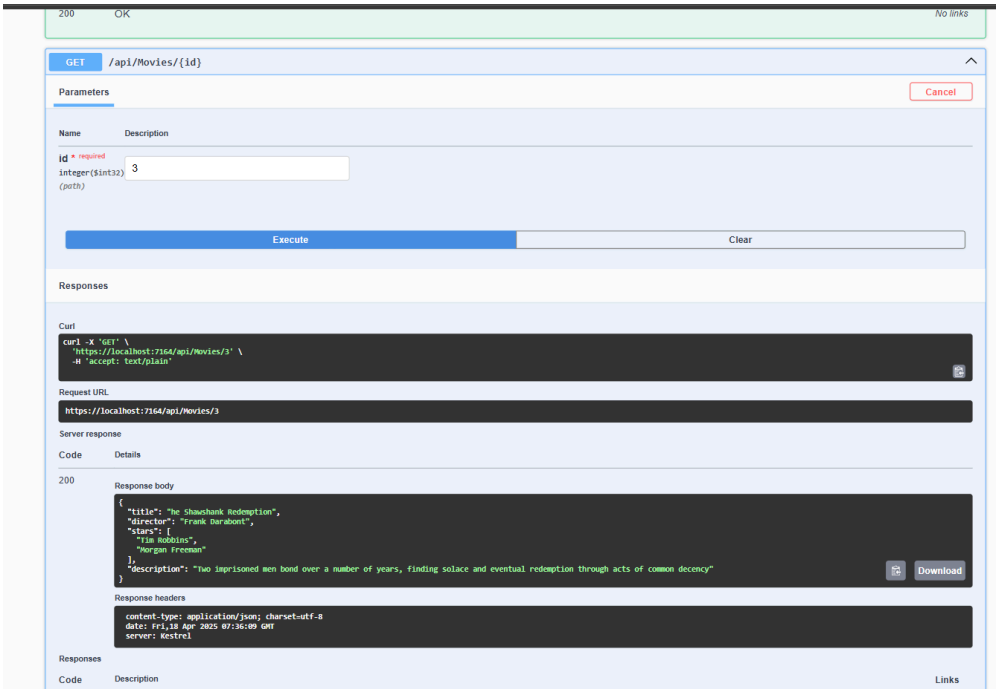
- Menambahkan Movie baru yaitu urutan ke-4 pada TOP IMDB list dengan memanggil API pada bagian “POST /api/Movies”



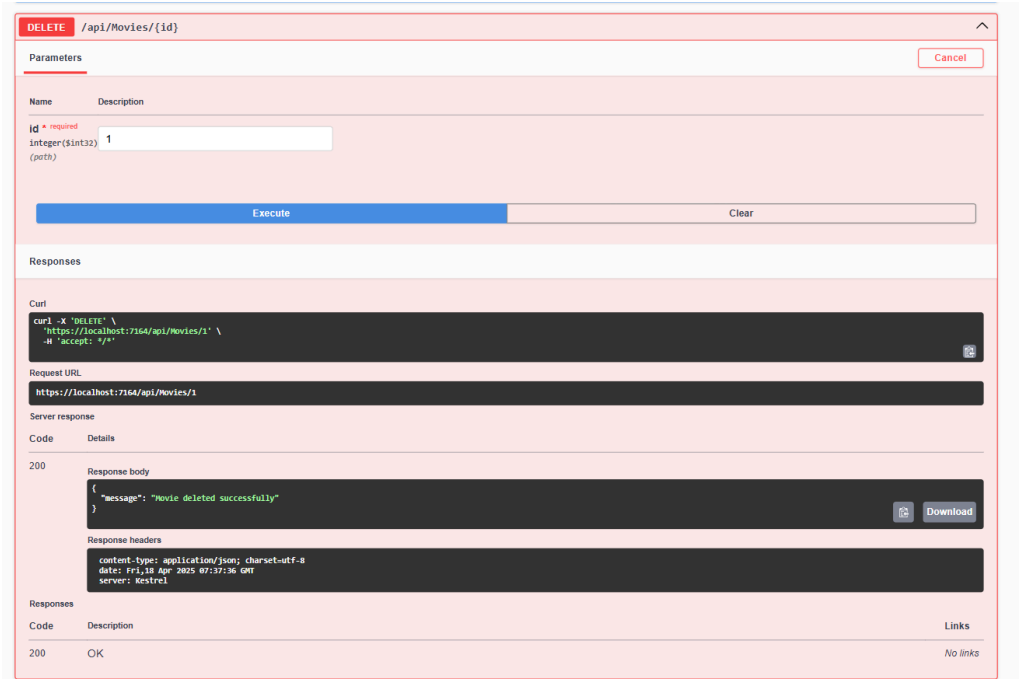
- Cek list/array dari semua Movie lagi dengan “GET /api/Movies”, pastikan Movie yang baru ditambahkan sebelumnya sudah ada:



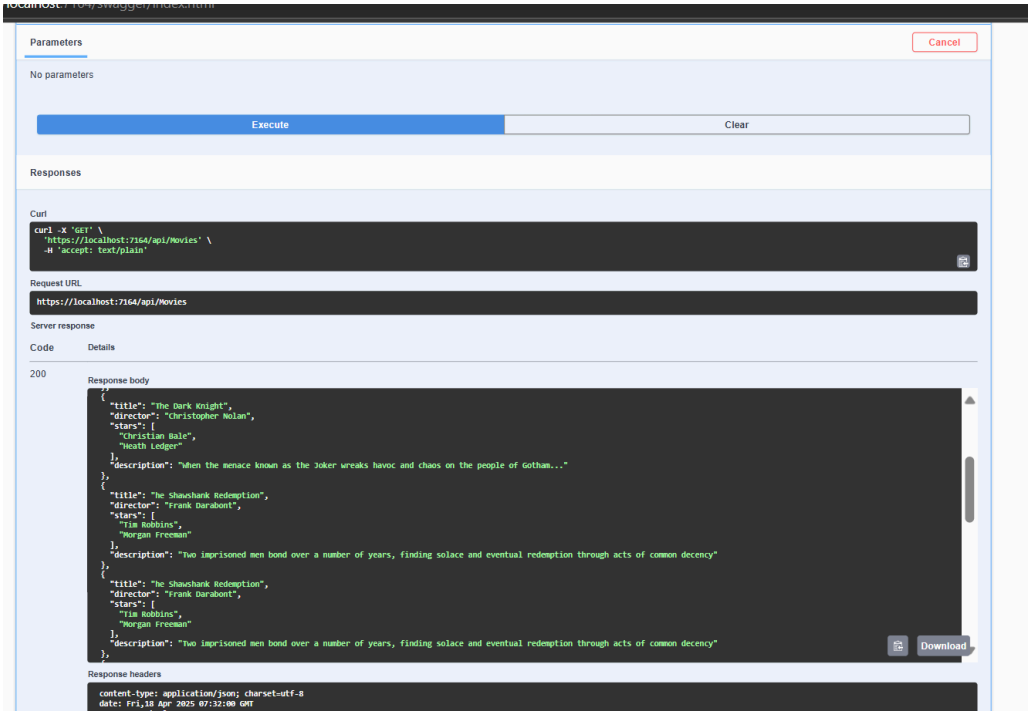
- Mencoba meminta Movie dengan index 3, “GET /api/Movies/3” yang seharusnya mengembalikan Movie yang baru saja ditambah:



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



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



Penjelasan Prorgam :

Program ini merupakan implementasi dari Web API menggunakan ASP.NET Core yang berfungsi untuk mengelola data film (Movie) secara sederhana tanpa menggunakan basis data. API ini dibuat sebagai bagian dari praktikum Modul 9 untuk mempelajari desain dan konstruksi API menggunakan Swagger. Dalam program ini, terdapat sebuah kelas bernama Movie yang merepresentasikan data film dengan atribut berupa judul film (Title), sutradara (Director), daftar pemeran utama (Stars), dan deskripsi film (Description). Data film disimpan menggunakan list statis (static List<Movie>) sehingga dapat diakses dan dimodifikasi selama program berjalan. Pengelolaan data dilakukan melalui MoviesController, yaitu sebuah controller yang menyediakan empat endpoint utama berbasis REST API. Endpoint tersebut meliputi GET /api/Movies untuk menampilkan seluruh data film, GET /api/Movies/{id} untuk menampilkan data film berdasarkan indeks, POST /api/Movies untuk menambahkan data film baru, serta DELETE /api/Movies/{id} untuk menghapus data film tertentu berdasarkan indeks. Semua data awal diinisialisasi dengan tiga film teratas dari daftar IMDb. Program ini juga mendukung dokumentasi otomatis melalui Swagger yang dapat diakses pada alamat https://localhost:<port>/swagger/index.html, sehingga pengujian API dapat dilakukan secara langsung melalui antarmuka web. Dengan pendekatan ini, pengembangan dan pengujian API menjadi lebih efisien dan terstruktur.