

**JURNAL
KONSTRUKSI PERANGKAT LUNAK**

**PERTEMUAN 8
API DESIGN DAN CONSTRUCTION
USING SWAGGER**



Disusun Oleh :
Muhammad Abdul Aziz
2211104026
SE0601

Asisten Praktikum :
Naufal El Kamil Aditya Pratama Rahman
Imelda

Dosen Pengampu :
Yudha Islami Sulistya, S.Kom., M.Cs.

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025

1. Link Github Repository

https://github.com/akuazizz/KPL_MUHAMMAD-ABDUL-AZIZ_2211104026_SE0601/tree/main/09_API_Design_dan_Construction_Using_Swagger/JURNAL

2. Screenshot hasil run (hasil console output dari proses run sesuai bagian 4) Penjelasan singkat dari kode implementasi yang dibuat (beserta screenshot dari potongan source code yang dijelaskan).

a. Implementasi Web API

Source Code

Program.cs :

```
modul9_2211104026
1  var builder = WebApplication.CreateBuilder(args);
2
3  // Add services to the container.
4
5  builder.Services.AddControllers();
6  // Learn more about configuring Swagger/OpenAPI at https://aka.ms/aspnetcore/swashbuckle
7  builder.Services.AddEndpointsApiExplorer();
8  builder.Services.AddSwaggerGen();
9
10 var app = builder.Build();
11
12 // Configure the HTTP request pipeline.
13 if (app.Environment.IsDevelopment())
14 {
15     app.UseSwagger();
16     app.UseSwaggerUI();
17 }
18
19 app.UseHttpsRedirection();
20
21 app.UseAuthorization();
22
23 app.MapControllers();
24
25 app.Run();
```

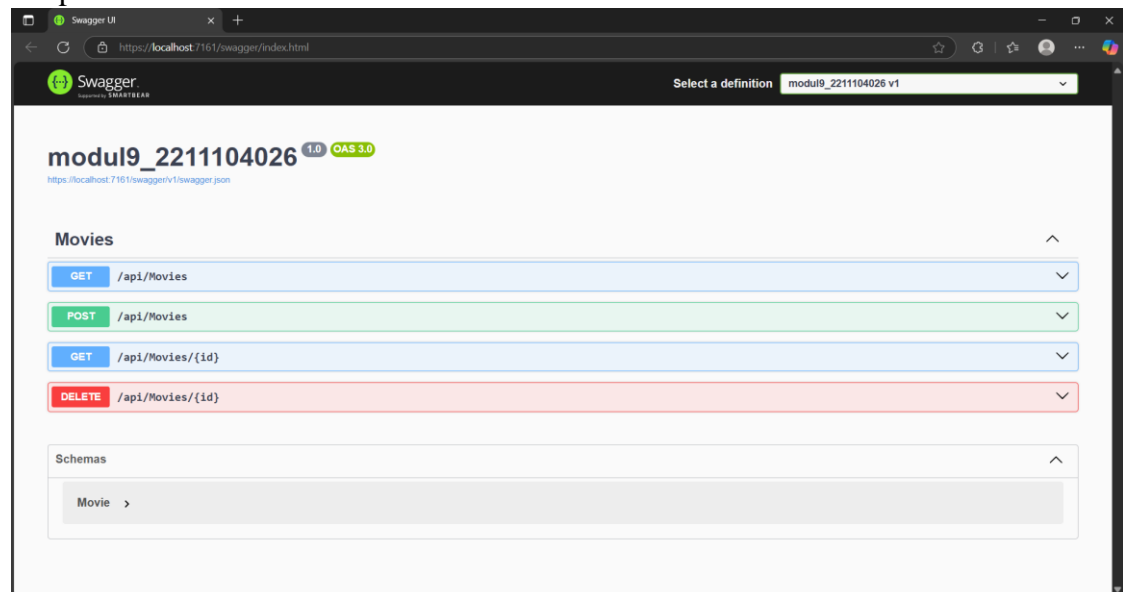
Movie.cs :

```
modul9_2211104026 modul9_2211104026.Movie
1  namespace modul9_2211104026
2  {
3      7 references
4      public class Movie
5      {
6          3 references
7          public string Title { get; set; }
8          3 references
9          public string Director { get; set; }
10         3 references
11         public List<string> Stars { get; set; }
12         3 references
13         public string Description { get; set; }
14
15         3 references
16         public Movie() { }
17     }
18 }
```

MovieController.cs :

```
1 namespace modul9_2211104026
2 {
3     using Microsoft.AspNetCore.Mvc;
4
5     namespace modul9_NIM.Controllers
6     {
7         [ApiController]
8         [Route("api/[controller]")]
9         public class MoviesController : ControllerBase
10         {
11             private static List<Movie> movies = new List<Movie>
12             {
13                 new Movie
14                 {
15                     Title = "The Shawshank Redemption",
16                     Director = "Frank Darabont",
17                     Stars = new List<string> { "Tim Robbins", "Morgan Freeman" },
18                     Description = "A banker convicted of uxoricide forms a friendship over a quarter century with a hardened convict, while maintaining his innocence."
19                 },
20                 new Movie
21                 {
22                     Title = "The Godfather",
23                     Director = "Francis Ford Coppola",
24                     Stars = new List<string> { "Marlon Brando", "Al Pacino" },
25                     Description = "The aging patriarch of an organized crime dynasty transfers control of his clandestine empire to his reluctant son."
26                 },
27                 new Movie
28                 {
29                     Title = "The Dark Knight",
30                     Director = "Christopher Nolan",
31                     Stars = new List<string> { "Christian Bale", "Heath Ledger" },
32                     Description = "When a menace known as the Joker wreaks havoc and chaos on the people of Gotham, Batman, James Gordon and Harvey Dent must work together to stop him."
33                 }
34             };
35
36             [HttpGet]
37             public IActionResult GetMovies() => Ok(movies);
38
39             [HttpGet("{id}")]
40             public IActionResult GetMovie(int id)
41             {
42                 if (id < 0 || id >= movies.Count)
43                     return NotFound();
44                 return Ok(movies[id]);
45             }
46
47             [HttpPost]
48             public IActionResult AddMovie([FromBody] Movie movie)
49             {
50                 movies.Add(movie);
51                 return Ok(movie);
52             }
53
54             [HttpDelete("{id}")]
55             public IActionResult DeleteMovie(int id)
56             {
57                 if (id < 0 || id >= movies.Count)
58                     return NotFound();
59                 var removed = movies[id];
60                 movies.RemoveAt(id);
61                 return Ok(removed);
62             }
63         }
64     }
65 }
```

Output :



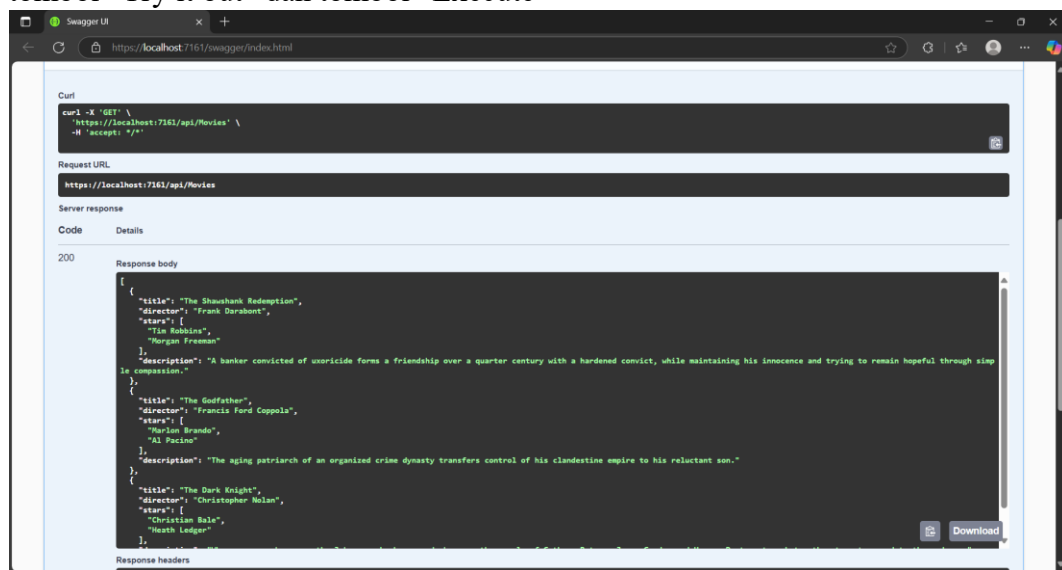
Penjelasan :

Aplikasi Web API ini dibuat menggunakan ASP.NET Core dengan struktur dasar yang terdiri dari file Program.cs, class Movie, dan controller MoviesController. File Program.cs berfungsi sebagai titik masuk aplikasi, yang mengatur layanan seperti controller dan Swagger untuk dokumentasi API. Di dalamnya, juga terdapat konfigurasi agar aplikasi dapat mengenali dan menjalankan semua endpoint dari controller. Class Movie merepresentasikan data film dengan properti seperti Title,

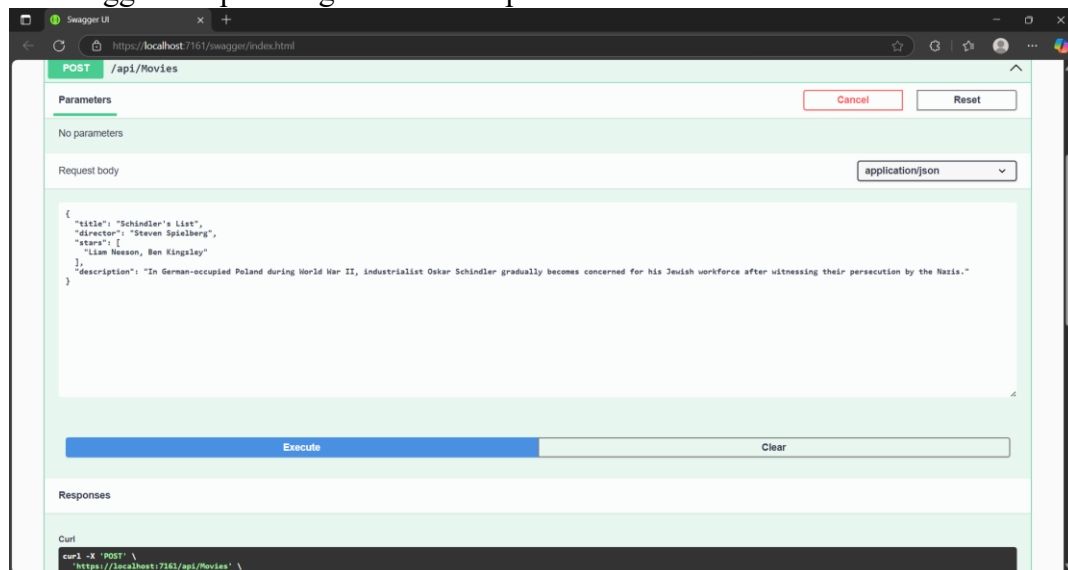
Director, Stars, dan Description. Sementara itu, MoviesController adalah komponen utama yang menangani permintaan HTTP melalui endpoint /api/Movies. Controller ini memiliki lima metode yaitu: GET untuk mengambil semua data film, GET/{id} untuk mengambil film berdasarkan indeks, POST untuk menambahkan film baru, dan DELETE/{id} untuk menghapus film berdasarkan indeks. Data film disimpan dalam list statis, tanpa menggunakan database, dan secara default sudah diisi dengan tiga film terbaik dari IMDb. Swagger digunakan sebagai antarmuka untuk menguji semua fungsi API secara langsung di browser.

b. Demonstrasi Web API

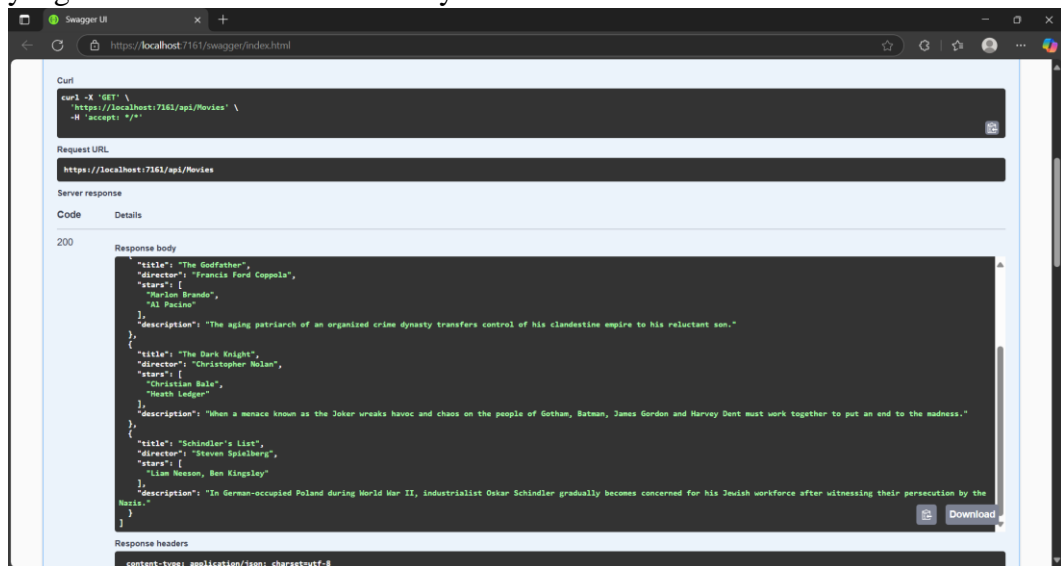
- a. Mencoba “GET /api/Movies” saat baru dijalankan yang mengeluarkan list film dari TOP 3 IMDB seperti pada tampilan berikut pada saat dicoba dengan menekan tombol “Try it out” dan tombol “Execute”



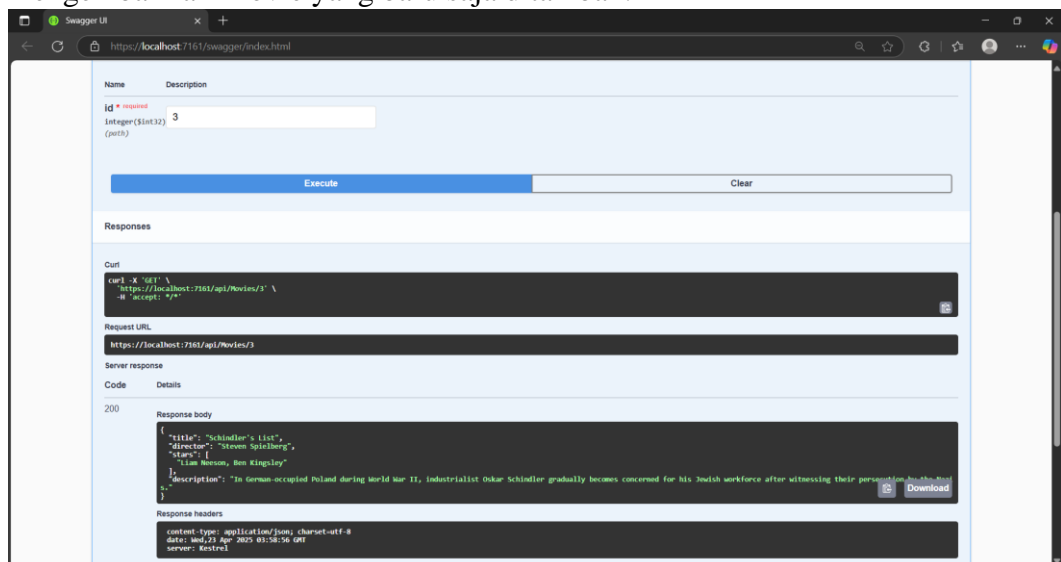
- b. Menambahkan Movie baru yaitu urutan ke-4 pada TOP IMDB list dengan memanggil API pada bagian “POST /api/Movies”



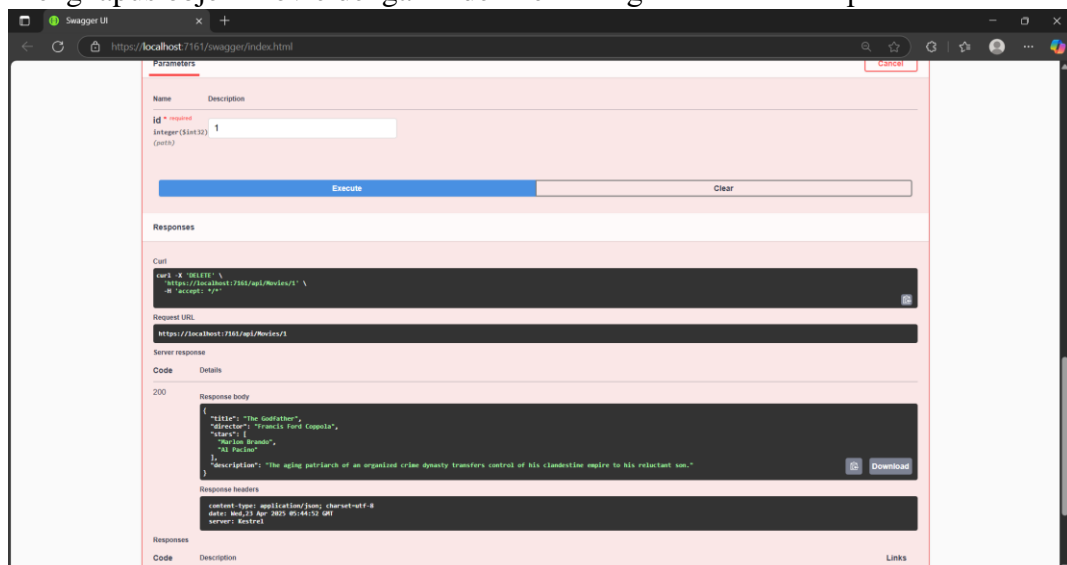
- c. Cek list/array dari semua Movie lagi dengan “GET /api/Movies”, pastikan Movie yang baru ditambahkan sebelumnya sudah ada:



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



- 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:

