

**JURNAL MANDIRI 9**  
**KONSTRUKSI PERANGKAT LUNAK**  
**S1 SOFTWARE ENGINEERING**



**ILHAM LII ASSIDAQ**

**2311104068**

**SE0702**

**DIREKTORAT TELKOM UNIVERSITY PURWOKERTO**

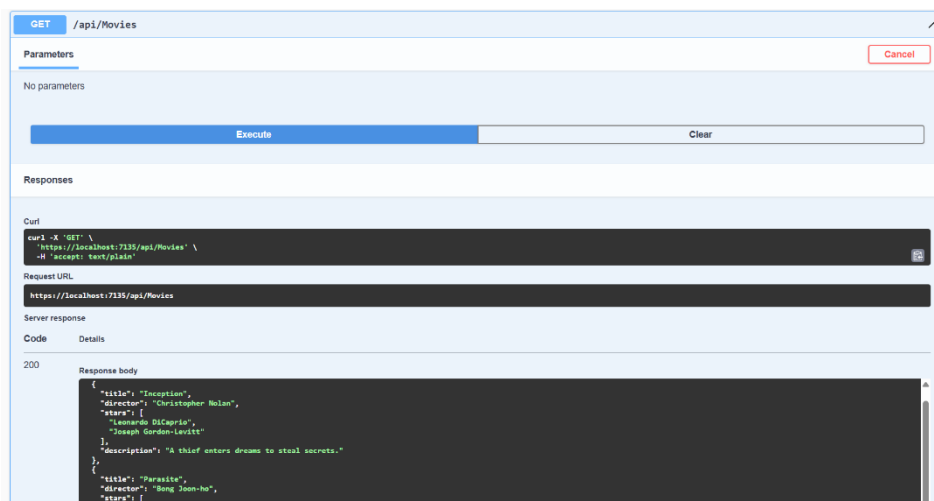
1. Link GitHub : [https://github.com/ilhamlii21/KPL\\_ILHAM\\_LII\\_ASSIDAQ-2311104068\\_S1SE-07-02](https://github.com/ilhamlii21/KPL_ILHAM_LII_ASSIDAQ-2311104068_S1SE-07-02)

## 2. A. MovieController.cs

```
2 namespace JMMODUL9
3 {
4     [ApiController]
5     [Route("api/[controller]")]
6     0 references
7     public class MoviesController : ControllerBase
8     {
9         private static List<Movie> Movies = new List<Movie>
10         {
11             new Movie { Title = "Inception", Director = "Christopher Nolan", Stars = new List<string> { "Leonardo DiCaprio", "Joseph Gordon-Levitt" }, Description = "A thief enters dreams to steal secrets." },
12             new Movie { Title = "Parasite", Director = "Bong Joon-ho", Stars = new List<string> { "Song Kang-ho", "Choi Woo-shik" }, Description = "A poor family schemes to become rich by infiltrating a wealthy household." },
13             new Movie { Title = "Spirited Away", Director = "Hayao Miyazaki", Stars = new List<string> { "Rumi Hiiragi", "Miyu Irino" }, Description = "A girl is transported to a spirit world and must work to free herself and her parents." },
14         };
15
16         [HttpGet]
17         0 references
18         public ActionResult<List<Movie>> Get() => Movies;
19
20         [HttpGet("{id}")]
21         0 references
22         public ActionResult<Movie> Get(int id) => (id < 0 || id >= Movies.Count) ? NotFound() : Movies[id];
23
24         [HttpPost]
25         0 references
26         public ActionResult Post([FromBody] Movie movie)
27         {
28             Movies.Add(movie);
29             return Ok();
30         }
31
32         [HttpDelete("{id}")]
33         0 references
34     }
```

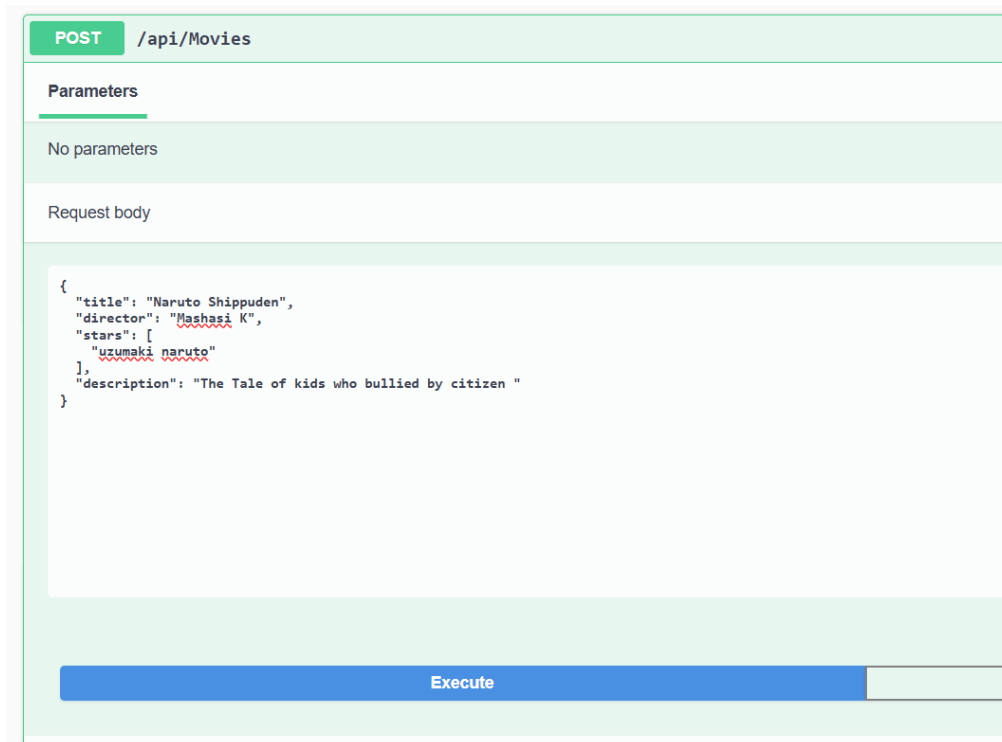
Kode di atas merupakan implementasi controller ASP.NET Core bernama MoviesController yang menyediakan RESTful API untuk mengelola daftar film (Movie). Controller ini memiliki empat endpoint utama: GET tanpa parameter untuk mengambil seluruh daftar film, GET dengan parameter indeks (id) untuk mengambil satu film berdasarkan posisinya dalam daftar, POST untuk menambahkan data film baru ke dalam list Movies, serta DELETE untuk menghapus film berdasarkan indeks yang diberikan. Data film disimpan sementara dalam list statis, yang berarti hanya tersedia selama server berjalan. Fitur ini cocok untuk keperluan pengujian atau prototipe awal sebelum integrasi ke basis data sesungguhnya.

## B. API GET



Endpoint GET `/api/Movies` pada API ini digunakan untuk mengambil seluruh daftar film yang tersedia dalam koleksi server. Saat dieksekusi, server akan merespons dengan kode status 200 OK dan mengembalikan data dalam format JSON yang berisi array objek film, di mana setiap objek memiliki atribut `title`, `director`, `stars`, dan `description`. Dalam contoh respons, terdapat tiga film: *Inception*, *Parasite*, dan *Spirited Away*, masing-masing dengan sutradara, daftar aktor, dan deskripsi singkatnya. Endpoint ini cocok digunakan oleh frontend untuk menampilkan seluruh data film yang tersedia kepada pengguna.

### C. API POST



The screenshot shows a REST client interface with a green header bar. On the left, a green button labeled 'POST' is next to the endpoint '/api/Movies'. Below the header, there are three sections: 'Parameters' (showing 'No parameters'), 'Request body' (showing a JSON object), and 'Execute' (a blue button). The JSON object in the request body is:

```
{
  "title": "Naruto Shippuden",
  "director": "Mashasi K",
  "stars": [
    "uzumaki naruto"
  ],
  "description": "The Tale of kids who bullied by citizen "
}
```

Endpoint POST `/api/Movies` digunakan untuk menambahkan data film baru ke dalam daftar koleksi film yang disimpan di server. Pada permintaan ini, klien mengirimkan data JSON yang berisi informasi film seperti `title`, `director`, `stars` (daftar aktor), dan `description`. Contoh film yang ditambahkan adalah *Naruto Shippuden* yang disutradarai oleh *Mashasi K* dengan pemeran *Uzumaki Naruto* dan deskripsi cerita tentang anak-anak yang dibully oleh warga. Setelah permintaan dikirim, server memberikan respons dengan kode status 200 OK tanpa isi badan respons, yang menunjukkan bahwa data berhasil ditambahkan meskipun tidak memberikan pesan balasan secara eksplisit.

### D. API GET by index film

GET
/api/Movies/{id}

Parameters

Name	Description
id * required	
integer(\$int32)	2
(path)	

Execute

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7135/api/Movies/2' \
  -H 'accept: text/plain'
```

Request URL

```
https://localhost:7135/api/Movies/2
```

Server response

Code	Details
200	<div>Response body</div> <pre>{   "title": "Spirited Away",   "director": "Hayao Miyazaki",   "stars": [     "Rumi Hiiragi",     "Miyu Irino"   ],   "description": "A girl enters a magical world ruled by spirits." }</pre>

Endpoint GET `/api/Movies/{id}` digunakan untuk mengambil data film berdasarkan indeks tertentu dari daftar film yang disimpan di server. Dalam contoh ini, permintaan dilakukan dengan ID 2, dan server merespons dengan detail film *Spirited Away*, disutradarai oleh *Hayao Miyazaki*, serta dibintangi oleh *Rumi Hiiragi* dan *Miyu Irino*. Film ini memiliki deskripsi tentang seorang gadis yang memasuki dunia magis yang dikuasai oleh roh. Respons dari server dikembalikan dalam format JSON dengan status kode 200 OK, yang berarti data berhasil ditemukan dan dikirimkan ke klien.

## E. API DELETE

DELETE
/api/Movies/{id}

Parameters

Cancel

Name	Description
id * required	
integer(\$int32)	1
(path)	

Execute

Clear

Responses

Curl

```
curl -X 'DELETE' \
  'https://localhost:7135/api/Movies/1' \
  -H 'accept: */*'
```

Request URL

```
https://localhost:7135/api/Movies/1
```

Server response

Code	Details
200	<div>Response headers</div> <pre>content-length: 0 date: Fri, 13 Jun 2025 13:24:04 GMT server: Kestrel</pre>

Endpoint DELETE /api/Movies/{id} digunakan untuk menghapus data film berdasarkan indeks yang diberikan dalam URL. Pada contoh ini, permintaan dikirim dengan ID 1, yang berarti film pada indeks ke-1 dalam daftar akan dihapus. Setelah eksekusi, server memberikan respons dengan status kode 200 OK, menunjukkan bahwa proses penghapusan berhasil dilakukan, meskipun tidak ada konten yang dikembalikan (content-length: 0). Proses ini umum digunakan dalam API berbasis REST untuk mengelola data secara dinamis melalui HTTP request.