

Nama : Candra Dinata

Nim : 2311104061

Kelas : SE0702

kode:

```
1 using JURNALMODUL9_2311104061.Models;
2 using Microsoft.AspNetCore.Mvc;
3
4
5 namespace modul9_NIM.Controllers
6 {
7     [Route("api/[controller]")]
8     [ApiController]
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", "Bob Gunton" },
18                Description = "Two imprisoned men bond over a number of years."
19            },
20            new Movie
21            {
22                Title = "The Godfather",
23                Director = "Francis Ford Coppola",
24                Stars = new List<string> { "Marlon Brando", "Al Pacino", "James Caan" },
25                Description = "The aging patriarch of an organized crime dynasty transfers control 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", "Aaron Eckhart" },
32                Description = "When the menace known as the Joker wreaks havoc, Batman must accept one of the greatest psychological and physical tests."
33            }
34        };
35
36        // GET: api/Movies
37        [HttpGet]
38        public ActionResult<IEnumerable<Movie>> GetMovies()
39        {
40            return movies;
41        }
42
43        // GET: api/Movies/1
44        [HttpGet("{id}")]
45        public ActionResult<Movie> GetMovie(int id)
46        {
47            if (id < 0 || id >= movies.Count)
48                return NotFound();
49
50            return movies[id];
51        }
52
53        // POST: api/Movies
54        [HttpPost]
55        public ActionResult<Movie> PostMovie(Movie movie)
56        {
57            movies.Add(movie);
58            return CreatedAtAction(nameof(GetMovie), new { id = movies.Count - 1 }, movie);
59        }
60
61        // DELETE: api/Movies/1
62        [HttpDelete("{id}")]
63        public IActionResult DeleteMovie(int id)
64        {
65            if (id < 0 || id >= movies.Count)
66                return NotFound();
67
68            movies.RemoveAt(id);
69            return NoContent();
70        }
71    }
72 }
73
```

Kode di atas merupakan implementasi dari **Web API controller** bernama MoviesController dalam proyek ASP.NET Core yang bertugas mengelola data film (Movie) secara sementara (tanpa database). Controller ini berada pada route api/Movies dan menggunakan atribut [ApiController] untuk mempermudah validasi dan routing. Di dalamnya terdapat list statis movies yang menyimpan tiga film populer sebagai data awal, masing-masing berisi informasi seperti judul, sutradara, daftar aktor, dan deskripsi. Terdapat empat endpoint utama: GET /api/Movies untuk menampilkan semua

data film, GET /api/Movies/{id} untuk menampilkan detail film berdasarkan indeks, POST /api/Movies untuk menambahkan film baru ke list, serta DELETE /api/Movies/{id} untuk menghapus film berdasarkan indeks. Semua operasi dilakukan langsung di memori melalui list statis movies, tanpa melibatkan database eksternal, sesuai dengan ketentuan tugas. Namespace dan direktori model (Movie.cs) mengacu pada JURNALMODUL9\_2311104061.Models, sehingga pastikan kesesuaian antara file model dan controller.

```
1 namespace JURNALMODUL9_2311104061.Models
2 {
3     public class Movie
4     {
5         public string Title { get; set; }
6         public string Director { get; set; }
7         public List<string> Stars { get; set; }
8         public string Description { get; set; }
9
10        public Movie() { }
11    }
12 }
13
```

Kode di atas mendefinisikan sebuah **class model** bernama Movie dalam namespace JURNALMODUL9\_2311104061.Models, yang digunakan untuk merepresentasikan data film dalam aplikasi Web API. Class ini memiliki empat properti publik: Title (judul film), Director (sutradara), Stars (daftar aktor, disimpan sebagai List<string>), dan Description (deskripsi film). Properti-properti ini menggunakan getter dan setter otomatis agar dapat dengan mudah diakses dan dimodifikasi dalam berbagai proses seperti input dari pengguna atau manipulasi data dalam controller. Selain itu, disediakan konstruktor default (public Movie() { }) yang memungkinkan pembuatan objek Movie tanpa parameter, yang penting agar framework ASP.NET Core dapat melakukan proses binding data JSON secara otomatis saat menerima input dari endpoint POST. Class ini menjadi struktur data utama yang digunakan di controller MoviesController untuk menyimpan, menampilkan, dan mengelola data film.

## Output

a.

Swagger UI interface showing a GET request to `https://localhost:7218/api/Movies`. The request is executed via curl. The server response is a 200 status code with a JSON array of movie data.

```
curl -X 'GET' \
  'https://localhost:7218/api/Movies' \
  -H 'accept: text/plain'
```

Request URL: `https://localhost:7218/api/Movies`

Server response: 200

Response body:

```
{
  {
    "title": "The Shawshank Redemption",
    "director": "Frank Darabont",
    "stars": [
      "Tim Robbins",
      "Morgan Freeman",
      "Bob Gunton"
    ],
    "description": "Two imprisoned men bond over a number of years."
  },
  {
    "title": "The Godfather",
    "director": "Francis Ford Coppola",
    "stars": [
      "Marlon Brando",
      "Al Pacino",
      "James Caan"
    ],
    "description": "The aging patriarch of an organized crime dynasty transfers control to his reluctant son."
  },
  {
    "title": "The Dark Knight",
    "director": "Christopher Nolan",
    "stars": [
      "Christian Bale",
      "Heath Ledger",
      "Aaron Eckhart"
    ],
    "description": "Batman returns to Gotham City to save the city from the evil of the Joker."
  }
]
```

b.

Swagger UI interface showing a POST request to `https://localhost:7218/api/Movies`. The request is executed via curl. The server response is a 201 status code with a JSON object of movie data.

```
curl -X 'POST' \
  'https://localhost:7218/api/Movies' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
    "title": "The Lord of the Rings: The Return of the King",
    "director": "Peter Jackson",
    "stars": [
      "Elijah Wood",
      "Viggo Mortensen",
      "Ian McKellen",
      "Orlando Bloom"
    ],
    "description": "Gandalf and Aragorn lead the World of Men against Sauron's army to draw his gaze from Frodo and Sam as they approach Mount Doom with the One Ring."
  }'
```

Request URL: `https://localhost:7218/api/Movies`

Server response: 201

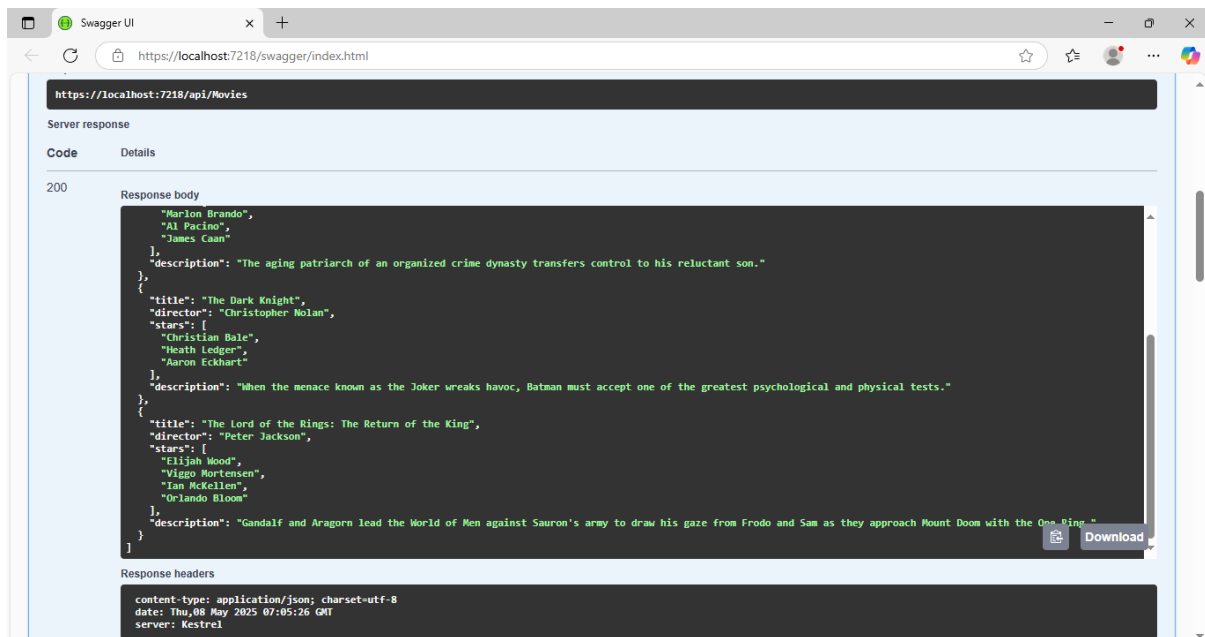
Response body:

```
{
  "title": "The Lord of the Rings: The Return of the King",
  "director": "Peter Jackson",
  "stars": [
    "Elijah Wood",
    "Viggo Mortensen",
    "Ian McKellen",
    "Orlando Bloom"
  ],
  "description": "Gandalf and Aragorn lead the World of Men against Sauron's army to draw his gaze from Frodo and Sam as they approach Mount Doom with the One Ring."
}
```

Response headers:

```
content-type: application/json; charset=utf-8
date: Thu, 08 May 2025 07:04:30 GMT
```

C.



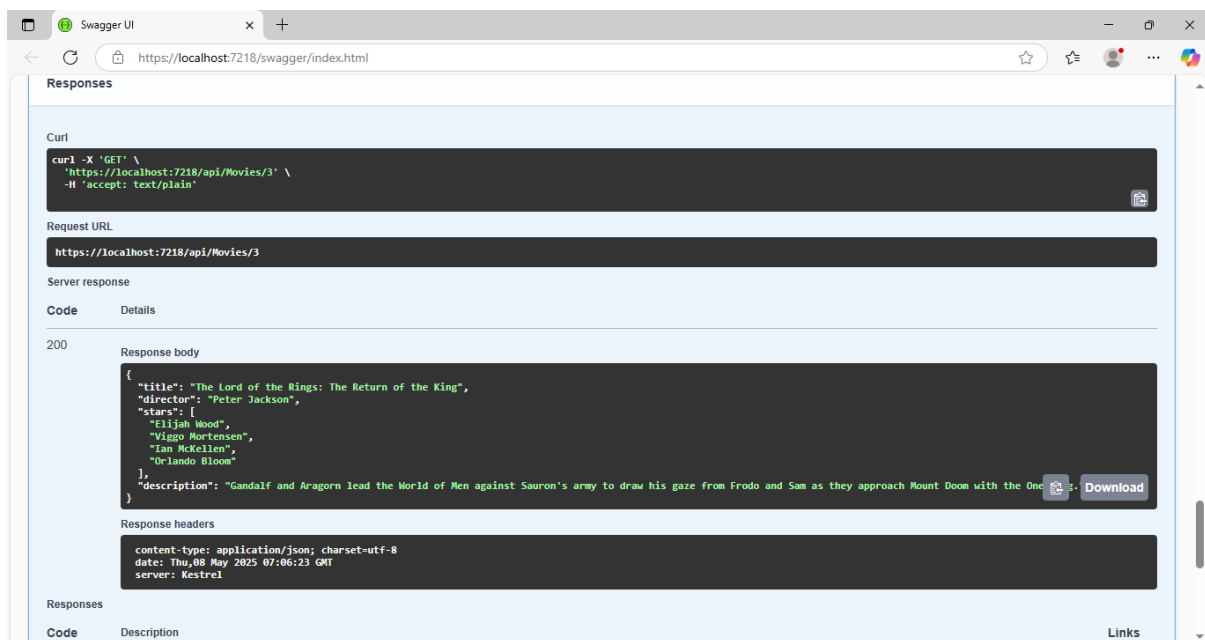
Swagger UI interface showing a GET request to `https://localhost:7218/api/Movies`. The response is a 200 status code with a JSON body. The response body contains an array of movie objects. The first movie is "The Godfather" and the second is "The Dark Knight".

```
["The Godfather", "The Dark Knight", "The Lord of the Rings: The Return of the King"]
```

Response headers:

```
content-type: application/json; charset=utf-8
date: Thu, 08 May 2025 07:05:26 GMT
server: Kestrel
```

d.



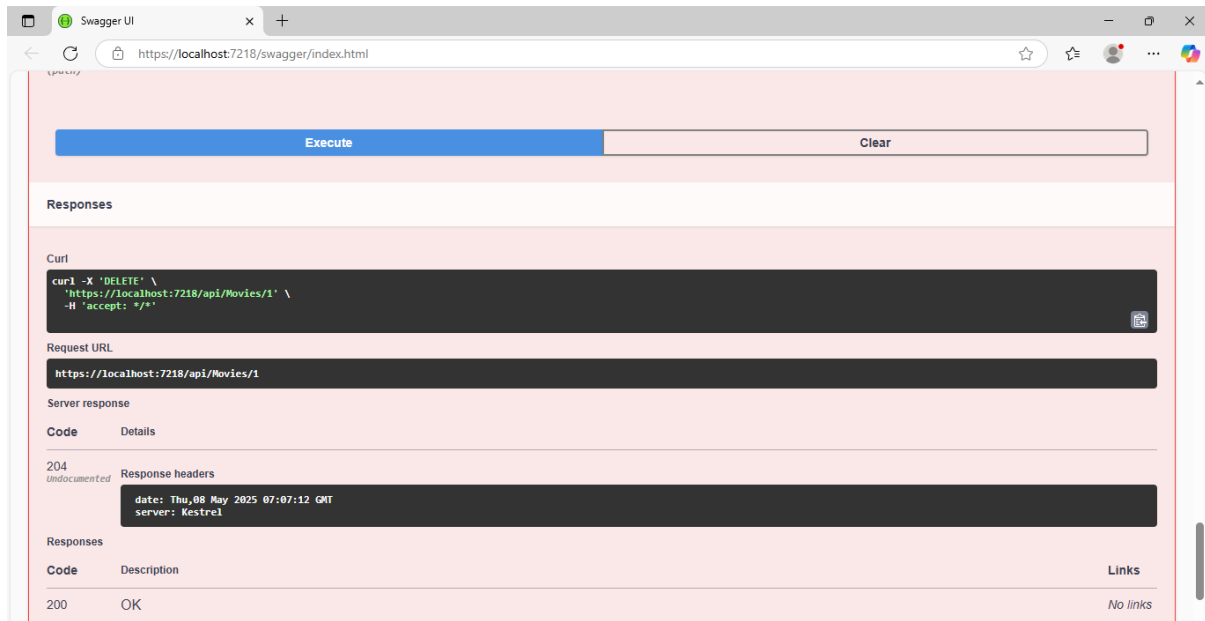
Swagger UI interface showing a GET request to `https://localhost:7218/api/Movies/3`. The response is a 200 status code with a JSON body. The response body contains a single movie object: "The Lord of the Rings: The Return of the King".

```
{ "title": "The Lord of the Rings: The Return of the King", "director": "Peter Jackson", "stars": [ "Elijah Wood", "Viggo Mortensen", "Ian McKellen", "Orlando Bloom" ], "description": "Gandalf and Aragorn lead the World of Men against Sauron's army to draw his gaze from Frodo and Sam as they approach Mount Doom with the One Ring." }
```

Response headers:

```
content-type: application/json; charset=utf-8
date: Thu, 08 May 2025 07:06:23 GMT
server: Kestrel
```

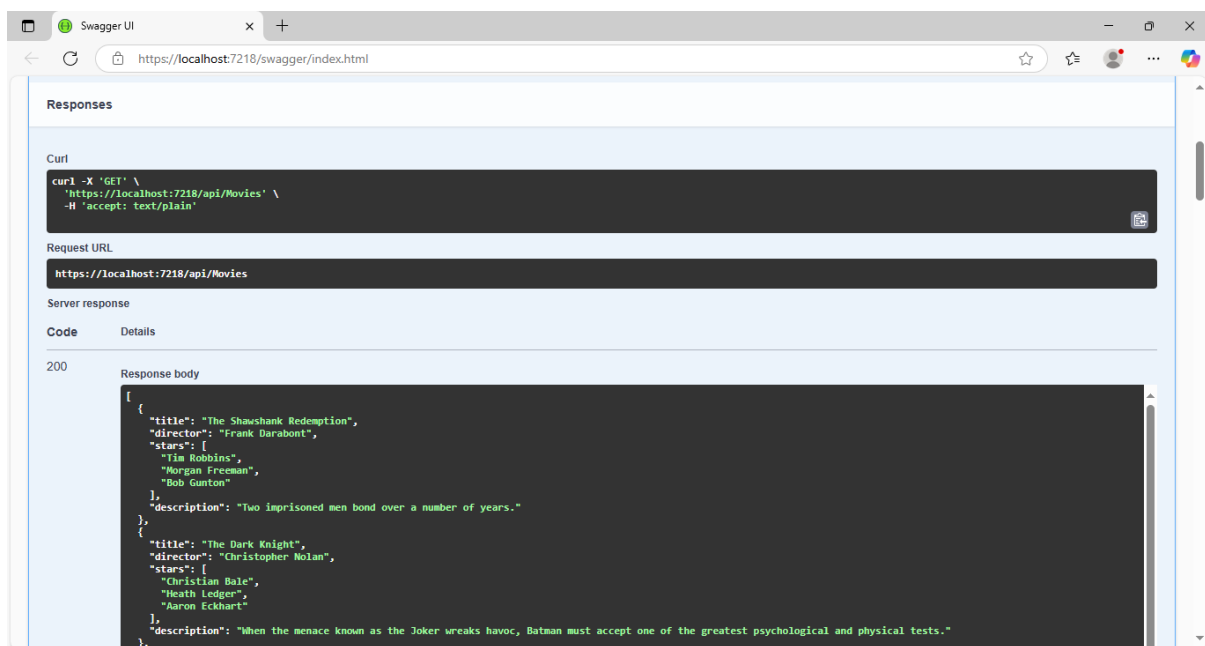
e.



A screenshot of the Swagger UI interface in a web browser. The browser's address bar shows the URL `https://localhost:7218/swagger/index.html`. The interface has a light pink background. At the top, there is a blue "Execute" button and a "Clear" button. Below this, the "Responses" section is active. It displays the following information:

- Curl:** `curl -X 'DELETE' \ 'https://localhost:7218/api/Movies/1' \ -H 'accept: */*'`
- Request URL:** `https://localhost:7218/api/Movies/1`
- Server response:** A table with two columns: "Code" and "Details". It shows a "204" status code with the text "Undocumented" below it.
- Response headers:** A table with two columns: "Header" and "Value". It shows `date: Thu, 09 May 2025 07:07:12 GMT` and `server: Kestrel`.
- Responses:** A table with three columns: "Code", "Description", and "Links". It shows a "200" status code with the description "OK" and "No links" in the Links column.

f.



A screenshot of the Swagger UI interface in a web browser. The browser's address bar shows the URL `https://localhost:7218/swagger/index.html`. The interface has a light blue background. The "Responses" section is active. It displays the following information:

- Curl:** `curl -X 'GET' \ 'https://localhost:7218/api/Movies' \ -H 'accept: text/plain'`
- Request URL:** `https://localhost:7218/api/Movies`
- Server response:** A table with two columns: "Code" and "Details". It shows a "200" status code.
- Response body:** A JSON array of movie objects. The first object is for "The Shawshank Redemption" (director: Frank Darabont, stars: Tim Robbins, Morgan Freeman, Bob Gunton, description: "Two imprisoned men bond over a number of years."). The second object is for "The Dark Knight" (director: Christopher Nolan, stars: Christian Bale, Heath Ledger, Aaron Eckhart, description: "When the menace known as the Joker wreaks havoc, Batman must accept one of the greatest psychological and physical tests.").