

## Laporan Tugas Jurnal Modul 9

### 1. Movie.cs

```
namespace modul9_2311104064.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; }
}
```

#### Penjelasan:

kode ini membuat sebuah "template" atau "cetakan" untuk data film. Setiap kali kamu ingin menyimpan informasi tentang sebuah film dalam programmu, kamu bisa membuat objek baru dari kelas Movie ini. Objek tersebut akan memiliki tempat untuk menyimpan judul, sutradara, daftar bintang, dan deskripsinya.

### 2. MoviesController.cs

```
using Microsoft.AspNetCore.Mvc;
using modul9_2311104064.Models;

namespace modul9_2311104064.Controllers;

[ApiController]
[Route("api/[controller]")]
public class MoviesController : ControllerBase
{
    private static List<Movie> movies = new()
    {
        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."
        },
        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."
        },
        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."
    }
};

[HttpGet]
public ActionResult<List<Movie>> GetAll() => Ok(movies);

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

[HttpPost]
public IActionResult Add(Movie movie)
{
    movies.Add(movie);
    return Ok("Movie berhasil ditambahkan.");
}

[HttpDelete("{id}")]
public IActionResult Delete(int id)
{
    if (id < 0 || id >= movies.Count) return NotFound();
    movies.RemoveAt(id);
    return Ok("Movie berhasil dihapus.");
}
}

```

Penjelasam:

kode ini menciptakan sebuah layanan web sederhana yang memungkinkan pengguna untuk melihat, menambah, dan menghapus data film melalui antarmuka HTTP. Ini adalah contoh dasar dari bagaimana API dibangun menggunakan ASP.NET Core.

### 3. Hasil Run

Movies

GET

/api/Movies

Try it out

Parameters

No parameters

Responses

Curl

curl -X 'GET' \
'https://localhost:7257/api/Movies' \
-H 'accept: text/plain'

Request URL

https://localhost:7257/api/Movies

Server response

Code

Details

200

Response body

[

curl -X 'GET' \
'https://localhost:7257/api/Movies' \
-H 'accept: text/plain'

Request URL

https://localhost:7257/api/Movies

Server response

Code

Details

200

Response body

{
 {
 "title": "The Shawshank Redemption",
 "director": "Frank Darabont",
 "stars": [
 "Tim Robbins",
 "Morgan Freeman"
 ],
 "description": "Two imprisoned men bond over a number of years."
 },
 {
 "title": "The Godfather",
 "director": "Francis Ford Coppola",
 "stars": [
 "Marlon Brando",
 "Al Pacino"
 ],
 "description": "The aging patriarch of an organized crime dynasty transfers control."
 },
 {
 "title": "The Dark Knight",
 "director": "Christopher Nolan",
 "stars": [
 "Christian Bale",
 "Heath Ledger"
 ],
 "description": "Batman faces the Joker, a criminal mastermind."
 }
}

Response headers

content-type: application/json; charset=utf-8
date: Thu, 05 Jun 2025 05:11:51 GMT
server: Kestrel

Responses

Code

Description

Links