

TUGAS JURNAL
KONSTRUKSI PERANGKAT LUNAK
MODUL IX
API DESIGN & CONSTRUCTION USING SWAGGER



Disusun Oleh:
Lintang Suminar Tyas Wening
2211104009
SE0601
Dosen Pengampu:
Yudha Islami Sulistya, S.Kom., M.Cs.

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025

1. IMPLEMENTASI WEB API

- A. Dari master/main branch dan class utama, buatlah program/aplikasi web API dari spesifikasi sebagai berikut ini:

Movie
+ Title : string
+ Director : string
+ Stars : List<string>
+ Description: string
+ Movie()

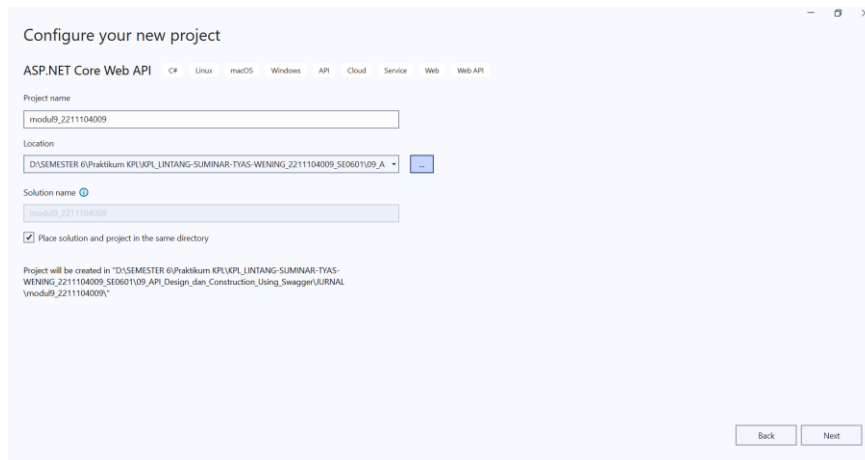
- B. API yang dibuat mempunyai lokasi sebagai berikut **‘/api/Movies**, URL domain boleh dari port mana saja (port bebas). Dengan menggunakan swagger API tersebut dapat menerima RESTful API dengan metoda sebagai berikut (halaman swagger dapat diakses pada <https://localhost:<PORT>/swagger/index.html>):

Movies	
GET	/api/Movies
POST	/api/Movies
GET	/api/Movies/{id}
DELETE	/api/Movies/{id}

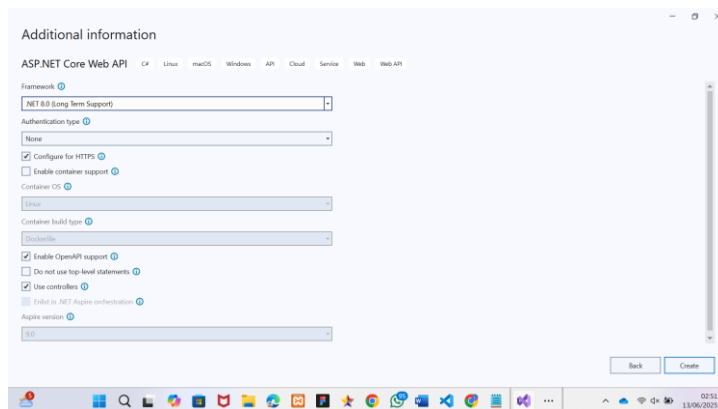
- GET /api/Movies: mengembalikan output berupa list/array dari semua objek Movies
 - ii. GET /api/Movies/{id}: mengembalikan output berupa objek Movie untuk index “id”
 - iii. POST /api/Movies: menambahkan objek Movie baru
 - iv. DELETE /api/Movies/{id}: menghapus objek Movie pada index “id”
- C. Secara default, program yang dibuat memiliki list film yang berasal dari TOP 3 film IMDB dari link: https://www.imdb.com/search/title/?groups=top_100&sort=user_rating,desc
- D. Implementasi yang dibuat tidak menggunakan database, cukup disimpan sebagai suatu variable, dan gunakan “static” di variable tersebut yang menyimpan list/array dari objek-objek Movie.
- E. Dalam pembuatan program/aplikasi ini, anda dapat mengasumsikan bahwa input dari user selalu benar dan sesuai dengan tipe data yang diharapkan.

Jawab :

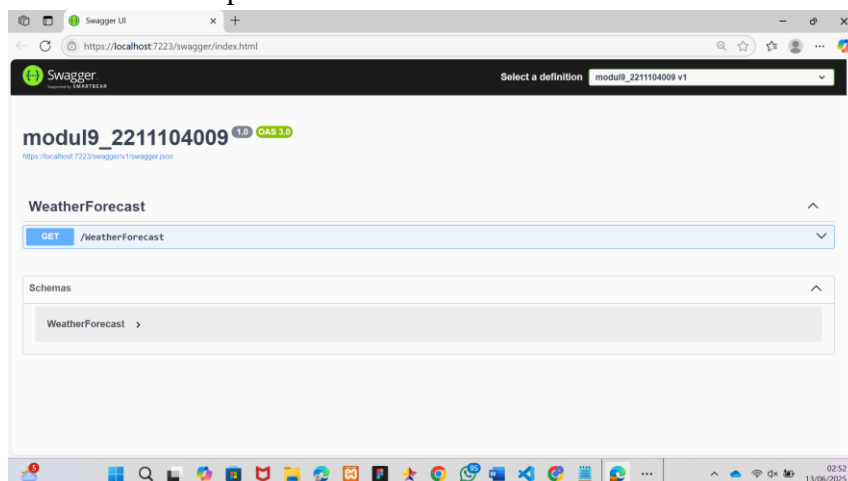
- Membuat file



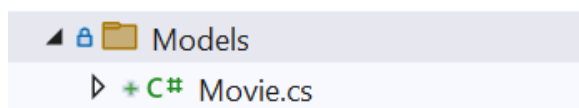
- Lalu klik create



- Lalu muncul tampilan berikut ini saat di run



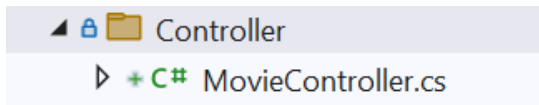
- Membuat Folder Models



- Source Code Movie.cs

```
09_API_Design_dan_Construction_Using_Swagger > JURNAL > modul9_2211104009 > Models > Movie.cs
1  using System.Collections.Generic;
2
3
4  namespace modul9_2211104009.Models
5  {
6      public class Movie
7      {
8          public string Title { get; set; }
9          public string Director { get; set; }
10         public List<string> Stars { get; set; }
11         public string Description { get; set; }
12     }
13 }
```

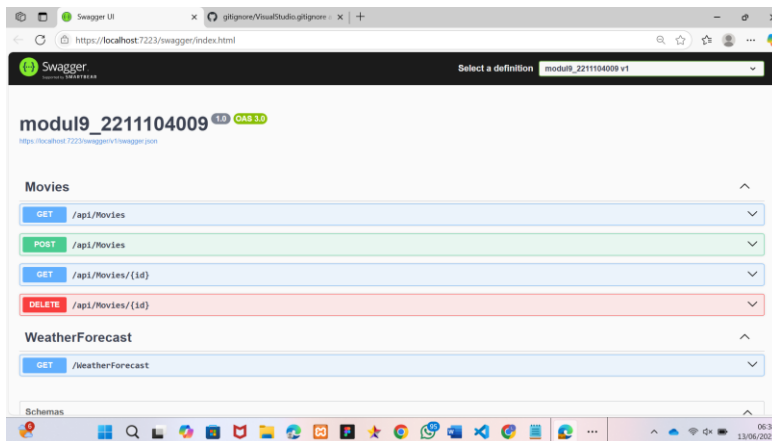
- Membuat Folder Controller



- Source Code MovieController.cs

```
09_API_Design_dan_Construction_Using_Swagger > JURNAL > modul9_2211104009 > Controller > MovieController.cs
1  using Microsoft.AspNetCore.Mvc;
2  using modul9_2211104009.Models;
3  using System.Collections.Generic;
4
5  namespace modul9_2211104009.Controller
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                 Title = "The Godfather",
15                 Director = "Francis Ford Coppola",
16                 Stars = new List<string> { "Marlon Brando", "Al Pacino" },
17                 Description = "The aging patriarch of an organized crime dynasty..."
18             },
19             new Movie {
20                 Title = "The Shawshank Redemption",
21                 Director = "Frank Darabont",
22                 Stars = new List<string> { "Tim Robbins", "Morgan Freeman" },
23                 Description = "Two imprisoned men bond over a number of years..."
24             },
25             new Movie {
26                 Title = "The Dark Knight",
27                 Director = "Christopher Nolan",
28                 Stars = new List<string> { "Christian Bale", "Heath Ledger" },
29                 Description = "Batman faces the Joker in a battle for Gotham City."
30             }
31         };
32
33         [HttpGet]
34         public ActionResult<IEnumerable<Movie>> Get() => movies;
35
36         [HttpGet("{id}")]
37         public ActionResult<Movie> Get(int id)
38         {
39             if (id < 0 || id >= movies.Count)
40                 return NotFound();
41             return movies[id];
42         }
43
44         [HttpPost]
45         public ActionResult Post([FromBody] Movie newMovie)
46         {
47             movies.Add(newMovie);
48             return Ok();
49         }
50
51         [HttpDelete("{id}")]
52         public ActionResult Delete(int id)
53         {
54             if (id < 0 || id >= movies.Count)
55                 return NotFound();
56             movies.RemoveAt(id);
57             return Ok();
58         }
59     }
60 }
```

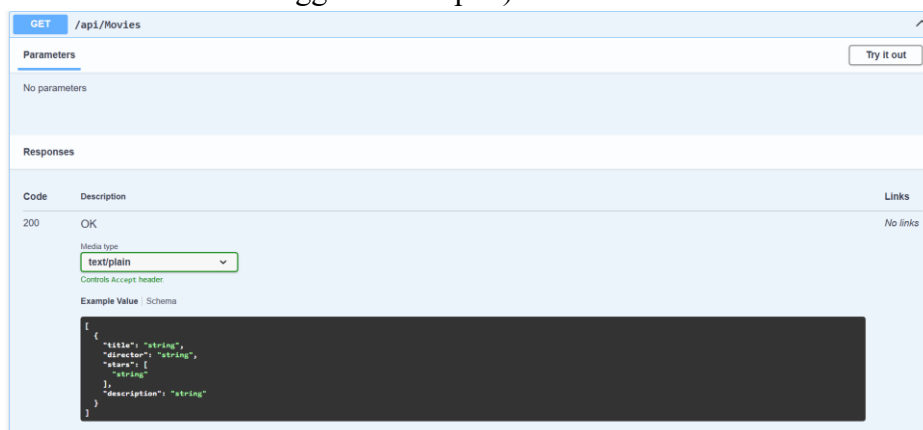
- Hasil output



- Penjelasan

Pada codingan nya membuatkan sebuah *Web API* berbasis ASP.NET Core yang didefinisikan dalam kelas *MoviesController* untuk mengelola data film. Data disimpan dalam list statis movies, yang berisi beberapa film lengkap dengan judul, sutradara, daftar pemeran, dan deskripsi. API ini menyediakan empat endpoint: GET untuk mengambil seluruh data film, GET/{id} untuk mengambil film berdasarkan indeks, POST untuk menambahkan film baru dari *request body*, dan DELETE/{id} untuk menghapus film berdasarkan indeks

2. MENDEMONSTRASI WEB API Beberapa skenario yang harus dicoba untuk memastikan jika program telah berjalan dengan baik. Buatlah dokumen yang berisi semua screenshot dari hasil uji coba scenario yang disebutkan pada list berikut ini:
 - A. Mencoba “GET /api/mahasiswa” saat baru dijalankan (mengeluarkan list nama mahasiswa dan nim anggota kelompok)



GET /api/Movies

Parameters

No parameters

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7223/api/Movies' \
  -H 'accept: text/plain'
```

Request URL

https://localhost:7223/api/Movies

Server response

Code Details

200

Response body

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

Response headers

```
content-type: application/json; charset=utf-8
date: Thu, 12 Jun 2025 20:07:50 GMT
server: Kestrel
```

Responses

Code	Description	Links
200	OK	No links

Media type: text/plain

Controls Accept header

Example Value Schema

```
{
  "title": "string",
  "director": "string",
  "stars": [
    "string"
  ],
  "description": "string"
}
```

B. Menambahkan mahasiswa => Nama: “John Doe” dan NIM: “1302199999” dengan “POST /api/mahasiswa”

POST /api/Movies

Parameters

No parameters

Try it out

Request body

application/json

Example Value Schema

```
{
  "title": "string",
  "director": "string",
  "stars": [
    "string"
  ],
  "description": "string"
}
```

Responses

Code	Description	Links
200	OK	No links

POST /api/Movies

Parameters

No parameters

Request body

application/json

```
{
  "title": "12 Angry Men",
  "director": "Sidney Lumet",
  "stars": [
    "Henry Fonda", "Lee J. Cobb"
  ],
  "description": "A jury holdout attempts to prevent a miscarriage of justice..."
}
```

Execute

Responses

Responses

Curl

```
curl -X 'POST' \
  'https://localhost:7223/api/Movies' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "title": "12 Angry Men",
    "director": "Sidney Lumet",
    "stars": [
      "Henry Fonda", "Lee J. Cobb"
    ],
    "description": "A jury holdout attempts to prevent a miscarriage of justice..."
  }'
```

Request URL

https://localhost:7223/api/Movies

Server response

Code Details

200

Response headers

```
content-length: 0
date: Thu, 12 Jun 2025 22:04:20 GMT
server: Kestrel
```

Responses

Code	Description	Links
200	OK	No links

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

Server response

Code Details

200

Response body

```
{
  "title": "The Dark Knight",
  "director": "Christopher Nolan",
  "stars": [
    "Christian Bale",
    "Heath Ledger"
  ],
  "description": "Batman faces the Joker in a battle for Gotham City."
},
{
  "title": "12 Angry Men",
  "director": "Sidney Lumet",
  "stars": [
    "Henry Fonda",
    "Lee J. Cobb"
  ],
  "description": "A jury holdout attempts to prevent a miscarriage of justice..."
},
{
  "title": "12 Angry Men",
  "director": "Sidney Lumet",
  "stars": [
    "Henry Fonda",
    "Lee J. Cobb"
  ],
  "description": "A jury holdout attempts to prevent a miscarriage of justice..."
}
}
```

Response headers

```
content-type: application/json; charset=utf-8
date: Thu, 12 Jun 2025 22:06:26 GMT
server: Kestrel
```

Responses

Code	Description	Links
------	-------------	-------

Responses

Curl

```
curl -X 'POST' \
  'https://localhost:7223/api/Movies' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "title": "12 Angry Men",
    "director": "Sidney Lumet",
    "stars": [
      "Henry Fonda",
      "Lee J. Cobb"
    ],
    "description": "A jury holdout attempts to prevent a miscarriage of justice..."
  }'
```

Request URL

https://localhost:7223/api/Movies

Server response

Code Details

200

Response headers

```
content-length: 0
date: Thu, 12 Jun 2025 22:02:06 GMT
server: Kestrel
```

Responses

Code	Description	Links
200	OK	No links

https://localhost:7223/swagger/index.html

description: "A jury holdout attempts to prevent a miscarriage of justice..."

```
{
  "title": "12 Angry Men",
  "director": "Sidney Lumet",
  "stars": [
    "Henry Fonda",
    "Lee J. Cobb"
  ],
  "description": "A jury holdout attempts to prevent a miscarriage of justice..."
}
```

Response headers

```
content-type: application/json; charset=utf-8
date: Thu, 12 Jun 2025 22:06:26 GMT
server: Kestrel
```

Responses

Code	Description	Links
200	OK	No links

Media type

text/plain

Controls Accept header

Example Value Schema

```
{
  "title": "string",
  "director": "string",
  "stars": [
    "string"
  ],
  "description": "string"
}
```

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

GET /api/Movies/{id}

Parameters

Try it out

Name	Description
id • required	id
integer(\$int32)	(path)

Responses

Code	Description	Links
200	OK	No links

Media type

text/plain

Controls Accept header

Example Value Schema

```
{
  "title": "string",
  "director": "string",
  "stars": [
    "string"
  ],
  "description": "string"
}
```

GET /api/Movies/{id}

Parameters

Cancel

Name	Description
id • required	3
integer(\$int32)	(path)

Execute

Responses

Code	Description	Links
200	OK	No links

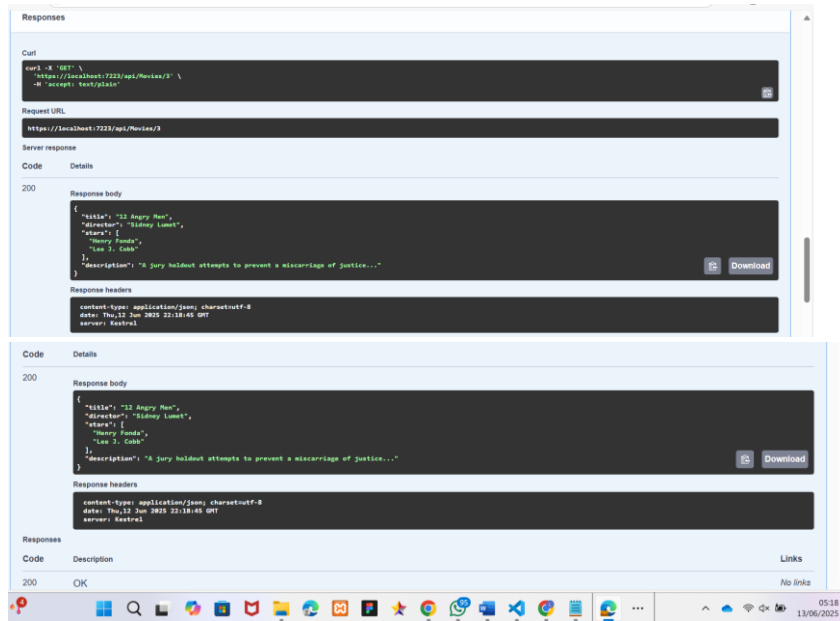
Media type

text/plain

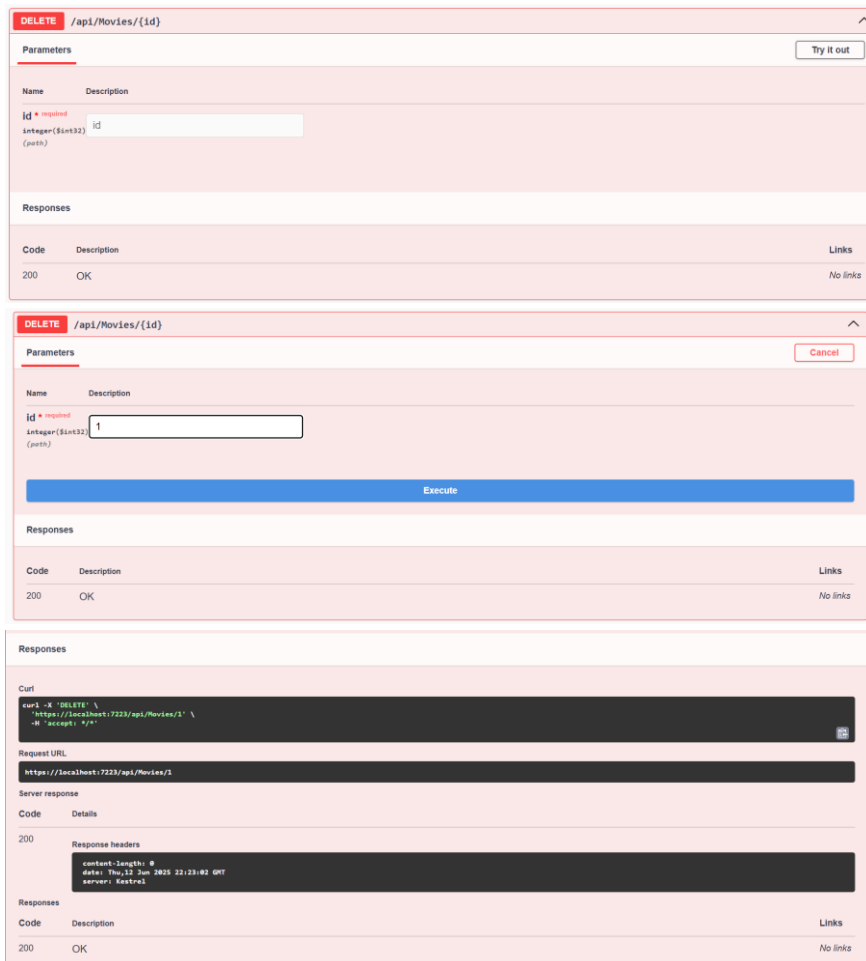
Controls Accept header

Example Value Schema

```
{
  "title": "string",
  "director": "string",
  "stars": [
    "string"
  ],
  "description": "string"
}
```

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 “The Shawsank Redemption” sudah tidak ada di list:

The screenshot shows a REST client interface with two panels. The top panel displays a GET request to `https://localhost:7223/api/Movies` with a `Content-Type: text/plain` header. The bottom panel shows the response, which is a JSON array of movie objects. The response status is 200 OK.

Request:

```
curl -X 'GET' \
  'https://localhost:7223/api/Movies' \
  -H 'accept: text/plain'
```

Request URL: `https://localhost:7223/api/Movies`

Server response:

Code: 200

Details:

Response body:

```
{
  "title": "The Godfather",
  "director": "Francis Ford Coppola",
  "stars": [
    "Marlon Brando",
    "Al Pacino"
  ],
  "description": "The aging patriarch of an organized crime dynasty..."
},
{
  "title": "12 Angry Men",
  "director": "Sidney Lumet",
  "stars": [
    "Henry Fonda",
    "Lee J. Cobb"
  ],
  "description": "A jury holdout attempts to prevent a miscarriage of justice..."
},
{
  "title": "12 Angry Men",
  "director": "Sidney Lumet",
  "stars": [
    "Henry Fonda",
    "Lee J. Cobb"
  ],
  "description": "A jury holdout attempts to prevent a miscarriage of justice..."
}
```

Response headers:

```
content-type: application/json; charset=utf-8
date: Thu, 12 Jun 2025 22:24:11 GMT
server: Fastify
```

Responses:

Code	Description	Links
200	OK	No links

Media type: `text/plain`

Example Value:

```
{
  "title": "string",
  "director": "string",
  "stars": [
    "string"
  ],
  "description": "string"
}
```