

LAPORAN PRAKTIKUM
Jurnal Modul 9
Api Design dan Construction Using Swagger



Disusun Oleh:
Berlian Seva Astryana 2311104067
S1SE-07-02
Dosen :
Yudha Islami Sulistya, S.Kom., M.Cs
PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY
PURWOKERTO
2024

1. Link GitHub

https://github.com/onyooooon/KPL_Berlian-Seva-Astryana_2311104067.git

2. Deskripsi Singkat

Tugas jurnal ini bertujuan untuk melatih praktikan dalam membangun sebuah aplikasi Web API sederhana menggunakan ASP.NET Core di Visual Studio 2022. Web API yang dikembangkan berfungsi untuk mengelola data film berdasarkan struktur class Movie, yang terdiri dari properti judul, sutradara, daftar pemain, dan deskripsi film. Data disimpan menggunakan list statis tanpa menggunakan database. Melalui tugas ini, praktikan diharapkan memahami cara membuat, menjalankan, dan menguji API dengan metode HTTP seperti GET, POST, dan DELETE.

3. Implementasi Program

3.1 Kode Movie(Movie.cs)

Class Movie dibuat untuk merepresentasikan struktur data dari sebuah film. Class ini memiliki empat properti, yaitu Title bertipe string untuk menyimpan judul film, Director bertipe string untuk nama sutradara, Stars berupa list string untuk menyimpan daftar pemain utama, dan Description bertipe string yang berisi deskripsi singkat film. Berikut merupakan kode dari class Movie:

```
namespace modul9_2311104067
{
    8 references
    public class Movie
    {
        3 references
        public string Title { get; set; }
        3 references
        public string Director { get; set; }
        3 references
        public List<string> Stars { get; set; }
        3 references
        public string Description { get; set; }
    }
}
```

3.2 Kode Controller Movie(MovieController.cs)

Controller MoviesController dibuat untuk menangani permintaan HTTP dari klien. Controller ini menggunakan atribut [ApiController] dan [Route("api/[controller]")] untuk mendefinisikan rute akses ke API. Di dalam controller terdapat list statis Movies yang berisi data default tiga film teratas dari IMDb. Controller ini mendukung empat endpoint utama: GET untuk mengambil seluruh data, GET dengan parameter ID untuk mengambil satu data berdasarkan indeks, POST untuk menambahkan data baru, dan DELETE untuk menghapus data berdasarkan indeks.

Berikut adalah kode lengkap controller:

```
using Microsoft.AspNetCore.Mvc;

namespace modul9_2311104067.Controllers
{
    [ApiController]
```

```

[Route("api/[controller]")]
public class MoviesController : ControllerBase
{
    private static List<Movie> Movies = new List<Movie>
    {
        new Movie {
            Title = "The Shawshank Redemption",
            Director = "Frank Darabont",
            Stars = new List<string> { "Tim Robbins", "Morgan Freeman" },
            Description = "A banker convicted of uxoricide forms a friendship over a
quarter century with a hardened convict, while maintaining his innocence and trying
to remain hopeful through simple compassion."
        },
        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 of his clandestine empire to his reluctant son."
        },
        new Movie {
            Title = "The Dark Knight",
            Director = "Christopher Nolan",
            Stars = new List<string> { "Christian Bale", "Heath Ledger" },
            Description = "When the menace known as the Joker wreaks havoc and
chaos on the people of Gotham, Batman, James Gordon and Harvey Dent must work
together to put an end to the madness."
        }
    };

    [HttpGet]
    public ActionResult<List<Movie>> GetAll()
    {
        return Movies;
    }

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

    [HttpPost]
    public ActionResult Add(Movie movie)
    {
        Movies.Add(movie);
        return Ok();
    }

    [HttpDelete("{id}")]
    public ActionResult Delete(int id)
    {
        if (id < 0 || id >= Movies.Count) return NotFound();
        Movies.RemoveAt(id);
    }
}

```

```
        return Ok();  
    }  
}  
}
```

3.3 Penjelasan Implementasi Kode

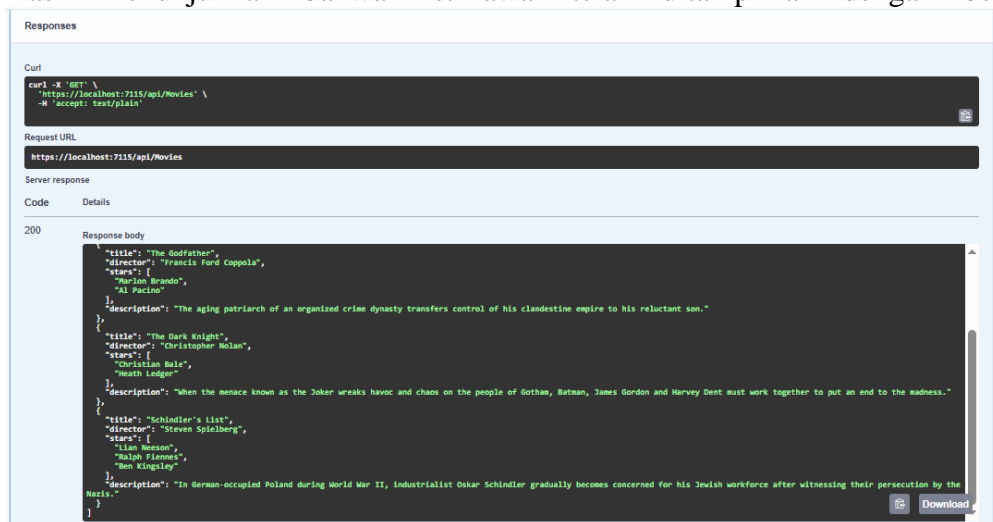
Implementasi kode terdiri dari dua bagian utama, yaitu class model Movie dan controller MoviesController. Berikut penjelasan detailnya:

- Class Movie digunakan sebagai model data yang menyimpan informasi mengenai film. Dengan adanya class ini, struktur data menjadi konsisten dan mudah digunakan di dalam API.
- List Movies disimpan sebagai variabel statis dalam controller untuk menyimpan data secara sementara selama aplikasi berjalan. List ini berisi data awal tiga film IMDb teratas.
- Method GetAll() mengembalikan seluruh isi list Movies dalam bentuk JSON melalui metode GET.
- Method GetById(int id) menerima parameter ID dan mengembalikan data pada indeks yang sesuai. Bila ID tidak valid, akan mengembalikan respons 404.
- Method Add(Movie movie) menambahkan data baru ke dalam list menggunakan metode POST.
- Method Delete(int id) menghapus data dari list berdasarkan indeks ID menggunakan metode DELETE.

4. Dokumentasi Pengujian API

Berikut hasil uji API menggunakan Swagger. Setiap screenshot menjelaskan hasil eksekusi dari masing-masing endpoint:

- Endpoint GET /api/Movies digunakan untuk menampilkan tiga film default. Hasil menunjukkan bahwa list awal telah ditampilkan dengan benar.



- Endpoint POST /api/Movies digunakan untuk menambahkan film "Schindler's List" ke dalam list. Data dikirim dalam format JSON, dan server memberikan respons 200 OK.

POST /api/Movies

Parameters Cancel Reset

No parameters

Request body application/json

```
{
  "title": "Schindler's List",
  "director": "Steven Spielberg",
  "stars": [
    "Liam Neeson",
    "Ralph Fiennes",
    "Ben Kingsley"
  ],
  "description": "In German-occupied Poland during World War II, industrialist Oskar Schindler gradually becomes concerned for his Jewish workforce after witnessing their persecution by the Nazis."
}
```

Execute Clear Activate Windows

- c. Setelah proses POST, endpoint GET /api/Movies digunakan kembali dan menampilkan data "Schindler's List" sebagai elemen keempat di dalam list.

Responses

Curl

```
curl -X 'POST' \
  https://localhost:7115/api/Movies \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "title": "Schindler\'s List",
    "director": "Steven Spielberg",
    "stars": [
      "Liam Neeson",
      "Ralph Fiennes",
      "Ben Kingsley"
    ],
    "description": "In German-occupied Poland during World War II, industrialist Oskar Schindler gradually becomes concerned for his Jewish workforce after witnessing their persecution by the Nazis."
  }'
```

Request URL

https://localhost:7115/api/Movies

Server response

Code Details

200

Response headers

```
content-length: 0
date: Sun, 25 May 2025 16:59:31 GMT
server: Kestrel
```

Responses

Code	Description	Links
200	OK	No links

Activate Windows

- d. Endpoint GET /api/Movies/3 berhasil mengambil data "Schindler's List"

Responses

Curl

```
curl -X 'GET' \
  https://localhost:7115/api/Movies \
  -H 'accept: text/plain'
```

Request URL

https://localhost:7115/api/Movies

Server response

Code Details

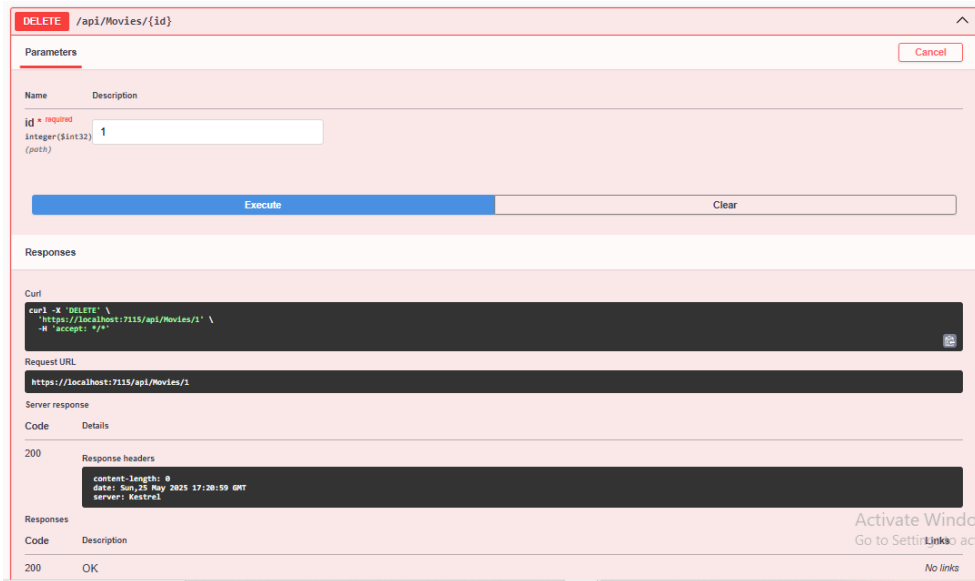
200

Response body

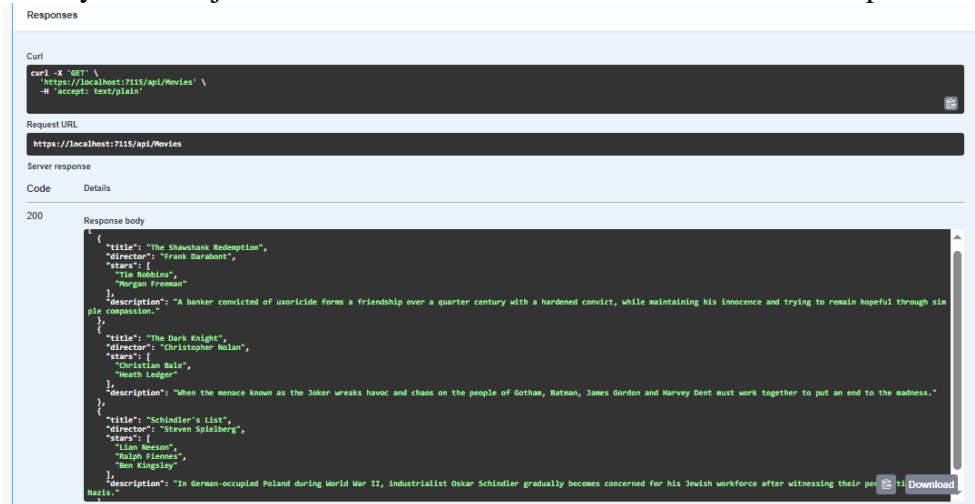
```
{
  "title": "The Godfather",
  "director": "Francis Ford Coppola",
  "stars": [
    "Marlon Brando",
    "Al Pacino"
  ],
  "description": "The aging patriarch of an organized crime dynasty transfers control of his clandestine empire to his reluctant son."
},
{
  "title": "The Dark Knight",
  "director": "Christopher Nolan",
  "stars": [
    "Christian Bale",
    "Heath Ledger"
  ],
  "description": "When the menace known as the Joker wreaks havoc and chaos on the people of Gotham, Batman, James Gordon and Harvey Dent must work together to put an end to the madness."
},
{
  "title": "Schindler's list",
  "director": "Steven Spielberg",
  "stars": [
    "Liam Neeson",
    "Ralph Fiennes",
    "Ben Kingsley"
  ],
  "description": "In German-occupied Poland during World War II, industrialist Oskar Schindler gradually becomes concerned for his Jewish workforce after witnessing their persecution by the Nazis."
}
]
```

Download

- e. Endpoint DELETE /api/Movies/1 digunakan untuk menghapus film "The Godfather" dari list.



- f. Endpoint GET `/api/Movies` terakhir digunakan untuk mengecek data, dan hasilnya menunjukkan bahwa film "The Godfather" telah terhapus dari list.



5. Kesimpulan

Tugas jurnal ini memberikan pengalaman langsung dalam membangun dan menguji Web API sederhana dengan ASP.NET Core. Praktikan berhasil membuat model dan controller, serta memahami cara kerja endpoint GET, POST, dan DELETE. Selama pengujian, API berfungsi dengan baik dan mampu menangani input serta output sesuai spesifikasi. Penggunaan list statis sebagai penyimpanan sementara sangat cocok untuk latihan awal tanpa basis data.