Nama:Faishal Arif Setiawan

Nim:2311104066

Jurnal 9 Code_Reuse_and_Libraries

I.Link Github

https://github.com/faishalstwn

II.Source code

Moviescontroller.cs

```
v using Microsoft.AspNetCore.Mvc;
using Jurnal9.Models;
using System.Collections.Generic;
```

- Microsoft.AspNetCore.Mvc: Digunakan untuk membuat Web API Controller.
- Jurnal9.Models: Namespace di mana class Movie berada.
- System.Collections.Generic: Untuk menggunakan List<T>.

```
[ApiController]
[Route("api/[controller]")]
0 references
public class MoviesController : ControllerBase
{
```

- ApiController]: Menandakan bahwa ini adalah controller khusus Web API.
- [Route("api/[controller]")]: Routing otomatis. [controller] akan diganti dengan nama controller (movies), jadi rute dasarnya adalah api/movies.
- ControllerBase: Kelas dasar untuk controller Web API (tanpa fitur view seperti MVC).

```
private static List<Movie> Movies = new List<Movie>
```

Daftar Movie disimpan dalam memori.

Masing-masing Movie memiliki: Title, Director, Stars (list nama aktor), dan Description.

```
[HttpGet]
0 references
public IActionResult GetAll()
{
    return Ok(Movies);
}
```

- endpoint: GET /api/movies
- Mengembalikan seluruh list movie dalam format JSON.

```
[HttpGet("{id}")]
1 reference
public IActionResult GetById(int id)
{
   if (id < 0 || id >= Movies.Count)
      return NotFound();
   return Ok(Movies[id]);
}
```

- Endpoint: GET /api/movies/{id}
- Mengambil 1 movie berdasarkan index-nya di list.
- Jika id tidak valid (di luar batas list), akan mengembalikan 404 NotFound.

```
[HttpPost]
0 references
public IActionResult Add(Movie movie)
{
    Movies.Add(movie);
    return CreatedAtAction(nameof(GetById), new { id = Movies.Count - 1 }, movie);
}
```

- Endpoint: POST /api/movies
- Menerima data movie dari body request (dalam format JSON), menambahkannya ke list
- CreatedAtAction mengembalikan 201 Created dengan lokasi dan data movie yang baru ditambahkan.

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

- Endpoint: DELETE /api/movies/{id}
- Menghapus movie dari list berdasarkan index.
- Jika tidak ditemukan, mengembalikan 404 NotFound. Jika berhasil, mengembalikan 204 NoContent.

Program.cs

```
var builder = WebApplication.CreateBuilder(args);

// Add services
builder.Services.AddControllers();
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();

var app = builder.Build();

// Enable Swagger

if (app.Environment.IsDevelopment())
{
    app.UseSwagger();
    app.UseSwaggerUI();
}

app.UseHttpsRedirection();
app.UseAuthorization();
app.MapControllers();
app.Run();
```

Membuat W

ebApplication Builder:

```
var builder = WebApplication.CreateBuilder(args);
```

Membuat builder untuk mengatur konfigurasi aplikasi, dependency injection, dan environment.

Menambahkan Layanan (Services) ke Container DI:

```
builder.Services.AddControllers();
```

Menambahkan support untuk Web API Controller ([ApiController]).

```
builder.Services.AddEndpointsApiExplorer();
```

• Mendukung eksplorasi endpoint untuk dokumentasi (digunakan oleh Swagger).

```
builder.Services.AddSwaggerGen();
```

Menambahkan layanan untuk Swagger, yaitu dokumentasi interaktif untuk API

Build Aplikasi:

```
var app = builder.Build();
```

• Membuat WebApplication dari konfigurasi yang sudah disiapkan.

Mengaktifkan Swagger di Mode Development:

```
if (app.Environment.IsDevelopment())
{
    app.UseSwagger();
    app.UseSwaggerUI();
}
```

- Hanya menjalankan Swagger jika environment-nya adalah Development.
- UseSwagger() untuk menghasilkan file JSON dokumentasi.
- UseSwaggerUI() untuk menampilkan halaman Swagger interaktif di browser.

III.Output

```
GET /api/Movies
```

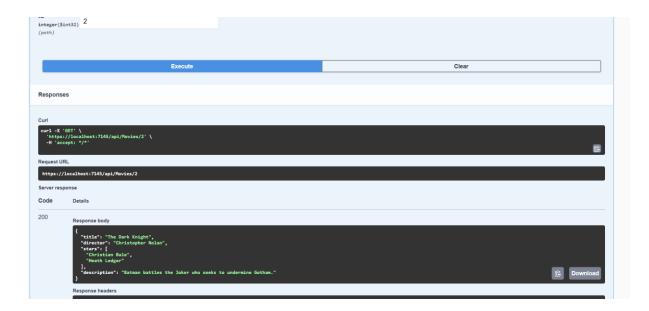
```
"Request URL

Notice That Taking page of the second of the
```

POST /api/Movies



GET /api/Movies/{id}



DELETE /api/Movies/{id}

