

**TUGAS JURNAL
MODUL 9**



Disusun Oleh :

Izzaty Zahara Br Barus – 23111040452

Kelas :

SE-07-02

Dosen :

Yudha Islami Sulistya

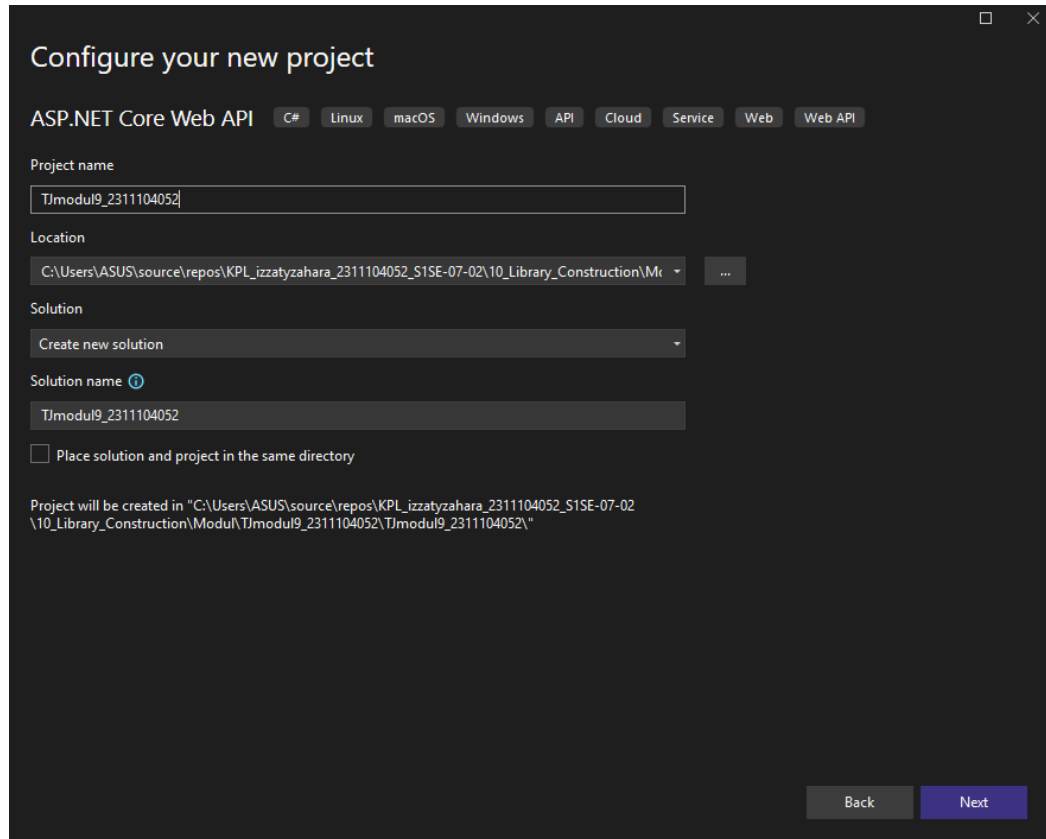
**PROGRAM STUDI SOFTWARE ENGINEERING
DIREKTORAT KAMPUS PURWOKERTO
TELKOM UNIVERSITY
PURWOKERTO
2025**

I. Link Github

- https://github.com/Izzaaaaaaaaa/KPL_izzatyzahara_2311104052_S1SE-07-02.git

II. Penjelasan

1. Membuat New Project



- ### 2. Membuat Package Model yang dimana terdapat Class Movie.cs
- Membuat File JSON “bank_transfer_config.json”.

```
using System.Collections.Generic;

namespace TJmodul9_2311104052.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 Singkat :

Bagian model dalam kode ini menjelaskan class Movie yang berfungsi untuk merepresentasikan struktur data sebuah film. Class ini terletak di dalam namespace TJmodul9_2311104052.Models dan memiliki empat properti utama, yaitu Title, Director, Stars, dan Description. Properti Title dan Director bertipe string, yang menyimpan judul dan nama sutradara film. Sementara itu, Stars bertipe List<string>, yang memungkinkan kita menyimpan lebih dari satu nama pemeran dalam satu film. Terakhir, Description juga bertipe string dan digunakan untuk memberikan ringkasan cerita atau penjelasan mengenai film tersebut. Model ini sederhana, tetapi sangat penting karena menjadi struktur acuan dalam pertukaran data antar bagian aplikasi, terutama saat menerima atau mengirim data melalui API.

3. Menambah Class di Package Controller dengan nama MovieController.cs

```
using Microsoft.AspNetCore.Mvc;
using TJmodul9_2311104052.Models;
using System.Collections.Generic;

namespace TJmodul9_2311104052.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 = "Two imprisoned men bond over a
                number of years, finding solace and eventual redemption
                through acts of common decency."
            },
            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 emerges, Batman must accept one of the greatest
psychological and physical tests."
    }
};

[HttpGet]
public ActionResult<IEnumerable<Movie>>
GetMovies()
{
    return Ok(Movies);
}

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

[HttpPost]
public ActionResult<Movie> AddMovie([FromBody]
Movie movie)
{
    Movies.Add(movie);
    return CreatedAtAction(nameof(GetMovie), new {
id = Movies.Count - 1 }, movie);
}

```

```

    }

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

```

Penjelasan Singkat:

Bagian model dalam kode ini menjelaskan class Movie yang berfungsi untuk merepresentasikan struktur data sebuah film. Class ini terletak di dalam namespace TJmodul9_2311104052.Models dan memiliki empat properti utama, yaitu Title, Director, Stars, dan Description. Properti Title dan Director bertipe string, yang menyimpan judul dan nama sutradara film. Sementara itu, Stars bertipe List<string>, yang memungkinkan kita menyimpan lebih dari satu nama pemeran dalam satu film. Terakhir, Description juga bertipe string dan digunakan untuk memberikan ringkasan cerita atau penjelasan mengenai film tersebut. Model ini sederhana, tetapi sangat penting karena menjadi struktur acuan dalam pertukaran data antar bagian aplikasi, terutama saat menerima atau mengirim data melalui API.

4. Class Program

```

var builder = WebApplication.CreateBuilder(args);

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

builder.Services.AddControllers();
// Learn more about configuring Swagger/OpenAPI at

```

```
https://aka.ms/aspnetcore/swashbuckle
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();

var app = builder.Build();

// Configure the HTTP request pipeline.
if (app.Environment.IsDevelopment())
{
    app.UseSwagger();
    app.UseSwaggerUI();
}

app.UseAuthorization();

app.MapControllers();

app.Run();
```

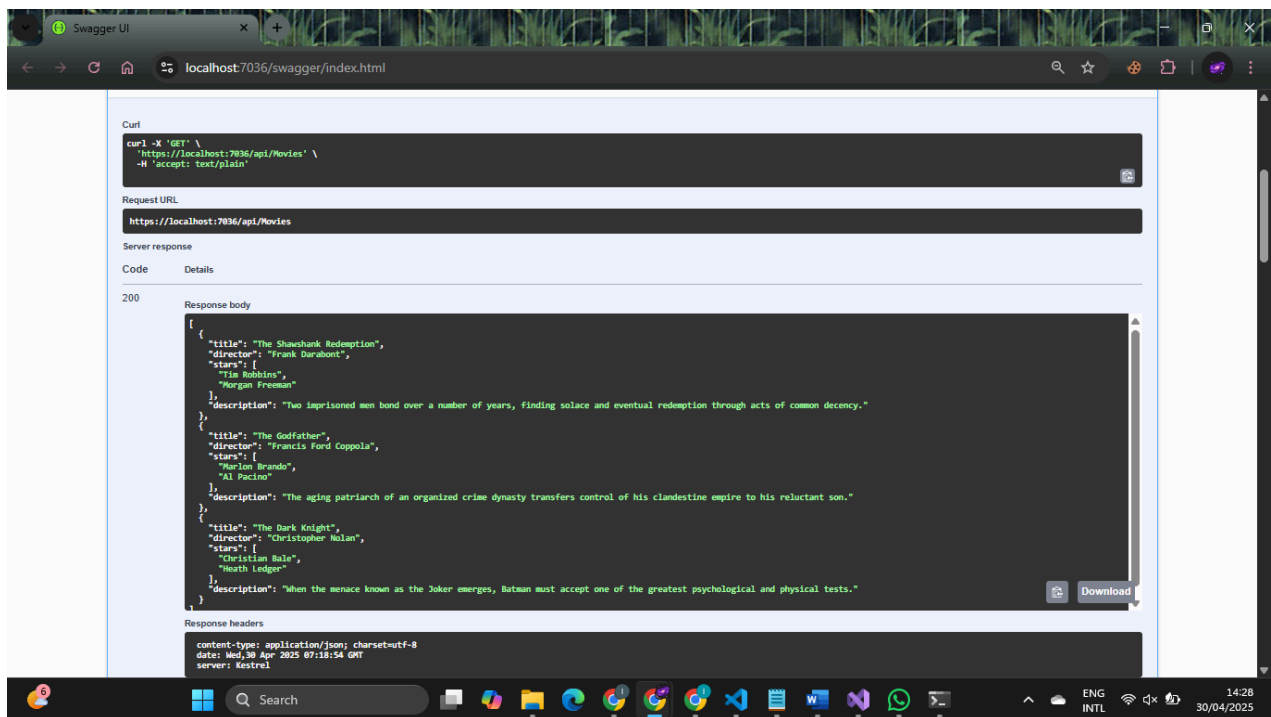
Penjelasan Singkat:

File Program.cs adalah titik masuk untuk aplikasi ASP.NET Core dan berfungsi untuk mengatur semua konfigurasi layanan serta alur HTTP. Di awal, `WebApplication.CreateBuilder(args)` digunakan untuk membuat builder aplikasi. Selanjutnya, `builder.Services.AddControllers()` menambahkan layanan controller, yang memungkinkan framework mengenali dan menjalankan endpoint dari controller. `AddSwaggerGen()` diaktifkan untuk mengaktifkan fitur Swagger, yang sangat berguna sebagai dokumentasi dan antarmuka pengujian API berbasis web. Setelah aplikasi dibangun (`builder.Build()`), bagian `if (app.Environment.IsDevelopment())` akan mengaktifkan Swagger dan SwaggerUI hanya jika aplikasi berjalan dalam mode pengembangan, sehingga API dapat diuji secara visual di browser. Kemudian, `app.UseAuthorization()` digunakan untuk memproses middleware otorisasi (meskipun belum digunakan dalam API ini). Terakhir, `app.MapControllers()` berfungsi untuk memetakan semua route controller agar dapat merespon permintaan HTTP, dan `app.Run()` mengeksekusi aplikasi sehingga dapat diakses melalui browser atau alat seperti Postman.

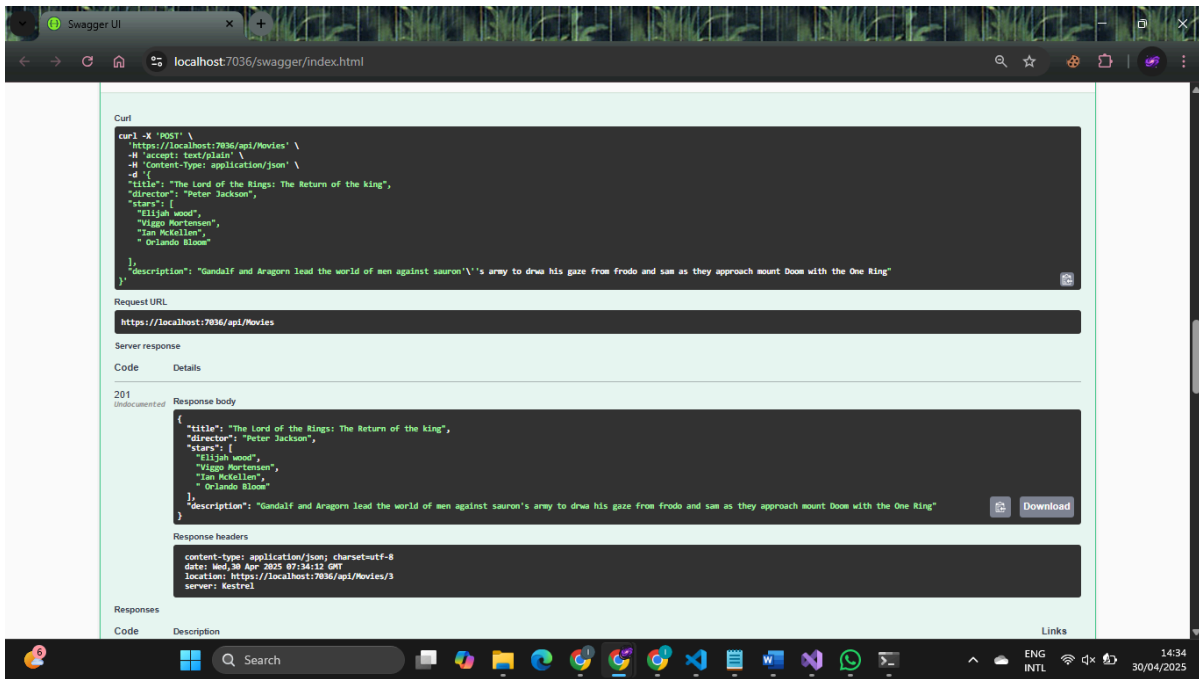
III. Hasil Running

1. Hasil Running

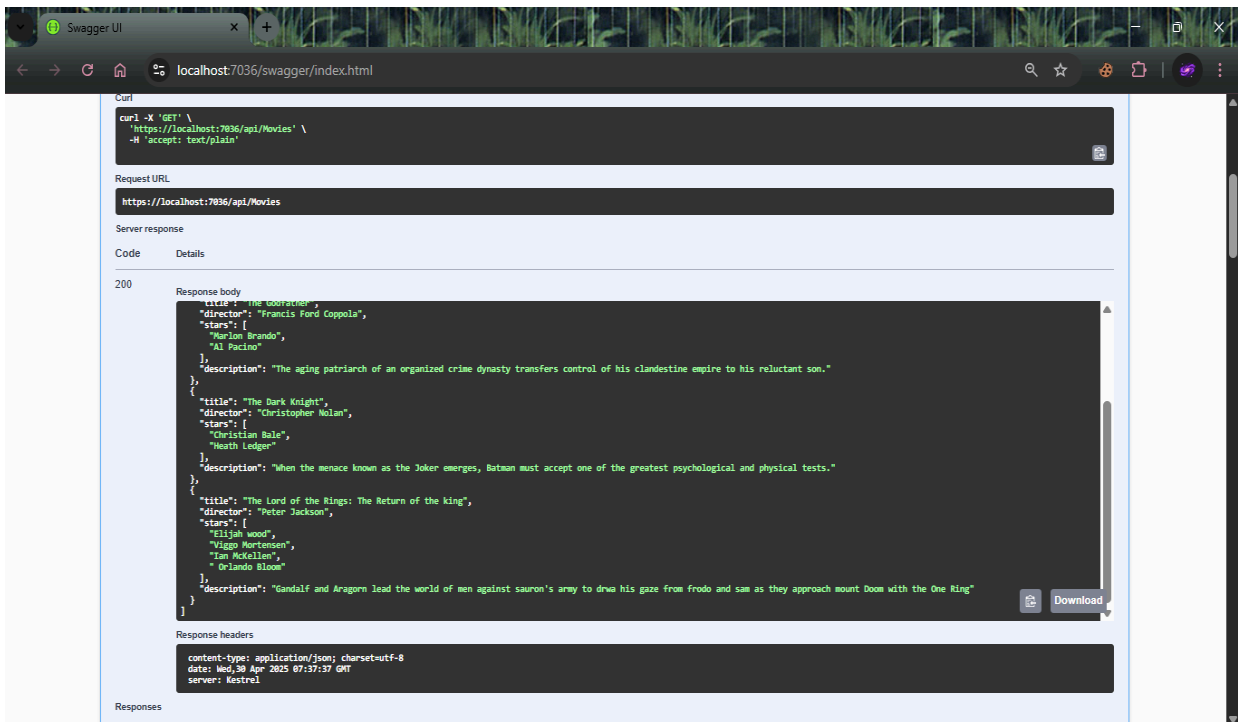
- Get



- Post



• Get



• Get Cari Berdasarkan ID

Name
Description

id * required
integer(\$int32)
(path)

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7036/api/Movies/3' \
  -H 'accept: text/plain'
```

Request URL

https://localhost:7036/api/Movies/3

Server response

Code

Details

200

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"
}
```

Download

Response headers

```
content-type: application/json; charset=utf-8
date: Wed,30 Apr 2025 07:39:37 GMT
server: Kestrel
```

• Delete

DELETE /api/Movies/{id}

Cancel

Parameters

Name

Description

id * required
integer(\$int32)
(path)

Execute

Clear

Responses

Curl

```
curl -X 'DELETE' \
  'https://localhost:7036/api/Movies/1' \
  -H 'accept: /*/*'
```

Request URL

https://localhost:7036/api/Movies/1

Server response

Code

Details

204

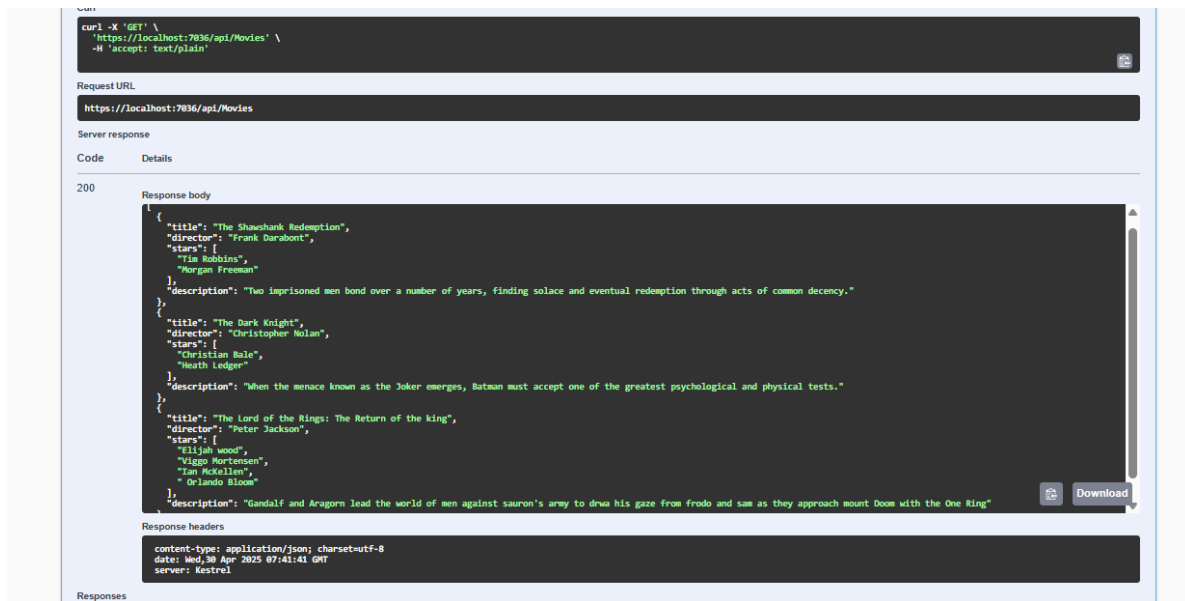
Undocumented

Response headers

```
date: Wed,30 Apr 2025 07:41:01 GMT
server: Kestrel
```

Responses

• Get



```
C:\WINDOWS\system32\cmd. x + v
*****
C:\Users\ASUS\source\repos\KPL_izzatyzahara_2311104052_S1SE-07-02\09_API_Design_dan_Construction_Using_Swagger\TJ\TJmodul9_2311104052>cd TJmodul9_2311104052
C:\Users\ASUS\source\repos\KPL_izzatyzahara_2311104052_S1SE-07-02\09_API_Design_dan_Construction_Using_Swagger\TJ\TJmodul9_2311104052>git add .
warning: in the working copy of '09_API_Design_dan_Construction_Using_Swagger\TJ\TJmodul9_2311104052\TJmodul9_2311104052\obj/Debug/net8.0/EndpointInfo/TJmodul9_2311104052.json', LF will be replaced by CRLF the next time Git touches it
C:\Users\ASUS\source\repos\KPL_izzatyzahara_2311104052_S1SE-07-02\09_API_Design_dan_Construction_Using_Swagger\TJ\TJmodul9_2311104052\TJmodul9_2311104052>git commit -m "menambahkan API Movies"
[master 720ab16] menambahkan API Movies
65 files changed, 1619 insertions(+)
create mode 100644 09_API_Design_dan_Construction_Using_Swagger\TJ\TJmodul9_2311104052\TJmodul9_2311104052\Controllers\MoviesController.cs
create mode 100644 09_API_Design_dan_Construction_Using_Swagger\TJ\TJmodul9_2311104052\TJmodul9_2311104052\Controllers\WeatherForecastController.cs
create mode 100644 09_API_Design_dan_Construction_Using_Swagger\TJ\TJmodul9_2311104052\TJmodul9_2311104052\Models\Movie.cs
create mode 100644 09_API_Design_dan_Construction_Using_Swagger\TJ\TJmodul9_2311104052\TJmodul9_2311104052\Program.cs
create mode 100644 09_API_Design_dan_Construction_Using_Swagger\TJ\TJmodul9_2311104052\TJmodul9_2311104052\Properties\LaunchSettings.json
create mode 100644 09_API_Design_dan_Construction_Using_Swagger\TJ\TJmodul9_2311104052\TJmodul9_2311104052\TJmodul9_2311104052.csproj
create mode 100644 09_API_Design_dan_Construction_Using_Swagger\TJ\TJmodul9_2311104052\TJmodul9_2311104052\TJmodul9_2311104052.csproj.user
create mode 100644 09_API_Design_dan_Construction_Using_Swagger\TJ\TJmodul9_2311104052\TJmodul9_2311104052\TJmodul9_2311104052.http
create mode 100644 09_API_Design_dan_Construction_Using_Swagger\TJ\TJmodul9_2311104052\TJmodul9_2311104052\WeatherForec
```

IV. Kesimpulan

Pratikum ini merupakan implementasi dasar dari RESTful API menggunakan ASP.NET Core yang terdiri dari tiga komponen utama: model, controller, dan konfigurasi aplikasi. Model Movie digunakan untuk merepresentasikan struktur data film secara sederhana namun cukup lengkap, mencakup judul, sutradara, pemeran, dan deskripsi. Controller MoviesController menangani berbagai operasi HTTP seperti mengambil semua data, mengambil data berdasarkan indeks, menambahkan, dan menghapus film, dengan data yang disimpan dalam list statis tanpa database. Sementara itu, file Program.cs bertugas mengatur layanan dan pipeline aplikasi, termasuk aktivasi Swagger untuk dokumentasi dan pengujian API. Secara keseluruhan, proyek ini memberikan fondasi yang kuat untuk memahami cara membangun dan mengelola Web API sederhana di

ASP.NET Core, yang dapat dikembangkan lebih lanjut dengan fitur seperti penyimpanan data berbasis database dan autentikasi pengguna.