

**TUGAS PENDAHULUAN MODUL 09
KONSTRUKSI PERANGKAT LUNAK**

API DESIGN DAN CONSTRUCTION USING SWAGGER



**DISUSUN OLEH:
KHOLIFAH DINA
2211104004**

SE 06 01

**S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY
2025**

1. MEMBUAT PROJECT CONSOLE/TANPA GUI

Buka IDE misalnya dengan Visual Studio

- Pilih New Project dan kemudian pilih ASP.NET Core Web API atau API (pastikan opsi 'Enable OpenAPI support' tercentang).
- Masukkan project baru dengan nama tpmodul9_NIM.

Configure your new project

ASP.NET Core Web API C# Linux macOS Windows API Cloud Service Web Web API

Project name
tpmodul9_2211104004

Location
D:\Kuliah\KPL_KHOLIFAHADINA_2211104004_SE0601\09_API_Design_dan_Construction_Using_Swag ...

Solution
Create new solution

Solution name ⓘ
tpmodul9_2211104004

☒ Place solution and project in the same directory

Project will be created in "D:\Kuliah\KPL_KHOLIFAHADINA_2211104004_SE0601\09_API_Design_dan_Construction_Using_Swagger\TP\tpmodul9_2211104004\"

2. SOURCE CODE

- Program.cs

```
1 var builder = WebApplication.CreateBuilder(args);
2
3 // Add services to the container.
4
5 builder.Services.AddControllers();
6 // Learn more about configuring Swagger/OpenAPI at https://aka.ms/aspnetcore/swashbuckle
7 builder.Services.AddEndpointsApiExplorer();
8 builder.Services.AddSwaggerGen();
9
10 var app = builder.Build();
11
12 // Configure the HTTP request pipeline.
13 if (app.Environment.IsDevelopment())
14 {
15     app.UseSwagger();
16     app.UseSwaggerUI();
17 }
18
19 app.UseHttpsRedirection();
20
21 app.UseAuthorization();
22
23 app.MapControllers();
24
25 app.Run();
26
```

b. MahasiswaController.cs

```
1 using Microsoft.AspNetCore.Mvc;
2 using System.Collections.Generic;
3 using tpmodul9_NIM.Models;
4
5 namespace tpmodul9_NIM.Controllers
6 {
7     [Route("api/[controller]")]
8     [ApiController]
9     public class MahasiswaController : ControllerBase
10     {
11         // Static list to simulate a database
12         private static List<Mahasiswa> daftarMahasiswa = new List<Mahasiswa>
13         {
14             new Mahasiswa { Nama = "Kholifahdina", Nim = "2211104004" },
15             new Mahasiswa { Nama = "Byeon Woo Seok", Nim = "2211109560" },
16             new Mahasiswa { Nama = "Lee Jong Suk", Nim = "2211110987" }
17         };
18
19         // GET: api/mahasiswa
20         [HttpGet]
21         public ActionResult<IEnumerable<Mahasiswa>> Get()
22         {
23             return daftarMahasiswa;
24         }
25
26         // GET: api/mahasiswa/{index}
27         [HttpGet("{index}")]
28         public ActionResult<Mahasiswa> GetByIndex(int index)
29         {
30             if (index < 0 || index >= daftarMahasiswa.Count)
31                 return NotFound("Index tidak ditemukan");
32             return daftarMahasiswa[index];
33         }
34
35         // POST: api/mahasiswa
36         [HttpPost]
37         public ActionResult Post([FromBody] Mahasiswa mhs)
38         {
39             daftarMahasiswa.Add(mhs);
40             return Ok("Mahasiswa ditambahkan");
41         }
42
43         // DELETE: api/mahasiswa/{index}
44         [HttpDelete("{index}")]
45         public ActionResult Delete(int index)
46         {
47             if (index < 0 || index >= daftarMahasiswa.Count)
48                 return NotFound("Index tidak ditemukan");
49             daftarMahasiswa.RemoveAt(index);
50             return Ok("Mahasiswa dihapus");
51         }
52     }
53 }
```

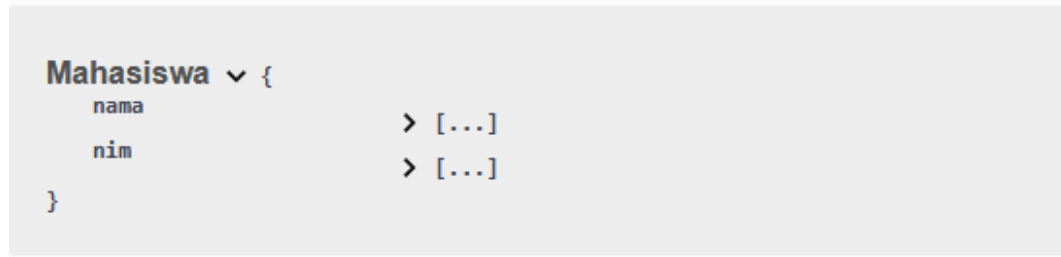
c. Mahasiswa.cs

```
1 namespace tpmodul9_NIM.Models
2 {
3     public class Mahasiswa
4     {
5         public string Nama { get; set; }
6         public string Nim { get; set; }
7     }
8 }
```

3. IMPLEMENTASI WEB API

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

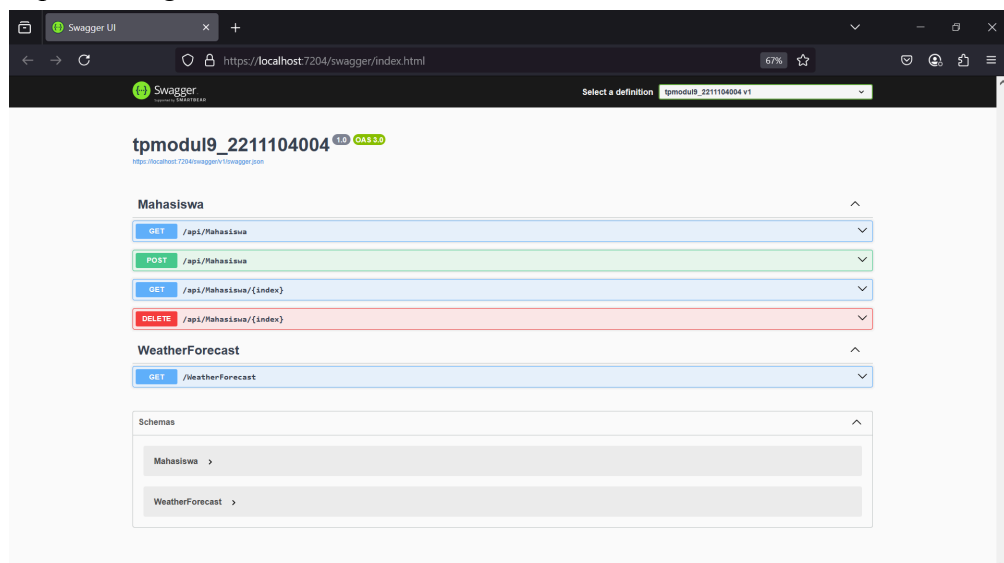
A. API yang dibuat menggunakan data dari kelas Mahasiswa.



B. Secara default, program yang dibuat memiliki array/list mahasiswa dari anggota kelompok anda (tuliskan nama anda di urutan pertama/paling atas), contohnya:

- Nama: “LeBron James”, Nim: “1302000001”
- Nama: “Stephen Curry”, Nim: “1302000002”
- dst.

C. Gunakan teknologi API sehingga program tersebut dapat menerima HTTP request sebagai berikut:

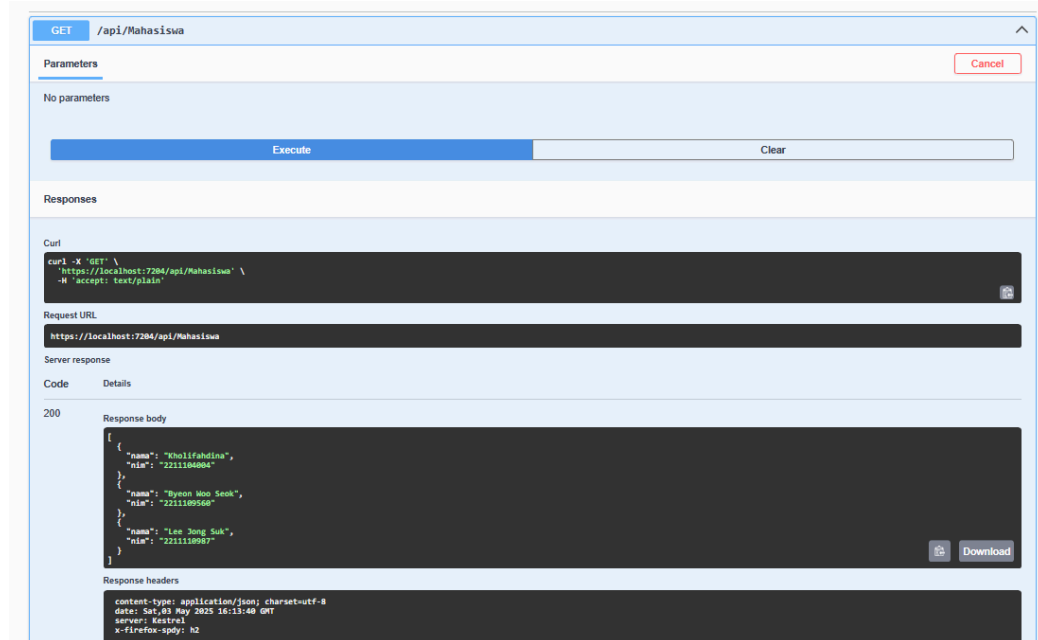


- GET /api/mahasiswa: mengembalikan output berupa list/array dari semua objek mahasiswa yang tersimpan
- GET /api/mahasiswa/{index}: mengembalikan output berupa objek mahasiswa untuk index ke-‘index’
- POST /api/mahasiswa: menambahkan objek mahasiswa baru dengan menyertakan nama dan nim
- DELETE /api/mahasiswa/{index}: menghapus objek mahasiswa dengan index ke-‘index’

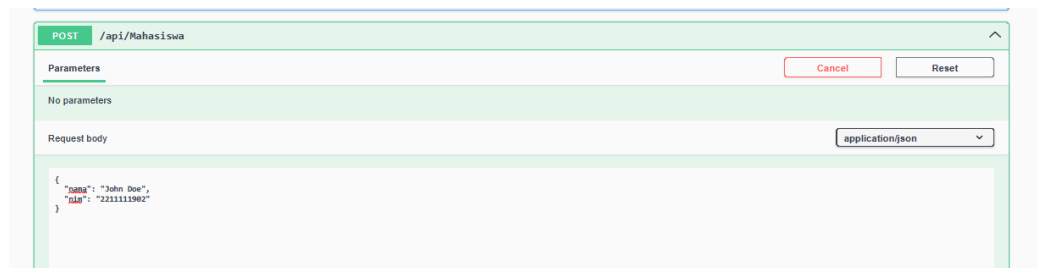
4. MENDEMONSTRASIKAN WEB API

Beberapa skenario yang harus dicoba untuk memastikan jika program telah berjalan dengan baik. Buatlah dokumen yang berisi semua screenshot dari hasil uji coba skenario yang disebutkan pada list berikut ini:

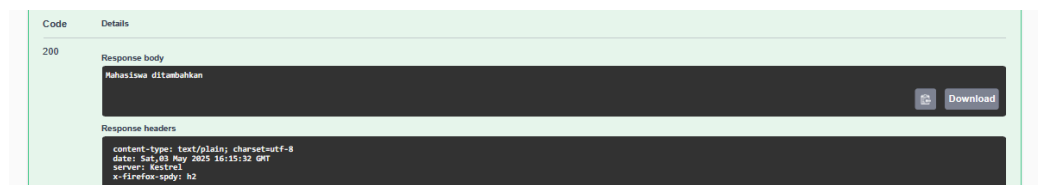
- A. Mencoba “GET /api/mahasiswa” saat baru dijalankan (mengeluarkan list nama mahasiswa dan nim anggota kelompok):



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



- C. Cek list/array dari semua mahasiswa lagi dengan “GET /api/mahasiswa”, pastikan mahasiswa yang baru ditambahkan sebelumnya ada di list mahasiswa:



- D. Mencoba meminta mahasiswa dengan index 0, “GET /api/mahasiswa/0” yang seharusnya mengeluarkan nama dan nim anda:

The screenshot shows a REST client interface with a GET request to `/api/mahasiswa/{index}`. The parameter `index` is set to `0`. The response is a 200 status code with a JSON body containing the student's name and nim.

Parameters

Name	Description
<code>index</code> * required	
integer(\$int32)	
(path)	

Execute **Clear**

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7204/api/mahasiswa/0' \
  -H 'accept: text/plain'
```

Request URL

```
https://localhost:7204/api/mahasiswa/0
```

Server response

Code	Details
200	<p>Response body</p> <pre>{ "nama": "Kholifahdina", "nim": "2211194004" }</pre> <p>Response headers</p> <pre>content-type: application/json; charset=utf-8 date: Sat, 03 May 2025 16:17:45 GMT server: Kestrel x-firefox-spy: h2</pre>

- E. Menghapus objek mahasiswa dengan index ke-0 dengan “DELETE /api/mahasiswa/0”

The screenshot shows a REST client interface with a DELETE request to `/api/mahasiswa/{index}`. The parameter `index` is set to `0`. The response is a 200 status code with a plain text body indicating the student was deleted.

Parameters

Name	Description
<code>index</code> * required	
integer(\$int32)	
(path)	

Execute **Clear**

Responses

Curl

```
curl -X 'DELETE' \
  'https://localhost:7204/api/mahasiswa/0' \
  -H 'accept: */'
```

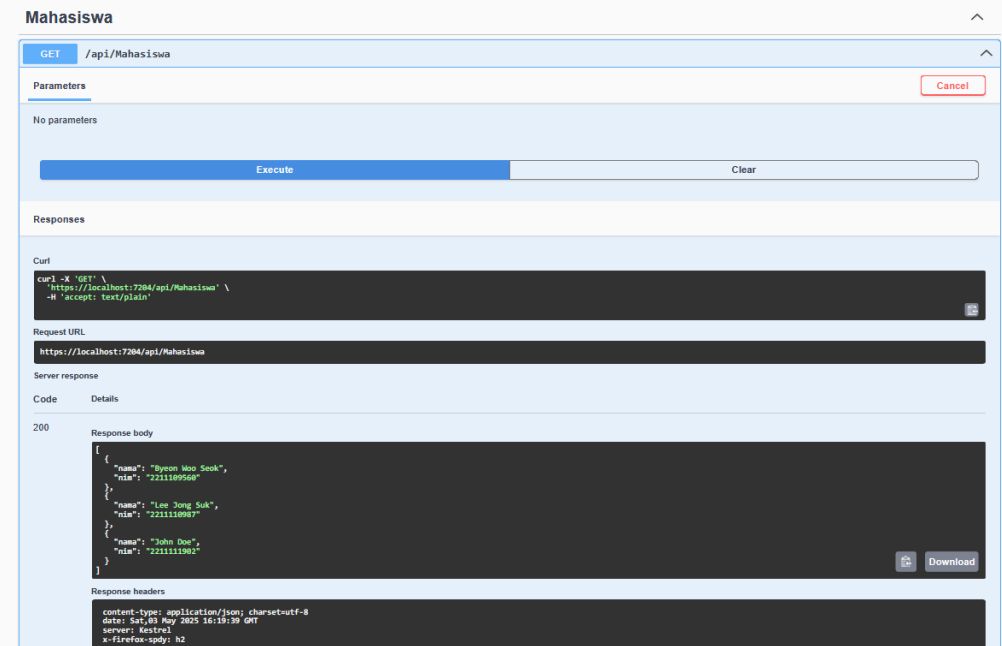
Request URL

```
https://localhost:7204/api/mahasiswa/0
```

Server response

Code	Details
200	<p>Response body</p> <pre>Mahasiswa dihapus</pre> <p>Response headers</p> <pre>content-type: text/plain; charset=utf-8 date: Sat, 03 May 2025 16:19:03 GMT server: Kestrel x-firefox-spy: h2</pre>

- F. Cek list/array dari semua mahasiswa sekali lagi dengan “GET /api/mahasiswa”, pastikan nama anda sudah tidak muncul di list tersebut:



The screenshot shows a REST client interface with the following sections:

- Method and URL:** GET /api/mahasiswa
- Parameters:** No parameters
- Execute Button:** A blue button labeled "Execute" and a grey button labeled "Clear".
- Responses:**
 - Curl:**

```
curl -X GET \
  https://localhost:7204/api/mahasiswa \
  -H 'accept: text/plain'
```
 - Request URL:** https://localhost:7204/api/mahasiswa
 - Server response:**
 - Code:** 200
 - Response body:**

```
{
  "nama": "Byeon Woo Seok",
  "nim": "2211109560"
},
{
  "nama": "Lee Jong Suk",
  "nim": "2211110087"
},
{
  "nama": "John Doe",
  "nim": "2211111902"
}

```
 - Response headers:**

```
content-type: application/json; charset=utf-8
date: Sat, 03 May 2025 16:19:39 GMT
server: Kestrel
x-frame-options: deny
```