

LAPORAN PENDAHULUAN MODUL 9



Disusun Oleh :

Zaenarif Putra 'Ainurdin – 2311104049

Kelas :

SE-07-02

Dosen :

Yudha Islami Sulistya

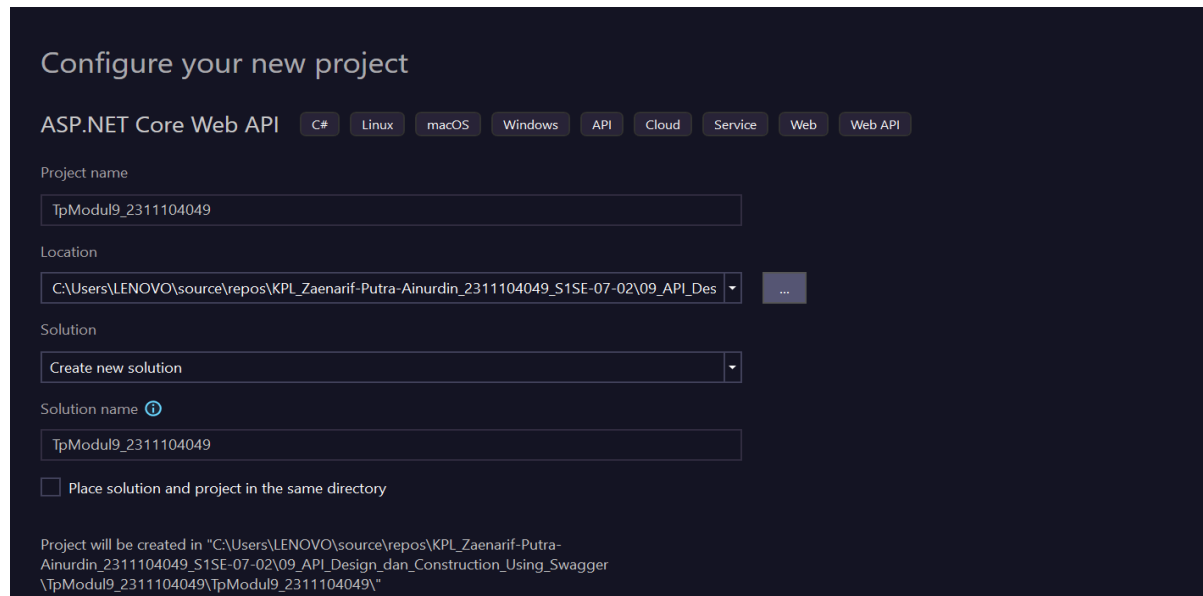
**PROGRAM STUDI SOFTWARE ENGINEERING
DIREKTORAT KAMPUS PURWOKERTO
TELKOM UNIVERSITY
PURWOKERTO
2025**

I. Link Github

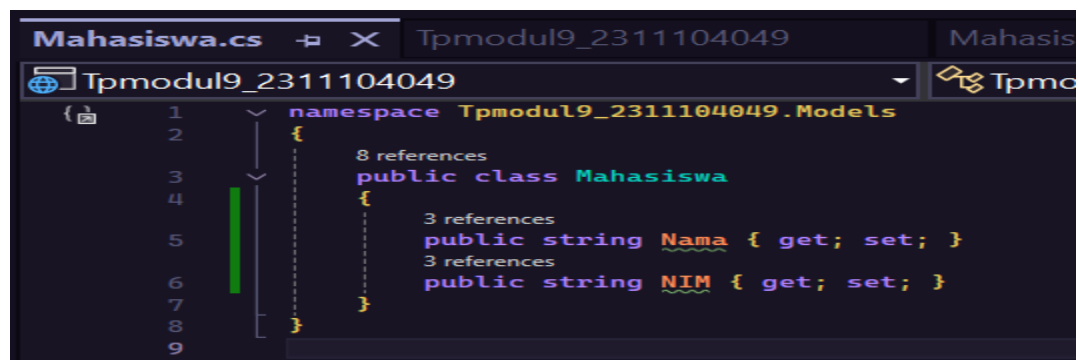
https://github.com/zaenarifputra/KPL_Zaenarif-Putra-Ainurdin_2311104049_S1SE-07-02/tree/20da781eb828996cac29e5b1bd074221c1d6a2a1/08_Runtime_Configuration_dan_Internation_alization/TP

II. Alur Pengerjaan

1. Membuat project baru yang dimana menggunakan ASP.NET Core Web Api yang diberi nama “TpModul9_2311104049”



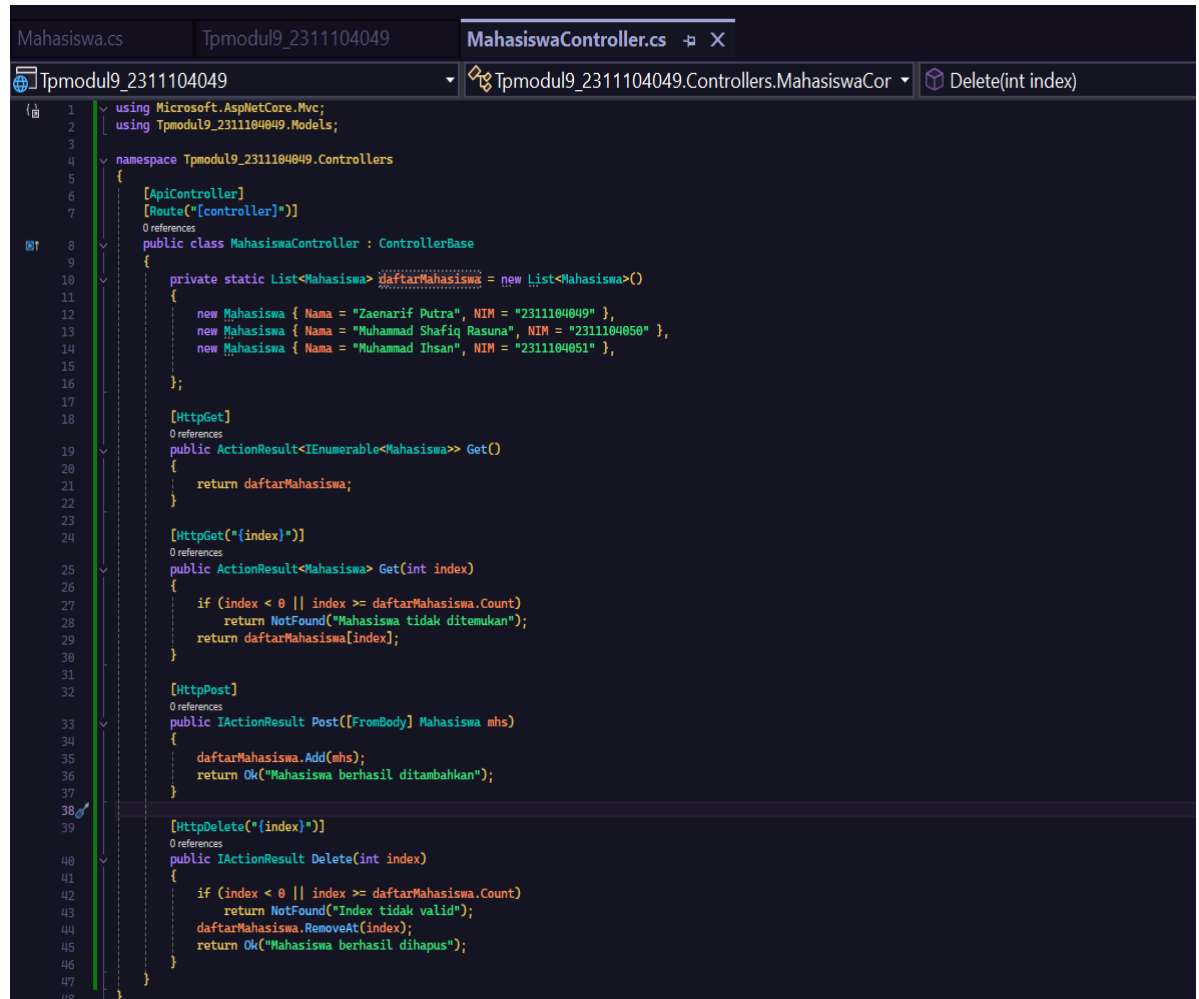
2. setelah membuat project kemudian melakukan pengerjaan yang pertama yaitu membuat package yang diberi nama models yang kemudian ditambah sebuah class yang diberi nama Mahasiswa.cs yang digunakan sebagai tempat penyimpanan representasi data yang kemudian diimplementasikan pada folder controller berikut foto dan juga penjelasan syntax pada class Mahasiswa.cs :



pada syntax berikut digunakan untuk mempresentasikan data mahasiswa dalam bentuk objek. Di dalamnya terdapat dua properti, yaitu Nama dan NIM, yang masing-masing bertipe data string dan dapat diakses atau diubah dari luar kelas menggunakan getter dan setter otomatis ({ get; set; }). Secara sederhana, kode ini membuat cetakan data mahasiswa

yang nanti bisa digunakan untuk menyimpan, menerima, dan mengirimkan data dalam bentuk JSON melalui API.

- Setelah melakukan penyelesaian pada models mahasiswa selanjutnya kita implementasikan pada folder controller yang diberi nama MahasiswaController.cs berikut penjelasan syntaxnya secara sederhana dan ss dari syntaxnya :



```

1  using Microsoft.AspNetCore.Mvc;
2  using Tpmodule9_2311104049.Models;
3
4  namespace Tpmodule9_2311104049.Controllers
5  {
6      [ApiController]
7      [Route("[controller]")]
8      public class MahasiswaController : ControllerBase
9      {
10         private static List<Mahasiswa> daftarMahasiswa = new List<Mahasiswa>()
11         {
12             new Mahasiswa { Nama = "Zaenarif Putra", NIM = "2311104049" },
13             new Mahasiswa { Nama = "Muhammad Shafiq Rasuna", NIM = "2311104060" },
14             new Mahasiswa { Nama = "Muhammad Ihsan", NIM = "2311104051" },
15         };
16
17         [HttpGet]
18         public ActionResult<IEnumerable<Mahasiswa>> Get()
19         {
20             return daftarMahasiswa;
21         }
22
23         [HttpGet("{index}")]
24         public ActionResult<Mahasiswa> Get(int index)
25         {
26             if (index < 0 || index >= daftarMahasiswa.Count)
27                 return NotFound("Mahasiswa tidak ditemukan");
28             return daftarMahasiswa[index];
29         }
30
31         [HttpPost]
32         public IActionResult Post([FromBody] Mahasiswa mhs)
33         {
34             daftarMahasiswa.Add(mhs);
35             return Ok("Mahasiswa berhasil ditambahkan");
36         }
37
38         [HttpDelete("{index}")]
39         public IActionResult Delete(int index)
40         {
41             if (index < 0 || index >= daftarMahasiswa.Count)
42                 return NotFound("Index tidak valid");
43             daftarMahasiswa.RemoveAt(index);
44             return Ok("Mahasiswa berhasil dihapus");
45         }
46     }
47 }

```

Pada class MahasiswaController berikut menerima dan merespon HTTP. Yang dimana didalamnya Ada sebuah daftar statis bernama daftarMahasiswa yang menyimpan informasi tentang tiga mahasiswa sebagai data awal. API ini memiliki empat endpoint utama: GET untuk mengambil semua data mahasiswa, GET(index) untuk mengambil data mahasiswa berdasarkan indeks tertentu, POST untuk menambahkan mahasiswa baru ke dalam daftar, dan DELETE(index) untuk menghapus mahasiswa dari daftar berdasarkan indeks. Setiap tindakan dilengkapi dengan validasi agar hanya indeks yang valid yang dapat diakses atau dihapus, serta akan memberikan pesan jika data tidak ditemukan.

- Setelah selesai melakukan semua konfigurasi baik dari package models dan controller kemudian run yang dimana akan menghasilkan output seperti berikut yang akan di tampilkan pada hasil running :

Tpmodul9_2311104049 1.0 OAS 3.0

<https://localhost:7086/swagger/v1/swagger.json>

Mahasiswa

GET	/Mahasiswa
POST	/Mahasiswa
GET	/Mahasiswa/{index}
DELETE	/Mahasiswa/{index}

Pada class program dimana membuat sebuah fungsi untuk mencoba membaca file covid_config.json menggunakan class CovidConfig yang sudah dibuat sebelumnya. Kalau file-nya gagal dibaca, program langsung kasih pesan error dan berhenti. Kalau berhasil, program menampilkan ucapan selamat datang, lalu meminta pengguna mengisi suhu tubuh dan berapa hari lalu terakhir demam. Setelah itu, program akan mengecek apakah suhu tubuh berada di rentang normal berdasarkan satuan suhu yang diatur di file config, dan juga mengecek apakah jumlah hari demam masih di bawah batas aman. Kalau kedua syarat ini terpenuhi, pengguna akan mendapat pesan diterima; kalau tidak, akan keluar pesan ditolak. Setelah screening selesai, program juga otomatis mengubah satuan suhu (misalnya dari celcius ke fahrenheit) dan memberitahu pengguna perubahan tersebut. Terakhir, program mengucapkan terima kasih atas partisipasi dan menutup dengan rapi.

5. setelah semua class yang dibutuhkan sudah berhasil di implementasikan semua selanjutnya melakukan push menuju cloud github yang dimana untuk melakukan push bisa melakukannya dengan perintah `git add .`, `git commit -m "menyelesaikan program dengan runtime configuration"` dan kemudian terakhir melakukan `git push -u origin master`. jika berhasil maka akan di tampilkan seperti gambar berikut :

III. Hasil Running

1. Hasil Running GET / Mahasiswa

GET

/Mahasiswa

Parameters

No parameters

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7086/Mahasiswa' \
  -H 'accept: text/plain'
```

Request URL

https://localhost:7086/Mahasiswa

Server response

Code

Details

200

Response body

```
{
  "nama": "Zaenarif Putra",
  "nim": "2311104049"
},
{
  "nama": "Muhammad Shafiq Rasuna",
  "nim": "2311104050"
},
{
  "nama": "Muhammad Ihsan",
  "nim": "2311104051"
}
```

Response headers

```
content-type: application/json; charset=utf-8
date: Fri,25 Apr 2025 08:34:02 GMT
server: Kestrel
```

2. POST / Mahasiswa

POST

/Mahasiswa

Parameters

No parameters

Request body

application/json

```
{
  "nama": "Rengganis Tantri",
  "nim": "2311104068"
}
```

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
  'https://localhost:7086/Mahasiswa' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "nama": "Rengganis Tantri",
    "nim": "2311104068"
  }'
```

Request URL

https://localhost:7086/Mahasiswa

Responses

Curl

```
curl -X 'POST' \
  'https://localhost:7086/Mahasiswa' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "nama": "Rengganis Tantri",
    "nim": "2311104068"
  }'
```

Request URL

https://localhost:7086/Mahasiswa

Server response

Code	Details
200	<p>Response body</p> <p>Mahasiswa berhasil ditambahkan</p> <p>Response headers</p> <pre>content-type: text/plain; charset=utf-8 date: Fri, 25 Apr 2025 08:36:57 GMT server: Kestrel</pre>

Responses

Code	Description	Links
200	OK	No links

3. GET / Mahasiswa / {index}

GET /Mahasiswa/{index}

Parameters

Cancel

Name	Description
index * required integer (\$int32) (path)	0

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7086/Mahasiswa/0' \
  -H 'accept: text/plain'
```

Request URL

https://localhost:7086/Mahasiswa/0

Server response

Code	Details
200	<p>Response body</p> <pre>{ "nama": "Zaenarif Putra", "nim": "2311104049" }</pre> <p>Response headers</p> <pre>content-type: application/json; charset=utf-8 date: Fri, 25 Apr 2025 08:38:55 GMT server: Kestrel</pre>

Responses

4. DELETE / Mahasiswa / {index}

DELETE /Mahasiswa/{index} ^

Parameters Cancel

Name	Description
index * required integer (\$int32) (path)	<input type="text" value="3"/>

Execute

Clear

Responses

Curl

```
curl -X 'DELETE' \
'https://localhost:7086/Mahasiswa/3' \
-H 'accept: */*'
```

Request URL

```
https://localhost:7086/Mahasiswa/3
```

Server response

Code	Details
200	<div>Response body</div> <div>Mahasiswa berhasil dihapus</div> <div>Download</div> <div>Response headers</div> <div>content-type: text/plain; charset=utf-8 date: Fri, 25 Apr 2025 08:40:06 GMT server: Kestrel</div>

Responses