

Ketentuan Tugas Pendahuluan

- Pengerjaan TP **DITULIS TANGAN** pada kertas bergaris (buku tulis, kertas polio), lalu di scan dan dimasukkan ke dalam 1 file PDF.
- Jika terdapat instruksi untuk screenshot, tidak perlu di print terlebih dahulu namun langsung disatukan ke file PDF dengan jawaban lainnya.
- TP dikumpulkan **MAKSIMAL SENIN, 17 April 2023 Pukul 08.00 WIB** di LMS.
- Asisten tidak bertanggung jawab terhadap TP yang tidak dikumpulkan pada tempat yang ditentukan sebelumnya.
- TP bersifat **WAJIB** Jika mengerjakan TP, nilai jurnal akan ditambah 10 poin (nilai jurnal maksimal 100 poin).
- **TIDAK ADA TOLERANSI KETERLAMBATAN TP.**
- Jawaban TP harus **BERKAITAN DAN SESUAI DENGAN INSTRUKSI SOAL**, jika jawaban tidak relevan dengan soal, maka **JAWABAN TERSEBUT DIANGGAP KOSONG**.
- Praktikan **WAJIB** menyertakan sumber referensi jawaban dari setiap soal TP.
- Kerjakan TP dengan jelas agar dapat dimengerti.
- **PLAGIARISME = E**

Jika terdapat masalah mengenai soal TP, dapat menghubungi:

Nama : {{ Abdullah Razqy Aqil }}

WA : {{ 081296272719 }}

SELAMAT MENERJAKAN!

TP MODUL 9

1. MEMBUAT PROJECT WEB API

Berhubung cara membuat project web api berbeda-beda untuk setiap bahasa pemrograman, langkah-langkah berikut hanya berlaku apabila dilakukan dengan menggunakan .NET dan Visual Studio. Untuk IDE dan bahasa pemrograman lain, yang terpenting adalah nama project yang dibuat yaitu "tpmodul8_NIM".

- A. Buka visual studio yang sudah terinstall dengan ASP.NET dan .NET 5.0 SDK atau setelahnya
- B. Pilih New Project dan kemudian pilih ASP.NET Core Web API atau API (pastikan opsi 'Enable OpenAPI support' tercentang).
- C. Pastikan untuk memilih .NET versi 5.0 atau yang lebih baru.
- D. Masukkan nama projek "tpmodul9_NIM".
- E. Langkah-langkah yang disertai gambar dapat dilihat pada link berikut ini (cukup dilihat pada bagian "Create a Web API project"):
<https://docs.microsoft.com/en-us/aspnet/core/tutorials/min-web-api?view=aspnetcore-6.0&tabs=visual-studio>
- F. Setelah project tersebut selesai dibuat, coba run programnya, dan tunggu sampai program selesai di-compile.

2. MELAKUKAN GIT COMMIT PADA PROJECT YANG DIBUAT

Task atau langkah-langkah yang perlu dikerjakan adalah sebagai berikut:

- A. Buatlah github public repository kosong (pastikan bagian "Initialize this repository with" tidak ada yang dicentang pada saat membuat repository baru) melalui <https://github.com/>
- B. Melakukan inisialisasi git repository di folder project yang dibuat.
- C. Pastikan untuk menambahkan file ".gitignore" baik manual atau dengan menggunakan visual studio/IDE. Untuk project dengan C# dapat melihat referensi file ".gitignore" pada link berikut ini:
<https://github.com/github/gitignore/blob/main/VisualStudio.gitignore>
- D. Membuat commit untuk versi pertama dari project yang dibuat dengan pesan commit bebas.
- E. Melakukan git push ke github repo.

SELAMAT MENGERJAKAN!

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.

Mahasiswa
- Nama: string
- Nim: string
+ Mahasiswa

- B. API yang dibuat mempunyai lokasi sebagai berikut **‘/api/mahasiswa’**, URL domain boleh dari port mana saja, misalnya <https://localhost:5001/api/mahasiswa> (port bebas)
- C. 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.
- D. Gunakan teknologi API sehingga program tersebut dapat menerima HTTP request sebagai berikut:

Mahasiswa	
GET	/api/Mahasiswa
POST	/api/Mahasiswa
GET	/api/Mahasiswa/{id}
DELETE	/api/Mahasiswa/{id}

- 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’
- E. Implementasi yang dibuat tidak menggunakan database, cukup disimpan sebagai suatu variable, dan gunakan “static” di variable tersebut yang menyimpan list/array dari objek-objek mahasiswa

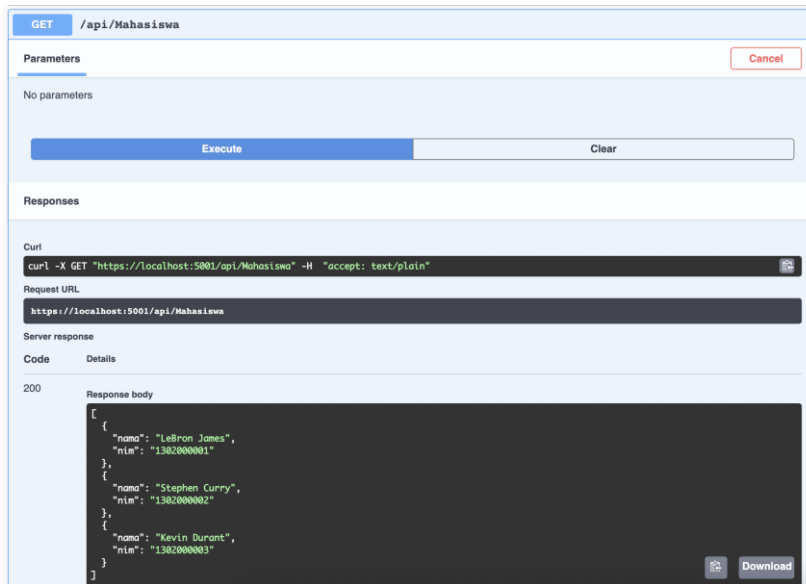
SELAMAT MENGERJAKAN!

- F. Dalam pembuatan program/aplikasi ini, anda dapat mengasumsikan bahwa input dari user selalu benar dan sesuai dengan tipe data yang diharapkan.

4. MENDEMONSTRASI WEB API

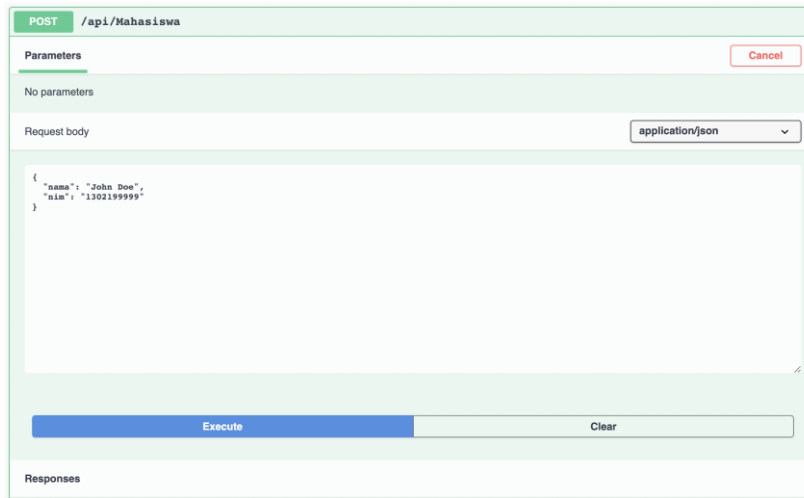
Beberapa skenario yang harus dicoba untuk memastikan jika program telah berjalan dengan baik. Buatlah dokumen yang berisi semua screenshot dari hasil uji coba scenario yang disebutkan pada list berikut ini:

- 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”

SELAMAT MENGERJAKAN!



POST /api/Mahasiswa

Parameters Cancel

No parameters

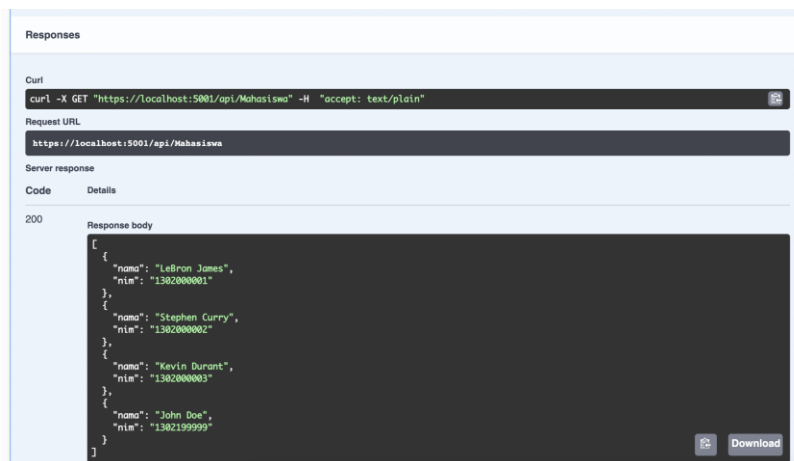
Request body application/json

```
{
  "nama": "John Doe",
  "nim": "1302199999"
}
```

Execute Clear

Responses

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



Responses

Curl

```
curl -X GET "https://localhost:5001/api/Mahasiswa" -H "accept: text/plain"
```

Request URL

```
https://localhost:5001/api/Mahasiswa
```

Server response

Code Details

200

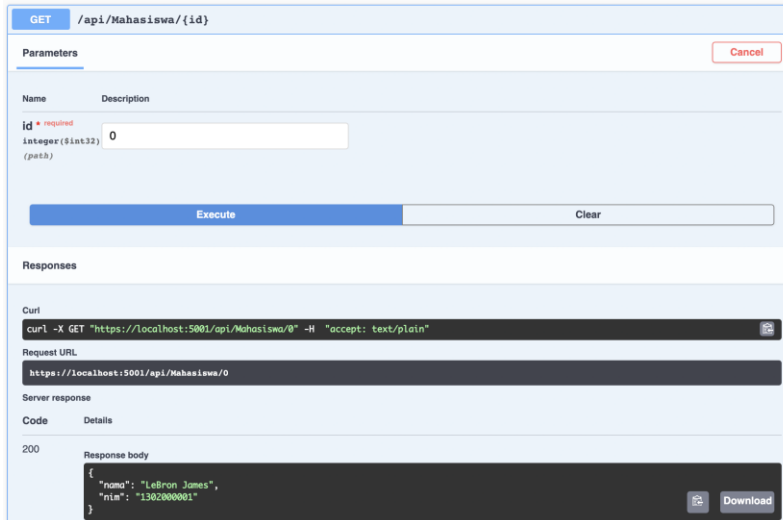
Response body

```
{
  "nama": "LeBron James",
  "nim": "1302000001"
},
{
  "nama": "Stephen Curry",
  "nim": "1302000002"
},
{
  "nama": "Kevin Durant",
  "nim": "1302000003"
},
{
  "nama": "John Doe",
  "nim": "1302199999"
}
}
```

Download

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

SELAMAT MENGERJAKAN!



GET /api/Mahasiswa/{id}

Parameters

Name Description

id • required
integer(5int32)
(path) 0

Execute Clear

Responses

Curl

```
curl -X GET "https://localhost:5001/api/Mahasiswa/0" -H "accept: text/plain"
```

Request URL

```
https://localhost:5001/api/Mahasiswa/0
```

Server response

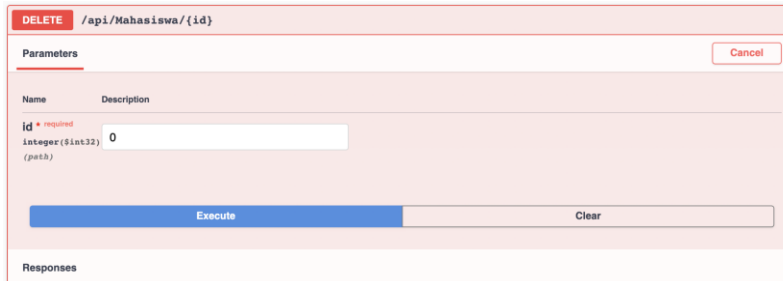
Code Details

200

Response body

```
{
  "nama": "Lebron James",
  "nim": "1302000001"
}
```

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



DELETE /api/Mahasiswa/{id}

Parameters

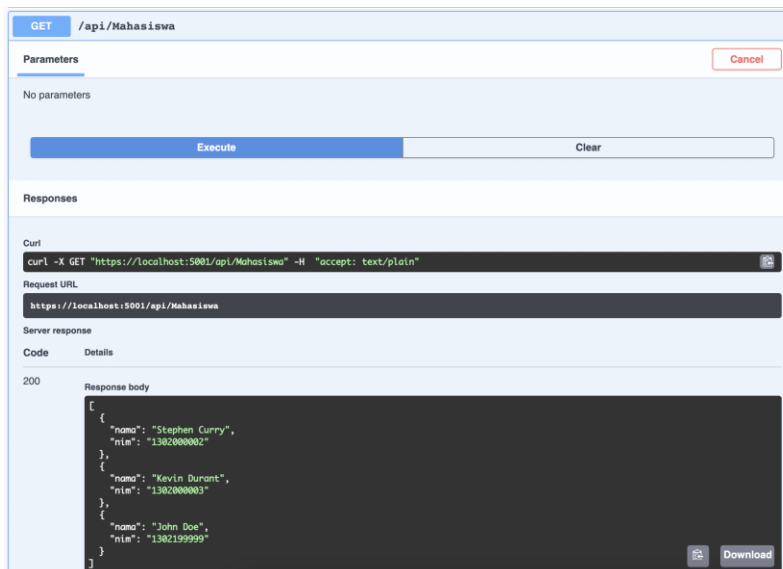
Name Description

id • required
integer(5int32)
(path) 0

Execute Clear

Responses

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



GET /api/Mahasiswa

Parameters

No parameters

Execute Clear

Responses

Curl

```
curl -X GET "https://localhost:5001/api/Mahasiswa" -H "accept: text/plain"
```

Request URL

```
https://localhost:5001/api/Mahasiswa
```

Server response

Code Details

200

Response body

```
[
  {
    "nama": "Stephen Curry",
    "nim": "1302000002"
  },
  {
    "nama": "Kevin Durant",
    "nim": "1302000003"
  },
  {
    "nama": "John Doe",
    "nim": "1302199999"
  }
]
```

SELAMAT MENGERJAKAN!

5. MELAKUKAN COMMIT

Pada branch master/main:

- A. Lakukan commit dengan pesan “menambahkan API mahasiswa”.
- B. Lakukan push ke github ke branch yang dibuat di bagian sebelumnya.

6. PENGUMPULAN FILE/TUGAS PENDAHULUAN

Kumpulkan semua file berikut dalam bentuk file zip/rar/7zip:

- A. Source code dari project yang dibuat
- B. File docx/pdf yang berisi:
 - i. Link github repository
 - ii. Screenshot hasil run (demonstrasi di bagian nomor 4)
 - iii. Penjelasan singkat dari kode implementasi yang dibuat (beserta screenshot dari potongan source code yang dijelaskan).

SELAMAT MENGERJAKAN!