

PERATURAN PRAKTIKUM ONLINE

1. Praktikum diampu oleh dosen kelas dan dibantu oleh asisten laboratorium dan asisten praktikum.
2. Praktikan wajib memastikan tidak ada kendala jaringan maupun perangkat (camera, mic, laptop) yang akan digunakan selama praktikum.
3. Praktikan wajib berada pada ruangan yang kondusif.
4. Praktikan wajib memakai pakaian berkerah selama kegiatan praktikum.
5. Praktikan wajib menyalakan **kamera dan mic** selama kegiatan praktikum.
6. Praktikan wajib hadir 10 menit sebelum jadwal praktikum yang ditentukan.
7. Praktikan wajib hadir minimal 75% dari seluruh pertemuan praktikum (sesuai ketentuan institusi).
8. Praktikum dilaksanakan di Ms Teams Kelas Praktikum sesuai jadwal yang ditentukan.
9. Durasi kegiatan praktikum S-1 = 2 jam (100 menit).
10. Jumlah pertemuan praktikum online sebanyak 2 kali pada modul 8 dan 9.
11. Dosen berhak melarang praktikan masuk ataupun mengeluarkan praktikan yang tidak mematuhi aturan praktikum.
12. Praktikan yang datang terlambat :
 - ≤ 30 menit : diperbolehkan mengikuti praktikum tanpa tambahan waktu pengerjaan praktikum.
 - > 30 menit : tidak diperbolehkan mengikuti praktikum.
13. Saat praktikum berlangsung praktikan:
 - Wajib menggunakan pakaian berkerah.
 - Wajib menyalakan kamera dan mic selama praktikum.
 - Wajib mematikan/ mengkondisikan semua alat komunikasi.
 - Dilarang membuka aplikasi yang tidak berhubungan dengan praktikum yang berlangsung.
 - Dilarang meninggalkan tempat selama tidak diizinkan oleh asisten.
 - Dilarang bekerjasama atau kecurangan lainnya selama praktikum baik secara langsung maupun tidak langsung (WA, Discord, dan media online lainnya).
 - Dilarang memberikan jawaban ke praktikan lain.
 - Dilarang menyebarkan soal praktikum.
 - Wajib menjaga kondusifitas ruangan selama praktikum.
14. Setiap praktikan dapat mengikuti praktikum susulan maksimal dua modul untuk satu mata kuliah praktikum.
 - Praktikan yang dapat mengikuti praktikum susulan hanyalah praktikan yang memenuhi syarat sesuai ketentuan institusi, yaitu: sakit (dibuktikan dengan surat keterangan medis), tugas dari institusi (dibuktikan dengan surat dinas atau dispensasi dari institusi), atau mendapat musibah atau keduakaan (menunjukkan surat keterangan dari orangtua/wali mahasiswa.)
 - Persyaratan untuk praktikum susulan diserahkan sesegera mungkin kepada asisten laboratorium untuk keperluan administrasi.
 - Praktikan yang diijinkan menjadi peserta praktikum susulan ditetapkan oleh Asman Lab dan Bengkel Informatika dan tidak dapat diganggu gugat.

KONSEKUENSI PELANGGARAN ATURAN PRAKTIKUM ONLINE

1. Keterlambatan menghadiri kelas praktikum (max 30 menit)
 - Tidak diperkenankan mengikuti kegiatan praktikum
2. Ketidakhadiran pada kelas praktikum
 - Absensi dibawah 75% = nilai '0' pada assessment akhir/tubes
3. Meminta, mendapatkan, dan menyebarkan soal atau kunci jawaban praktikum
 - Penyebar soal dan kunci jawaban : Pengajuan sanksi kepada Komisi Disiplin Fakultas
 - Penerima soal dan kunci jawaban : Nilai '0' pada (seluruh assessment) praktikum
4. Lupa menghapus file praktikum
 - Pengurangan nilai assessment

JURNAL 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 "modul8_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 "modul9_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.

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 Movie.

Movie
+ Title : string
+ Director : string
+ Stars : List<string>
+ Description: string
+ Movie()

- B. API yang dibuat mempunyai lokasi sebagai berikut **‘/api/Movies**, URL domain boleh dari port mana saja (port bebas). Dengan menggunakan swagger API tersebut dapat menerima RESTful API dengan metoda sebagai berikut (halaman swagger dapat diakses pada <https://localhost:<PORT>/swagger/index.html>):

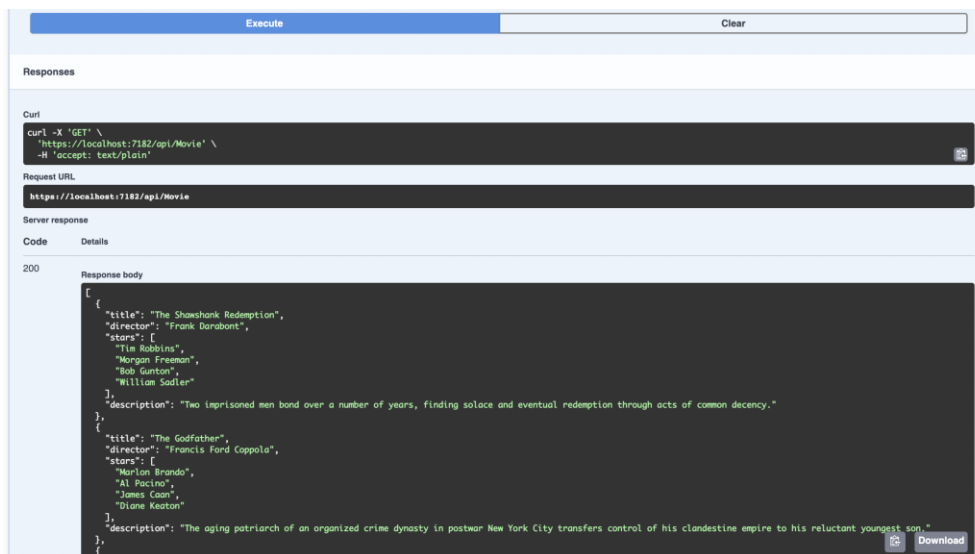
Movies	
GET	/api/Movies
POST	/api/Movies
GET	/api/Movies/{id}
DELETE	/api/Movies/{id}

- GET /api/Movies: mengembalikan output berupa list/array dari semua objek Movies
 - GET /api/Movies/{id}: mengembalikan output berupa objek Movie untuk index “id”
 - POST /api/Movies: menambahkan objek Movie baru
 - DELETE /api/Movies/{id}: menghapus objek Movie pada index “id”
- C. Secara default, program yang dibuat memiliki list film yang berasal dari TOP 3 film IMDB dari link: https://www.imdb.com/search/title/?groups=top_100&sort=user_rating_desc
- D. Implementasi yang dibuat tidak menggunakan database, cukup disimpan sebagai suatu variable, dan gunakan “static” di variable tersebut yang menyimpan list/array dari objek-objek Movie.
- E. Dalam pembuatan program/aplikasi ini, anda dapat mengasumsikan bahwa input dari user selalu benar dan sesuai dengan tipe data yang diharapkan.

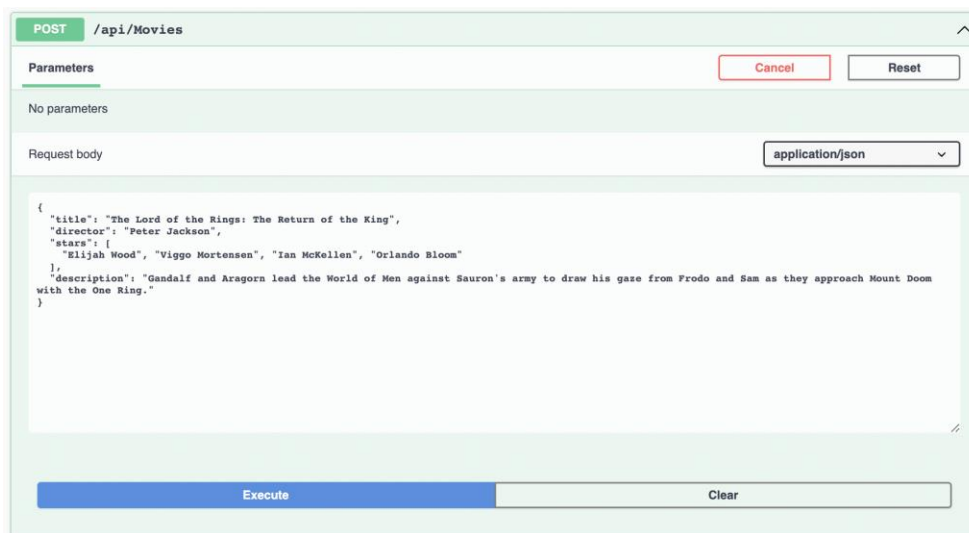
4. MENDEMONSTRASI WEB API

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:

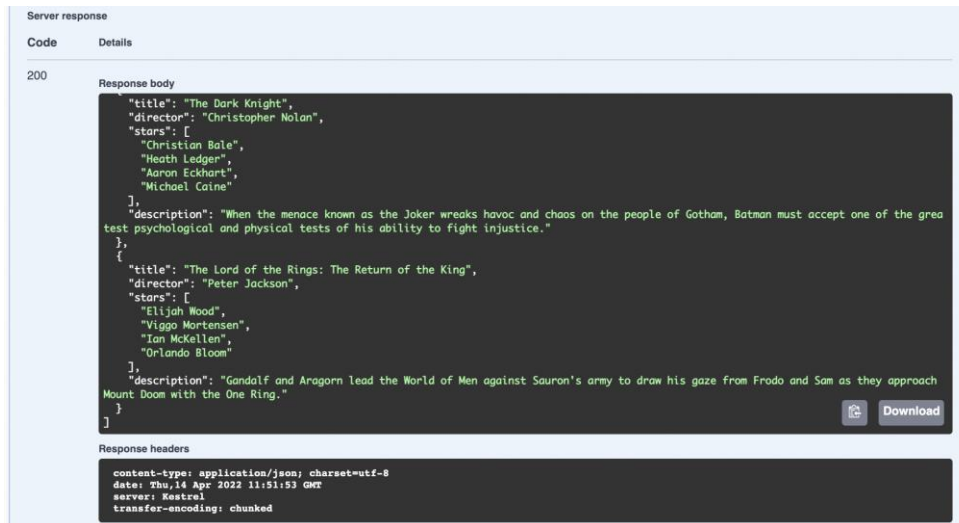
- A. Mencoba “GET /api/Movies” saat baru dijalankan yang mengeluarkan list film dari TOP 3 IMDB seperti pada tampilan berikut pada saat dicoba dengan menekan tombol “Try it out” dan tombol “Execute”



- B. Menambahkan Movie baru yaitu urutan ke-4 pada TOP IMDB list dengan memanggil API pada bagian “POST /api/Movies”



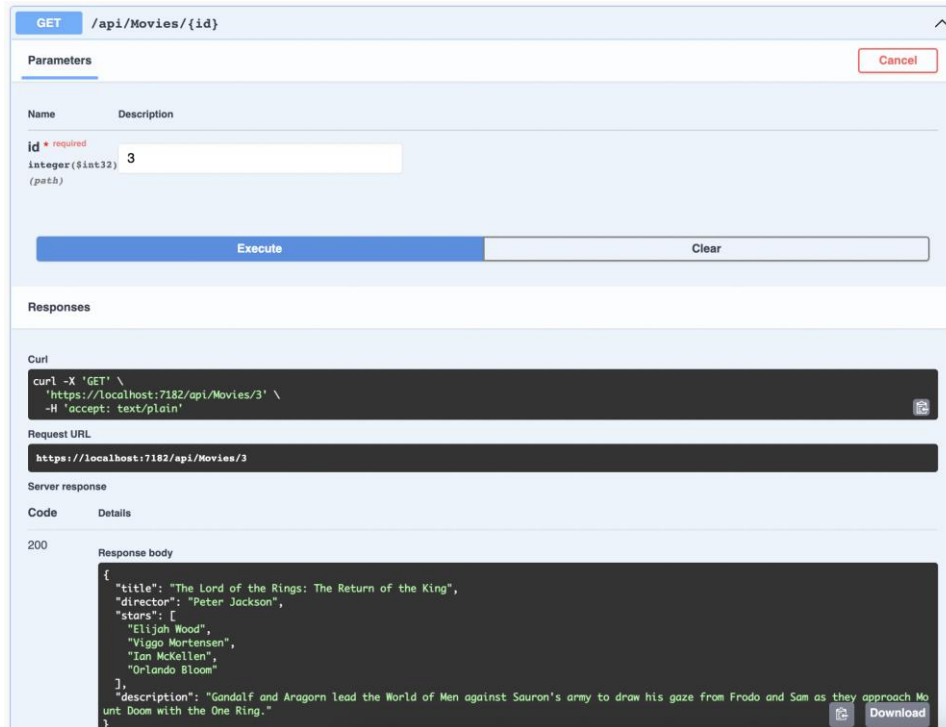
- C. Cek list/array dari semua Movie lagi dengan “GET /api/Movies”, pastikan Movie yang baru ditambahkan sebelumnya sudah ada:



```

Server response
Code    Details
200
Response body
{
  "title": "The Dark Knight",
  "director": "Christopher Nolan",
  "stars": [
    "Christian Bale",
    "Heath Ledger",
    "Aaron Eckhart",
    "Michael Caine"
  ],
  "description": "When the menace known as the Joker wreaks havoc and chaos on the people of Gotham, Batman must accept one of the greatest psychological and physical tests of his ability to fight injustice."
},
{
  "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."
}
]
Response headers
content-type: application/json; charset=utf-8
date: Thu, 14 Apr 2022 11:51:53 GMT
server: Kestrel
transfer-encoding: chunked
  
```

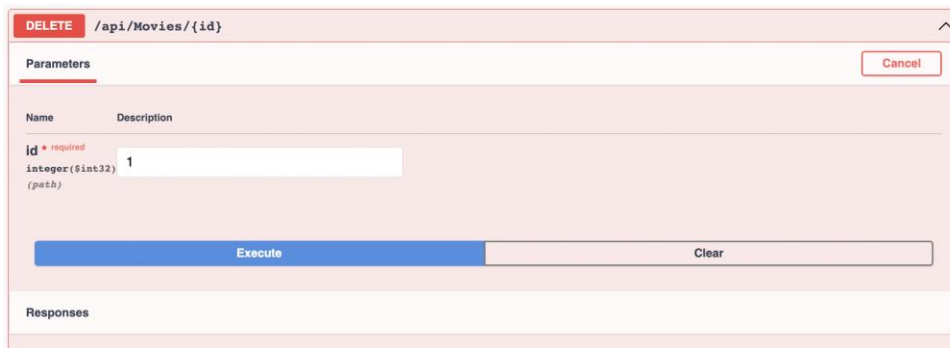
- D. Mencoba meminta Movie dengan index 3, “GET /api/Movies/3” yang seharusnya mengembalikan Movie yang baru saja ditambah:



```

GET /api/Movies/{id}
Parameters
Name    Description
id * required
integer(5int32)
(path)  3
Execute Clear
Responses
Curl
curl -X 'GET' \
'https://localhost:7182/api/Movies/3' \
-H 'accept: text/plain'
Request URL
https://localhost:7182/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."
}
  
```

- E. Menghapus objek Movie dengan index ke-1 dengan “DELETE /api/Movies/1”



DELETE /api/Movies/{id}

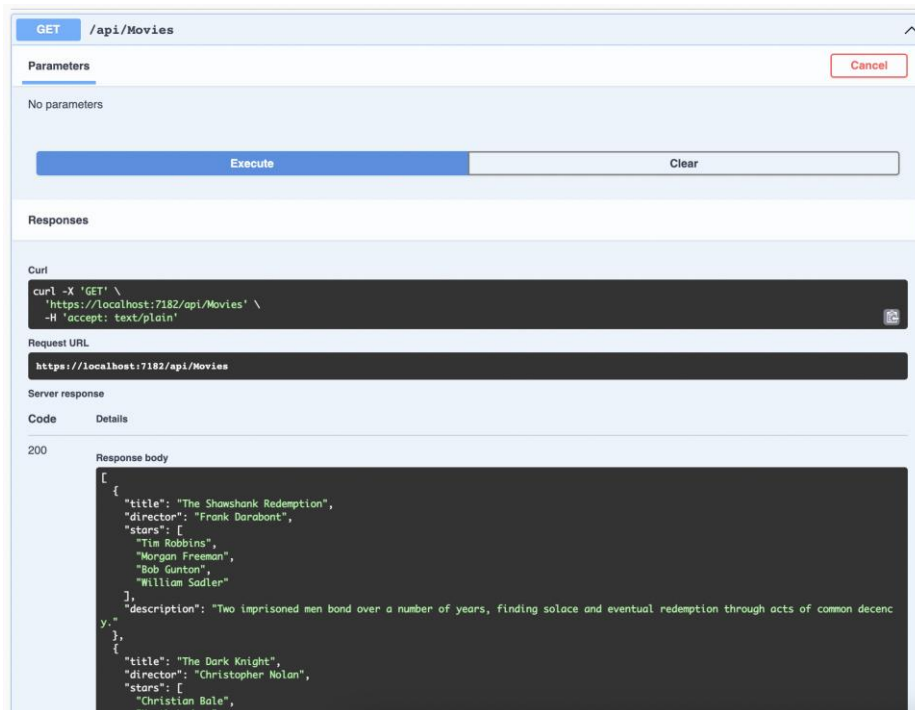
Parameters

Name	Description
id * required	
integer(\$int32)	1
(path)	

Execute Clear

Responses

- F. Cek list/array dari semua Movie sekali lagi dengan “GET /api/Movies”, film dengan ranking kedua “Godfather” sudah tidak ada di list:



GET /api/Movies

Parameters

No parameters

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7182/api/Movies' \
  -H 'accept: text/plain'
```

Request URL

https://localhost:7182/api/Movies

Server response

Code	Details
200	Response body

```
[
  {
    "title": "The Shawshank Redemption",
    "director": "Frank Darabont",
    "stars": [
      "Tim Robbins",
      "Morgan Freeman",
      "Bob Gunton",
      "William Sadler"
    ],
    "description": "Two imprisoned men bond over a number of years, finding solace and eventual redemption through acts of common decency."
  },
  {
    "title": "The Dark Knight",
    "director": "Christopher Nolan",
    "stars": [
      "Christian Bale",
      "Heath Ledger"
    ]
  }
]
```

5. MELAKUKAN COMMIT

Pada branch master/main:

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

6. PENGUMPULAN FILE/TUGAS JURNAL

Sebelum pengumpulan, praktikan wajib menunjukkan hasil run via share screen ke asprak. 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 kelompok
 - ii. Screenshot hasil run (hasil console output dari proses run sesuai bagian 4)
 - iii. Penjelasan singkat dari kode implementasi yang dibuat (beserta screenshot dari potongan source code yang dijelaskan).

Catatan: Tidak ada file docx/pdf (screenshot dan penjelasan) yang dikumpulkan DAN juga tidak melakukan demo/menunjukkan hasil run ke asprak maka nilai jurnal akan 0

KOMPONEN PENILAIAN

- A. Penggunaan Git dan Github **[10 pts]**
- B. Implementasi Web API **[45 pts]**
- C. Demo program/aplikasi **[30 pts]**
- D. Laporan jurnal **[15 pts]**