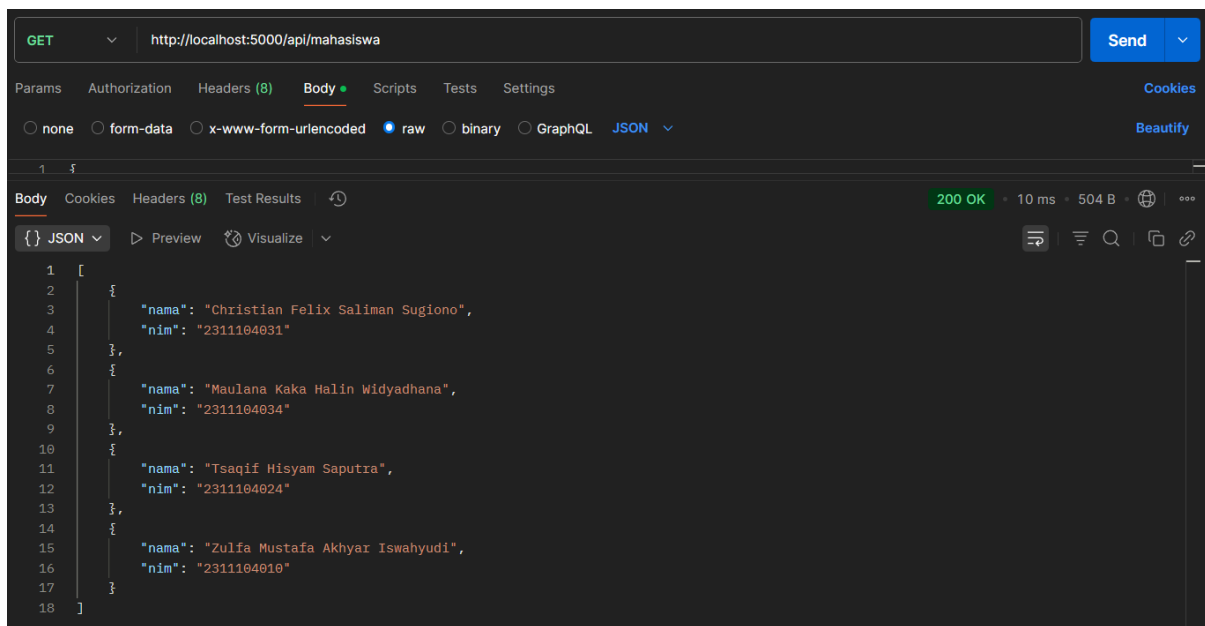
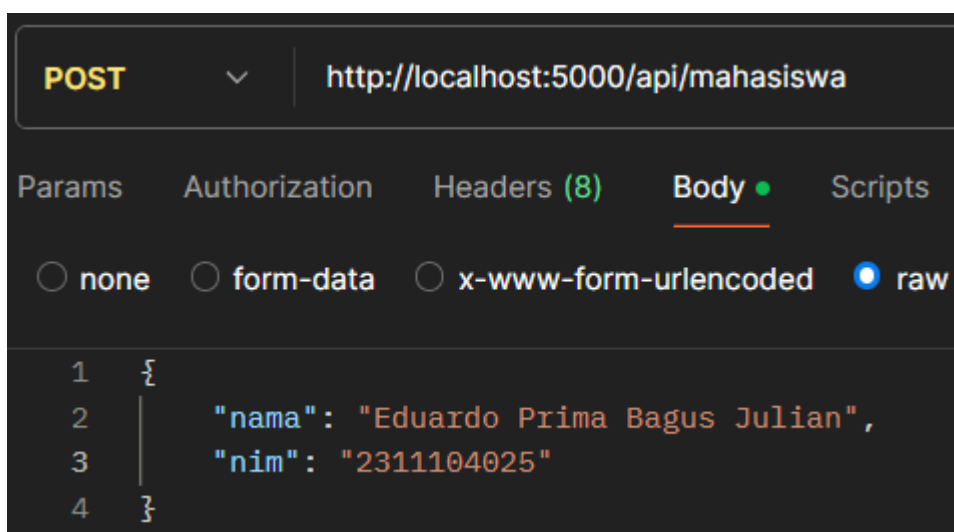


Tugas Pendahuluan Modul 9

Kodingan di TP ini adalah contoh sederhana server menggunakan Node.js dengan framework Express. Server ini mengatur API untuk mengelola data mahasiswa yang disimpan dalam array statis. Dengan middleware cors dan express.json(), server mendukung permintaan dari domain lain dan dapat memproses data JSON dari body request. Ada beberapa endpoint yang disediakan: GET /api/mahasiswa untuk mengambil seluruh daftar mahasiswa, GET /api/mahasiswa/:index untuk mengambil data mahasiswa berdasarkan indeks array, POST /api/mahasiswa untuk menambahkan mahasiswa baru dengan nama dan NIM, serta DELETE /api/mahasiswa/:index untuk menghapus mahasiswa berdasarkan indeks. Server berjalan pada port 5000 (atau bisa diganti sesuai kebutuhan), dan akan mencetak pesan ke konsol ketika berhasil dijalankan.



Untuk testing API pertama dilakukan dengan get untuk pemanggilan semua entity(yang bernama mahasiswa) dan dapat dilihat saya memasukan 4 mahasiswa di dalam kodingan saya.



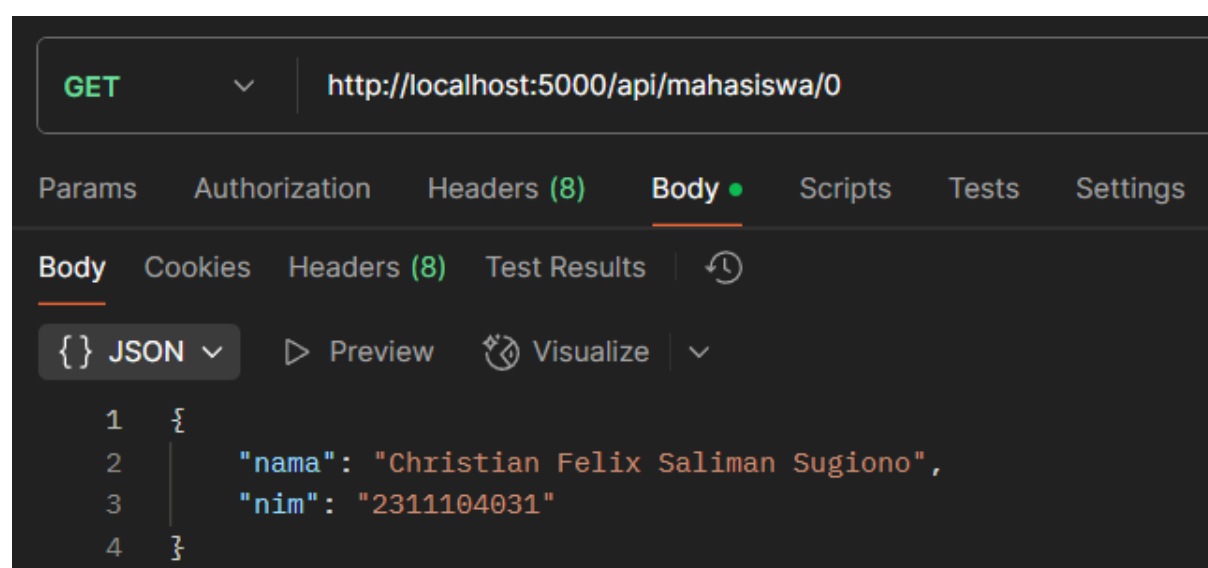
Lalu kita akan post (membuat mahasiswa baru), dengan end point post yang akan kita berikan atribut yang sama dengan mahasiswa yang lain agar tidak error.

```

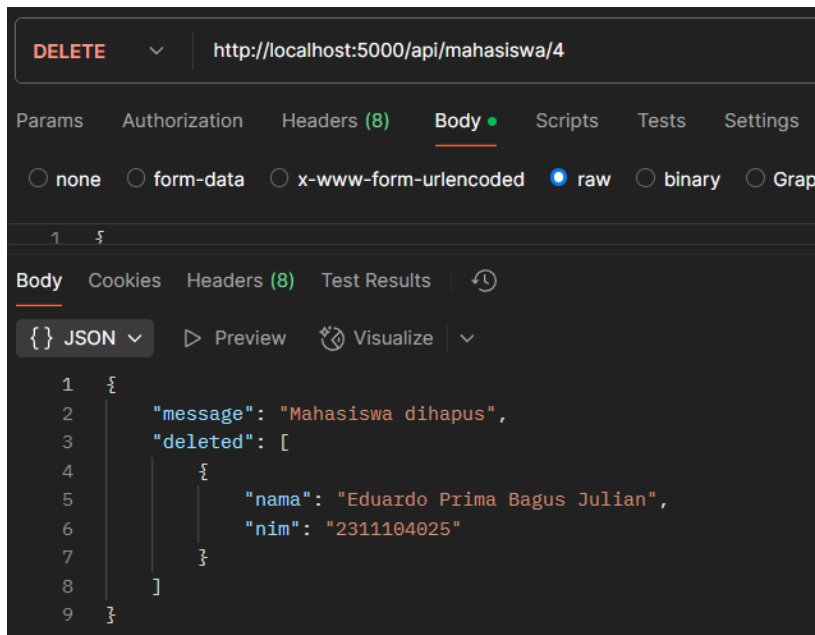
1  {
2      "message": "Mahasiswa ditambahkan",
3      "mahasiswa": [
4          {
5              "nama": "Christian Felix Saliman Sugiono",
6              "nim": "2311104031"
7          },
8          {
9              "nama": "Maulana Kaka Halin Widyadhana",
10             "nim": "2311104034"
11          },
12          {
13              "nama": "Tsaqif Hisyam Saputra",
14              "nim": "2311104024"
15          },
16          {
17              "nama": "Zulfa Mustafa Akhyar Iswahyudi",
18              "nim": "2311104010"
19          },
20          {
21              "nama": "Eduardo Prima Bagus Julian",
22              "nim": "2311104025"
23          }
24      ]
25  }

```

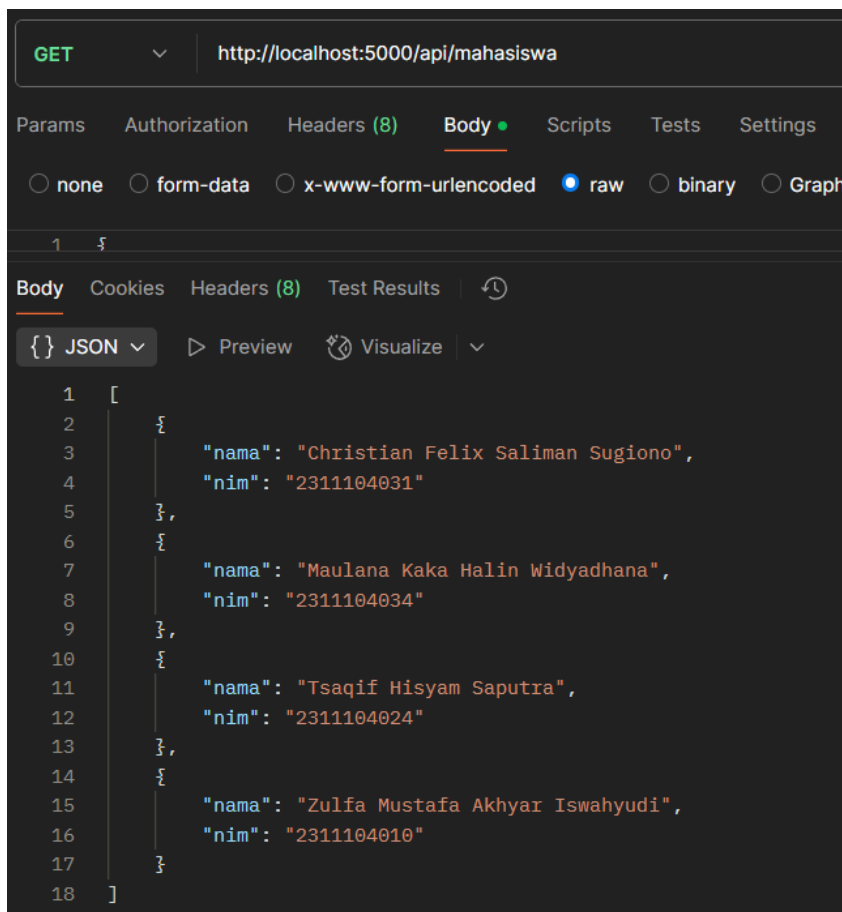
Dan ini output dari post yang kita lakukan.



Gambar diatas pemanggilan dari mahasiswa berdasarkan id yang kita pilih adalah id 0



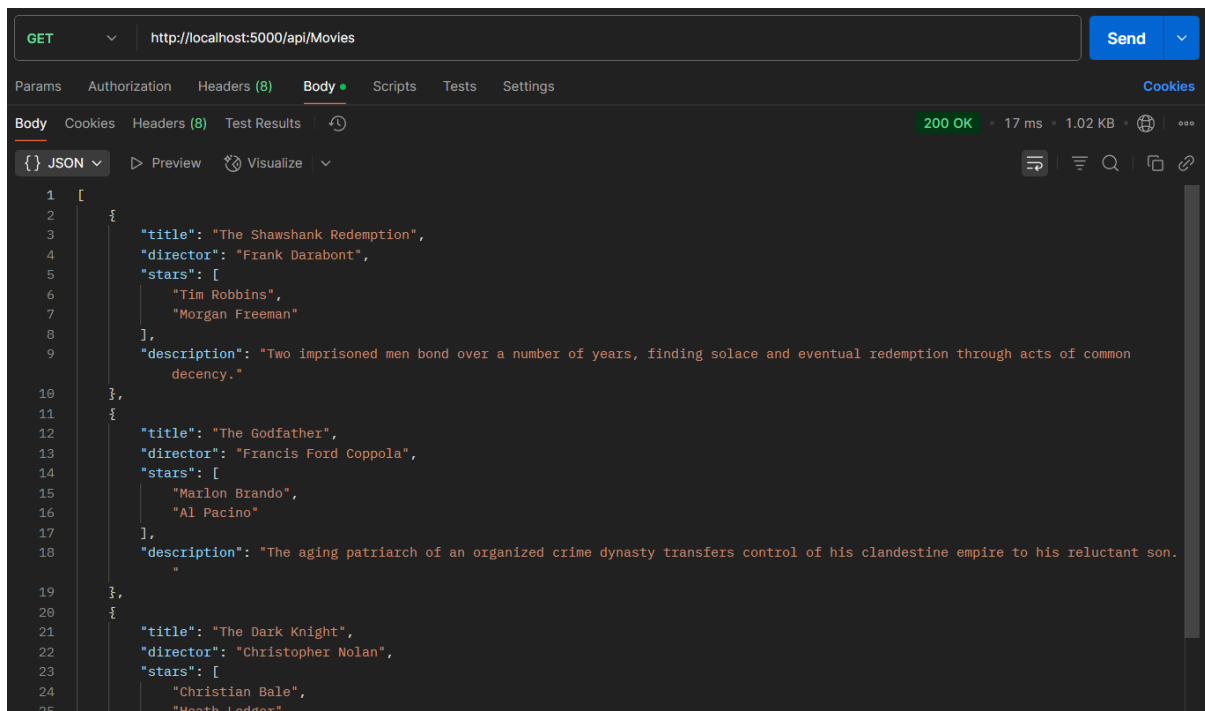
Penghapusan user berdasarkan id yang telar kita pilih, dari gambar ini kita memilih dengan user dengan ID 4.



Output setelah melakukan delete

Tugas Jurnal

Kode ini adalah sebuah server sederhana menggunakan Node.js dan Express untuk mengelola data film. Server menggunakan middleware cors agar dapat menerima request dari domain lain dan `express.json()` untuk memproses data JSON yang dikirim melalui body request. Data film disimpan dalam array statis bernama `movies`, dengan masing-masing film memiliki atribut `title`, `director`, `stars` (daftar aktor), dan `description`. API menyediakan beberapa endpoint: `GET /api/Movies` untuk mengambil semua film, `GET /api/Movies/:id` untuk mengambil detail film berdasarkan indeks array, `POST /api/Movies` untuk menambahkan film baru dengan data lengkap, dan `DELETE /api/Movies/:id` untuk menghapus film berdasarkan indeks. Server berjalan di port 5000 dan akan menampilkan pesan di konsol saat berhasil aktif. Struktur ini berguna untuk latihan membuat REST API dasar berbasis data film.

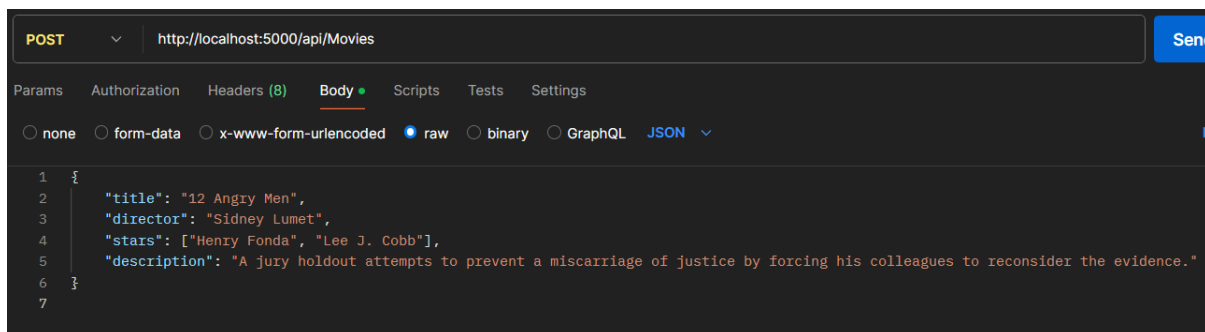


```
GET http://localhost:5000/api/Movies

200 OK · 17 ms · 1.02 KB

[
  {
    "title": "The Shawshank Redemption",
    "director": "Frank Darabont",
    "stars": [
      "Tim Robbins",
      "Morgan Freeman"
    ],
    "description": "Two imprisoned men bond over a number of years, finding solace and eventual redemption through acts of common decency."
  },
  {
    "title": "The Godfather",
    "director": "Francis Ford Coppola",
    "stars": [
      "Marlon Brando",
      "Al Pacino"
    ],
    "description": "The aging patriarch of an organized crime dynasty transfers control of his clandestine empire to his reluctant son."
  },
  {
    "title": "The Dark Knight",
    "director": "Christopher Nolan",
    "stars": [
      "Christian Bale",
      "Heath Ledger"
    ],
    "description": ""
  }
]
```

Pemanggilan semua elemen dari apa yang sudah dibuat di kodingan, atau static data



```
POST http://localhost:5000/api/Movies

{
  "title": "12 Angry Men",
  "director": "Sidney Lumet",
  "stars": ["Henry Fonda", "Lee J. Cobb"],
  "description": "A jury holdout attempts to prevent a miscarriage of justice by forcing his colleagues to reconsider the evidence."
}
```

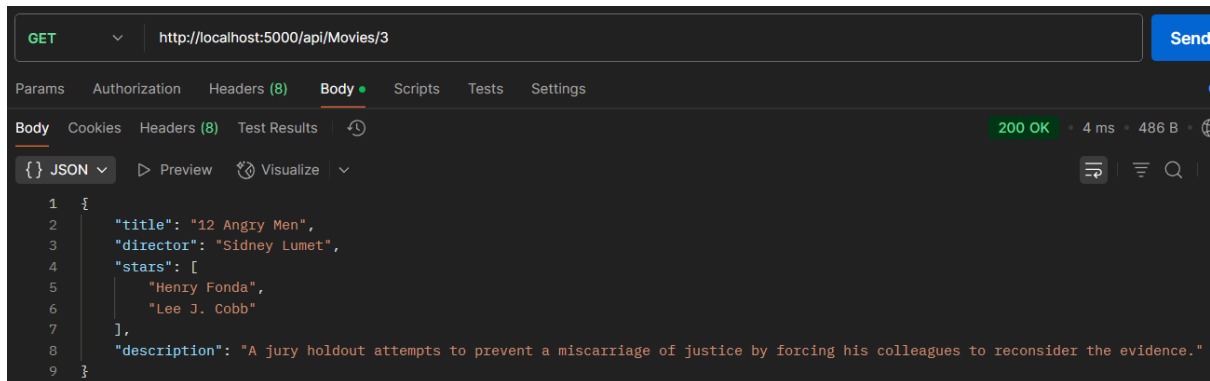
Pembuatan postingan movie yang baru

```

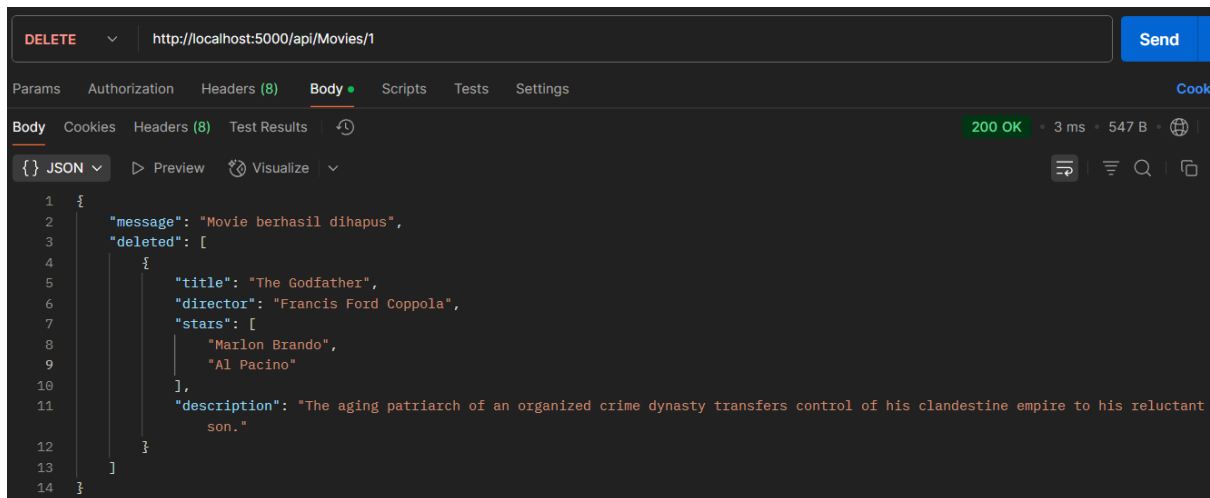
    {
      "title": "12 Angry Men",
      "director": "Sidney Lumet",
      "stars": [
        "Henry Fonda",
        "Lee J. Cobb"
      ],
      "description": "A jury holdout attempts to prevent a miscarriage of justice by forcing his colleagues to reconsider the evidence."
    }
  ]
}

```

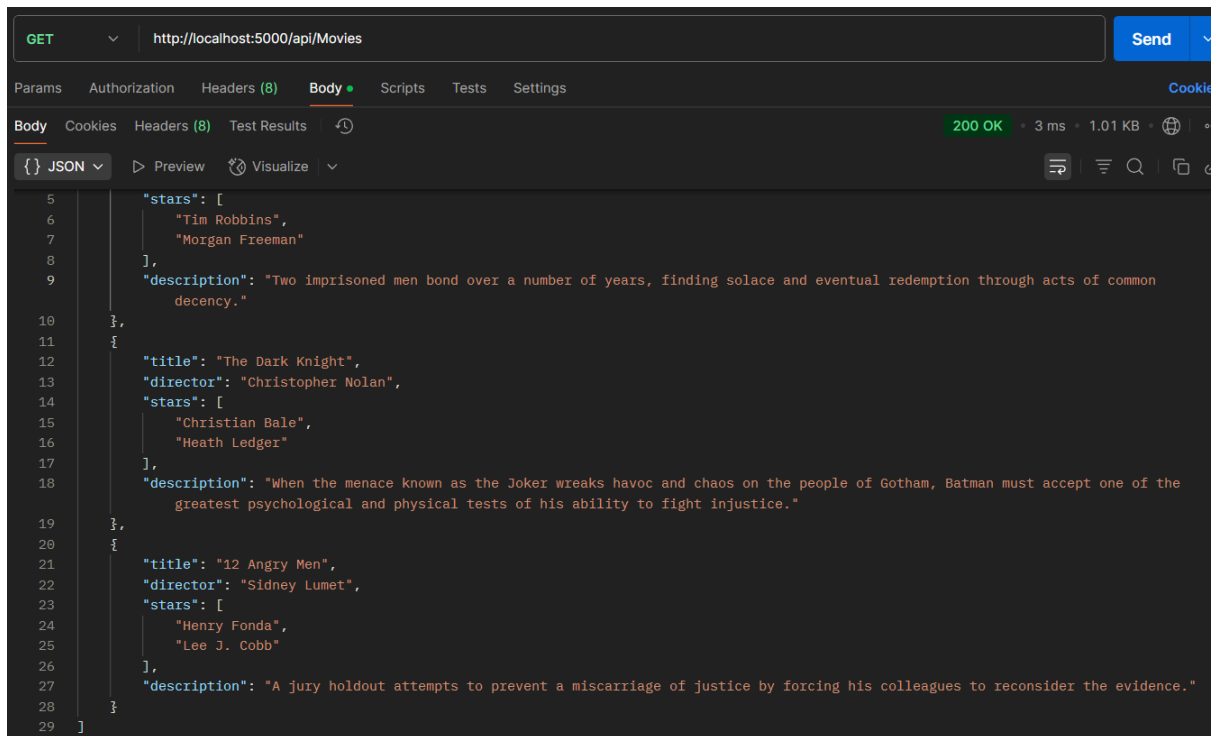
Tampilanya dalam body back end



Pemanggilan berdasarkan id film/urutan



Penghapusan dari suatu postingan/data dari list



The screenshot shows a REST client interface with a GET request to `http://localhost:5000/api/Movies`. The response is a 200 OK status with a 3 ms response time and 1.01 KB of data. The response body is a JSON array containing three movie objects. The first object is for "The Dark Knight" (id 1), the second for "12 Angry Men" (id 2), and the third for "The Godfather" (id 3). The first object is missing the "id" field.

```
5  {
6    "stars": [
7      "Tim Robbins",
8    ],
9    "description": "Two imprisoned men bond over a number of years, finding solace and eventual redemption through acts of common
10   decency."
11  },
12  {
13    "title": "The Dark Knight",
14    "director": "Christopher Nolan",
15    "stars": [
16      "Christian Bale",
17      "Heath Ledger"
18    ],
19    "description": "When the menace known as the Joker wreaks havoc and chaos on the people of Gotham, Batman must accept one of the
20    greatest psychological and physical tests of his ability to fight injustice."
21  },
22  {
23    "title": "12 Angry Men",
24    "director": "Sidney Lumet",
25    "stars": [
26      "Henry Fonda",
27      "Lee J. Cobb"
28    ],
29    "description": "A jury holdout attempts to prevent a miscarriage of justice by forcing his colleagues to reconsider the evidence."
30  }
31 ]
```

Pengecekan movie yang dengan id 1 sudah hilang