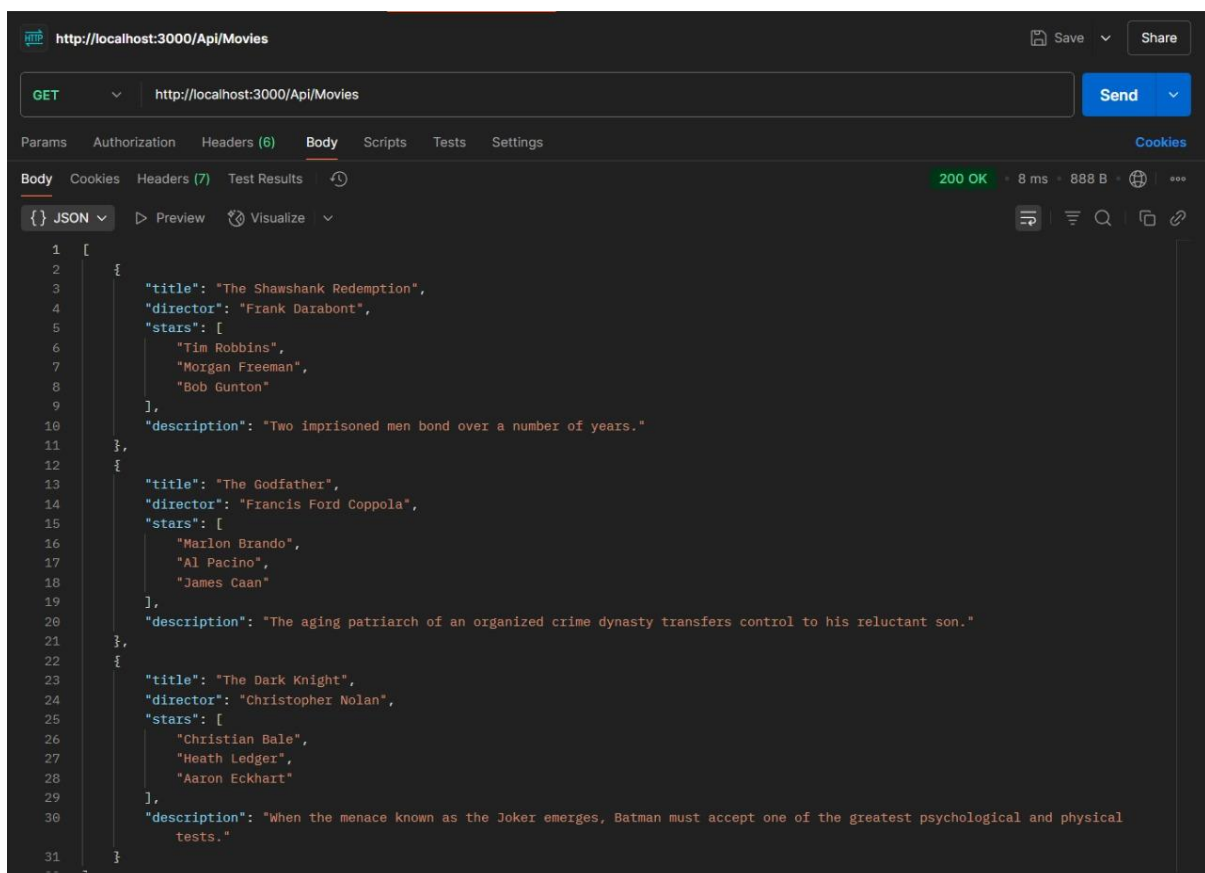


Jurnal Modul 9

Di bawah ini merupakan output dari aplikasi backend sederhana menggunakan **Node.js** dengan framework **Express** untuk membuat RESTful API yang mengelola data film. Data film disimpan dalam file `movies.json` dan dibaca saat server dijalankan menggunakan modul bawaan Node.js yaitu `fs`. API ini memiliki beberapa endpoint, yakni: `GET /api/Movies` untuk mengambil seluruh data film, `GET /api/Movies/:id` untuk mengambil data film berdasarkan indeks array-nya, `POST /api/Movies` untuk menambahkan data film baru ke dalam array, dan `DELETE /api/Movies/:id` untuk menghapus film berdasarkan indeksnya. Server berjalan di port 3000, dan setiap endpoint mengembalikan data dalam format JSON. Data film yang ditambahkan atau dihapus hanya berlaku selama server berjalan, karena perubahan tidak disimpan kembali ke file `movies.json`.

Pemanggilan semua elemen



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

- URL:** `http://localhost:3000/Api/Movies`
- Method:** `GET`
- Status:** `200 OK` (8 ms, 888 B)
- Response Format:** JSON
- Response Body:** A JSON array containing three movie objects.

```
1 [
2   {
3     "title": "The Shawshank Redemption",
4     "director": "Frank Darabont",
5     "stars": [
6       "Tim Robbins",
7       "Morgan Freeman",
8       "Bob Gunton"
9     ],
10    "description": "Two imprisoned men bond over a number of years."
11  },
12  {
13    "title": "The Godfather",
14    "director": "Francis Ford Coppola",
15    "stars": [
16      "Marlon Brando",
17      "Al Pacino",
18      "James Caan"
19    ],
20    "description": "The aging patriarch of an organized crime dynasty transfers control to his reluctant son."
21  },
22  {
23    "title": "The Dark Knight",
24    "director": "Christopher Nolan",
25    "stars": [
26      "Christian Bale",
27      "Heath Ledger",
28      "Aaron Eckhart"
29    ],
30    "description": "When the menace known as the Joker emerges, Batman must accept one of the greatest psychological and physical tests."
31  }
32 ]
```

Pembuatan postingan baru

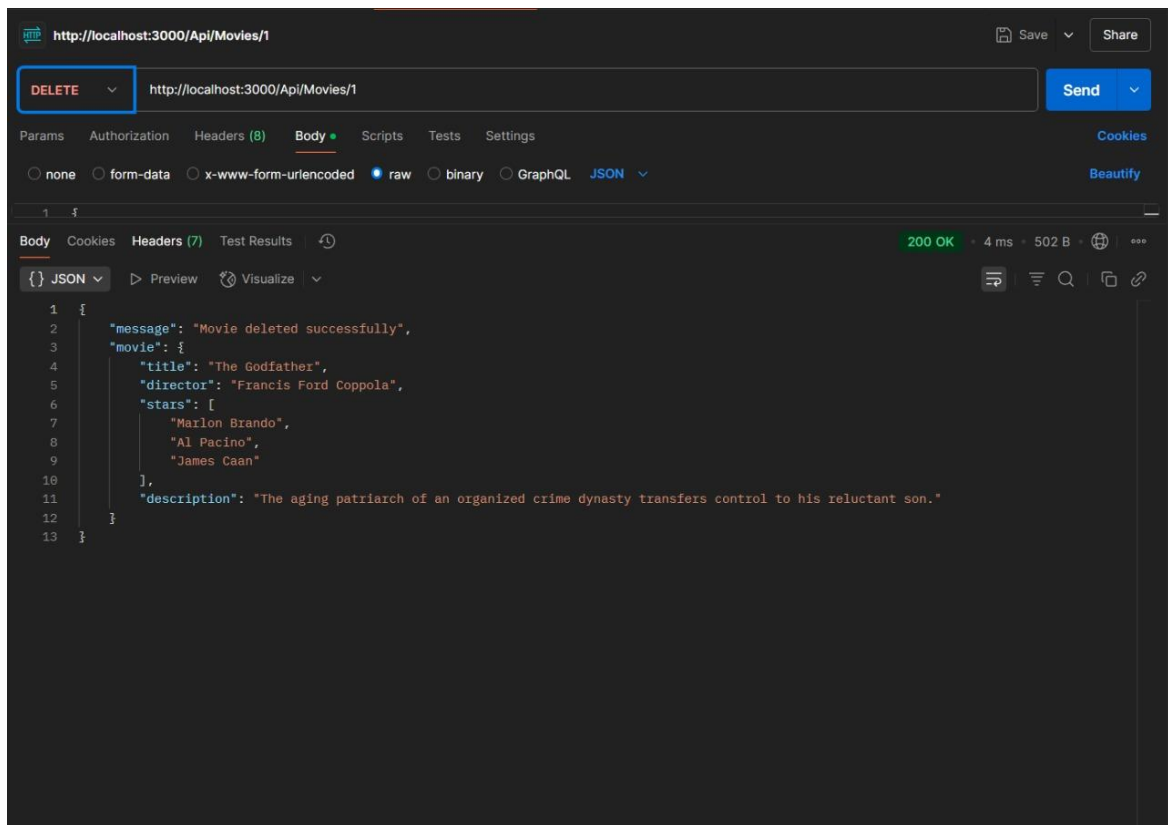
The screenshot displays a REST client interface with the following details:

- URL:** `http://localhost:3000/api/movies`
- Method:** `POST`
- Body Type:** `raw` (selected), with `JSON` as an alternative.
- Request Body (raw):**

```
1 {
2   "title": "American Psycho",
3   "director": "Mary Harron",
4   "stars": ["Christian Bale", "Willem Dafoe", "Jared Leto"],
5   "description": "A wealthy New York City investment banking executive hides his psychopathic ego from his circle of friends as he delves deeper into violent, hedonistic fantasies."
6 }
```
- Response Status:** `201 Created` (15 ms, 575 B)
- Response Body (JSON):**

```
1 {
2   "message": "Movie added successfully",
3   "movie": {
4     "title": "American Psycho",
5     "director": "Mary Harron",
6     "stars": [
7       "Christian Bale",
8       "Willem Dafoe",
9       "Jared Leto"
10    ],
11     "description": "A wealthy New York City investment banking executive hides his psychopathic ego from his circle of friends as he delves deeper into violent, hedonistic fantasies."
12   }
13 }
```

Hasil Delete



DELETE http://localhost:3000/Api/Movies/1

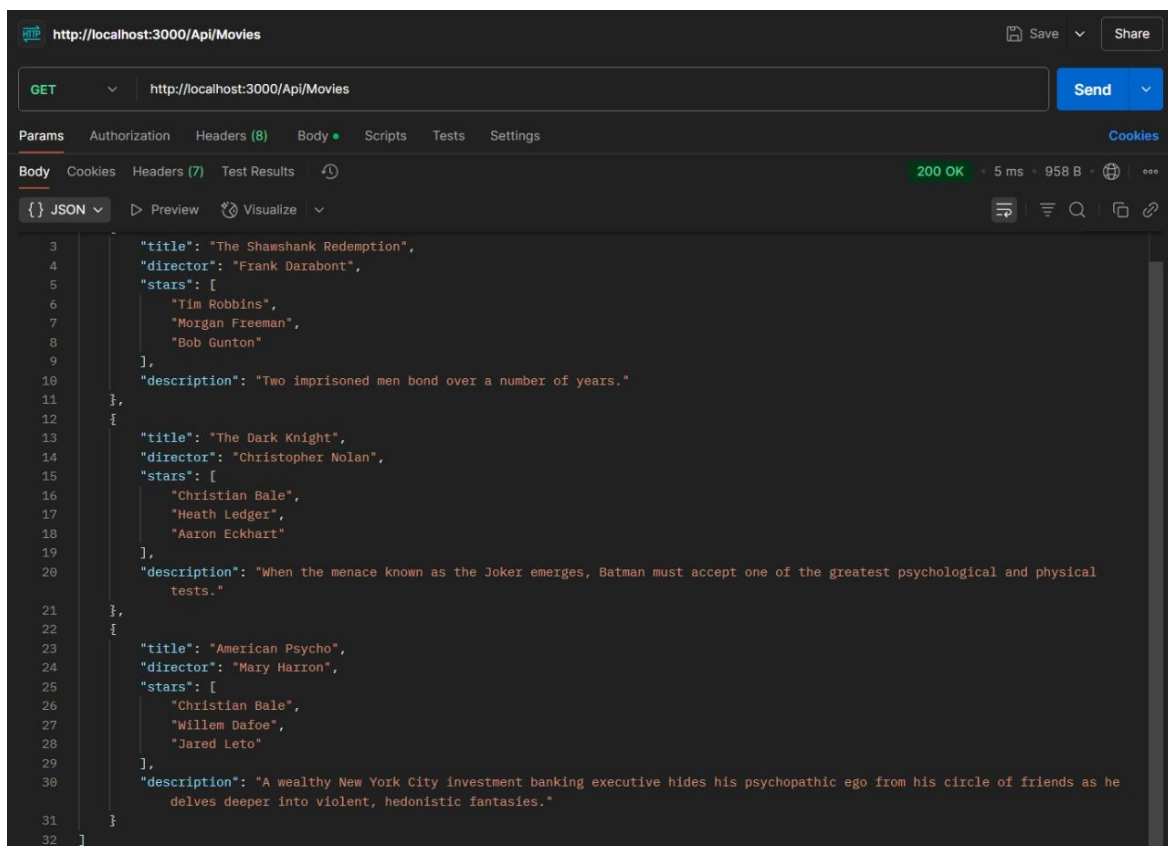
Params Authorization Headers (8) Body Scripts Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

Body Cookies Headers (7) Test Results

200 OK · 4 ms · 502 B

```
{
  "message": "Movie deleted successfully",
  "movie": {
    "title": "The Godfather",
    "director": "Francis Ford Coppola",
    "stars": [
      "Marlon Brando",
      "Al Pacino",
      "James Caan"
    ],
    "description": "The aging patriarch of an organized crime dynasty transfers control to his reluctant son."
  }
}
```



GET http://localhost:3000/Api/Movies

Params Authorization Headers (8) Body Scripts Tests Settings Cookies

Body Cookies Headers (7) Test Results

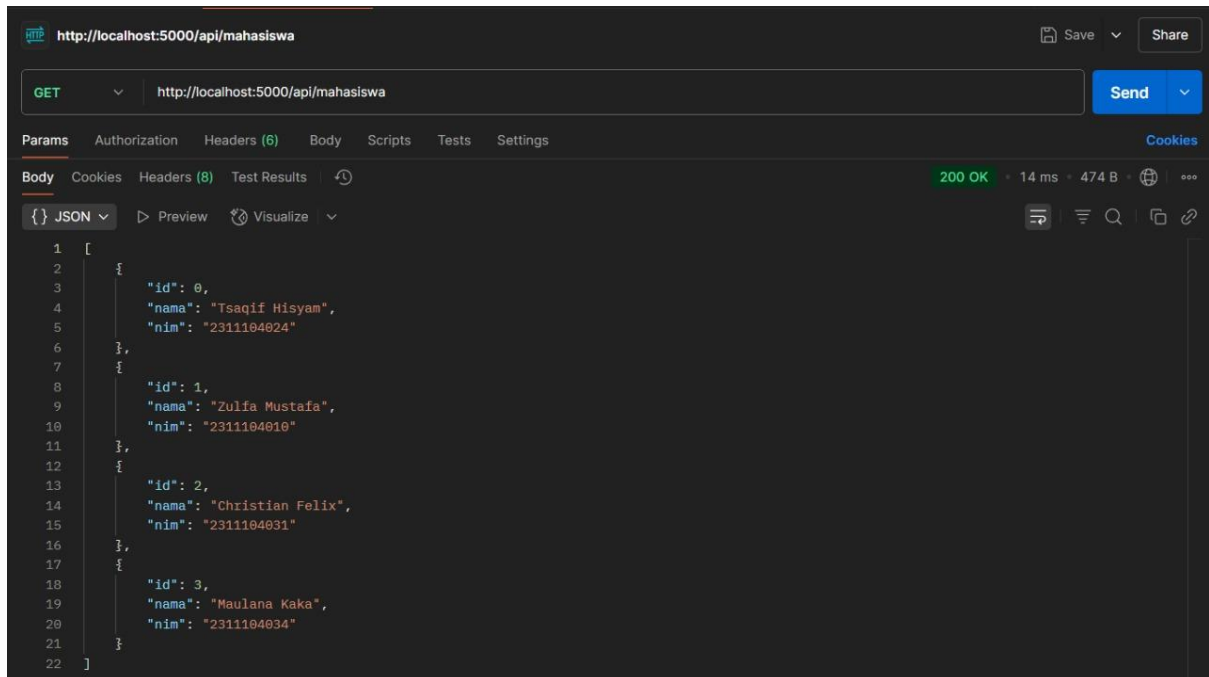
200 OK · 5 ms · 958 B

```
[
  {
    "title": "The Shawshank Redemption",
    "director": "Frank Darabont",
    "stars": [
      "Tim Robbins",
      "Morgan Freeman",
      "Bob Gunton"
    ],
    "description": "Two imprisoned men bond over a number of years."
  },
  {
    "title": "The Dark Knight",
    "director": "Christopher Nolan",
    "stars": [
      "Christian Bale",
      "Heath Ledger",
      "Aaron Eckhart"
    ],
    "description": "When the menace known as the Joker emerges, Batman must accept one of the greatest psychological and physical tests."
  },
  {
    "title": "American Psycho",
    "director": "Mary Harron",
    "stars": [
      "Christian Bale",
      "Willem Dafoe",
      "Jared Leto"
    ],
    "description": "A wealthy New York City investment banking executive hides his psychopathic ego from his circle of friends as he delves deeper into violent, hedonistic fantasies."
  }
]
```

Tugas Pendahuluan 9

Kurang lebih sama seperti yang terdapat pada jurnal tetapi data yang dimanipulasi disini adalah data Mahasiswa.

Pemanggilan semua elemen



Hasil Post

Postman interface showing a successful POST request to `http://localhost:5000/api/mahasiswa`. The request body is a JSON object:

```
1 {
2   "nama": "Yoga Eka Pratama",
3   "nim": "2311104023"
4 }
```

The response is a **201 Created** status with a response time of 21 ms and a body size of 374 B. The response body is a JSON object:

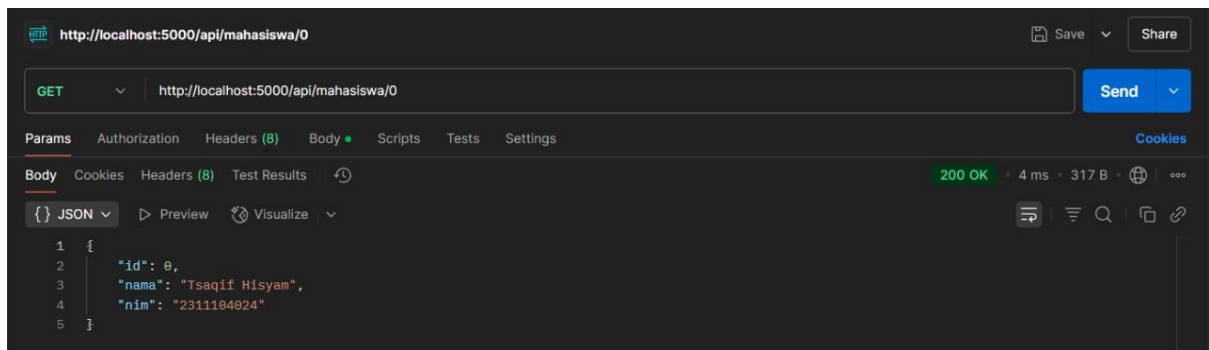
```
1 {
2   "message": "Mahasiswa ditambahkan",
3   "mahasiswa": {
4     "id": 4,
5     "nama": "Yoga Eka Pratama",
6     "nim": "2311104023"
7   }
8 }
```

Hasil Semua Post

Postman interface showing a successful GET request to `http://localhost:5000/api/mahasiswa`. The response is a **200 OK** status with a response time of 4 ms and a body size of 529 B. The response body is a JSON array containing all student data:

```
1 [
2   {
3     "id": 0,
4     "nama": "Tsaqif Hisyam",
5     "nim": "2311104024"
6   },
7   {
8     "id": 1,
9     "nama": "Zulfa Mustafa",
10    "nim": "2311104010"
11  },
12  {
13    "id": 2,
14    "nama": "Christian Felix",
15    "nim": "2311104031"
16  },
17  {
18    "id": 3,
19    "nama": "Maulana Kaka",
20    "nim": "2311104034"
21  },
22  {
23    "id": 4,
24    "nama": "Yoga Eka Pratama",
25    "nim": "2311104023"
26  }
27 ]
```

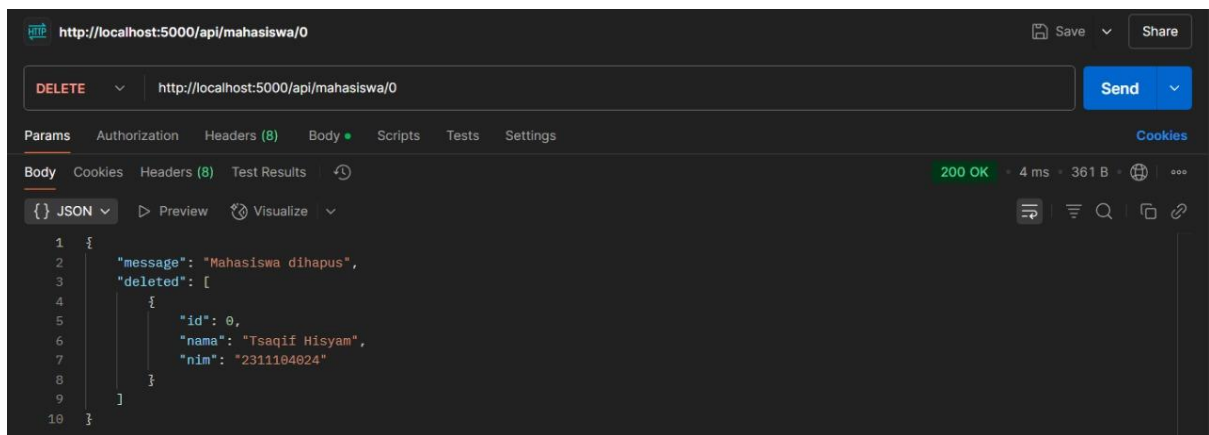
Mencari data mahasiswa berdasarkan id



REST client interface showing a GET request to `http://localhost:5000/api/mahasiswa/0`. The response is a 200 OK status with a JSON body containing student data for id 0.

```
1 {
2   "id": 0,
3   "nama": "Tsaqif Hisyam",
4   "nim": "2311104024"
5 }
```

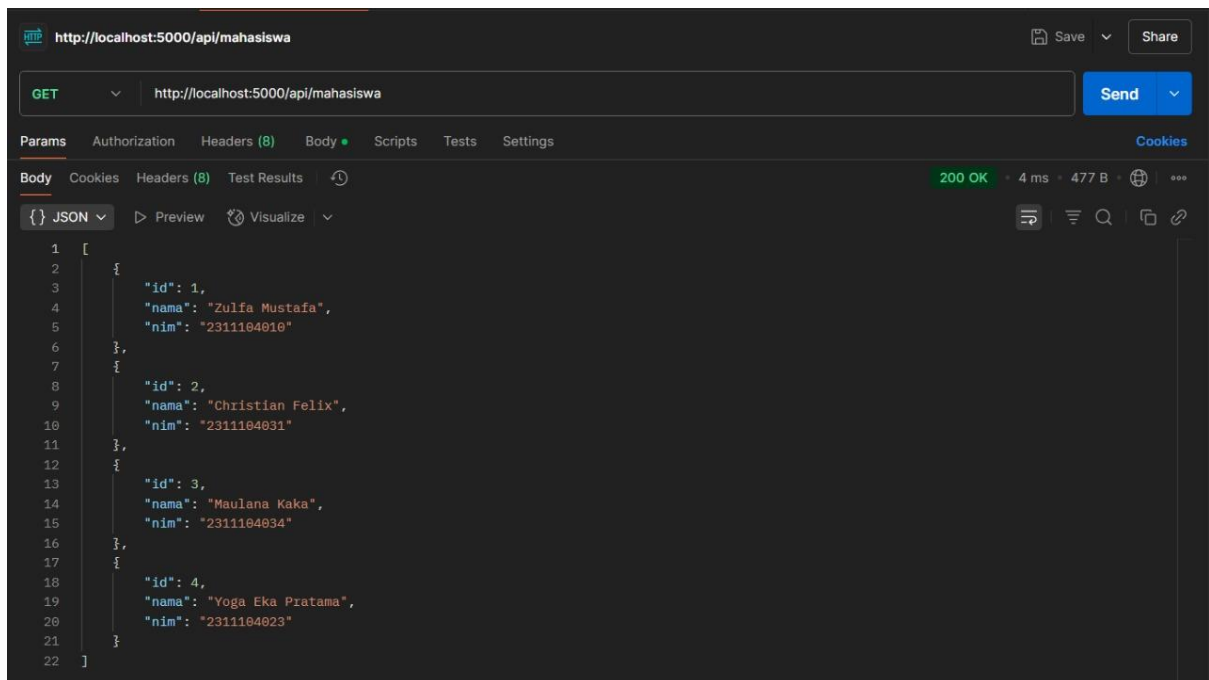
Hasil Delete



REST client interface showing a DELETE request to `http://localhost:5000/api/mahasiswa/0`. The response is a 200 OK status with a JSON body indicating the student was deleted.

```
1 {
2   "message": "Mahasiswa dihapus",
3   "deleted": [
4     {
5       "id": 0,
6       "nama": "Tsaqif Hisyam",
7       "nim": "2311104024"
8     }
9   ]
10 }
```

Hasil Semua Delete



REST client interface showing a GET request to `http://localhost:5000/api/mahasiswa`. The response is a 200 OK status with a JSON array containing student data for ids 1 through 4.

```
1 [
2   {
3     "id": 1,
4     "nama": "Zulfa Mustafa",
5     "nim": "2311104010"
6   },
7   {
8     "id": 2,
9     "nama": "Christian Felix",
10    "nim": "2311104031"
11  },
12  {
13    "id": 3,
14    "nama": "Maulana Kaka",
15    "nim": "2311104034"
16  },
17  {
18    "id": 4,
19    "nama": "Yoga Eka Pratama",
20    "nim": "2311104023"
21  }
22 ]
```