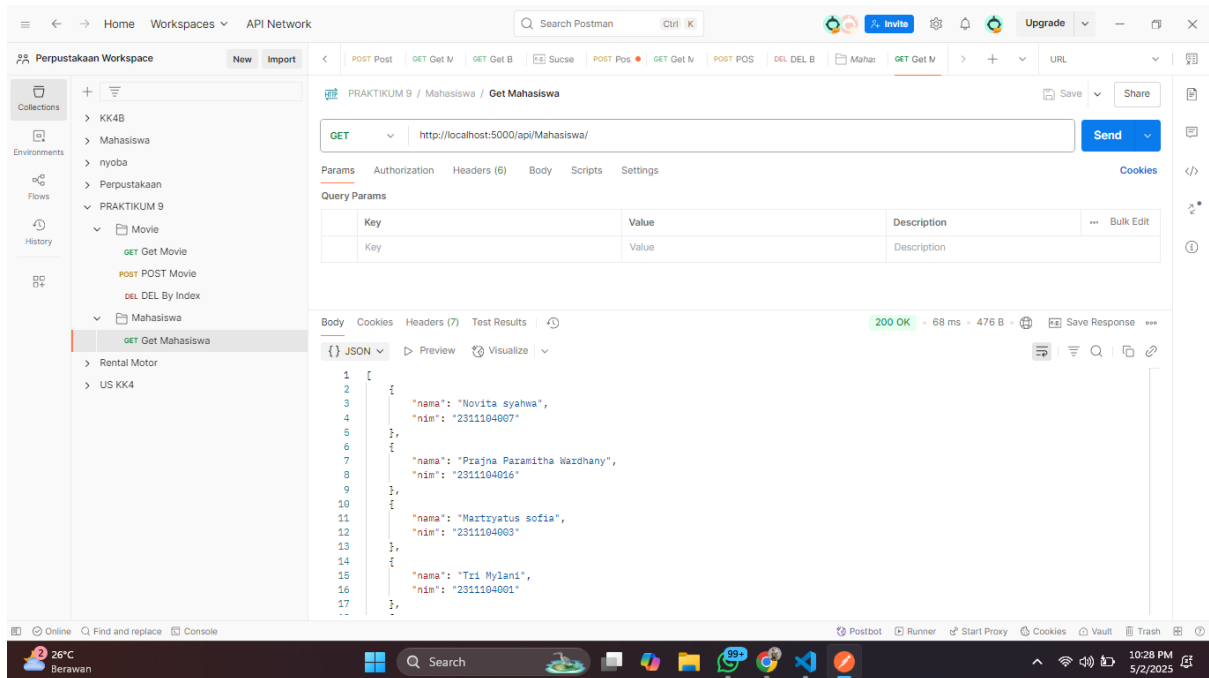


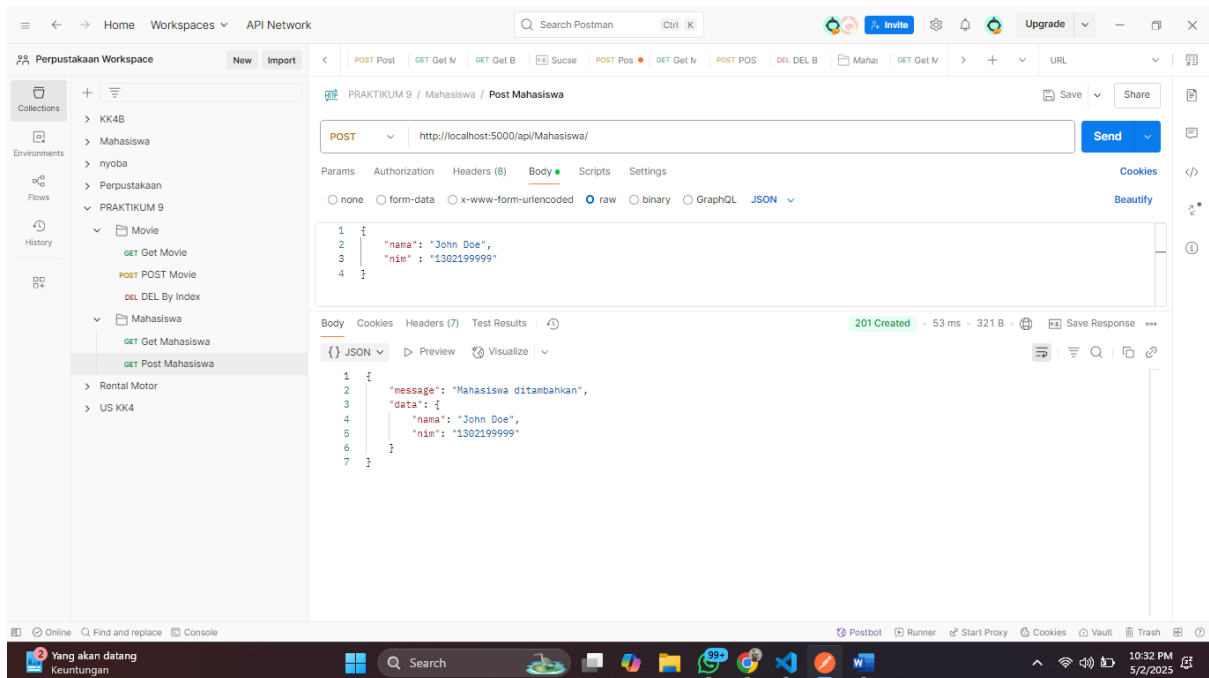
Novita Syahwa Tri Hapsari

2311104007/SE07-01

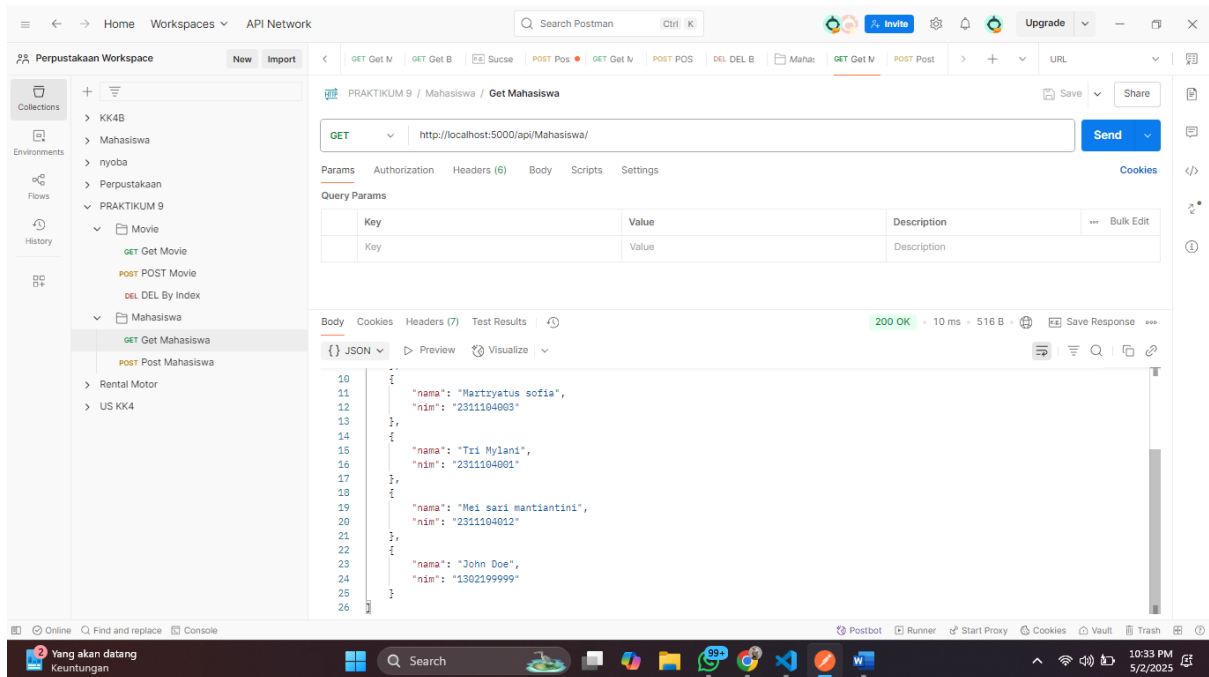
Mencoba “GET /api/mahasiswa” saat baru dijalankan (mengeluarkan list nama mahasiswa dan nim anggota kelompok):



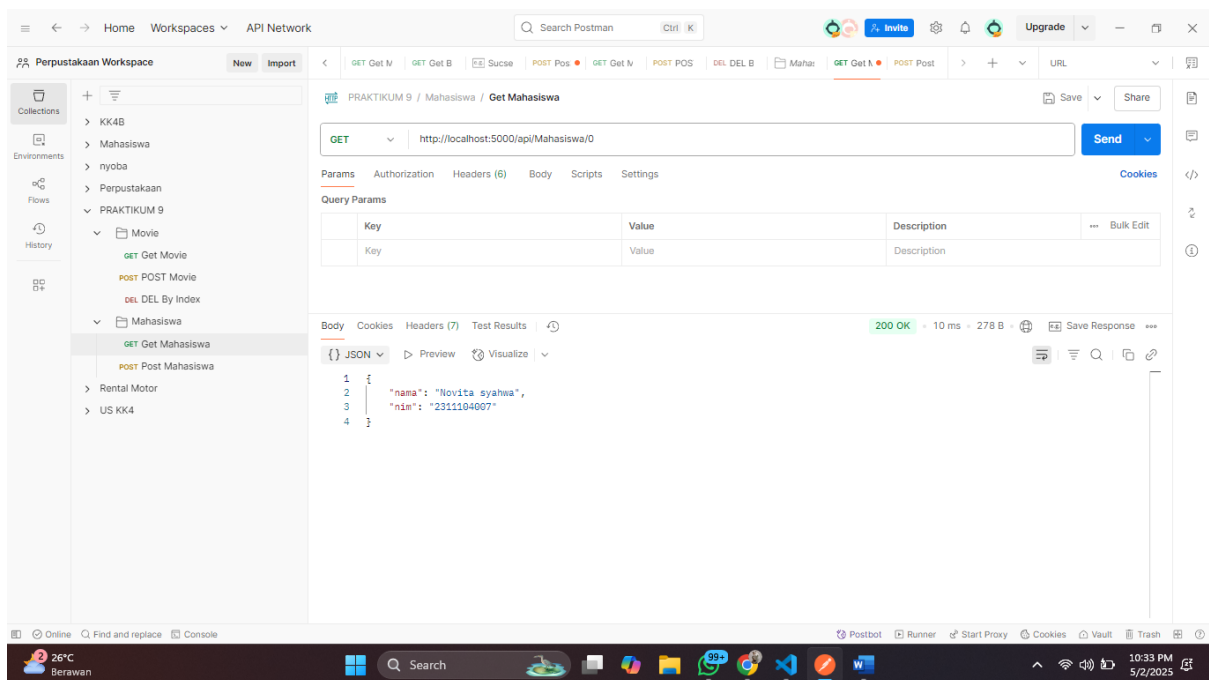
Menambahkan mahasiswa => Nama: “John Doe” dan NIM: “1302199999” dengan “POST /api/mahasiswa”



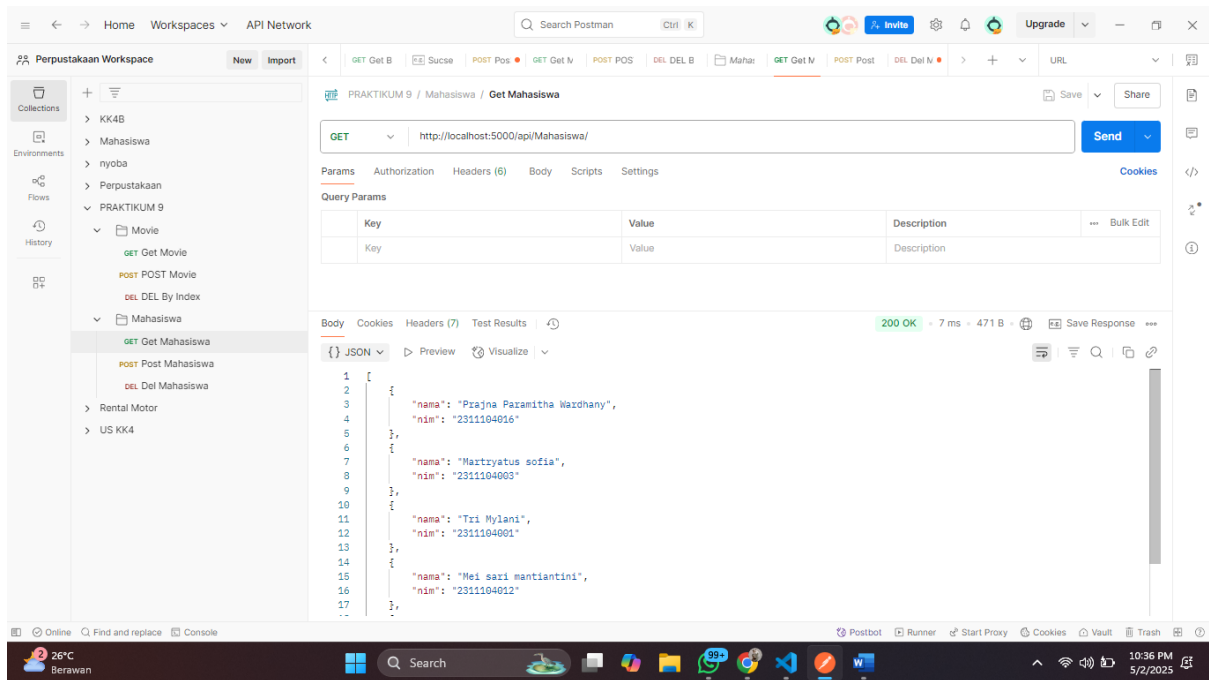
Cek list/array dari semua mahasiswa lagi dengan “GET /api/mahasiswa”, pastikan mahasiswa yang baru ditambahkan sebelumnya ada di list mahasiswa:



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



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

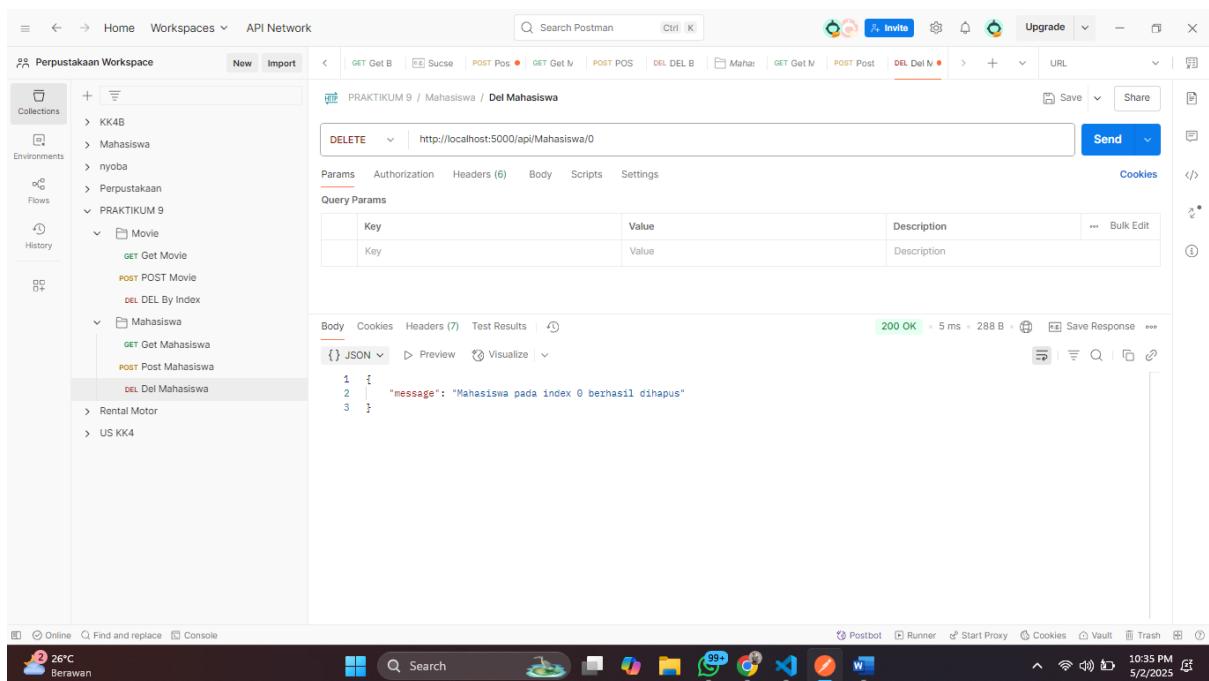


The screenshot shows the Postman interface with a workspace named 'Perpustakaan Workspace'. The selected collection is 'PRAKTIKUM 9', and the endpoint is 'Mahasiswa'. The request method is 'GET' and the URL is 'http://localhost:5000/api/mahasiswa/'. The response status is '200 OK' with a response time of 7 ms and a body size of 471 B. The response body is a JSON array of 5 student objects.

Key	Value	Description
Key	Value	Description

```
1 {
2   {
3     "nama": "Prajna Paramitha Wardhany",
4     "nim": "2311104016"
5   },
6   {
7     "nama": "Martryatus sofia",
8     "nim": "2311104003"
9   },
10  {
11    "nama": "Tri Mylani",
12    "nim": "2311104061"
13  },
14  {
15    "nama": "Nai sari mantiantini",
16    "nim": "2311104012"
17  },
18 }
```

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



The screenshot shows the Postman interface with a workspace named 'Perpustakaan Workspace'. The selected collection is 'PRAKTIKUM 9', and the endpoint is 'Mahasiswa'. The request method is 'DELETE' and the URL is 'http://localhost:5000/api/mahasiswa/0'. The response status is '200 OK' with a response time of 5 ms and a body size of 288 B. The response body is a JSON object with a success message.

Key	Value	Description
Key	Value	Description

```
1 {
2   "message": "Mahasiswa pada index 0 berhasil dihapus"
3 }
```

Code:

```
const express = require('express');
const app = express();
const PORT = 5000;
app.use(express.json());

let mahasiswaList = [
```

```

{ nama: "Novita syahwa", nim: "2311104007" },
{ nama: "Prajna Paramitha Wardhany", nim: "2311104016" },
{ nama: "Martryatus sofia", nim: "2311104003" },
{ nama: "Tri Mylani", nim: "2311104001" },
{ nama: "Mei sari mantiantini", nim: "2311104012" },
];

app.get('/api/mahasiswa', (req, res) => {
  res.json(mahasiswaList);
});

// GET /api/mahasiswa/:index Untuk engembalikan mahasiswa berdasarkan index
app.get('/api/mahasiswa/:index', (req, res) => {
  const index = parseInt(req.params.index);
  if (index < 0 || index >= mahasiswaList.length) {
    return res.status(404).json({ message: 'Index tidak ditemukan' });
  }
  res.json(mahasiswaList[index]);
});

// POST /api/mahasiswa Untuk enambahkan mahasiswa baru
app.post('/api/mahasiswa', (req, res) => {
  const { nama, nim } = req.body;
  mahasiswaList.push({ nama, nim });
  res.status(201).json({ message: 'Mahasiswa ditambahkan', data: { nama, nim } });
});

// DELETE /api/mahasiswa/:index Untuk menghapus mahasiswa berdasarkan index
app.delete('/api/mahasiswa/:index', (req, res) => {
  const index = parseInt(req.params.index);
  if (index < 0 || index >= mahasiswaList.length) {
    return res.status(404).json({ message: 'Index tidak ditemukan' });
  }
  mahasiswaList.splice(index, 1);
  res.json({ message: `Mahasiswa pada index ${index} berhasil dihapus` });
});

// Running server
app.listen(PORT, () => {
  console.log(`Server berjalan di http://localhost:${PORT}/api/mahasiswa`);
});

```

Penjelasan code:

Code tersebut berjalan di port 5000 dan berfungsi untuk mengelola data mahasiswa. Data disimpan dalam array mahasiswaList yang berisi nama dan NIM. Program ini memiliki beberapa endpoint API: GET /api/mahasiswa untuk menampilkan seluruh data mahasiswa, GET /api/mahasiswa/:index untuk menampilkan data mahasiswa berdasarkan index, POST /api/mahasiswa untuk menambahkan mahasiswa baru, dan DELETE /api/mahasiswa/:index untuk menghapus data mahasiswa berdasarkan index. Jika index yang diminta tidak valid, server akan mengembalikan pesan error.