LAPORAN PRAKTIKUM
KONSTRUKSI PERANGKAT LUNAK

MODUL IX

REST API NODE.JS

Disusun Oleh :

Satria Ariq Adelard Dompas

S1SE-06-02

Asisten Praktikum :

Muhamad Taufiq Hidayat

Dosen Pengampu :

Riyan Dwi Yulian Prakoso, S.Kom., M.Kom.

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK

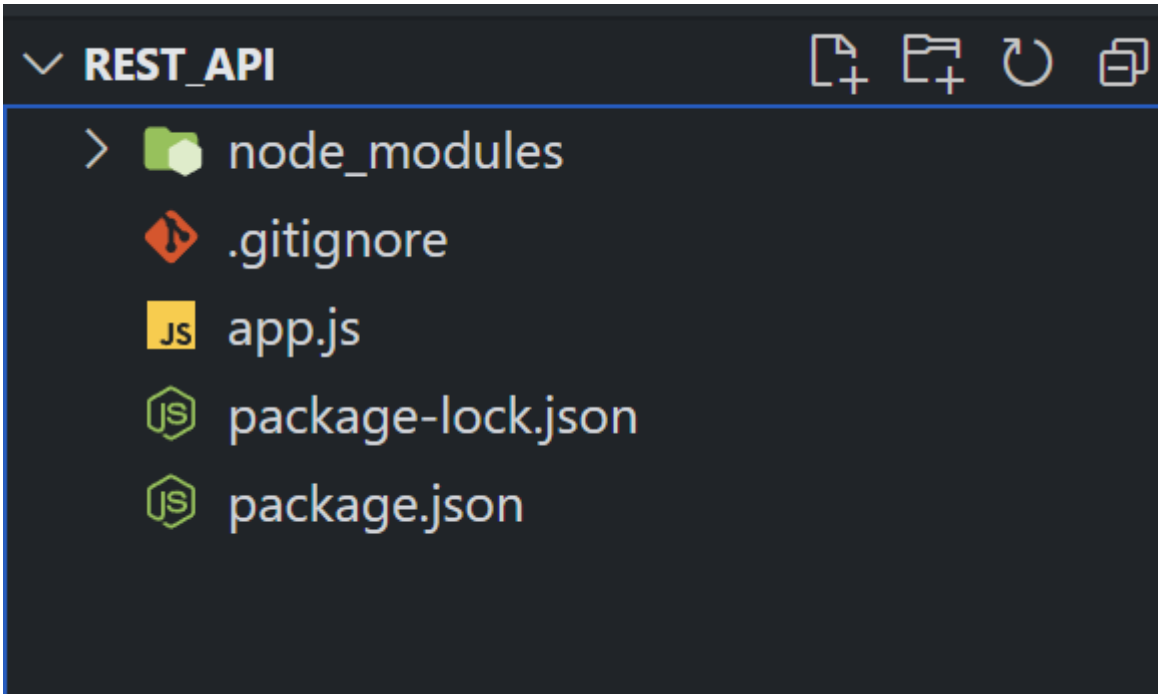
DIREKTORAT TELKOM KAMPUS PURWOKERTO

2025

BAB I
GUIDED

Hasil Program:

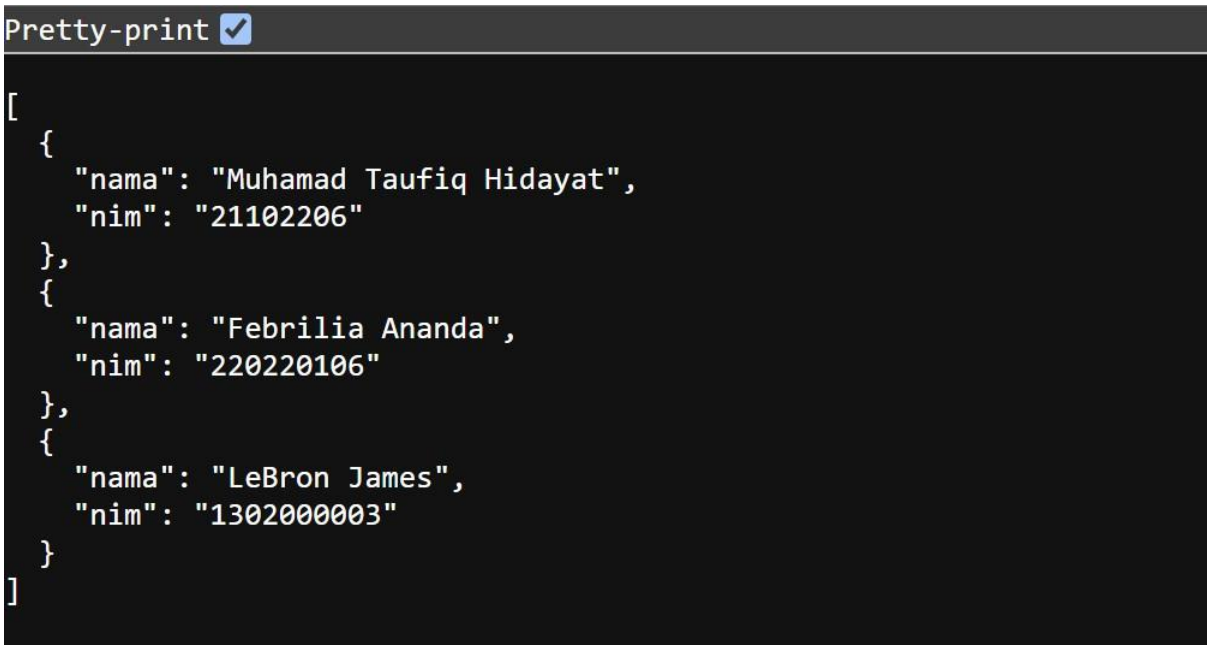
1. Inisialisasi Struktur Folder



2. app.js

```js
const express = require('express');
const app = express();
const port = 3000;

app.use(express.json());

// Simpan data mahasiswa di array static
let mahasiswa = [
  { nama: "Muhamad Taufiq Hidayat", nim: "21102206" },
  { nama: "Febrilia Ananda", nim: "220220106" },
  { nama: "LeBron James", nim: "1302000003" }
];

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

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

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

// DELETE mahasiswa berdasarkan index
app.delete('/api/mahasiswa/:index', (req, res) => {
  const index = parseInt(req.params.index);
  if (index >= 0 && index < mahasiswa.length) {
    mahasiswa.splice(index, 1);
    res.json({ message: 'Data berhasil dihapus' });
  } else {
    res.status(404).json({ message: 'Data tidak ditemukan' });
  }
});

app.listen(port, () => {
  console.log(`Server berjalan di http://localhost:${port}`);
});
```

Contoh output:



Penjelasan Kode:

Kode di atas merupakan implementasi API sederhana menggunakan Express.js untuk mengelola data mahasiswa yang disimpan dalam array statis. API ini mendukung operasi dasar seperti membaca semua data mahasiswa (GET /api/mahasiswa), membaca satu mahasiswa berdasarkan index (GET /api/mahasiswa/:index), menambah data mahasiswa baru dengan mengirim JSON lewat body (POST /api/mahasiswa), serta menghapus data mahasiswa berdasarkan index (DELETE /api/mahasiswa/:index). Middleware express.json() digunakan agar server dapat membaca data JSON dari body permintaan. Server berjalan pada port 3000, dan saat dijalankan akan menampilkan pesan bahwa server aktif di http://localhost:3000.

BAB II
UNGUIDED

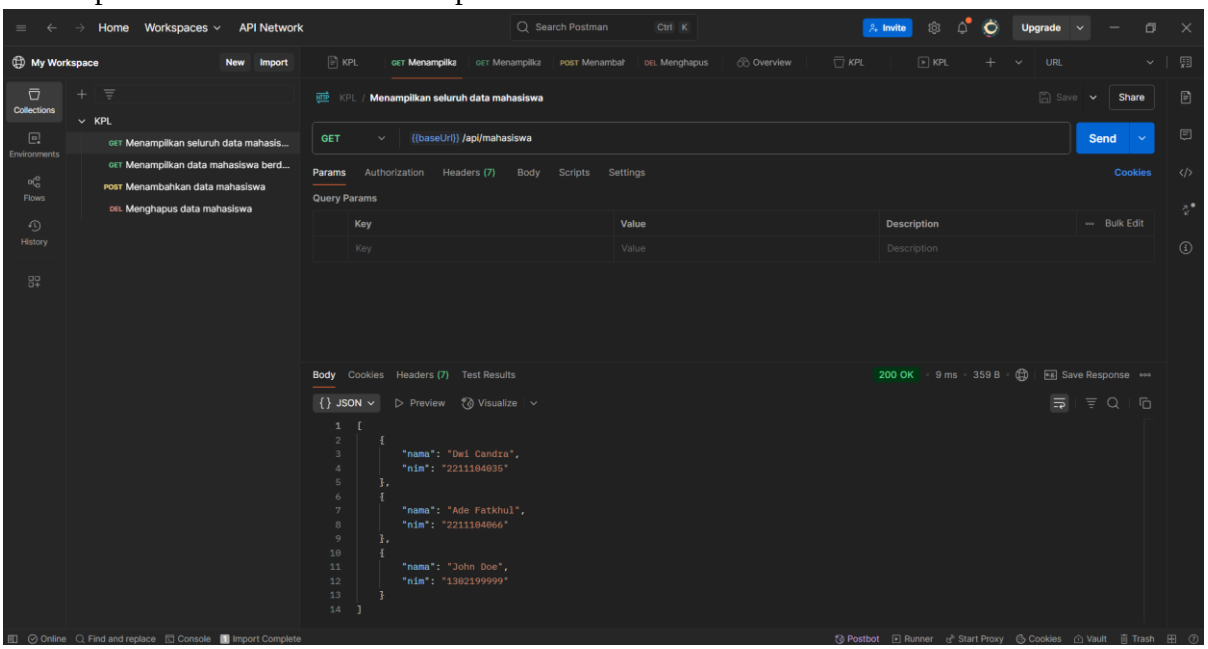
Dokumentasi REST API dengan Postman

1.app.js

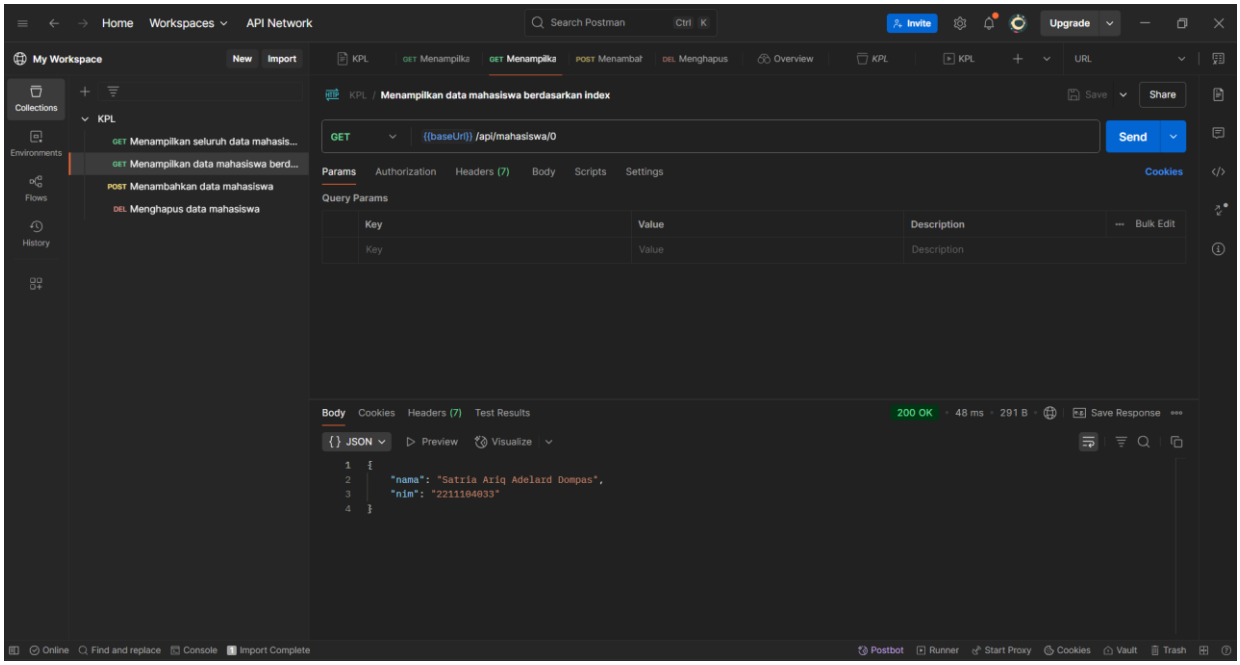
```
const express = require("express");
const app = express();
const port = 3000;

app.use(express.json());

// Data awal mahasiswa
let mahasiswa = [
    { nama: "Satria Ariq Adelard Dompas", nim: "2211104033" },
    { nama: "Dwi Candra", nim: "2211104035" },
    { nama: "Ade Fatkhul", nim: "2211104066" },
  ];


// GET semua mahasiswa
app.get("/api/mahasiswa", (req, res) => {
  res.json(mahasiswa);
});

// GET mahasiswa berdasarkan index
app.get("/api/mahasiswa/:index", (req, res) => {
  const index = parseInt(req.params.index);
  if (index >= 0 && index < mahasiswa.length) {
    res.json(mahasiswa[index]);
  } else {
    res.status(404).json({ message: "Data tidak ditemukan" });
  }
});

// POST mahasiswa baru
app.post("/api/mahasiswa", (req, res) => {
  const { nama, nim } = req.body;
  mahasiswa.push({ nama, nim });
  res.status(201).json({ message: "Data berhasil ditambahkan" });
});

// DELETE mahasiswa berdasarkan index
app.delete("/api/mahasiswa/:index", (req, res) => {
  const index = parseInt(req.params.index);
  if (index >= 0 && index < mahasiswa.length) {
    mahasiswa.splice(index, 1);
    res.json({ message: "Data berhasil dihapus" });
  } else {
```

```
    res.status(404).json({ message: "Data tidak ditemukan" });
  }
});

app.listen(port, () => {
  console.log(`Server berjalan di http://localhost:${port}`);
});
```

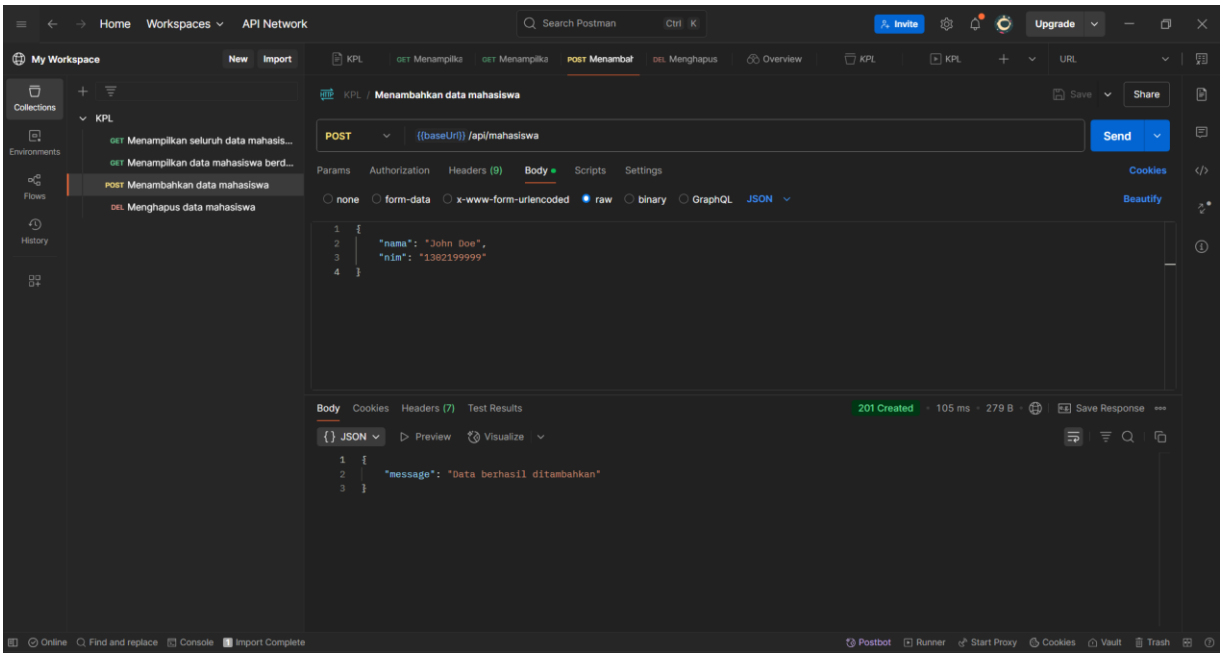
2. Dokumentasi Postman
   a. GET /api/mahasiswa untuk Menampilkan seluruh data mahasiswa



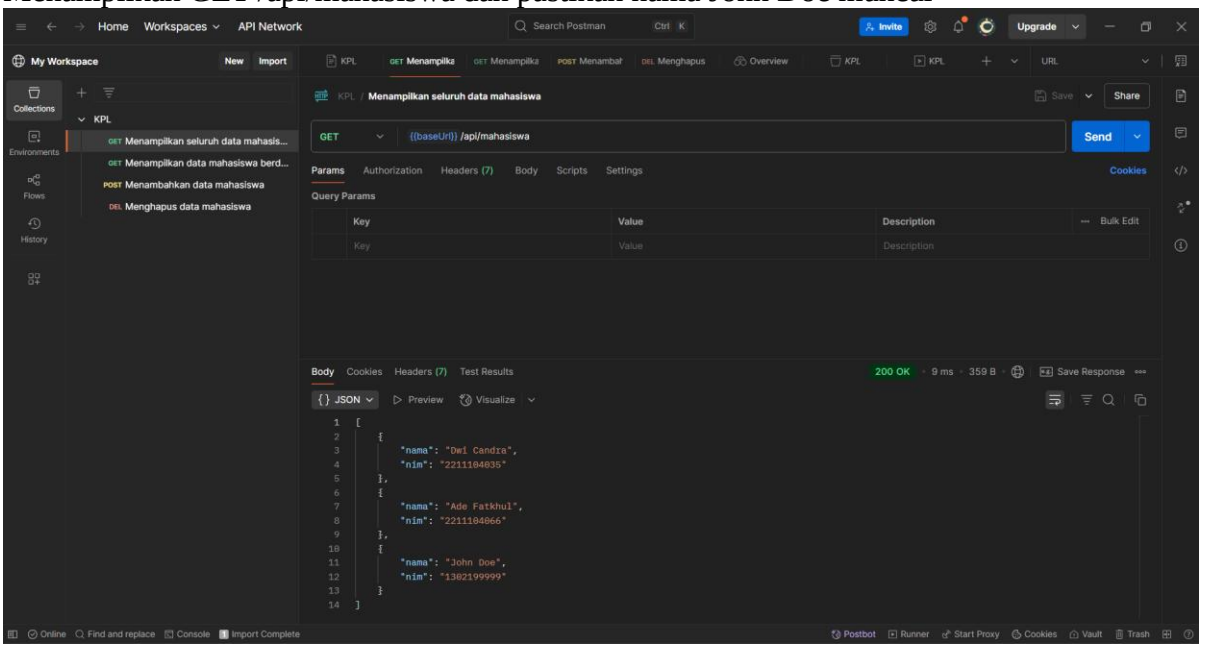
   b. GET /api/mahasiswa/:index untuk Menampilkan data mahasiswa berdasarkan indeks array



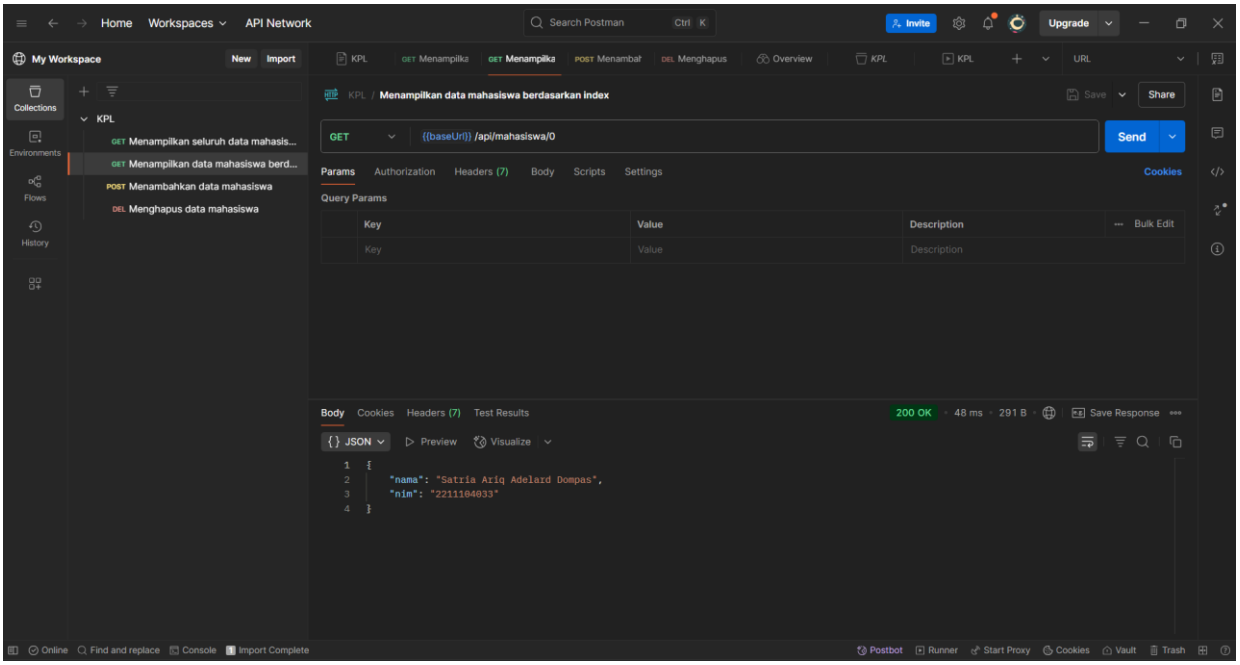
c. Menjalankan POST /api/mahasiswa untuk menambahkan John Doe – 1302199999



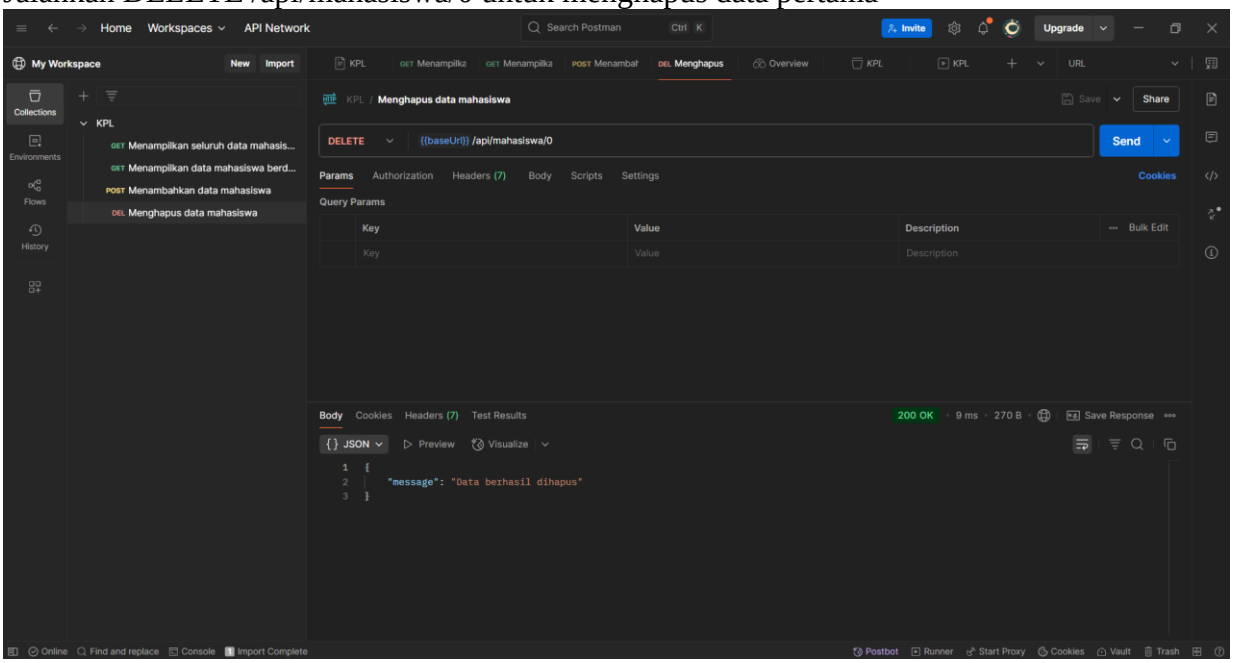
d. Menampilkan GET /api/mahasiswa dan pastikan nama John Doe muncul



e. Jalankan GET /api/mahasiswa/0 dan pastikan data pertama (nama Anda) muncul



f. Jalankan DELETE /api/mahasiswa/0 untuk menghapus data pertama



g. Jalankan GET /api/mahasiswa lagi dan pastikan data Anda sudah hilang