

Nama : Salman Alfarisi

Kelas : S1SE-07-01

NIM : 2311104036

```
const Mahasiswa = require('./Mahasiswa');

let mahasiswa = [
  new Mahasiswa("Salman Alfarisi", "231110436"),
  new Mahasiswa("Ajeng Nur Laila", "2311100001"),
  new Mahasiswa("Danang Aji Setiawan", "2311100002")
];

module.exports = mahasiswa;
```

```
class Mahasiswa {
  constructor(nama, nim) {
    this.nama = nama;
    this.nim = nim;
  }
}

module.exports = Mahasiswa;
```

```

const express = require('express');
const bodyParser = require('body-parser');
const Mahasiswa = require('./Mahasiswa');
const mahasiswa = require('./dataMahasiswa');

const app = express();
app.use(bodyParser.json());

// 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).send("Mahasiswa tidak ditemukan");
  }
});

// POST mahasiswa baru
app.post('/api/mahasiswa', (req, res) => {
  const { nama, nim } = req.body;
  const mhsBaru = new Mahasiswa(nama, nim);
  mahasiswa.push(mhsBaru);
  res.status(201).send("Mahasiswa 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.send("Mahasiswa dihapus");
  } else {
    res.status(404).send("Mahasiswa tidak ditemukan");
  }
});

const PORT = 3000;
app.listen(PORT, () => {
  console.log(`API berjalan di http://localhost:${PORT}`);
});

app.get('/', (req, res) => {
  res.send('API Mahasiswa aktif. Gunakan endpoint /api/mahasiswa');
});

```

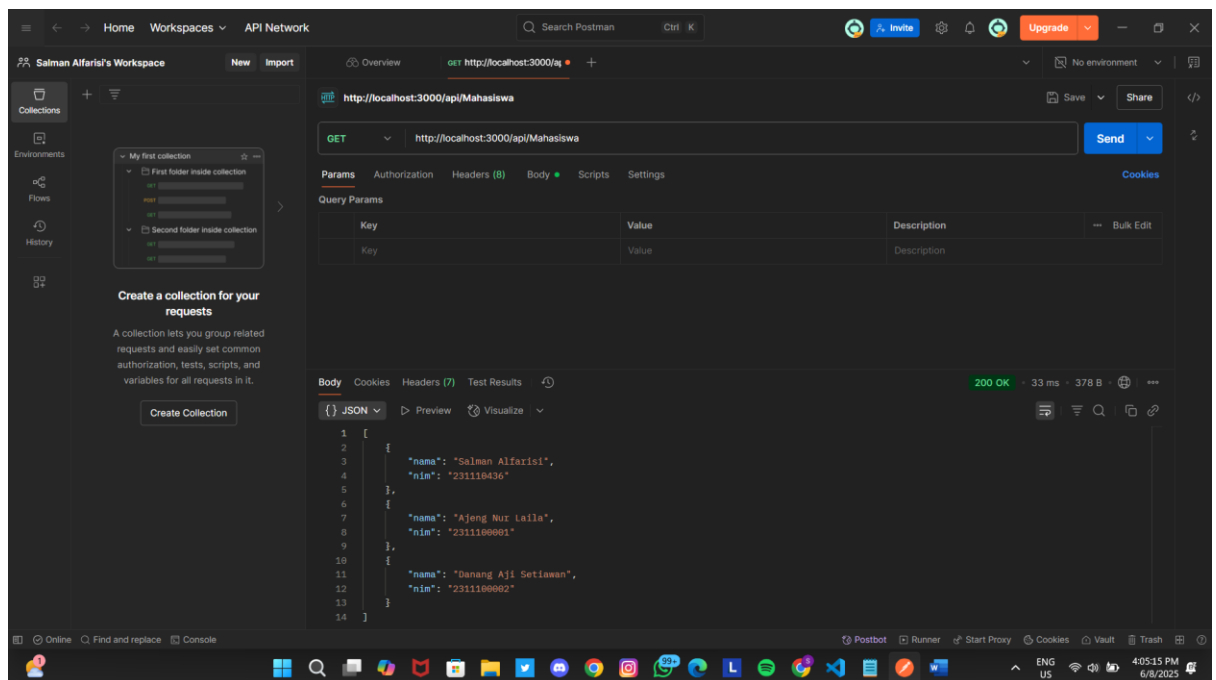
Kode di atas adalah implementasi backend menggunakan Node.js dengan framework Express untuk membuat API sederhana yang mengelola data mahasiswa. Modul express digunakan untuk membuat server dan routing, sementara body-parser digunakan untuk

memproses data JSON yang dikirim melalui request body. Dua modul lokal, yaitu Mahasiswa dan dataMahasiswa, diimpor untuk mengakses struktur data dan daftar mahasiswa awal yang sudah didefinisikan.

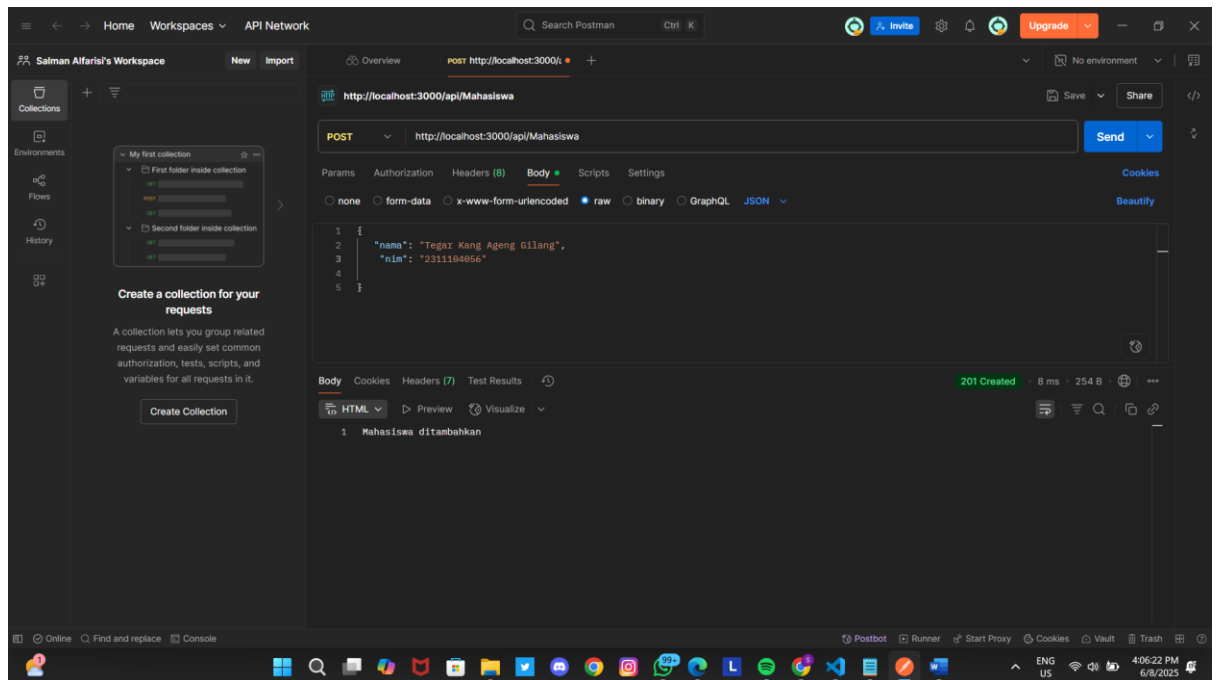
API ini menyediakan beberapa endpoint. Endpoint GET /api/mahasiswa mengembalikan seluruh daftar mahasiswa dalam bentuk JSON. Endpoint GET /api/mahasiswa/:index digunakan untuk mengambil data mahasiswa berdasarkan indeks array; jika indeks valid, maka data mahasiswa ditampilkan, jika tidak, akan dikembalikan pesan error 404. Endpoint POST /api/mahasiswa memungkinkan penambahan mahasiswa baru ke dalam array. Data mahasiswa dikirim melalui body request, lalu digunakan untuk membuat instance baru dari kelas Mahasiswa, dan ditambahkan ke array mahasiswa. Endpoint DELETE /api/mahasiswa/:index memungkinkan penghapusan data mahasiswa berdasarkan indeks. Jika indeks valid, data akan dihapus dan dikirimkan respon sukses, jika tidak, dikembalikan respon 404.

Server berjalan di port 3000, dan jika diakses melalui root (/), akan memberikan pesan informasi bahwa API mahasiswa aktif. API ini sangat berguna untuk latihan dasar pembuatan RESTful API, mencakup operasi CRUD (Create, Read, Update, Delete) sederhana dengan struktur data berbasis array.

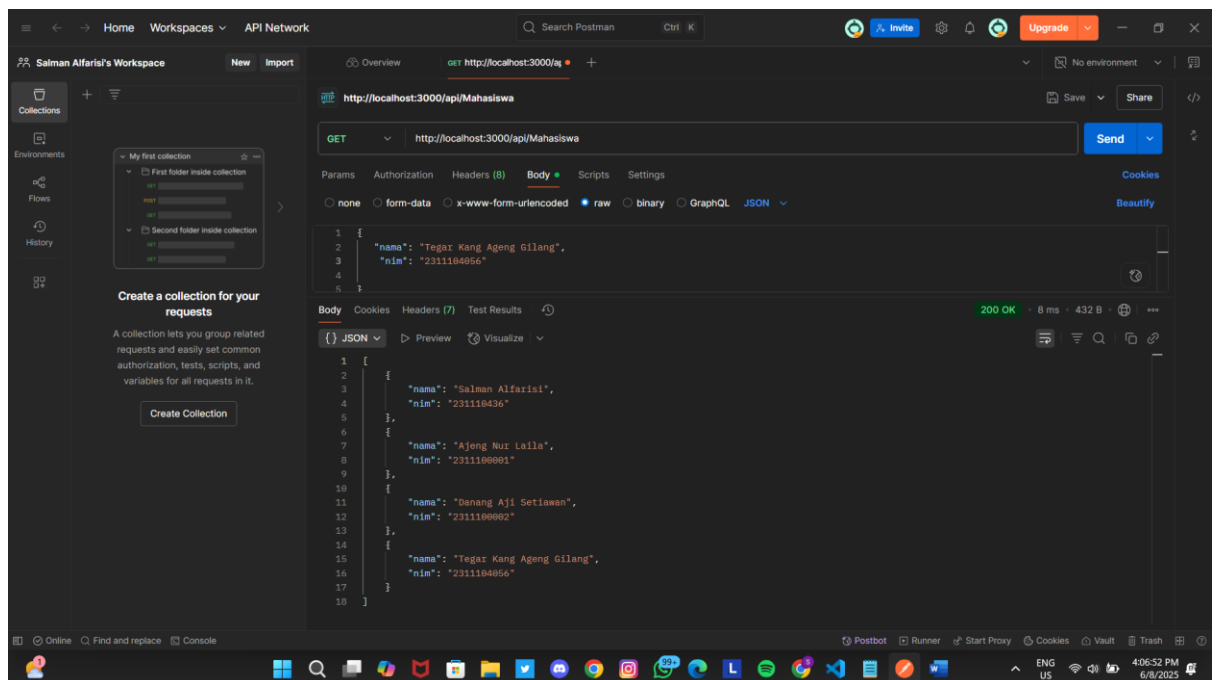
A. Cek



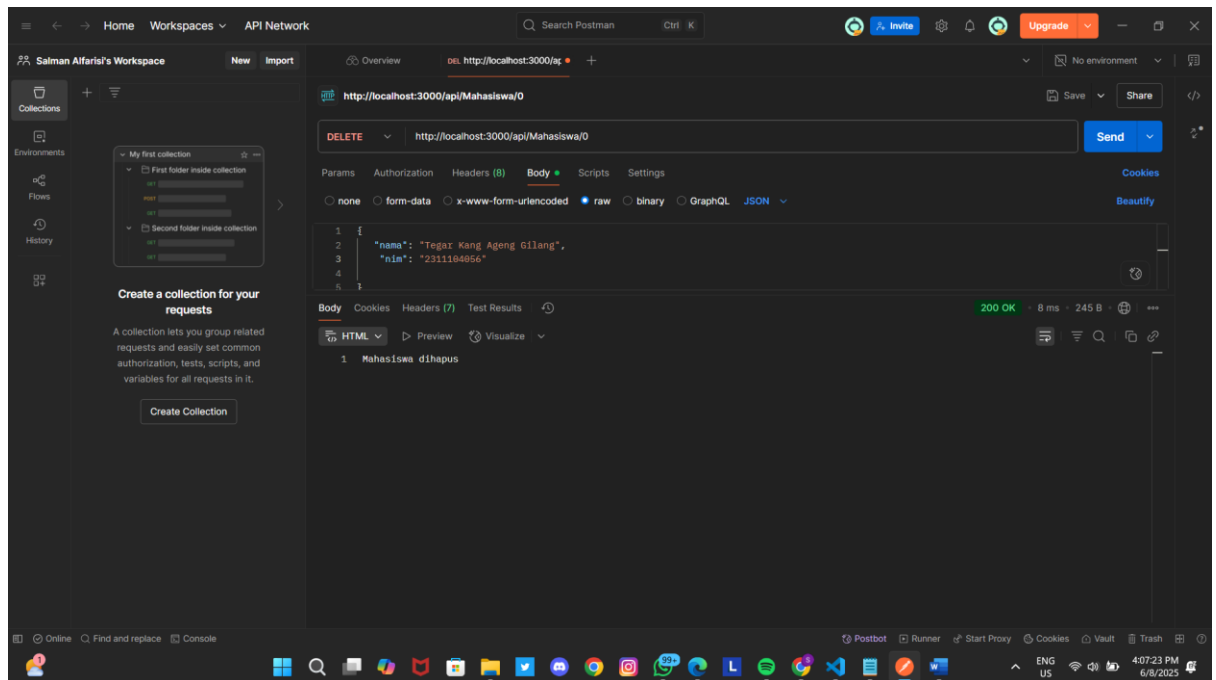
B. Post



C. Cek



D. Delete/api/mahasiswa/0



E. cek

