

Nama : M. Faris

Nim : 2311104017

## Jurnal Modul 9\

### 1. Source Kode :

Mahasiswa.js

```
const express = require('express');
const router = express.Router();
```

```
let mahasiswaList = [
  { Nama: 'Nama Kamu', Nim: '23111040' },
  { Nama: 'Teman A', Nim: '23111040' },
  { Nama: 'Teman B', Nim: '23111040' }
];
```

```
// GET semua mahasiswa
router.get('/', (req, res) => {
  res.json(mahasiswaList);
});
```

```
// GET mahasiswa by index
router.get('/:index', (req, res) => {
  const i = parseInt(req.params.index);
  if (i >= 0 && i < mahasiswaList.length) {
    res.json(mahasiswaList[i]);
  } else {
    res.status(404).send('Mahasiswa tidak ditemukan');
  }
});
```

```
// POST mahasiswa baru
router.post('/', (req, res) => {
  const { Nama, Nim } = req.body;
  if (!Nama || !Nim) {
    return res.status(400).send('Data tidak lengkap');
  }
  mahasiswaList.push({ Nama, Nim });
  res.status(201).send('Mahasiswa berhasil ditambahkan');
});
```

```
// DELETE mahasiswa by index
router.delete('/:index', (req, res) => {
  const i = parseInt(req.params.index);
  if (i >= 0 && i < mahasiswaList.length) {
    mahasiswaList.splice(i, 1);
    res.send('Mahasiswa berhasil dihapus');
  } else {
    res.status(404).send('Mahasiswa tidak ditemukan');
  }
});
```

```
module.exports = router;
```

index.js

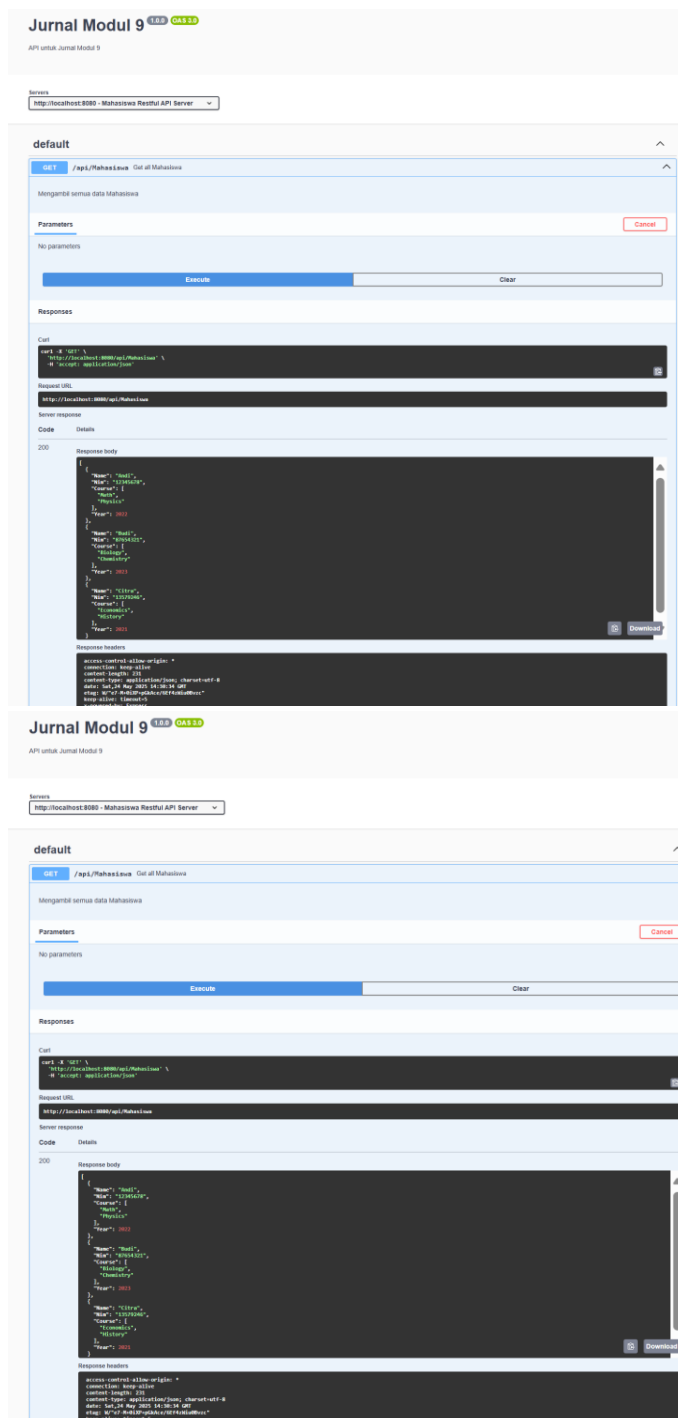
```
const express = require('express');
const cors = require('cors');
const mahasiswaRoutes = require('./mahasiswa');
const swaggerUi = require('swagger-ui-express');
const swaggerDocument = require('./swagger.json');
```

```
const app = express();
const PORT = 8080;
```

```
app.use(cors());
app.use(express.json());
app.use('/api/mahasiswa', mahasiswaRoutes);
app.use('/swagger', swaggerUi.serve, swaggerUi.setup(swaggerDocument));
```

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

## 2. Output :



## 3. Penjelasan :

Kode di atas merupakan implementasi Web API menggunakan Node.js dan Express yang menyediakan endpoint untuk mengelola data mahasiswa secara sederhana tanpa database. Data mahasiswa disimpan dalam array statis, dan API mendukung operasi GET untuk mengambil semua data atau data berdasarkan index, POST untuk menambahkan mahasiswa baru, serta DELETE untuk menghapus mahasiswa

berdasarkan index. Selain itu, dokumentasi API disediakan menggunakan Swagger di endpoint /swagger, yang memudahkan pengujian dan pemahaman struktur API.