

## 1. Tujuan

Dokumen ini bertujuan untuk memberikan dokumentasi lengkap mengenai RESTful API Mahasiswa yang dibangun menggunakan Express.js dan MongoDB Atlas. Dokumen ini disusun sebagai bagian dari tugas Ujian Tengah Semester (UTS) Mata Kuliah Pemrograman III (Web Services).

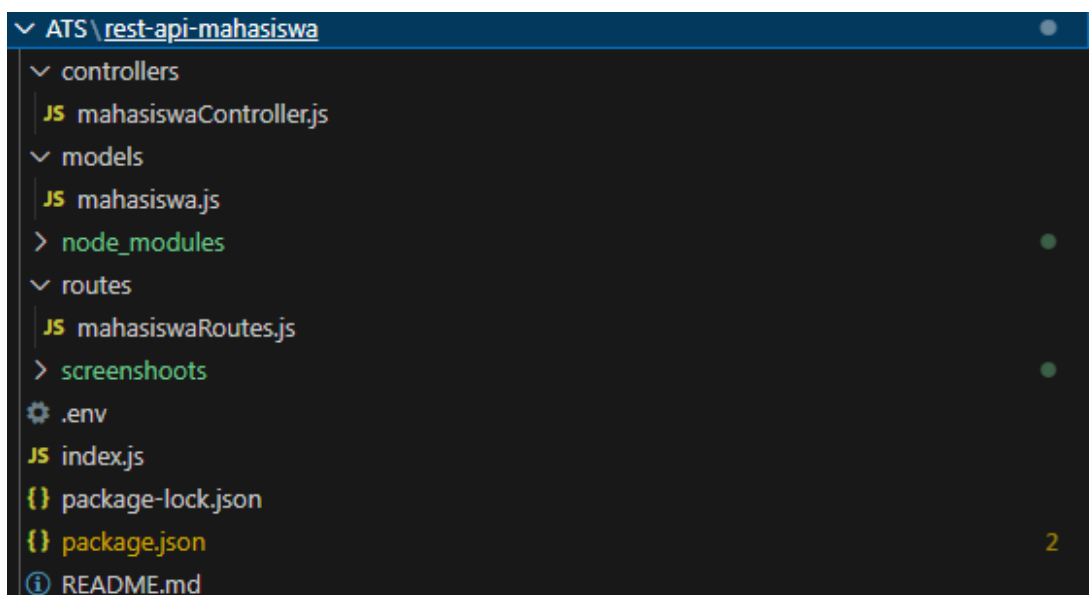
## 2. Teknologi yang Digunakan

API ini dikembangkan menggunakan teknologi modern yang sering digunakan dalam pengembangan web services, memastikan fleksibilitas, skalabilitas, dan kemudahan pengujian. Teknologi yang digunakan untuk membuat RESTful API sederhana untuk mengelola data mahasiswa diantaranya:

Komponen	Teknologi
Backend	Node.js + Express.js
Database	MongoDB Atlas
Testing	Postman
Tools Pendukung	Mongoose, dotenv, cors, morgan

## 3. Struktur Folder Proyek

Proyek ini memiliki struktur folder yang terorganisir dengan baik, sehingga memudahkan proses pengembangan dan pemeliharaan aplikasi.



## 4. File .env – Konfigurasi Lingkungan

Untuk menjaga keamanan konfigurasi aplikasi, kredensial database dan variabel lingkungan lainnya dikelola dalam file .env.

```
ATS > rest-api-mahasiswa > .env
1 PORT=3000
2 MONGO_URI=mongodb+srv://iday:10091996yadi@cluster-iday.wanofyr.mongodb.net/mahasiswa_db?retryWrites=true&w=majority&appName=Cluster-IdayS
```

## 5. File index.js – Server Utama

File index.js berperan sebagai entry point dari aplikasi dan bertanggung jawab untuk menjalankan server serta menghubungkan aplikasi ke database.

```
ATS > rest-api-mahasiswa > JS index.js > ...
1 const express = require('express');
2 const mongoose = require('mongoose');
3 const cors = require('cors');
4 const morgan = require('morgan');
5 require('dotenv').config();
6
7 const app = express();
8 const PORT = process.env.PORT || 3000;
9
10 // Middleware
11 app.use(cors());
12 app.use(morgan('dev'));
13 app.use(express.json());
14
15 // Routes
16 const mahasiswaRoutes = require('./routes/mahasiswaRoutes');
17 app.use('/api/mahasiswa', mahasiswaRoutes);
18
19 // Connect to MongoDB
20 mongoose.connect(process.env.MONGO_URI)
21   .then(() => console.log("Connected to MongoDB"))
22   .catch(err => console.error("MongoDB connection error:", err));
23
24 // Start server
25 app.listen(PORT, () => {
26   console.log(`Server running on port ${PORT}`);
27 });
```

## 6. Model Mahasiswa (models/Mahasiswa.js)

Model Mahasiswa mendefinisikan struktur data mahasiswa yang akan disimpan di dalam database MongoDB.

```
ATS > rest-api-mahasiswa > models > JS mahasiswa.js > ...
1 const mongoose = require('mongoose');
2
3 const MahasiswaSchema = new mongoose.Schema({
4   nama: { type: String, required: true },
5   npm: { type: String, required: true, unique: true },
6   prodi: { type: String, required: true },
7   angkatan: { type: Number, required: true }
8 });
9
10 module.exports = mongoose.model('Mahasiswa', MahasiswaSchema);
```

## 7. Controller (controllers/mahasiswaController.js)

Controller berisi logika bisnis dari masing-masing endpoint, termasuk operasi CRUD terhadap data mahasiswa.

```
ATS > rest-api-mahasiswa > controllers > JS mahasiswaController.js > ...
1 // GET semua mahasiswa
2
3
4 exports.getAllMahasiswa = async (req, res) => {
5   try {
6     const mahasiswa = await Mahasiswa.find();
7     res.status(200).json(mahasiswa);
8   } catch (err) {
9     res.status(500).json({ message: err.message });
10  }
11 };
12
13 // GET detail mahasiswa
14 exports.getMahasiswaById = async (req, res) => {
15   try {
16     const mahasiswa = await Mahasiswa.findById(req.params.id);
17     if (!mahasiswa) return res.status(404).json({ message: 'Mahasiswa tidak ditemukan' });
18     res.status(200).json(mahasiswa);
19   } catch (err) {
20     res.status(500).json({ message: err.message });
21   }
22 };
23
24 // POST tambah mahasiswa
25 exports.createMahasiswa = async (req, res) => {
26   const { nama, npm, prodi, angkatan } = req.body;
27
28   // Validasi input
29   if (!nama || !npm || !prodi || !angkatan) {
30     return res.status(400).json({ message: 'Semua field wajib diisi' });
31   }
32
33   const mahasiswaBaru = new Mahasiswa({ nama, npm, prodi, angkatan });
34
35   try {
36     const savedMahasiswa = await mahasiswaBaru.save();
37     res.status(201).json(savedMahasiswa);
38   } catch (err) {
39     res.status(500).json({ message: err.message });
40   }
41 };
42
```

## 8. Routes (routes/mahasiswaRoutes.js)

File routes mengatur jalur akses (endpoint) dan menghubungkannya dengan controller yang sesuai

```
ATS > rest-api-mahasiswa > routes > JS mahasiswaRoutes.js > ...
1 const express = require('express');
2 const router = express.Router();
3 const mahasiswaController = require('../controllers/mahasiswaController');
4
5 router.get('/', mahasiswaController.getAllMahasiswa);
6 router.get('/:id', mahasiswaController.getMahasiswaById);
7 router.post('/', mahasiswaController.createMahasiswa);
8 router.put('/:id', mahasiswaController.updateMahasiswa);
9 router.delete('/:id', mahasiswaController.deleteMahasiswa);
10
11 module.exports = router;
```

9. Daftar Endpoint API

Berikut adalah daftar endpoint yang tersedia pada RESTful API Mahasiswa beserta metode HTTP, URL, dan fungsinya.









Method	URL	Deksripsi
GET	/api/mahasiswa	Menampilkan semua mahasiswa
GET	/api/mahasiswa/:id	Detail satu mahasiswa
POST	/api/mahasiswa	Tambah data mahasiswa
PUT	/api/mahasiswa/:id	Update data mahasiswa
DELETE	/api/mahasiswa/:id	Hapus data mahasiswa

10. Screenshot Hasil Pengujian

Pengujian semua endpoint dilakukan menggunakan Postman untuk memastikan API berjalan sesuai harapan. Terlampir pada folder **screenshoot** adalah hasil uji dari masing-masing endpoint.

11. Lampiran

Sebagai pelengkap, beberapa lampiran penting seperti file **README.md**, koleksi Postman, dan screenshot hasil uji juga disertakan dalam pengumpulan tugas.

Name	Date modified	Type	Size
 controllers	17/05/2025 9:41	File folder	
 models	17/05/2025 9:42	File folder	
 node_modules	17/05/2025 9:46	File folder	
 routes	17/05/2025 9:42	File folder	
 .env	17/05/2025 9:48	ENV File	1 KB
 index.js	17/05/2025 9:59	JSFile	1 KB
 package	17/05/2025 9:47	JSON Source File	1 KB
 package-lock	17/05/2025 9:46	JSON Source File	51 KB