**LAPORAN TUGAS UTS BACKEND**



**Disusun Oleh**

**Achmad Aldino /3124101290**

**Program Studi D3 Manajamen Informatika  
STIKOM PGRI Banyuwangi  
Tahun 2025**

**Daftar isi**

[A. Soal: 1](#_Toc197077113)

[B. Struktur Folder: 2](#_Toc197077114)

[C. Penjelasan Isi Folder dan File 3](#_Toc197077115)

[D. Code 4](#_Toc197077116)

[E. Hasil 7](#_Toc197077117)

# Soal:

1. Buatlah file data.json isi dengan data dengan skema berikut:

{

"id":number,

"namaBarang":text,

"jumlah Barang":number,

"terjual":number,

"sisa":number

"updateAt": datetime

}

1. Buatlah sebuah api untuk crud database diatas menggunakan express.js
2. Lakukan ujicoba dengan postman atau insomnia
3. Upload project github.
4. Buatkan laporan.

# Struktur Folder:

Project-Barang-CRUD/

│

├── 📁 controller/

│ └── BarangC.js

│

├── 📁 data/

│ └── data.json

│

├── 📁 node\_modules/

│ └── (berisi file-file dependency npm, dibuat otomatis oleh npm install)

│

├── 📁 routes/

│ └── BarangR.js

│

├── 📄 index.js

├── 📄 package.json

├── 📄 package-lock.json

└── 📄 README.md (opsional untuk dokumentasi GitHub)

# Penjelasan Isi Folder dan File

1. Folder Controller berisi Code barangC yang berfungsi mengatur Respons dan Request dari semua aksi Membaca Data Membaca Data Berdasarkan Id Membuat data baru Memperbarui data dan Menghapus Data
2. Folder Routes berisi Code barangR yang berfungsi menjalankan File Controller dan aksi aksi yang kita buat
3. Folder Data berisi data.json yang berfungsi sebagai database dumy yang berisikan data barang
4. Folder node\_modules berisi package express js untuk menjalankan semua proses dari express js
5. Index.js adalah file utama untuk menjalankan Project yang kita sudah buat
6. Berisi metadata proyek dan daftar dependensi yang digunakan, serta script untuk menjalankan server
7. Berfungsi untuk mengunci versi dari setiap dependensi agar instalasi proyek tetap konsisten di setiap lingkungan.

# Code

1. Index.js

const express = require('express');

const cors = require('cors');

const bodyParser = require('body-parser');

const Barangroute = require('./routes/BarangR');

const app = express();

const port = 3000;

app.use(cors());

app.use(bodyParser.json());

app.use('/Barang', Barangroute);

app.listen(port, () => {

console.log(`Server is running on http://localhost:${port}`);

});

1. data.json

[

{

"id": 1,

"namaBarang": "Pensil dan Penghapus",

"jumlahBarang": 200,

"terjual": 100,

"sisa": 100,

"updateAt": "2025-03-01T00:00:00Z"

},

{

"id": 2,

"namaBarang": "Kabel Lan",

"jumlahBarang": 150,

"terjual": 100,

"sisa": 50,

"updateAt": "2025-03-10T00:00:00Z"

},

{

"id": 3,

"namaBarang": "TV 24 Inch",

"jumlahBarang": 200,

"terjual": 200,

"sisa": 0,

"updateAt": "2025-02-11T00:00:00Z"

},

{

"id": 4,

"namaBarang": "Laptop",

"jumlahBarang": 100,

"terjual": 20,

"sisa": 80,

"updateAt": "2025-04-12T00:00:00Z"

},

{

"id": 5,

"namaBarang": "Monitor",

"jumlahBarang": 50,

"terjual": 10,

"sisa": 40,

"updateAt": "2025-03-10T00:00:00Z"

}

]

1. BarangC.js

const fs = require('fs');

const path = require('path');

const dataPath = path.join(\_\_dirname, '../data/data.json');

const readBarangData = () => {

const barangData = fs.readFileSync(dataPath);

return JSON.parse(barangData);

};

const writeBarangData = (data) => {

fs.writeFileSync(dataPath, JSON.stringify(data, null, 2));

}

exports.getAllBarang = (req, res) => {

const barangData = readBarangData();

res.status(200).json(barangData);

};

exports.getBarangById = (req, res) => {

const barangData = readBarangData();

const id = parseInt(req.params.id, 10);

const barang = barangData.find(item => item.id === id);

if (!barang) {

return res.status(404).json({ message: 'Barang not found' });

}

res.status(200).json(barang);

};

exports.createBarang = (req, res) => {

const barangData = readBarangData();

const newBarang = req.body;

barangData.push(newBarang);

writeBarangData(barangData);

res.status(201).json({ message: 'Barang created successfully', barang: newBarang });

};

exports.updateBarang = (req, res) => {

const barangData = readBarangData();

const id = parseInt(req.params.id, 10);

const index = barangData.findIndex(item => item.id === id);

if (index === -1) {

return res.status(404).json({ message: 'Barang not found' });

}

const updatedBarang = { ...barangData[index], ...req.body };

barangData[index] = updatedBarang;

writeBarangData(barangData);

res.status(200).json({ message: 'Barang updated successfully', barang: updatedBarang });

};

exports.deleteBarang = (req, res) => {

const barangData = readBarangData();

const id = parseInt(req.params.id, 10);

const index = barangData.findIndex(item => item.id === id);

if (index === -1) {

return res.status(404).json({ message: 'Barang not found' });

}

barangData.splice(index, 1);

writeBarangData(barangData);

res.status(200).json({ message: 'Barang deleted successfully' });

};

1. BarangR.js

const express = require('express');

const router = express.Router();

const barangController = require('../controller/BarangC');

router.get('/', barangController.getAllBarang);

router.get('/:id', barangController.getBarangById);

router.post('/', barangController.createBarang);

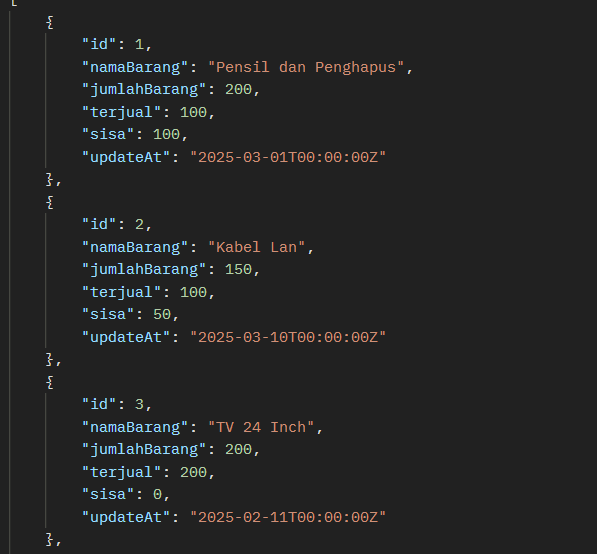
router.put('/:id', barangController.updateBarang);

router.delete('/:id', barangController.deleteBarang);

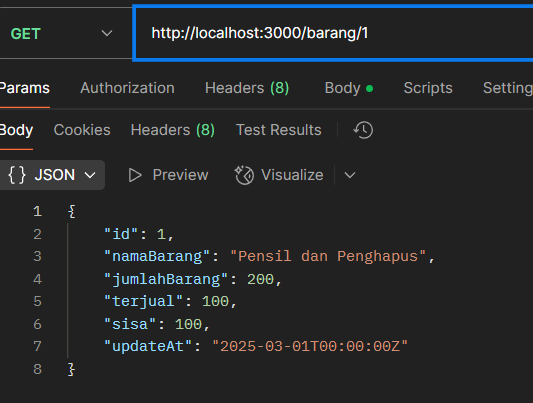
module.exports = router;

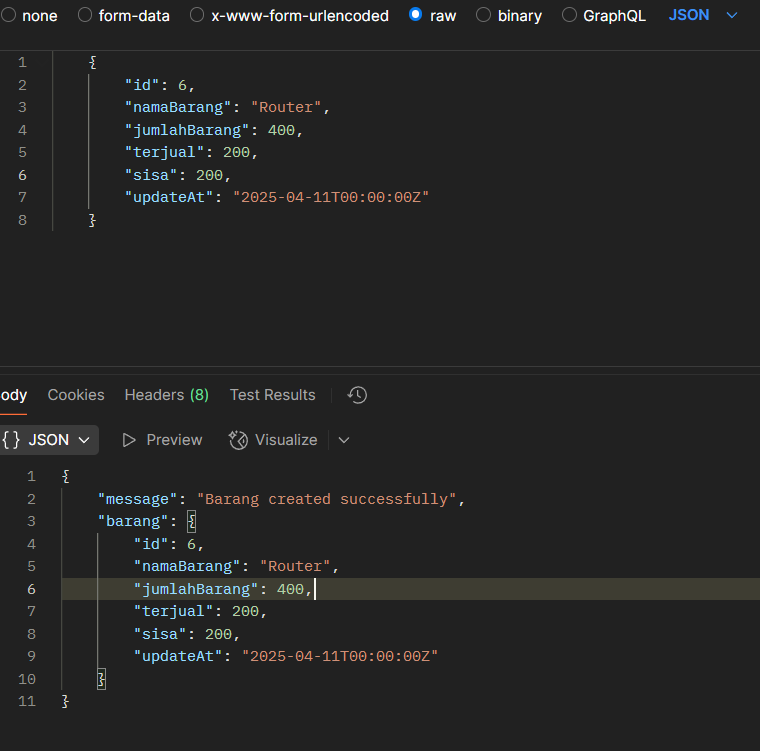
# Hasil

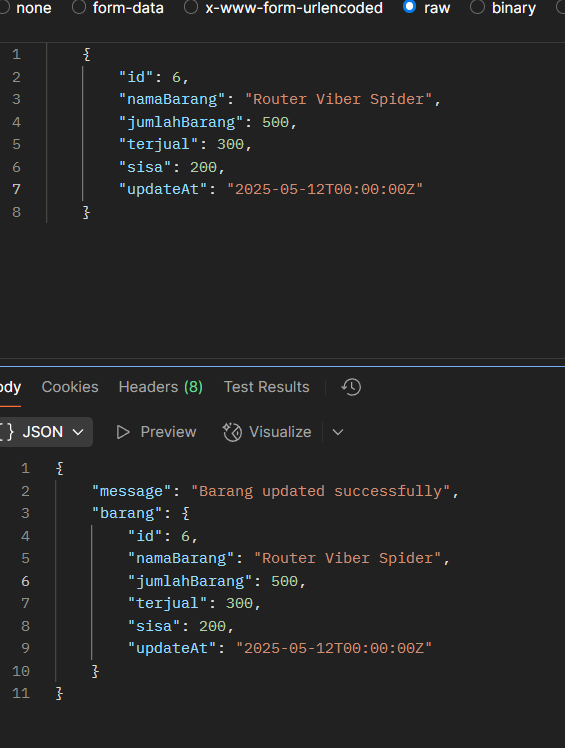
1. Read data



1. Read data Berdasarkan Id



1. Create Data
2. Update data



1. Delete data

