LAPORAN TUGAS UTS BACKEND

Disusun Oleh

Wildan Ramadhani /3124101285

Program Studi D3 Manajamen Informatika
STIKOM PGRI Banyuwangi
Tahun 2025

Daftar isi

A. Soal: 1
B. Struktur Folder: 2
C. Penjelasan Isi Folder dan File 3
D. Code 4
E. Hasil 7

A. Soal:

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

   {

   "id":number,

   "namaBarang":text,

   "jumlah Barang":number,

   "terjual":number,

   "sisa":number

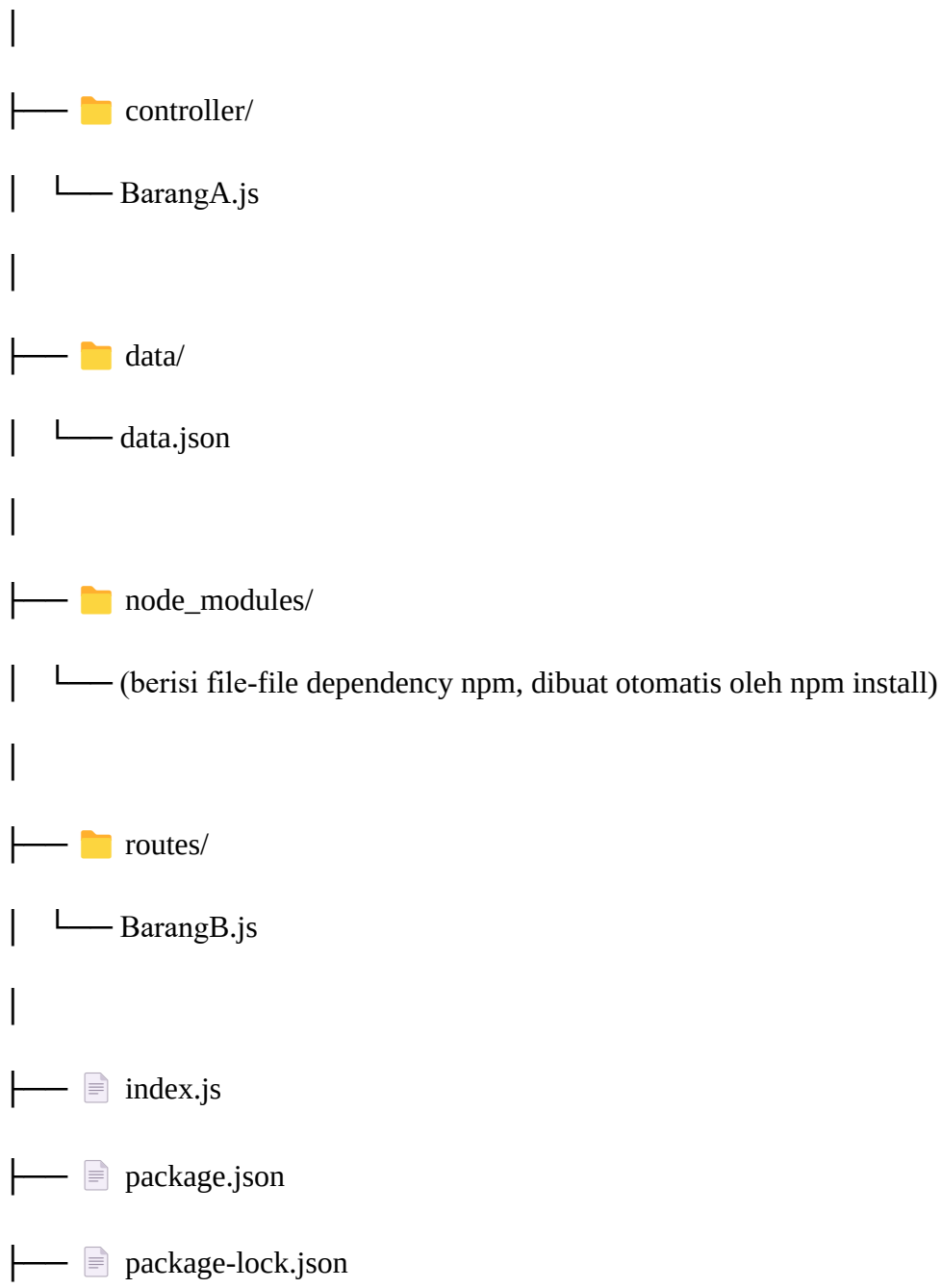
   "updateAt": datetime

   }

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

B. Struktur Folder:

Project-Barang-CRUD/



C. Penjelasan Isi Folder dan File

1. Folder Controller berisi Code barangA yang berfungsi mengatur Respons dan Request dari semua aksi (Membaca Data Membaca Data Membuat data baru Memperbarui data dan Menghapus Data)
2. Folder Routes berisi Code barangB 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.

D. Code

1. Index.js

```
const express = require('express');
const app = express();
const barangRoute = require('./routes/BarangB');

app.use((err, req, res, next) => {
    console.error(err.stack);
    res.status(500).send('Terjadi kesalahan pada server');
})

app.use(express.json());
app.use('/barang', barangRoute);

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

2. data.json

```
[
  {
    "id": 1,
    "namaBarang": "Buku Tulis",
    "jumlahBarang": 100,
    "terjual": 30,
    "sisa": 70,
    "updateAt": "2024-05-05T10:00:00Z"
  },
  {
    "id": 2,
    "namaBarang": "Pensil",
    "jumlahBarang": 200,
    "terjual": 50,
    "sisa": 150,
    "updateAt": "2024-05-05T10:00:00Z"
  },
  {
    "id": 3,
    "namaBarang": "Penghapus",
    "jumlahBarang": 200,
    "terjual": 30,
    "sisa": 170,
    "updateAt": "2024-05-05T10:00:00Z"
```

```
  },
  {
    "id": 5,
    "namaBarang": "Pulpen",
    "jumlahBarang": 100,
    "terjual": 30,
    "sisa": 70,
    "updateAt": "2025-05-05T05:43:12.358Z"
  }
]
```

3. BarangA.js

```
const fs = require('fs');
const path = require('path');
const dataPath = path.join(__dirname, '../data/data.json');

const getAll = (req, res) => {
  const data = JSON.parse(fs.readFileSync(dataPath));
  res.json(data);
};

const create = (req, res) => {
  console.log("POST request body:", req.body);

  if (!req.body.namaBarang || !req.body.jumlahBarang
||!req.body.terjual || req.body.sisa === undefined) {
    return res.status(400).json({ message: "Data tidak lengkap" });
  }

  const data = JSON.parse(fs.readFileSync(dataPath));
  const newData = {
    id: Date.now(),
    ...req.body,
    updateAt: new Date().toISOString(),
  };
  data.push(newData);
  fs.writeFileSync(dataPath, JSON.stringify(data, null, 2));
  res.status(201).json(newData);
};

const update = (req, res) => {
  const { id } = req.params;
  const data = JSON.parse(fs.readFileSync(dataPath));
  const index = data.findIndex(item => item.id == id);
```

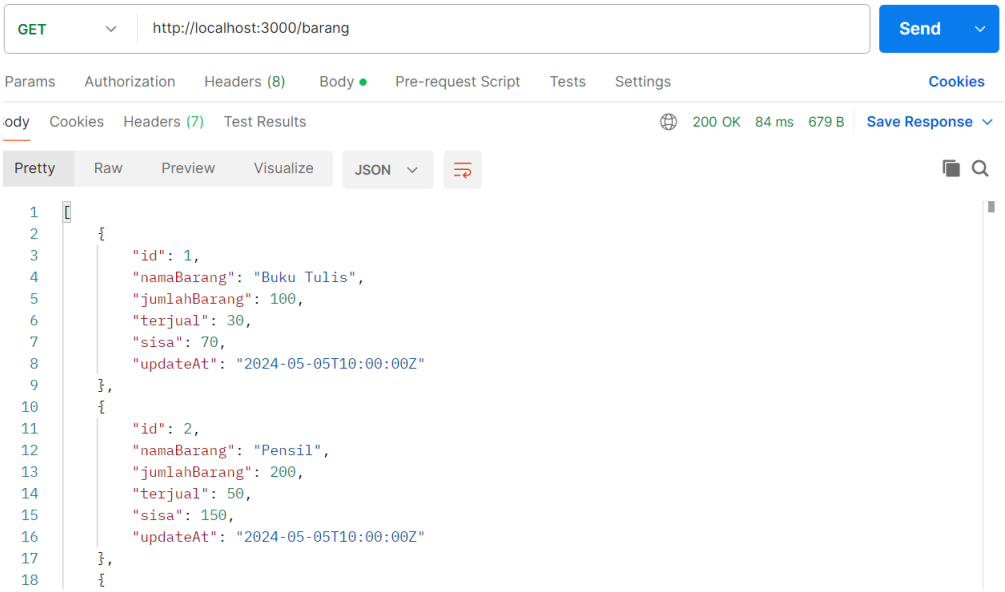
```
  if (index === -1) {
    return res.status(404).json({ message: "Data tidak ditemukan"});
  }

  data[index] = {
    ...data[index],
    ...req.body,
    updateAt: new Date().toISOString()
  };

  fs.writeFileSync(dataPath, JSON.stringify(data, null, 2));
  res.json(data[index]);
};

const remove = (req, res) => {
  const { id } = req.params;
  let data = JSON.parse(fs.readFileSync(dataPath));
  data = data.filter(item => item.id != id);
  fs.writeFileSync(dataPath, JSON.stringify(data, null, 2));
  res.json({ message: "Data berhasil dihapus" });
};

module.exports = { getAll, create, update, remove };
```

4. BarangB.js

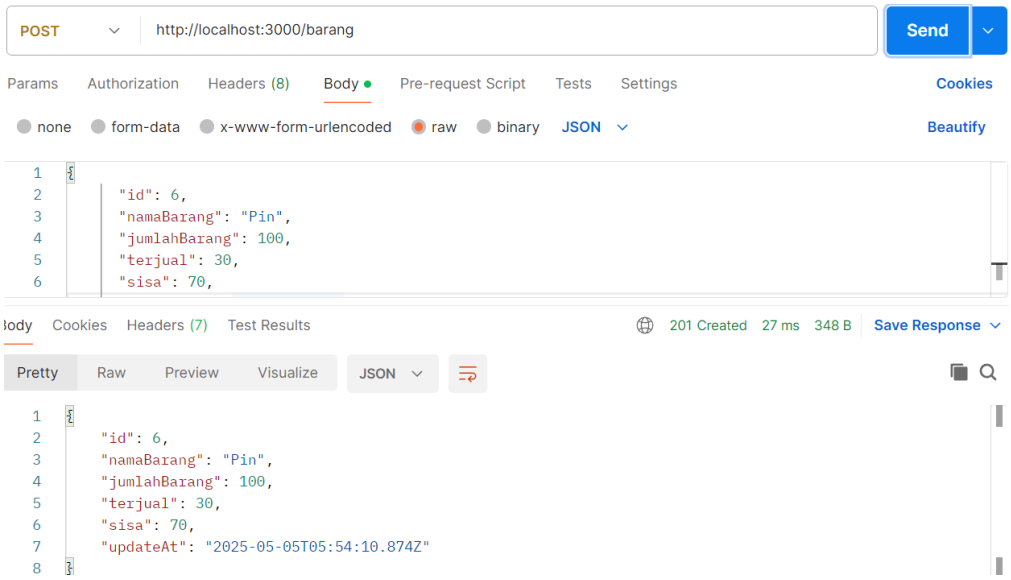
```
const express = require('express');
const router = express.Router();
const controller = require('../controller/BarangA');

router.get('/', controller.getAll);
router.post('/', controller.create);
router.put('/:id', controller.update);
router.delete('/:id', controller.remove);

module.exports = router;
```

E. Hasil

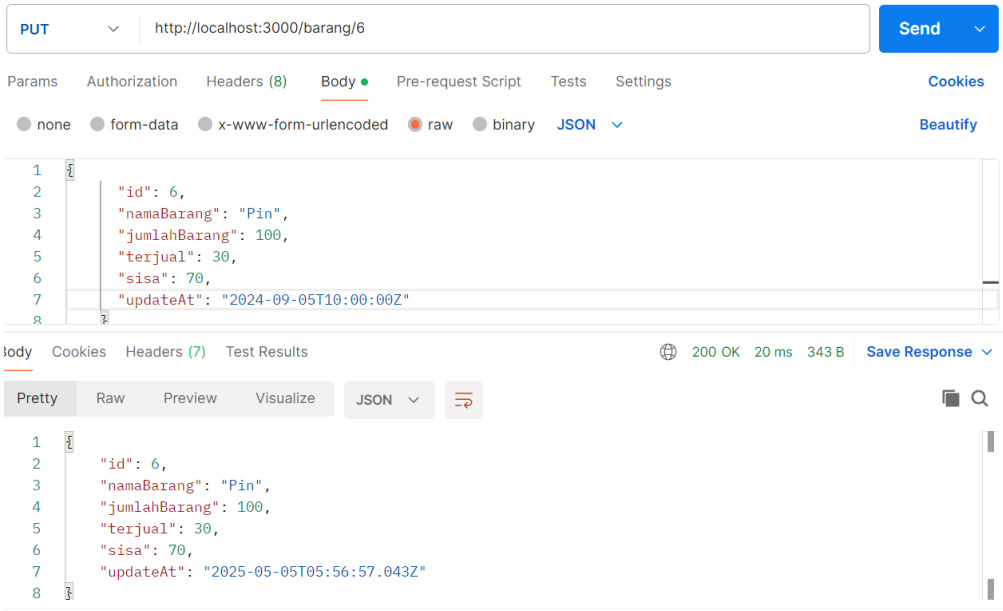
1. Read data



2. Create Data



3. Update data



4. Delete data