**LAPORAN TUGAS UTS BACKEND**



**Disusun Oleh**

**Muhammad Samsudin /3124101298**

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

**Daftar isi**

[A. Soal: 1](#_Toc197370178)

[B. Membuat Database dan Tabel di Mysql 2](#_Toc197370179)

[C. Code 3](#_Toc197370180)

[D. Strukture 9](#_Toc197370181)

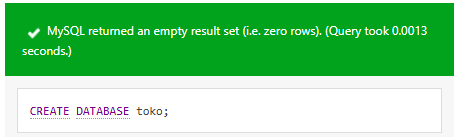
[E. Hasil 10](#_Toc197370182)

# Soal:

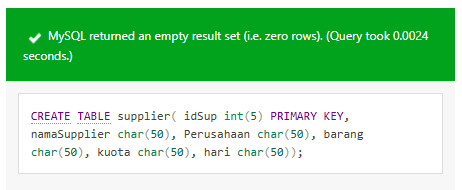
1. Buatlah sebuah database MySQL dengan ketentuan  
   a. Nama database : toko  
   b. Nama Tabel : supplier  
   c. Field : idSup, namaSupplier, Perusahaan, barang, kuota, hari
2. Buatlah sebuah api untuk crud database diatas menggunakan express.js
3. Lakukan ujicoba dengan postman atau insomnia
4. Upload project dan database ke github.
5. Buatkan laporan.
6. Kirimkan link laporan ke elearning

# Membuat Database dan Tabel di Mysql

1. Membuat Database



1. Membuat Tabel



# Code

1. **index.js**

const express = require("express");

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

const cors = require('cors');

const supplierroutes = require('./routes/tokoroutes');

const app = express();

const port = 3000;

app.use(cors());

app.use(bodyParser.json());

app.use('/api/supplier', supplierroutes);

app.listen(port, () => {

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

});

1. **db.js**

const {Sequelize} = require('sequelize');

const sequelize = new Sequelize('toko', 'root', '', {

host: 'localhost',

dialect: 'mysql',

port: 3306,

});

module.exports = sequelize;

1. **tokocontroller.js**

const Supplier = require('../models/supplier');

// GET semua atau satu Supplier

const get = async (req, res) => {

try {

if (req.params.id) {

const supplier = await Supplier.findByPk(req.params.id);

if (supplier) {

res.json(supplier);

} else {

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

}

} else {

const suppliers = await Supplier.findAll();

res.json(suppliers);

}

} catch (error) {

res.status(500).json({ error: error.message });

}

};

// POST - buat Supplier baru

const post = async (req, res) => {

const { idSupplier, namaSupplier, Perusahaan, barang, kuota, hari } = req.body;

try {

const supplier = await Supplier.create({ idSupplier, namaSupplier, Perusahaan, barang, kuota, hari });

res.status(201).json(supplier);

} catch (error) {

res.status(400).json({ error: error.message });

}

};

// PUT - update Supplier berdasarkan idSupplier

const put = async (req, res) => {

const idSup = req.params.id;

const { namaSupplier, Perusahaan, barang, kuota, hari } = req.body;

try {

const [updated] = await Supplier.update(

{ namaSupplier, Perusahaan, barang, kuota, hari },

{ where: { idSup } }

);

if (updated) {

res.json({ message: 'Supplier updated successfully' });

} else {

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

}

} catch (error) {

res.status(400).json({ error: error.message });

}

};

// DELETE - hapus Supplier berdasarkan idSupplier

const del = async (req, res) => {

const idSupplier = req.params.id;

try {

const supplier = await Supplier.findByPk(idSupplier);

if (supplier) {

await supplier.destroy();

res.json({ message: 'Supplier deleted successfully' });

} else {

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

}

} catch (error) {

res.status(500).json({ error: error.message });

}

};

module.exports = { get, post, put, delete: del };

1. **supplier.js**

const { DataTypes } = require('sequelize');

const sequelize = require('../config/db.js');

const Supplier = sequelize.define('supplier', {

idSup: {

type: DataTypes.INTEGER,

primaryKey: true,

autoIncrement: true,

},

namaSupplier: {

type: DataTypes.STRING,

allowNull: false,

},

Perusahaan: {

type: DataTypes.STRING,

allowNull: false,

},

barang: {

type: DataTypes.STRING,

allowNull: false,

},

kuota: {

type: DataTypes.STRING,

allowNull: false,

},

hari: {

type: DataTypes.STRING,

allowNull: false,

},

}, {

tableName: 'supplier',

timestamps: false,

});

module.exports = Supplier;

1. **tokoroutes.js**

const express = require('express');

const router = express.Router();

const supplierroutes = require('../controller/tokocontroller');

router.get('/', supplierroutes.get);

router.get('/:id', supplierroutes.get);

router.post('/', supplierroutes.post);

router.put('/:id', supplierroutes.put);

router.delete('/:id', supplierroutes.delete);

module.exports = router;

# Strukture

uts-toko/

│

├── config/

│ └── db.js

│

├── models/

│ └── supplier.js

│

├── routes/

│ └── supplierRoutes.js

│

├── migrations/

│

│

├── seeders/

│

├── node\_modules/

│

├── index.js

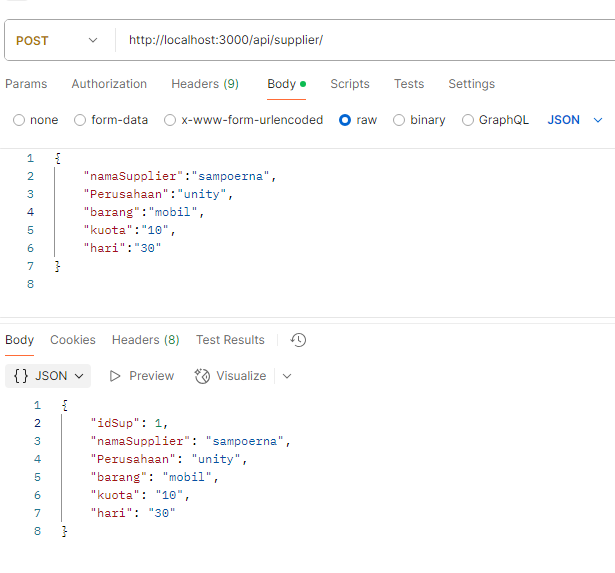
│

├── package.json

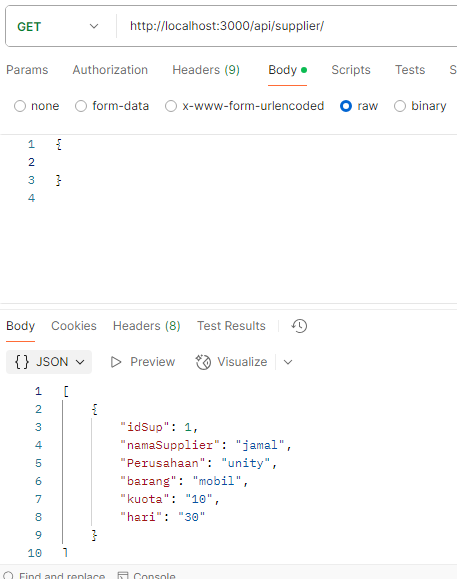
└── package-lock.json

# Hasil

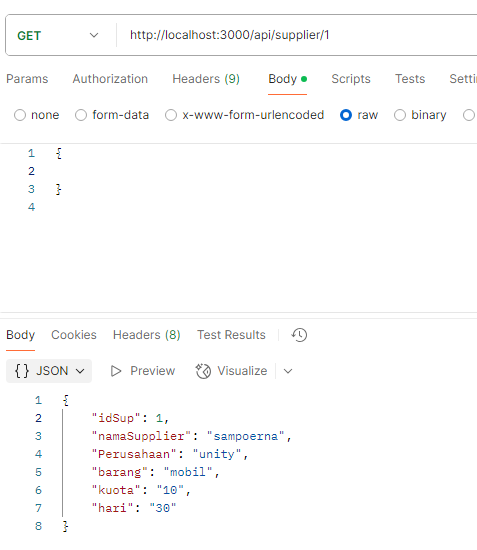
1. Create



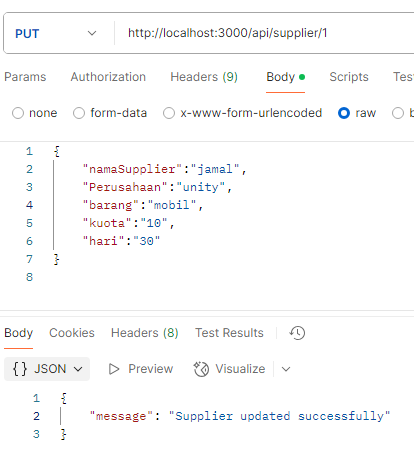
1. Read

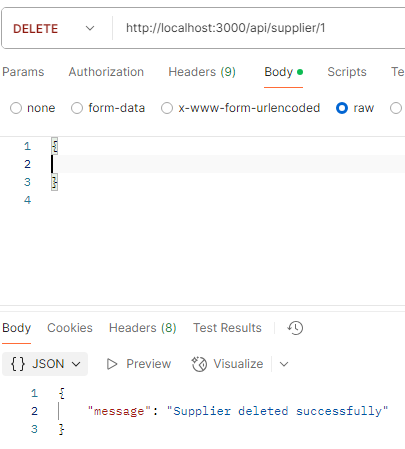


1. Read By Id



1. Update



1. Delete