MODUL PERTEMUAN 5

PRAKTIKUM FULLSTACK DEVELOPMENT

“Pembuatan Backend Express dan Database MySQL”

SEKOLAH TINGGI ILMU KOMPUTER PGRI BANYUWANGI

STIKOM BANYUWANGI

2024

A. Pendahuluan

Untuk modul kali ini kita kan membahas backend dengan menggunakan express.js dan database mysql sebagai tempat penyimpananya. Modul ini bertujuan untuk memberikan pemahaman dasar tentang cara membuat backend menggunakan Express dan menghubungkannya dengan database MySQL. Peserta diharapkan dapat melakukan operasi CRUD (Create, Read, Update, Delete) pada tabel `mahasiswa` dalam database `siakad` yang digunakan dalam contoh dibawah dan mengembangkan lebih lanjut.

B. Persiapan/Prasyarat

1. **Node.js dan npm** sudah terinstal di komputer Anda.
2. **MySQL Server** sudah terinstal dan berjalan di komputer Anda boleh menggunakan paket XAMPP.
3. **Text editor** (misalnya VSCode) atau IDE yang mendukung JavaScript.
4. Komputer dengan sistem operasi Windows, MacOS, atau Linux.
5. Koneksi internet untuk mengunduh dependensi.

C. Langkah-langkah Praktikum

1. Bagian 1: Persiapan Lingkungan

a) Instalasi Node.js dan npm

Pastikan Node.js dan npm telah terinstal. Anda dapat memeriksanya dengan perintah berikut:

Perintah pada CMD

```
node -v
npm -v
```

b) Instalasi MySQL

Pastikan MySQL telah terinstal dan berjalan. Anda dapat memeriksanya dengan mencoba masuk ke MySQL melalui terminal:

Perintah pada CMD

```
Mysql -u root -p
```

Nb:
Untuk pengguna xampp bisa menggunakan perintah diatas pada folder c://xampp/mysql/bin sesuaikan dengan folder Dimana anda menginstall xampp

2. Bagian 2: Pembuatan Database dan Tabel

a) Json yang digunakan sebagai acuan untuk pembuatan table adalah sebagai berikut :

json

```
{
  "mahasiswa": [
    {
      "nim": "1122101010",
      "nama": "Ahmad Budi",
      "jurusan": "Teknik Informatika",
      "angkatan": 2020,
```

```
            "email": "ahmad.budi@example.com"
          },
          {
            "nim": "1122101011",
            "nama": "Siti Aminah",
            "jurusan": "Teknik Informatika",
            "angkatan": 2020,
            "email": "siti.aminah@example.com"
          },
          {
            "nim": "1122101012",
            "nama": "Rudi Hartono",
            "jurusan": "Teknik Informatika",
            "angkatan": 2020,
            "email": "rudi.hartono@example.com"
          },
          {
            "nim": "1122101013",
            "nama": "Dewi Sartika",
            "jurusan": "Teknik Informatika",
            "angkatan": 2020,
            "email": "dewi.sartika@example.com"
          }
        ]
}
```

b) Maka jika dibuat dalam transact SQL json diata maka akan seperti berikut :

Transact SQl

```
CREATE TABLE mahasiswa (
      nim VARCHAR(20) PRIMARY KEY,
      nama VARCHAR(100),
      jurusan VARCHAR(50),
      angkatan INT,
      email VARCHAR(100)
);
```

c) Membuat database `siakad` dan tabel `mahasiswa`

Buka MySQL dan jalankan perintah berikut untuk membuat database siakad dan membua table mahasiswa:

Perintah pada CMD

```
CREATE DATABASE siakad;
USE siakad;
CREATE TABLE mahasiswa (
      nim VARCHAR(20) PRIMARY KEY,
      nama VARCHAR(100),
      jurusan VARCHAR(50),
      angkatan INT,
      email VARCHAR(100)
   );
```

Nb:
Untuk pengguna xampp bisa menggunakan phpMyAdmin pada Alamat https://localhost/phpmyadmin silahkan buat database siakad dan table sesuai ketentuan diatas

3. **Bagian 3: Pembuatan Proyek Express**
   a) Membuat direktori proyek
   b) Inisialisasi proyek Node.js dengan perintah "**npm init -y**"
   c) Instalasi dependensi "***npm install express mysql2 body-parser***"

4. **Bagian 4: Implementasi Backend dengan Express**
   a) Buat file `index.js` tambahkan kode berikut:

index.js

```
//import dependencies
   const express = require('express');
 const mysql = require('mysql2');
 const bodyParser = require('body-parser');

//deklarasikan express
const app = express();
const port = 3000;
app.use(bodyParser.json());

// Konfigurasi koneksi ke database
const db = mysql.createConnection({
     host: 'localhost',
     user: 'root',
     password: '',
     database: 'siakad'
});

// Koneksi ke database
db.connect((err) => {
     if (err) {
       console.error('Error connecting to the database:', err);
       return;
     }
     console.log('Connected to the database.');
});
// CREATE: Tambah data mahasiswa
app.post('/mahasiswa', (req, res) => {
     const { nim, nama, jurusan, angkatan, email } = req.body;
     const sql = 'INSERT INTO mahasiswa (nim, nama, jurusan,
angkatan, email) VALUES (?, ?, ?, ?, ?)';
     db.query(sql, [nim, nama, jurusan, angkatan, email], (err,
result) => {
       if (err) {
         return res.status(500).json({ error: err.message });
       }
       res.status(201).json({ message: 'Mahasiswa added
successfully', data: result });
```

```
    });
});

// READ: Ambil semua data mahasiswa
app.get('/mahasiswa', (req, res) => {
    const sql = 'SELECT  FROM mahasiswa';
    db.query(sql, (err, results) => {
      if (err) {
        return res.status(500).json({ error: err.message });
      }
      res.status(200).json({ data: results });
    });
});

// READ: Ambil data mahasiswa berdasarkan NIM
app.get('/mahasiswa/:nim', (req, res) => {
    const { nim } = req.params;
    const sql = 'SELECT  FROM mahasiswa WHERE nim = ?';
    db.query(sql, [nim], (err, result) => {
      if (err) {
        return res.status(500).json({ error: err.message });
      }
      if (result.length === 0) {
        return res.status(404).json({ message: 'Mahasiswa not
found' });
      }
      res.status(200).json({ data: result[0] });
    });
});

// UPDATE: Ubah data mahasiswa
app.put('/mahasiswa/:nim', (req, res) => {
    const { nim } = req.params;
    const { nama, jurusan, angkatan, email } = req.body;
    const sql = 'UPDATE mahasiswa SET nama = ?, jurusan = ?,
angkatan = ?, email = ? WHERE nim = ?';
    db.query(sql, [nama, jurusan, angkatan, email, nim], (err,
result) => {
      if (err) {
        return res.status(500).json({ error: err.message });
      }
      if (result.affectedRows === 0) {
        return res.status(404).json({ message: 'Mahasiswa not
found' });
      }
      res.status(200).json({ message: 'Mahasiswa updated
successfully' });
    });
});

// DELETE: Hapus data mahasiswa
app.delete('/mahasiswa/:nim', (req, res) => {
    const { nim } = req.params;
    const sql = 'DELETE FROM mahasiswa WHERE nim = ?';
    db.query(sql, [nim], (err, result) => {
      if (err) {
        return res.status(500).json({ error: err.message });
```

```
        }
        if (result.affectedRows === 0) {
          return res.status(404).json({ message: 'Mahasiswa not
found' });
        }
        res.status(200).json({ message: 'Mahasiswa deleted
successfully' });
      });
});

// Jalankan server
app.listen(port, () => {
      console.log(`Server running at http://localhost:${port}/`);
});
```

b) Jalankan server

Perintah pada CMD

```
node index.js
```

Nb:
Untuk pengguna yang sudah menginstall nodemon dapat menjalankan dengan perintah : "***nodemon index.js***"

5. Bagian 5: Pengujian API

a) Menggunakan Postman

Skenario

- ✓ **Tambah data mahasiswa (POST)**
  - Endpoint: `POST http://localhost:3000/mahasiswa`
  - Body (JSON):
  - Inputkan json berikut :

json

```
{
  "nim": "12345678",
  "nama": "Ahmad Budi",
  "jurusan": "Teknik Informatika",
  "angkatan": 2020,
  "email": "ahmad.budi@example.com"
}
```

- ✓ Ambil semua data mahasiswa (GET)
  - Endpoint: `GET http://localhost:3000/mahasiswa`
- ✓ Ambil data mahasiswa berdasarkan NIM (GET)
  - Endpoint: `GET http://localhost:3000/mahasiswa/12345678`
- ✓ Ubah data mahasiswa (PUT)
  - Endpoint: `PUT http://localhost:3000/mahasiswa/12345678`
  - Body (JSON):

json

```
{
```

```
    "nama": "Ahmad Budi Santoso",
    "jurusan": "Teknik Informatika",
    "angkatan": 2020,
    "email": "ahmad.budi.santoso@example.com"
  }
```

✓ Hapus data mahasiswa (DELETE)

- Endpoint: `DELETE http://localhost:3000/mahasiswa/12345678`

6. Praktikum

a. Buatlah table baru dengan nama matakuliah dengan struktur sebagai berikut :

No	Nama Field	Tipe data	Keterangan
1.	kodeMK	Char(6)	Primary key, not null
2.	namaMK	Varchar(100)	Not null
3.	sks	Int	Not null
4.	semester	Char(2)	Not null

b. Buatkan kode untuk melakukan proses CRUD pada data diatas dibackend yang sudah dibuat.

c. Buatkan route untuk mengakses halaman tersebut.

d. Lakukan percobaan dengan postman berikan scenario sperti dalam modul diatas.

e. Buatlah laporan praktikumnya