

Laporan Praktikum
Mata Kuliah Pemrograman WEB



Pertemuan 5
“CRUD Node JS + MySQL”

Dosen Pengampu :
Willdan Aprizal Arifin, S.Pd., M.Kom..

Disusun Oleh :
Hilwa Nur Falah
(2312509)

PROGRAM STUDI SISTEM INFORMASI KELAUTAN
UNIVERSITAS PENDIDIKAN INDONESIA
2024

I. Pendahuluan

Dalam pengembangan aplikasi berbasis web, integrasi antara server dan database merupakan hal yang sangat penting untuk memastikan data yang dikelola dapat diolah dengan baik. Salah satu cara populer untuk menghubungkan aplikasi dengan database adalah melalui teknik CRUD (Create, Read, Update, Delete). CRUD adalah operasi dasar yang dilakukan dalam pengelolaan database, dan ini bisa diimplementasikan menggunakan berbagai teknologi. Pada praktikum kali ini, kita akan memanfaatkan Node.js sebagai server-side platform dan MySQL sebagai database untuk membangun aplikasi CRUD. Dengan bantuan Express.js sebagai framework web dan EJS sebagai template engine, aplikasi akan mampu menampilkan, menambah, memperbarui, dan menghapus data pengguna secara dinamis.

II. Tujuan

1. Memahami cara menghubungkan aplikasi Node.js dengan MySQL.
2. Mengimplementasikan fungsi dasar CRUD (Create, Read, Update, Delete) pada aplikasi web.
3. Membangun aplikasi web sederhana menggunakan Express.js dan EJS yang terhubung dengan database MySQL.
4. Mengetahui cara mengelola data pengguna dengan cara menambahkan, membaca, memperbarui, dan menghapus data dari database MySQL.

III. Alat dan Bahan

Perangkat Lunak:

- Node.js: Sebagai platform untuk menjalankan server aplikasi.
- MySQL: Sebagai sistem database yang digunakan untuk menyimpan data pengguna.
- Express.js: Framework web untuk mempermudah pembuatan rute server.
- EJS (Embedded JavaScript): Template engine untuk menampilkan halaman web secara dinamis.
- Body-parser: Middleware untuk parsing request data dari form HTML.
- MySQL2: Library untuk menghubungkan Node.js dengan database MySQL.

2. Langkah Kerja

1. Instalasi dan Persiapan Lingkungan Kerja

- Node.js: Pastikan sudah terinstal di sistem. Jika belum, instal melalui situs resminya.
- Buat direktori baru untuk proyek, lalu inisialisasi proyek Node.js dengan perintah `npm init`.
- Instal paket yang dibutuhkan:

```
```bash  

npm install express mysql2 ejs body-parser

```
```

2. Membuat Database MySQL

- Buka MySQL dan buat database bernama `pertemuan5`:

```
```sql  

CREATE DATABASE pertemuan5;

```
```

- Selanjutnya, buat tabel `users` untuk menyimpan data pengguna:

```
```sql  

CREATE TABLE users (

 id INT AUTO_INCREMENT PRIMARY KEY,

 name VARCHAR(100),

 email VARCHAR(100),

 phone VARCHAR(20)

);

```
```

3. Membuat Server Node.js

- Buat file `server.js` untuk konfigurasi server Node.js.
- Kode di dalam `server.js` mengatur koneksi ke MySQL dan menangani operasi CRUD:

```
````javascript
```

```
const express = require('express');
```

```
const mysql = require('mysql2');
```

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

```
const app = express();
```

```
app.use(bodyParser.urlencoded({ extended: false }));
```

```
app.use(bodyParser.json());
```

```
const connection = mysql.createConnection({
```

```
 host: 'localhost',
```

```
 user: 'root',
```

```
 password: '',
```

```
 database: 'pertemuan5'
```

```
});
```

```
connection.connect((err) => {
```

```
 if (err) {
```

```
 console.error("Koneksi gagal:", err.stack);
```

```
 return;
```

```
 }
```

```
 console.log("Koneksi berhasil dengan id " + connection.threadId);
```

```
});
```

```
app.set('view engine', 'ejs');
```

// Read Data

```
app.get('/', (req, res) => {
 const query = 'SELECT * FROM users';
 connection.query(query, (err, result) => {
 res.render('index', { users: result });
 });
});
```

// Create Data

```
app.post('/add', (req, res) => {
 const { name, email, phone } = req.body;
 const query = 'INSERT INTO users (name, email, phone) VALUES (?, ?, ?)';
 connection.query(query, [name, email, phone], (err, result) => {
 if (err) throw err;
 res.redirect('/');
 });
});
```

// Update Data

```
app.get('/edit/:id', (req, res) => {
 const { id } = req.params;
 const query = 'SELECT * FROM users WHERE id = ?';
 connection.query(query, [id], (err, result) => {
 if (err) throw err;
 res.render('edit', { user: result[0] });
 });
});
```

```

app.post('/update/:id', (req, res) => {
 const { id } = req.params;
 const { name, email, phone } = req.body;
 const query = 'UPDATE users SET name = ?, email = ?, phone = ? WHERE id = ?';
 connection.query(query, [name, email, phone, id], (err, result) => {
 if (err) throw err;
 res.redirect('/');
 });
});

// Delete Data
app.get('/delete/:id', (req, res) => {
 const { id } = req.params;
 const query = 'DELETE FROM users WHERE id = ?';
 connection.query(query, [id], (err, result) => {
 if (err) throw err;
 res.redirect('/');
 });
});

app.listen(3000, () => {
 console.log("Server berjalan di port 3000");
});

```

#### 4. Membuat Tampilan Frontend dengan EJS

- Buat file `views/index.ejs` untuk menampilkan daftar pengguna dan form tambah pengguna baru.
- Buat juga file `views/edit.ejs` untuk form edit data pengguna.
- Kode lengkap tampilan frontend sudah dijelaskan sebelumnya.

## 5. Pengujian CRUD

- Jalankan server Node.js dengan perintah:

```
```bash  
  
node server.js  
  
```
```

- Buka browser dan akses `http://localhost:3000`. Di halaman tersebut, akan muncul daftar pengguna, form untuk menambah pengguna, serta opsi untuk mengedit dan menghapus data.
- Coba tambahkan data, edit data yang sudah ada, dan hapus data melalui aplikasi web.

## IV. Kesimpulan

Dalam praktikum ini, kami telah berhasil membangun aplikasi web sederhana berbasis Node.js dengan MySQL sebagai databasenya. Aplikasi ini memungkinkan pengguna untuk melakukan operasi CRUD (Create, Read, Update, Delete) pada data pengguna. Kami menggunakan Express.js untuk membuat server dan EJS untuk menampilkan data secara dinamis di halaman web. Dengan memahami cara kerja CRUD, kami dapat membangun aplikasi yang mampu mengelola data pengguna dengan baik. Pengalaman praktikum ini juga menunjukkan pentingnya integrasi yang baik antara backend dan frontend dalam aplikasi berbasis web.