

Laporan Praktikum
Mata Kuliah PEMROGRAMAN BERORIENTASI OBJEK



Pertemuan 6
"CRUD"

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

Disusun Oleh :
Dini Dwi Andini
2308802

PROGRAM STUDI SISTEM INFORMASI KELAUTAN
UNIVERSITAS PENDIDIKAN INDONESIA
2024

I. PENDAHULUAN

Pada praktikum kelima mata kuliah Pemrograman Berorientasi Objek, kita akan mempelajari dan mengimplementasikan operasi CRUD (Create, Read, Update, Delete). Operasi ini merupakan dasar dalam pengelolaan data pada aplikasi berbasis objek. Melalui praktikum ini, mahasiswa diharapkan dapat memahami dan menerapkan konsep CRUD dalam pengembangan aplikasi yang efisien dan interaktif.

II. ALAT DAN BAHAN

- Laptop
- Mouse(Optional)
- Bahasa pemrograman JS, HTML, CSS.

III. LANGKAH KERJA

1. Membuat folder Node.JS
2. Buka terminal dan jalankan perintah: `npm init -y`
Ini akan membuat file "package.json."
3. Install package yang diperlukan: `npm install express mysql2 ejs body-parser`
 - Express: Framework untuk menangani routing dan server.
 - MySQL2: Untuk berkomunikasi dengan database MySQL.
 - EJS: Template engine untuk membuat view.
 - Body-parser: Untuk menangani parsing data form.

4. Membuat Server dengan Express

5. Buat file `app.js` dan tambahkan kode berikut untuk membuat server sederhana:

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

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

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

```
// Menjalankan server
app.listen(3000, () => {
  console.log('Server running on port 3000');
});
```

6. Koneksi ke Database MySQL

7. Buat koneksi ke MySQL:

```
const connection = mysql.createConnection({
  host: 'localhost',
  user: 'root',
  password: "",
```

```

    database: 'your_database_name'
  });

  connection.connect((err) => {
    if (err) {
      console.log('Error connecting to MySQL:', err);
    } else {
      console.log('Connected to MySQL');
    }
  });
});

```

8. Buat database dan tabel di MySQL:

```

CREATE DATABASE your_database_name;
USE your_database_name;

```

```

CREATE TABLE member (
  id INT AUTO_INCREMENT PRIMARY KEY,
  nama VARCHAR(255),
  posisi VARCHAR(255),
  luckyNumber INT
);

```

9. Membuat routing CRUD

10. Read Data (Menampilkan data dari database):

```

// Read
app.get('/', (req, res) => {
  const query = 'SELECT * FROM member';
  connection.query(query, (err, results) => {
    if (err) throw err;
    res.render('index', { member: results, message: req.query.message });
  });
});

```

11. Create Data (Menambahkan data baru):

```

app.post('/add', (req, res) => {
  const { nama, posisi, luckyNumber } = req.body;
  const query = 'INSERT INTO member (nama, posisi, luckyNumber)
VALUES (?, ?, ?)';
  connection.query(query, [nama, posisi, luckyNumber], (err, result) => {
    if (err) throw err;
    res.redirect('/');
  });
});

```

12. Update Data (Mengedit data yang ada):

```

app.post('/update/:id', (req, res) => {
  const { nama, posisi, luckyNumber } = req.body;
  const query = 'UPDATE member SET nama = ?, posisi = ?, luckyNumber
= ? WHERE id = ?';

```

```

        connection.query(query, [nama, posisi, luckyNumber, req.params.id],
        (err, result) => {
            if (err) throw err;
            res.redirect('/');
        });
    });

```

13. Delete Data (Menghapus data):

```

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

```

14. Buat file views, lalu masukkan kode ini ke index.ejs untuk menampilkan data:

```

<h1>Daftar Member</h1>
<table border="1">
    <tr>
        <th>ID</th>
        <th>Nama</th>
        <th>Posisi</th>
        <th>LuckyNumber</th>
        <th>Aksi</th>
    </tr>
    <% member.forEach(member => { %>
    <tr>
        <td><%= member.id %></td>
        <td><%= member.nama %></td>
        <td><%= member.posisi %></td>
        <td><%= member.luckyNumber %></td>
        <td>
            <a href="/edit/<%= member.id %>">Edit</a>
            <a href="/delete/<%= member.id %>">Delete</a>
        </td>
    </tr>
    <% }) %>
</table>

<h2>Tambah Member</h2>
<form action="/add" method="post">
    <input type="text" name="nama" placeholder="Nama" required>
    <input type="text" name="posisi" placeholder="Posisi">
    <input type="number" name="luckyNumber" placeholder="Lucky
Number">
    <button type="submit">Tambah</button>
</form>

```

15. Jalankan program di terminal dengan menekan ctrl+c terlebih dahulu dilanjut dengan perintah: node index.js

16. Akses aplikasi di browser pada <http://localhost:3000>. Dengan kode berikut ini :

```
app.listen(3000,() => {  
  console.log("Server berjalan di port 3000, buka web melalui  
http://localhost:3000")  
})
```

17. Jangan lupa membuat file edit.ejs :

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-  
scale=1.0">  
  <title>Edit Data Member</title>  
</head>  
<body>  
  <h2>Edit Data Member Baru</h2>  
  <form action="/update/<%= member.id %>" method="post">  
    <label for="nama">Nama:</label>  
    <input type="text" id="nama" name="nama" required value="<%=  
member.nama %>"><br>  
    <label for="Posisi">Posisi:</label>  
    <input type="text" name="Posisi" id="Posisi" value="<%=  
member.Posisi %>"><br>  
    <label for="LuckyNumber">Nomor Keberuntungan</label>  
    <input type="number" name="LuckyNumber" id="LuckyNumber"  
value="<%= member.LuckyNumber %>"><br>  
    <button type="submit">Simpan</button>  
  </form>  
</body>  
</html>
```

18. UPDATE adalah perintah SQL yang digunakan untuk memperbarui data yang sudah ada di dalam tabel. Contohnya ketik : UPDATE member SET nama = "Mark", Posisi = "Center", LuckyNumber = "8" WHERE id = 6

19. Jika ingin menambahkan atau memasukkan data lagi bisa dengan mengetik seperti berikut : UPDATE member SET nama = "Mark", Posisi = "Center", LuckyNumber = "8" WHERE id = 6 pada .sql

20. Dan jika ingin menghapus data bisa dengan mengetik : DELETE FROM member WHERE id = 1 pada .sql

21. Jika ingin menampilkan semua data dari tabel "member," perintahnya adalah: SELECT * FROM member

22. Jika ingin menambahkan data baru ke tabel "member," perintahnya adalah: INSERT INTO member (nama, Posisi, LuckyNumber) VALUES ("Taeyong","Leader","1")

23. Berikut ini kode index.ejs yang sudah ditambah bootstrap, css, alert, dan prompt:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Crud Node JS - MYSQL</title>
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min
.css" rel="stylesheet">
</head>
<body>
  <style>
    body {
      /* Gradasi dari biru tua ke biru muda */
      background: linear-gradient(to bottom, #1e3c72, #2a5298);
      color: #15c2d2;
      font-family: Arial, sans-serif;
    }

    h1 {
      color: hwb(202 41% 15%);
      margin-bottom: 30px;
    }

    table {
      background-color: #21d1f9cc;
      color: hsl(0, 82%, 41%);
    }

    th, td {
      padding: 10px;
      text-align: center;
    }

    .btn-warning, .btn-danger {
      margin-right: 5px;
    }

  </style>
  <div class="container">
    <h1 class="my-4 text-center">Daftar Member Idol Kpop</h1>

    <% if (typeof message !== 'undefined' && message) { %>
```

```

        <div class="alert alert-success alert-dismissible fade show"
role="alert">
            <%= message %>
            <button type="button" class="btn-close" data-bs-
dismiss="alert" aria-label="Close"></button>
        </div>
    <% } %>

```

```

<table class="table table-bordered table-hover">
    <thead>
        <tr>
            <th>ID</th>
            <th>Nama</th>
            <th>Posisi</th>
            <th>LuckyNumber</th>
            <th>Aksi</th>
        </tr>
    </thead>
    <tbody>
        <% member.forEach(member => { %>
            <tr>
                <td><%= member.id %></td>
                <td><%= member.nama %></td>
                <td><%= member.Posisi %></td>
                <td><%= member.LuckyNumber %></td>
                <td>
                    <a href="/edit/<%= member.id %>" class="btn btn-
warning btn-sm">Edit</a>
                    <a href="/delete/<%= member.id %>" class="btn btn-
danger btn-sm" onclick="return confirm('Apakah Anda yakin ingin
menghapus data ini?')">Hapus</a>
                </td>
            </tr>
        <% }) %>
    </tbody>
</table>

```

```

<h2 class="mt-5">Tambah Member Baru</h2>
<form action="/add" method="post">
    <div class="mb-3">
        <label for="nama" class="form-label">Nama:</label>
        <input type="text" id="nama" name="nama" class="form-
control" required>
    </div>
    <div class="mb-3">
        <label for="Posisi" class="form-label">Posisi:</label>

```

```

        <input type="text" name="Posisi" id="Posisi" class="form-
control">
    </div>
    <div class="mb-3">
        <label for="LuckyNumber" class="form-
label">LuckyNumber:</label>
        <input type="number" name="LuckyNumber"
id="LuckyNumber" class="form-control">
    </div>
    <button type="submit" class="btn btn-
primary">Tambah</button>
</form>
</div>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle
.min.js"></script>
</body>
</html>

```

24. Berikut ini tampilan Halaman page nya:

Daftar Member Idol Kpop

ID	Nama	Posisi	LuckyNumber	Aksi
1	Sungchan	Rapper	9	<button>Edit</button> <button>Hapus</button>
2	Anton	Vocal	3	<button>Edit</button> <button>Hapus</button>
3	Eunseok	Vocal	5	<button>Edit</button> <button>Hapus</button>
4	Shotaro	Lead Dance	5	<button>Edit</button> <button>Hapus</button>
5	Mark	Center	8	<button>Edit</button> <button>Hapus</button>
6	Ahyeon	Vocal, Rapper	4	<button>Edit</button> <button>Hapus</button>
7	Taeyong	Leader	1	<button>Edit</button> <button>Hapus</button>
8	Wonbin	Center	27	<button>Edit</button> <button>Hapus</button>
9	Sua	Rap	4	<button>Edit</button> <button>Hapus</button>
10	Taeyong	Leader	1	<button>Edit</button> <button>Hapus</button>

Tambah Member Baru

Nama:

Posisi:

LuckyNumber:

IV. KESIMPULAN

Pada pertemuan ini, kami berhasil memahami dan mengimplementasikan konsep CRUD (Create, Read, Update, Delete) dalam aplikasi menggunakan Node.js dan MySQL. Kami membuat aplikasi sederhana untuk mengelola data member yang terdiri dari id, nama, posisi, dan lucky number. Melalui praktikum ini, kami belajar Proses create memungkinkan pengguna untuk menambahkan data baru dengan atribut yang telah ditentukan. Fungsi read

digunakan untuk menampilkan data yang sudah tersimpan di database dalam bentuk tabel. Kemudian, update memungkinkan pengguna untuk mengubah informasi pada data yang sudah ada, dan delete digunakan untuk menghapus data berdasarkan id member. Dari sini juga akhirnya tau bagaimana menghubungkan aplikasi dengan database, membuat, menampilkan, mengubah, dan menghapus data secara dinamis. Implementasi CRUD ini menunjukkan pentingnya pemahaman dasar pengelolaan data dalam pengembangan aplikasi berbasis objek.