

**LAPORAN PRAKTIKUM MATA KULIAH
PEMROGRAMAN BERORIENTASI OBJEK**



Pertemuan 6 Praktikum 5

Dosen Pengampu :

Willdan Aprizal Arifin, S.Pd., M.Kom.

Disusun Oleh :

Syifa Fauzia (2309845)

SIK B/3

PROGRAM STUDI SISTEM INFORMASI KELAUTAN

UNIVERSITAS PENDIDIKAN INDONESIA

2024

PENJELASAN CODE APP.JS

1. Memulai Koneksi dan Konfigurasi

```
JS app.js > ...
1  const express = require('express');
2  const mysql = require('mysql2');
3  const bodyParser = require('body-parser');
4
5  const app = express();
6  app.use(bodyParser.urlencoded({extended : false}));
7  app.use(bodyParser.json());
8  app.use(express.static('foto'));
9
10 const connection = mysql.createConnection({
11   host: 'localhost',
12   user: 'root',
13   password: '',
14   database: 'pbo5'
15 });
16
17 connection.connect((err) => {
18   if(err) {
19     console.error("Terjadi kesalahan dalam koneksi ke MySQL:", err.stack);
20     return;
21   }
22   console.log("Koneksi MySQL berhasil dengan id" + connection.threadId)
23 });
24
25 app.set('view engine', 'ejs');
```

- `express`: Digunakan untuk membuat aplikasi web.
- `mysql2`: Digunakan untuk menghubungkan aplikasi dengan database MySQL.
- `bodyParser`: Middleware untuk mengurai body dari request HTTP menjadi format yang dapat dibaca.
- `app.use(...)`: Mengaktifkan middleware `bodyParser` untuk menguraikan data dari form (URL-encoded) dan format JSON.
- `express.static('foto')`: Mengatur folder foto untuk menyimpan file statis (misalnya gambar).
- `mysql.createConnection`: Membuat koneksi ke database MySQL dengan detail host, user, password, dan database.
- `connection.connect`: Memulai koneksi ke database dan menampilkan pesan error jika koneksi gagal.
- Menentukan `ejs` sebagai view engine untuk merender halaman HTML yang dinamis.

2. Routing (CRUD - Create, Read, Update, Delete)

```
29 app.get('/', (req, res) => {
30   const query = 'SELECT * FROM pulau';
31   connection.query(query, (err, results) => {
32     res.render('index', {pulau:results});
33   });
34 });
35
36 app.post('/add', (req, res) => {
37   const {name, lokasi, harga, phone} = req.body;
38   const query = 'INSERT INTO pulau (name, lokasi, harga, phone) VALUES (?, ?, ?, ?)';
39   connection.query(query, [name, lokasi, harga, phone], (err, result) => {
40     if (err) {
41       console.error("Terjadi kesalahan saat menambahkan data:", err);
42       return res.status(500).send("Gagal menambahkan data");
43     }
44     res.redirect('/');
45   });
46 });
47
48
49 app.get('/edit/:id', (req, res) => {
50   const query = 'SELECT * FROM pulau WHERE id = ?';
51   connection.query(query, [req.params.id], (err, result) => {
52     if (err) throw err;
53     res.render('edit', {pulau:result[0]});
54   });
55 })
56
57 app.post('/update/:id', (req, res) =>{
58   const {name, lokasi, harga, phone} = req.body;
59   const query = 'UPDATE pulau SET name = ?, lokasi = ?, harga = ?, phone = ? WHERE id = ?';
60   connection.query(query,[name, lokasi, harga, phone, req.params.id], (err, result) =>{
61     if (err) throw err;
62     res.redirect('/');
63   });
64 })
65
66 app.get('/delete/:id', (req, res) => {
67   const query = 'DELETE FROM pulau WHERE id = ?';
68   connection.query(query, [req.params.id], (err, result) => {
69     if (err) throw err;
70     res.redirect('/');
71   });
72 })
73
```

- Endpoint GET / untuk mengambil semua data dari tabel pulau dan menampilkannya pada halaman index.ejs.
- connection.query: Melakukan query SQL SELECT * FROM pulau untuk mengambil data pulau.
- res.render('index', { pulau: results }): Merender data pulau ke template index.ejs.
- Endpoint POST /add untuk menambahkan data baru ke tabel pulau.
- Data diambil dari body request (req.body), yang berisi name, lokasi, harga, dan phone.
- Query SQL INSERT INTO pulau (name, lokasi, harga, phone) digunakan untuk memasukkan data baru ke tabel.
- Endpoint GET /edit/:id digunakan untuk menampilkan data yang ingin diubah berdasarkan id.
- Query SQL SELECT * FROM pulau WHERE id = ? digunakan untuk mengambil data dari database sesuai id.
- Endpoint POST /update/:id untuk mengupdate data yang sudah ada di tabel pulau.

- Query SQL UPDATE pulau SET ... WHERE id = ? digunakan untuk memperbarui data yang sesuai dengan id.
- Endpoint GET /delete/:id digunakan untuk menghapus data pulau berdasarkan id.
- Query SQL DELETE FROM pulau WHERE id = ? menghapus baris data yang sesuai dengan id.

3. Menjalankan Server

```
74 app.listen(3000, () =>{
75   console.log("Server berjalan di prt 3000, buka web melalui http://localhost:3000 ")
76 });
```

- app.listen(3000): Menjalankan server pada port 3000. Ketika aplikasi diakses melalui http://localhost:3000, pengguna akan melihat halaman utama.

Kesimpulan:

Aplikasi ini menggunakan Express.js untuk membuat aplikasi web CRUD yang sederhana dengan MySQL sebagai basis data. Fungsi utamanya adalah menampilkan daftar pulau dari tabel `pulau`, menambahkan pulau baru, memperbarui informasi pulau yang sudah ada, dan menghapus data pulau dari database.

PENJELASAN CODE INDEX.EJS

1. Struktur Dasar HTML

```
views > index.ejs > html > body > table > tr > th
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Crud Node JS - MySQL</title>
7   <link rel="stylesheet" href="style.css">
8 </head>
9 <body>
10  <h1> Destinasi Pulau Terbaik di Indonesia </h1>
```

- **DOCTYPE dan Struktur HTML:** Menandai bahwa ini adalah dokumen HTML5.
- **<meta> Tag:** Menentukan karakter set halaman (UTF-8) dan mengatur agar halaman tampil responsif di perangkat dengan lebar layar yang berbeda melalui viewport.
- **Title:** Judul dari halaman adalah "*Crud Node JS - MySQL*" yang akan muncul di tab browser.
- **Link ke CSS:** Menghubungkan file `style.css` yang berisi aturan gaya (CSS) untuk tampilan halaman.
- **<h1>:** Menampilkan judul besar "Destinasi Pulau Terbaik di Indonesia" di bagian atas halaman.

3. Tabel Data Destinasi Pulau

```
11 <table border="1">
12 <tr>
13 <th>ID</th>
14 <th>name</th>
15 <th>lokasi</th>
16 <th>harga</th>
17 <th>telephone</th>
18 <th>Aksi</th>
19 </tr>
20 <% pulau.forEach(Pengguna => {%>
21 <tr>
22 <td><%= Pengguna.id %></td>
23 <td><%= Pengguna.name %></td>
24 <td><%= Pengguna.lokasi %></td>
25 <td><%= Pengguna.harga %></td>
26 <td><%= Pengguna.phone %></td>
27 <td>
28 <a href="/edit/<%= Pengguna.id %>">Edit</a>
29 <a href="/delete/<%= Pengguna.id %>">Hapus</a>
30 </td>
31 </tr>
32 <% }%>
33 </table>
34
```

- **<table>**: Menampilkan tabel yang memiliki border (bingkai) dengan atribut `border="1"`. Ini digunakan untuk menampilkan data destinasi pulau yang tersimpan di dalam database.
- **Header Kolom (<th>)**: Bagian ini mendefinisikan nama kolom dalam tabel:
 - ID: Nomor identitas pulau.
 - Name: Nama pulau.
 - Lokasi: Lokasi dari pulau.
 - Harga: Kisaran harga tiket masuk ke pulau.
 - Telephone: Nomor telepon terkait destinasi pulau.
 - Aksi: Menampilkan tautan untuk "Edit" dan "Hapus".
- **Looping Data**:
 - Menggunakan EJS (Embedded JavaScript) untuk melakukan looping pada array pulau yang berisi data dari database.
 - `pulau.forEach()` akan membuat baris (<tr>) untuk setiap data pulau yang ada dalam array.
- **Menampilkan Data Pulau**:
 - Setiap data pulau ditampilkan di dalam kolom (<td>) masing-masing, misalnya id, name, lokasi, harga, dan phone.
- **Aksi Edit dan Hapus**:
 - Membuat tautan untuk setiap data pulau. Tautan ini digunakan untuk:
 - Edit: Mengarahkan pengguna ke halaman edit untuk memperbarui data berdasarkan id pulau.
 - Hapus: Menghapus data pulau berdasarkan id dari database.

4. Formulir Tambah Destinasi Pulau Dan Penutup

```
34
35 <h2 id="Tambah">Tambah Destinasi Pulau </h2>
36 <form action="/add" method="post">
37   <label for="name">name:</label>
38   <input type="text" Name="name" placeholder="Nama Pulau">
39   <label for="lokasi">lokasi:</label>
40   <input type="text" name="lokasi" placeholder="Lokasi Pulau">
41   <label for="harga">harga:</label>
42   <input type="text" name="harga" placeholder="Harga Tiket">
43   <label for="phone">telephone:</label>
44   <input type="text" name="phone" placeholder="Nomor Telepon">
45   <button type="submit">Tambah</button>
46 </form>
47
48 </body>
49 </html>
```

- Menampilkan judul formulir untuk menambah destinasi baru.
- Formulir ini dikirim ke server menggunakan metode POST ke endpoint /add untuk menambahkan destinasi baru ke dalam database.
- Terdapat beberapa input untuk mengisi data destinasi pulau yang baru:
 - **name:** Nama pulau.
 - **lokasi:** Lokasi dari pulau.
 - **harga:** Harga tiket masuk.
 - **phone:** Nomor telepon yang terkait.
- Mengirimkan data ke server untuk disimpan ke dalam database.
- Menutup struktur HTML yang dimulai dari tag <html>.

Kesimpulan:

Halaman ini berfungsi sebagai bagian tampilan dari aplikasi CRUD (Create, Read, Update, Delete) yang menampilkan data pulau dari database MySQL dan menyediakan formulir untuk menambah destinasi baru dan menggunakan EJS untuk merender data secara dinamis dari server ke HTML kemudia CRUD operasi Melalui tabel, pengguna bisa mengedit dan menghapus data yang ada, serta menambahkan data baru melalui formulir di bagian bawah.

PENJELASAN CODE INDEX.EJS

1. Struktur Dasar HTML Sampai Menutup struktur HTML.

```
views > edit.ejs > html > body
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Edit Destinasi Pulau</title>
7   <link rel="stylesheet" href="style.css">
8 </head>
9 <body>
10  <h2>Edit Destinasi Pulau</h2>
11  <form action="/update/<%= pulau.id %>" method="post">
12    <label for="name">name:</label>
13    <input type="text" id="name" name="name" required><br>
14    <label for="lokasi">lokasi:</label>
15    <input type="text" name="lokasi" id="lokasi"><br>
16    <label for="harga">harga:</label>
17    <input type="text" name="harga" id="harga"><br>
18    <label for="phone">telephone:</label>
19    <input type="text" name="phone" id="phone"><br>
20    <button type="submit">Tambah</button>
21  </form>
22
23 </body>
24 </html>
```

- DOCTYPE dan Struktur HTML: Menandakan bahwa ini adalah dokumen HTML5.
- <meta> Tag: Menentukan karakter set halaman (UTF-8) dan mengatur agar halaman dapat ditampilkan responsif di berbagai ukuran perangkat.
- <title>: Memberikan judul halaman, yaitu "Edit Destinasi Pulau", yang akan muncul di tab browser.
- <link rel="stylesheet">: Menghubungkan ke file CSS style.css yang berisi aturan gaya untuk halaman ini.
- <h2>: Menampilkan judul "Edit Destinasi Pulau" yang akan muncul di bagian atas halaman.
- <form>: Formulir ini akan mengirim data ke server menggunakan metode POST ke endpoint /update/<%= pulau.id %>. Bagian <%= pulau.id %> adalah EJS (Embedded JavaScript) yang akan dirender dengan ID dari pulau yang sedang di-edit. Artinya, formulir ini terkait dengan data pulau tertentu berdasarkan id.
 - Action /update/<%= pulau.id %>: Formulir ini akan mengirimkan data yang diperbarui ke route /update/:id di mana id adalah ID dari pulau yang ingin diubah.
- Di bawah ini adalah beberapa input field untuk mengisi data pulau yang akan di-update. Field ini mengandung data yang ada sebelumnya, atau pengguna dapat mengedit datanya.

- Label dan Input Field: Label ini untuk input field "Nama Pulau". Pengguna dapat mengubah nama pulau di field ini.
- required: Input ini wajib diisi sebelum formulir dapat dikirimkan.
- Label dan Input Field: Input ini digunakan untuk mengubah lokasi dari pulau.
- Label dan Input Field: Input ini digunakan untuk mengubah harga tiket pulau.
- Label dan Input Field: Input ini digunakan untuk mengubah nomor telepon yang terkait dengan destinasi pulau.
- Tombol Submit: Setelah pengguna melakukan perubahan data, mereka bisa menekan tombol ini untuk mengirim data yang telah diperbarui ke server. Tombol ini mengirimkan data melalui POST ke URL /update/:id.
- Menutup struktur HTML.

Kesimpulan:

Halaman ini digunakan untuk mengedit data destinasi pulau. Setiap input field memiliki data yang bisa diubah oleh pengguna dan data dikirim melalui metode POST ke server, dan server akan memperbarui data pulau berdasarkan ID di database kemudian EJS Template: EJS digunakan untuk merender nilai dinamis ke dalam HTML, seperti ID pulau yang sedang diedit.

HASIL DARI WEB YANG DI BUAT

The screenshot shows a web application running on localhost:3000. The main heading is "Destinasi Pulau Terbaik di Indonesia". Below it is a table with 7 rows of island data. Each row has columns for ID, name, lokasi, harga, telephone, and Aksi. The Aksi column contains "Edit" and "Hapus" links. Below the table is a form titled "Tambah Destinasi Pulau" with input fields for name, lokasi, harga, and telephone, and a "Tambah" button. The background of the page is a scenic image of a beach with a large tree root in the foreground and turquoise water.

ID	name	lokasi	harga	telephone	Aksi
1	Pulau Bali	Provinsi Bali	Rp 20.000 - Rp 50.000	081222334455	Edit Hapus
2	Pulau Weh	Aceh, Sumatra Utara	Rp 10.000 - Rp 20.000	081233445566	Edit Hapus
3	Pulau Tidung	Kepulauan Seribu, Jakarta	Rp 40.000 - Rp 70.000	081344556677	Edit Hapus
4	Pulau Bintan	Kepulauan Riau	Rp 25.000 - Rp 50.000	081355667788	Edit Hapus
5	Pulau Gili Trawangan	Lombok, Nusa Tenggara Barat	Rp 100.000 - Rp 150.000	081366778899	Edit Hapus
6	Pulau Belitung	Kepulauan Bangka Belitung	Rp 10.000 - Rp 25.000	081377889900	Edit Hapus
7	Pulau Komodo	Nusa Tenggara Timur	Rp 300.000 - Rp 500.000	081211223344	Edit Hapus

Tambah Destinasi Pulau

name:

lokasi:

harga:

telephone:

[Tambah](#)

Maksud dari di buatnya WEB ini untuk upgrade wisata destinasi pulau di indonesia dan untuk wisata pulau yang sudah tidak digunakan sebagai wisata bisa di hapus lalu bisa juga untuk wisata pulau apabila memiliki upgrade terbaru mengenai harga tiket dan nomer telephone bisa untuk di edit.

