Laporan Praktikum

Mata Kuliah Pemrograman Web

Pertemuan 5.Praktikum5

“Pembuatan Website Sederhana dengan Node.js dan HTML”

Dosen Pengampu :

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

Disusun Oleh :

Siti Nurkholizah 2308807

3A - Sistem Informasi Kelautan

PROGRAM STUDI SISTEM INFORMASI KELAUTAN

UNIVERSITAS PENDIDIKAN INDONESIA

2024

I. PENDAHULUAN

CRUD (Create, Read, Update, Delete) merupakan konsep fundamental dalam pengelolaan data yang memungkinkan interaksi dengan database. Dalam pengembangan aplikasi web, Node.js sering digunakan sebagai server backend karena sifatnya yang cepat dan mendukung operasi asinkron, sedangkan MySQL dipilih sebagai database relasional untuk penyimpanan data. Dengan menggabungkan Node.js dan framework express, operasi CRUD dapat dijalankan untuk memproses permintaan HTTP, seperti menambah, melihat, mengubah, dan menghapus data. MySQL berfungsi sebagai penyimpanan data utama, dan EJS digunakan untuk menghasilkan halaman web dinamis, memadukan data dengan tampilan antarmuka yang interaktif.

II. TUJUAN

Praktikum ini bertujuan untuk mempelajari dan memahami cara membuat aplikasi CRUD (Create, Read, Update, Delete) menggunakan **Node.js** dan **MySQL**. Aplikasi ini memungkinkan pengguna untuk menambah, melihat, memperbarui, dan menghapus data pengguna yang disimpan dalam database MySQL.

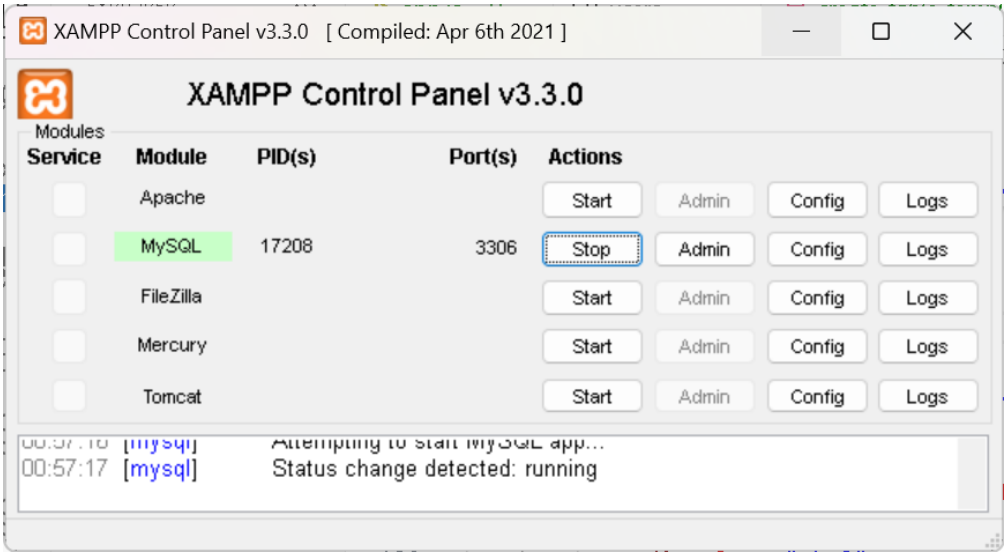
III. ALAT DAN BAHAN

- Laptop
- Node.js yang sudah terinstall di komputer.
- Visual Studio Code
- Browser web (Google Chrome)
- Terminal atau Command Prompt.
- File JS, EJS, dan CSS.

IV. LANGKAH KERJA

1. Memanggil npm init -y dalam terminal agar node js bisa dijalankan atau node_modules muncul dalam folder explorer
2. Menginstall npm express, mysql, body-parser, dan ejs dalam terminal fungsinya agar package.json dan package-lock.json muncul pada folder explorer
3. Setelah itu buat folder views dan buat 2 file “edit.ejs” “index.ejs” dimasukkan pada folder views. Serta buat 1 file terpisah “app.js”
4. Kemudian buat database pada server xampp dengan nama “pertemuan5” atau boleh buat baru server mysqlnya. Setelah dibuat databasenya input tabel “users” di db pertemuan5

5. Sebelum koneksi database ke server aktifkan terlebih dahulu aplikasi XAMPP centang mysql sampai muncul port 3306



Source code create table:

```
CREATE TABLE users(
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100),
    email VARCHAR(100),
    phone VARCHAR(12)
)
```

Klik run maka muncul affectrow 1.5 ms

6. Input kodingan index.ejs dan edit.ejs satu persatu
   Source code index.ejs yang sudah berisikan lengkap dengan css, bootstrap, alert dan promptnya

```
<!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>
    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
```

```
<style>
    body {
        background-color: #f5e8d4;
    }

    .container {
        margin-top: 50px;
    }
    h1 {
        color: #795548;
        margin-bottom: 30px;
        font-weight: bold;
        text-transform: uppercase;
    }
    h2 {
        color: #795548;
        margin-bottom: 30px;
        font-weight: bold;
        text-transform: uppercase;
    }

    table {
        width: 100%;
        margin-bottom: 30px;
    }
    th, td {
        text-align: center;
        vertical-align: middle;
        color: #6c757d;
    }
    .table-warning {
        background-color: #f7c094 !important;
    }

    .btn-edit {
        color: #fff;
        background-color: #795548;
        border: none;
        transition: background-color 0.3s ease;
    }
    .btn-edit:hover {
        background-color: #5d4037;
    }
```

```
.btn-delete {
    color: #fff;
    background-color: #ff7043;
    border: none;
    transition: background-color 0.3s ease;
}
.btn-delete:hover {
    background-color: #e64a19;
}
form {
    background-color: #fff3e0;
    padding: 20px;
    border-radius: 10px;
    box-shadow: 0px 4px 12px rgba(0, 0, 0, 0.1);
    margin-top: 30px;
}
.btn-warning {
    width: 100%;
    background-color: #ff9800;
    border: none;
    transition: background-color 0.3s ease;
}

.btn-warning:hover {
    background-color: #fb8c00;
}
.form-label {
    color: #6d4c41;
}
/* Style for the prompt */
.prompt {
    background-color: #ffcc80;
    border-left: 5px solid #ff9800;
    padding: 15px;
    margin-bottom: 20px;
    color: #6d4c41;
    font-size: 18px;
    border-radius: 5px;
    font-weight: bold;
}
```

```
    </style>
</head>
<body>
    <div class="container">
        <!-- Prompt -->
        <div class="prompt">
            Selamat datang di halaman daftar pengguna! Anda dapat
mengelola data pengguna dengan mudah melalui tabel ini.
        </div>

        <h1 class="text-center">Tabel Daftar Pengguna</h1>
        <table class="table table-bordered table-striped">
            <thead class="table-warning">
                <tr>
                    <th>ID</th>
                    <th>Nama</th>
                    <th>Email</th>
                    <th>Telepon</th>
                    <th>Aksi</th>
                </tr>
            </thead>
            <tbody>
                <% users.forEach(pengguna => {%>
                <tr>
                    <td><%= pengguna.id %></td>
                    <td><%= pengguna.name %></td>
                    <td><%= pengguna.email %></td>
                    <td><%= pengguna.phone %></td>
                    <td>

                        <a href="/edit/<%= pengguna.id %>" class="btn btn-
edit btn-sm" onclick="showEditAlert()">Edit</a>
                        <a href="/delete/<%= pengguna.id %>" class="btn btn-
delete btn-sm" onclick="showDeleteAlert()">Hapus</a>
                    </td>
                </tr>
                <% }) %>
            </tbody>
        </table>
```

```
        <h2 class="text-center">Tambah Pengguna Baru</h2>
        <form action="/add" method="post">
            <div class="mb-3">
                <label for="id" class="form-label">ID</label>
                <input type="number" id="id" name="id" class="form-
control" required>
            </div>
            <div class="mb-3">
                <label for="name" class="form-label">Nama:</label>
                <input type="text" id="name" name="name" class="form-
control" required>
            </div>
            <div class="mb-3">
                <label for="email" class="form-label">Email:</label>
                <input type="email" name="email" id="email" class="form-
control">
            </div>
            <div class="mb-3">
                <label for="phone" class="form-label">Telepon:</label>
                <input type="tel" name="phone" id="phone" class="form-
control">
            </div>
            <button type="submit" class="btn btn-
warning">Tambah</button>
        </form>
    </div>

    <!-- Bootstrap JS -->
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/js/bootstrap.bundle.min.js"></script>

    <!-- Alert Script -->
    <script>
        function showEditAlert() {
            alert('Sepertinya anda ingin mengedit. Lanjutkan?');
        }

        function showDeleteAlert() {
            alert('Data pengguna akan dihapus. Lanjutkan?');
        }
    </script>
</body>
</html>
```

Source code edit.ejs yang sudah lengkap dengan css, bootstrap, dan alertnya

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Edit Data Pengguna</title>
    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
    <style>
        body {
            background-color: #f8f9fa;
        }
        .container {
            margin-top: 50px;
            max-width: 600px;
        }
        h2 {
            margin-bottom: 20px;
        }
        label {
            font-weight: bold;
        }
        form {
            background-color: #efd1a1;
            padding: 20px;
            border-radius: 8px;
            box-shadow: 0 4px 8px rgba(240, 159, 61, 0.911);
        }

        button {
            width: 100%;
        }
```

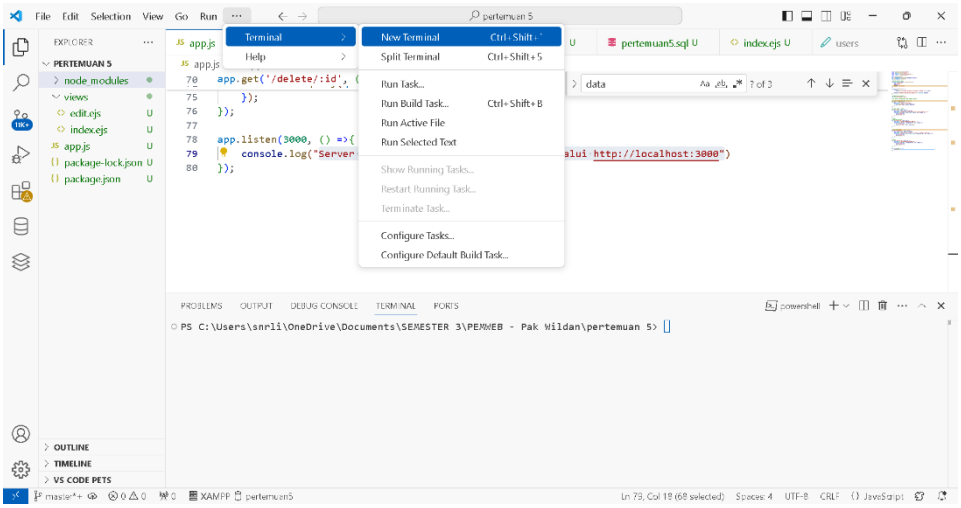
```
    </style>
</head>
<body>
    <div class="container">
        <h2 class="text-center">Edit Data Pengguna</h2>
        <form action="/update/<%= user.id %>" method="post"
onsubmit="return showSaveAlert()">
            <div class="mb-3">
                <label for="id" class="form-label">ID:</label>
                <input type="number" id="id" name="id" class="form-
control" required value="<%= user.id %>">
            </div>
            <div class="mb-3">
                <label for="name" class="form-label">Nama:</label>
                <input type="text" id="name" name="name" class="form-
control" required value="<%= user.name %>">
            </div>
            <div class="mb-3">
                <label for="email" class="form-label">Email:</label>
                <input type="email" name="email" id="email" class="form-
control" value="<%= user.email %>">
            </div>
            <div class="mb-3">
                <label for="phone" class="form-label">Telepon:</label>
                <input type="tel" name="phone" id="phone" class="form-
control" value="<%= user.phone %>">
            </div>
            <button type="submit" class="btn btn-
warning">Simpan</button>
        </form>
    </div>

    <!-- Bootstrap JS -->
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/js/bootstrap.bundle.min.js"></script>
    <!-- Alert Script -->
    <script>
        function showSaveAlert() {
            alert('Data telah berhasil diedit !');
            return true; // Membiarkan form tetap submit setelah alert
ditampilkan
        }
    </script>
</body>
</html>
```

Source code app.js

```
const express = require('express');
const mysql = require('mysql');
const bodyparser = require('body-parser');
const { name } = require('ejs');

const app = express();
app.use(bodyparser.urlencoded({extended : false}));
app.use(bodyparser.json());

//membuat koneksi mysql
const connection = mysql.createConnection({
   host: 'localhost',
   user: 'root',
   password: '',
   database: 'pertemuan5'
});

connection.connect((err) =>{
   if(err){
      console.error("Terjadi kesalahan dalam kondisi ke MySQL:",
err.stack);
   }
   console.log("Koneksi MySQL berhasil dengan id" +
connection.threadId)
});

//menggunakan view engine
app.set('view engine', 'ejs');

//ini adalah routing(Create, Read, Update, Delete)

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

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

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

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

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

7. Untuk menjalankan server buka terminal dan jalankan perintah berikut

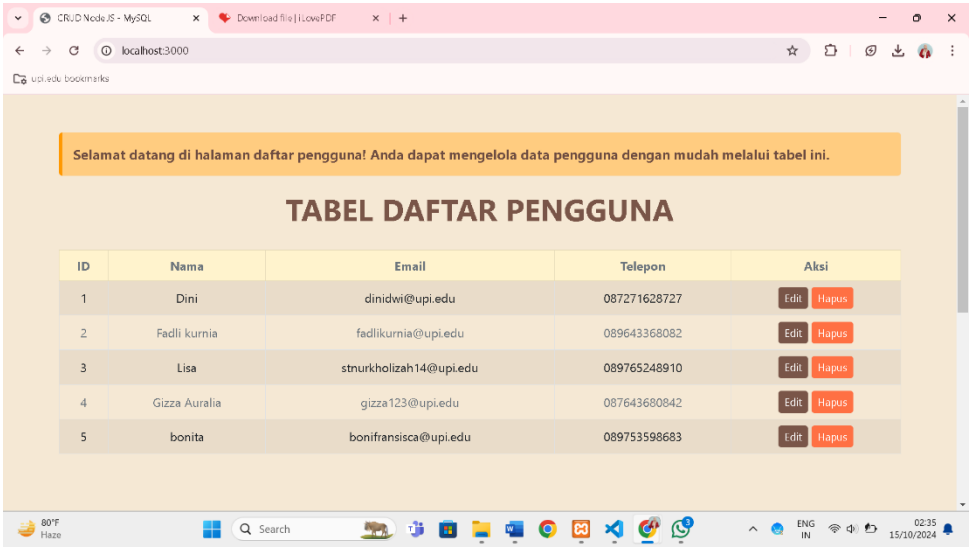


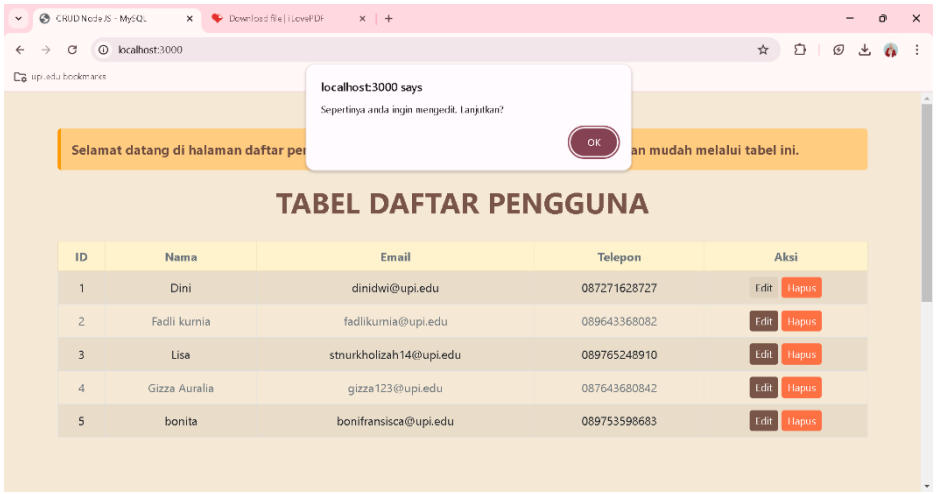
Kemudian server akan berjalan diport 3000 dan dapat mengakses halaman web dibrowser melalui URL : http ://localhost:3000

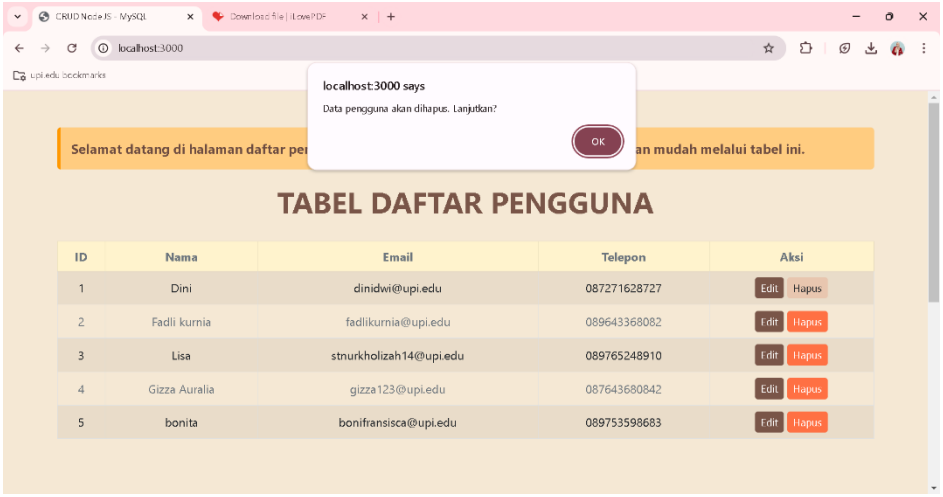
V. HASIL DAN PEMBAHASAN

Setelah server dijalankan, halaman **index.ejs** akan tampil di browser ketika mengakses http://localhost:3000. Proses ini menunjukkan bagaimana server Node.js membaca file dari sistem dan mengirimkannya sebagai tanggapan ke browser. Dalam praktik ini, saya berhasil membuat server sederhana dan memahami bagaimana data diambil dari file sistem serta ditampilkan sebagai halaman web.

Tampilan index.ejs beserta alert dan prompt dari aksi



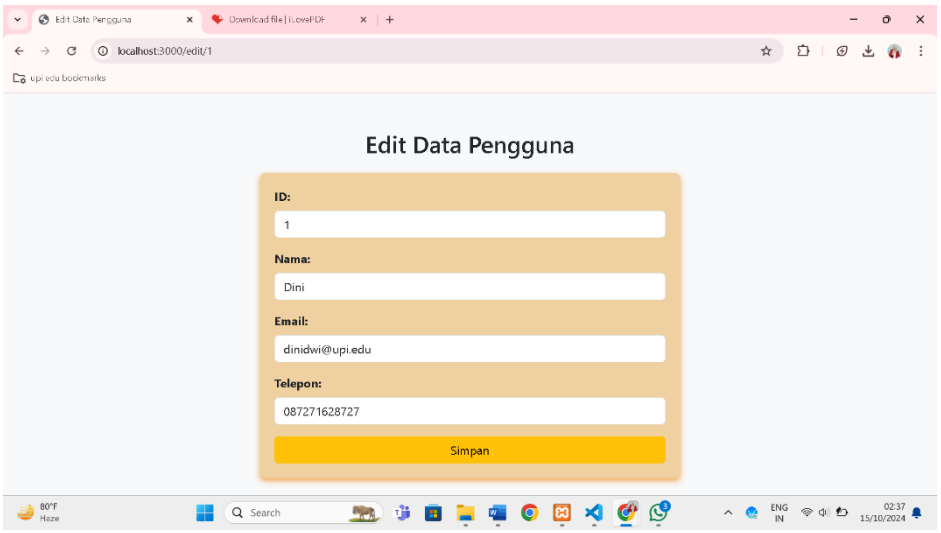


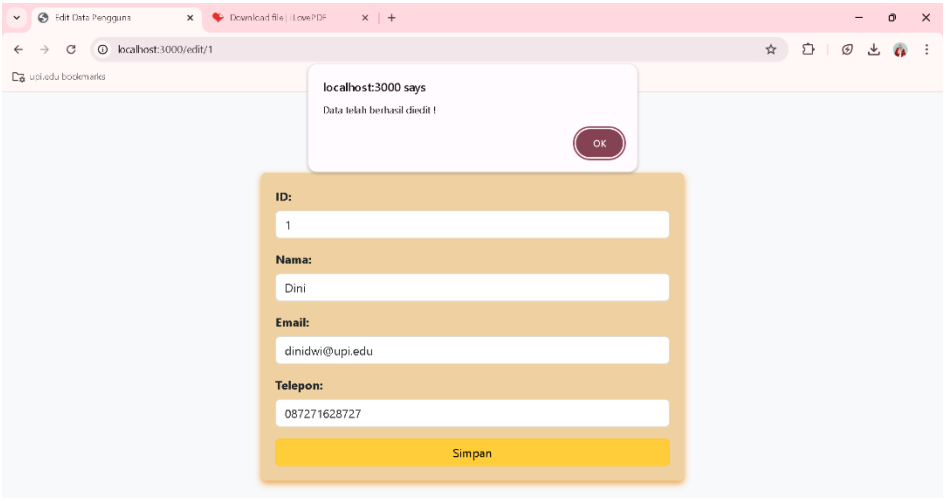


Tampilan tambah pengguna baru

Ketika diisi dan klik button tambah maka otomatis akan muncul dalam tabel data pengguna

Tampilan edit.ejs beserta alert jika diklik button simpan





VI. KESIMPULAN

Implementasi CRUD menggunakan Node.js dan MySQL memberikan solusi yang efisien untuk mengelola data dalam aplikasi web. Dengan memanfaatkan Express sebagai framework, operasi CRUD dapat diproses dengan cepat, memungkinkan pengembangan aplikasi yang dinamis dan interaktif. Selain itu, MySQL sebagai database relasional memudahkan pengelolaan data dengan struktur tabel yang terorganisir, serta mendukung berbagai fitur untuk menjaga integritas data.

Integrasi antara Node.js, Express, MySQL, dan EJS memungkinkan pembuatan aplikasi web yang dapat menangani interaksi pengguna secara real-time dengan database. Melalui pendekatan ini, pengembang dapat dengan mudah membuat, menampilkan, memperbarui, dan menghapus data dengan lebih cepat dan terstruktur. Secara keseluruhan, kombinasi teknologi ini sangat berguna untuk membangun aplikasi yang efisien dan mudah di-maintain di masa depan.