

LAPORAN PRAKTIKUM MATA KULIAH
PEMOGRAMAN WEB



Pertemuan 5 (Tugas 2) 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

1. Pendahuluan

Pada praktikum ini, kita akan membuat sebuah aplikasi web sederhana dengan fitur CRUD (Create, Read, Update, Delete) menggunakan Node.js dengan framework Express, MySQL sebagai basis data, serta EJS (Embedded JavaScript) sebagai template engine untuk menampilkan data di sisi frontend. CRUD adalah operasi dasar yang biasa digunakan dalam pengelolaan basis data. Dengan implementasi CRUD, kita dapat membuat (create), membaca (read), memperbarui (update), dan menghapus (delete) data dari basis data.

Tujuan Praktikum

- Mempelajari cara membangun aplikasi web sederhana dengan operasi CRUD.
- Memahami cara menghubungkan aplikasi Node.js dengan database MySQL.
- Menerapkan template engine EJS untuk rendering data dari server ke halaman web.

2. Teknologi Utama

2.1 Node.js

Node.js adalah lingkungan runtime JavaScript berbasis Chrome V8 yang memungkinkan kita menjalankan JavaScript di sisi server. Node.js sering digunakan untuk membangun aplikasi web yang cepat dan efisien.

2.2 Express.js

Express.js adalah framework minimalis untuk Node.js yang memudahkan pengelolaan routing dan middleware, sehingga mempercepat pengembangan aplikasi web. Dalam proyek ini, Express digunakan sebagai server untuk menangani request HTTP dan menyediakan endpoint untuk operasi CRUD.

2.3 MySQL

MySQL adalah sistem manajemen basis data relasional (RDBMS) yang banyak digunakan untuk menyimpan dan mengelola data dalam tabel. MySQL dipilih dalam proyek ini untuk menyimpan informasi pengguna seperti nama, email, dan nomor telepon.

2.4 EJS (Embedded JavaScript)

EJS adalah template engine untuk Node.js yang memungkinkan kita menyisipkan logika JavaScript ke dalam HTML. EJS akan membantu kita untuk menampilkan data dari server (hasil query MySQL) ke halaman web.

3. Penjelasan Kode

Penjelasan Detail mengenai app.js

3.1 Koneksi ke MySQL

```
9   const connection = mysql.createConnection({
10     host: 'localhost',
11     user: 'root',
12     password: '',
13     database: 'pertemuan5'
14   });
15
```

Kode ini mendefinisikan koneksi ke server MySQL lokal dengan menggunakan paket `mysql2`. Koneksi dibuat dengan mengatur `host`, `user`, `password`, dan nama database (`pertemuan5`).

3.2 Pengaturan Middleware

```
5   const app = express();
6   app.use(bodyParser.urlencoded({extended : false}));
7   app.use(bodyParser.json());
8
```

Middleware `body-parser` digunakan untuk memproses data yang dikirimkan melalui form. `urlencoded` menangani data form yang dikirimkan dalam format URL, dan `json` menangani data JSON.

3.3 Routing CRUD

3.3.1 Read (Menampilkan Data)

```
28   //Read
29   app.get('/', (req, res) => {
30     const query = 'SELECT * FROM users';
31     connection.query(query, (err, results) => {
32       res.render('index', {users:results});
33     });
34   });
35
```

Endpoint ini menangani permintaan GET ke root URL (`/`). Kode ini menjalankan query SQL untuk mengambil semua data dari tabel `users` dan kemudian menampilkan hasilnya pada halaman `index.ejs`.

3.3.2 Create (Menambah Data)

```
37 //create / input / insert
38 v app.post('/add', (req, res) => {
39     const {nama, email, phone} = req.body;
40     const query = 'INSERT INTO users (nama, email, phone) VALUES (?, ?, ?)';
41 v     connection.query(query, [nama, email, phone], (err, result) => {
42         if (err) throw err;
43         res.redirect('/');
44     });
45 });
46
```

Kode ini menangani permintaan POST untuk menambah data baru. Ketika pengguna mengirimkan data melalui form, nama, email, dan nomor telepon dari body request diambil dan dimasukkan ke dalam tabel `users` menggunakan perintah SQL `INSERT`.

3.3.3 Update (Memperbarui Data)

- Menampilkan Form Update:

```
46
47 //update
48 //untuk akses halaman
49 app.get('/edit/:id', (req, res) => {
50     const query = 'SELECT * FROM users WHERE id = ?';
51     connection.query(query, [req.params.id], (err, result) => {
52         if (err) throw err;
53         res.render('edit', {user:result[0]});
54     });
55 })
56
```

Permintaan GET ke `/edit/:id` akan menampilkan data pengguna berdasarkan id yang dikirimkan di URL. Data ini kemudian akan ditampilkan pada form di halaman `edit.ejs`.

- Mengirim Data untuk Diperbarui:

```
56
57 //untuk update data
58 app.post('/update/:id', (req, res) =>{
59     const {nama, email, phone} = req.body;
60     const query = 'UPDATE users SET nama = ?, email = ?, phone = ? WHERE id = ?';
61     connection.query(query, [nama, email, phone, req.params.id], (err, result) =>{
62         if (err) throw err;
63         res.redirect('/');
64     });
65 })
66
```

Kode ini menangani permintaan POST untuk memperbarui data pengguna di database berdasarkan id yang dikirimkan. Query `UPDATE` akan mengubah data pengguna sesuai input baru yang dimasukkan ke dalam form.

3.3.4 Delete (Menghapus Data)

```
67 //hapus
68 app.get('/delete/:id', (req, res) => {
69   const query = 'DELETE FROM users WHERE id = ?';
70   connection.query(query, [req.params.id], (err, result) => {
71     if (err) throw err;
72     res.redirect('/');
73   });
74 })
75
```

Endpoint ini menghapus data dari tabel `users` berdasarkan id yang dikirimkan melalui URL. Query SQL `DELETE` digunakan untuk menghapus data tersebut.

3.4 Menjalankan Server

```
76 app.listen(3000, () => {
77   console.log("Server berjalan di prt 3000, buka web melalui http://localhost:3000 ")
78 });
```

Penjelasan Detail mengenai edit.ejs

```
views > edit.ejs > html
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 Data Pengguna</title>
7 </head>
8 <body>
9   <h2>Edit Data Pengguna</h2>
10  <form action="/update/<%= user.id %>" method="post">
11    <label for="nama">Nama:</label>
12    <input type="text" id="nama" name="nama" required value="<%= user.nama %>"><br>
13    <label for="email">Email:</label>
14    <input type="email" name="email" id="email" value="<%= user.email %>"><br>
15    <label for="phone">Telephon:</label>
16    <input type="text" name="phone" id="phone" value="<%= user.phone %>"><br>
17    <button type="submit">Tambah</button>
18  </form>
19
20 </body>
21 </html>
```

Bagian ini menampilkan judul "Edit Data Pengguna" dan mendefinisikan formulir dengan metode POST yang akan mengirimkan data ke endpoint di sini merujuk pada ID pengguna yang datanya akan diperbarui. Setiap field input menampilkan nilai pengguna yang sudah ada (nama, email, dan telepon) dari objek user. Nilai ini dirender dengan **EJS** (<%= user.nama %> dan seterusnya), sehingga saat halaman ditampilkan, field sudah terisi dengan data lama pengguna, siap untuk diperbarui. Tombol submit memungkinkan pengguna mengirimkan perubahan yang telah dilakukan di form ke server untuk diperbarui di database.

Penjelasan Detail mengenai style.css

```
foto > # style.css > ...
1  body {
2      font-family: Arial, sans-serif;
3      background-color: #f4f4f9;
4      background-image: url("bg1.jpeg");
5      background-size: cover;
6      background-position: center;
7      background-repeat: no-repeat;
8      color: #776070;
9      margin: 0;
10     padding: 20px;
11 }
12
13 h1, h2 {
14     color: #776070;
15     text-align: center;
16 }
17
18 table {
19     width: 100%;
20     border-collapse: collapse;
21     margin: 20px 0;
22 }
23
24 table th, table td {
25     border: 1px solid #ddd;
26     padding: 12px;
27     text-align: center;
28 }
```

```
30 table th {
31     background-color: #776070;
32     color: #fff;
33 }
34
35 table tr:nth-child(even) {
36     background-color: #f2f2f2;
37 }
38
39 table tr:hover {
40     background-color: #e9e9e9;
41 }
42
43 a {
44     text-decoration: none;
45     color: #776070;
46     font-weight: bold;
47 }
48
49 a:hover {
50     color: #776070;
51 }
52
53 form {
54     background-color: #fff;
55     padding: 20px;
56     margin: 20px auto;
57     border-radius: 8px;
58     box-shadow: 0 0 10px #776070;
59     max-width: 600px;
60 }
```

```

62 form label {
63     display: block;
64     margin-bottom: 8px;
65     font-weight: bold;
66 }
67
68 form input {
69     width: calc(100% - 22px);
70     padding: 10px;
71     margin-bottom: 12px;
72     border: 1px solid #ccc;
73     border-radius: 4px;
74     font-size: 14px;
75 }
76
77 form button {
78     background-color: #776070;
79     color: #fff;
80     border: none;
81     padding: 10px 20px;
82     border-radius: 4px;
83     font-size: 16px;
84     cursor: pointer;
85     transition: background-color 0.3s ease;
86 }
87
88 form button:hover {
89     background-color: #776070;
90 }
91

```

1. Body Styling

- Menggunakan font Arial atau sans-serif secara default.
- Background berupa gambar (bg1.jpeg) dengan penyesuaian penuh pada layar (cover), berpusat (center), dan tidak berulang (no-repeat).
- Warna teks utama adalah #776070 (abu-abu keunguan), dengan margin dan padding diatur.

2. Heading (h1, h2)

- Warna teks diatur sama seperti body (#776070) dan ditampilkan di tengah (center).

3. Table Styling

- Tabel menggunakan lebar penuh (100%), tanpa celah antar-baris.
- Menambahkan border pada setiap sel dan memberikan padding.
- Header tabel memiliki background #776070 dan teks berwarna putih.
- Baris genap memiliki warna background berbeda untuk memberikan efek zebra (#f2f2f2).
- Warna background baris berubah saat kursor diarahkan di atasnya.

4. Link Styling (a)

- Link tidak memiliki garis bawah (tidak terdekorasi) dan berwarna #776070 dengan ketebalan font yang lebih.
- Warna link tidak berubah saat hover.

5. Form Styling

- Form memiliki background putih, padding, margin tengah, sudut melengkung, dan bayangan lembut.
- Lebar nya maksimal 600px.
- Setiap label form diberi margin bawah dan font tebal.
- Input form dirancang lebar penuh dengan padding, margin bawah, dan border melengkung.

6. Button Styling

- Tombol memiliki background #776070, teks berwarna putih, tanpa border, dengan padding, sudut melengkung, dan transisi saat warna berubah.
- Warna tombol tetap saat hover.

Penjelasan Detail mengenai index.ejs

1. html

```
views > <> index.ejs > <html> <head> <script> <validateForm>
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">
```

- **<!DOCTYPE html>**: Menunjukkan bahwa dokumen ini menggunakan HTML5.
- **<html lang="en">**: Membuka tag HTML dan menentukan bahwa bahasa yang digunakan adalah bahasa Inggris.
- **<meta charset="UTF-8">**: Menetapkan karakter encoding dokumen menjadi UTF-8, yang mendukung banyak karakter internasional.
- **<meta name="viewport" content="width=device-width, initial-scale=1.0">**: Mengatur tampilan pada perangkat mobile agar responsif, dengan lebar viewport disesuaikan dengan lebar perangkat.
- **<title>Crud Node JS - MySQL</title>**: Judul halaman yang muncul di tab browser.
- **<link rel="stylesheet" href="style.css">**: Menghubungkan file CSS eksternal untuk menambahkan gaya pada halaman.
- **<script>**: Bagian ini berisi JavaScript yang mengatur perilaku interaktif dari halaman.

2. Penjelasan notifikasi nama,email,phone setelah di tambahkan.

```
10 // bagian notifikasi tampilan atas
11 const nama = document.getElementById('nama').value.trim();
12 const email = document.getElementById('email').value.trim();
13 const phone = document.getElementById('phone').value.trim();
14 const notification = document.getElementById('notification');
15
16 if (!nama && !email && !phone) {
17     notification.innerText = 'Please fill in at least one field to add a new user.';
18     notification.style.display = 'block';
19     return false;
20 }
21
22 let summaryMessage = 'Are you sure you want to add the following user?\n';
23 if (nama) summaryMessage += `Name: ${nama}\n`;
24 if (email) summaryMessage += `Email: ${email}\n`;
25 if (phone) summaryMessage += `Phone: ${phone}\n`;
26
27 const confirmAddition = confirm(summaryMessage);
28 if (!confirmAddition) {
29     notification.innerText = 'User addition cancelled.';
30     notification.style.display = 'block';
31     return false;
32 }
33
34 notification.style.display = 'none';
35 return true;
36
```

- **Pengambilan Nilai:** Mengambil nilai dari tiga input (nama, email, telepon) dan menghapus spasi di awal/akhir.
- **Validasi Input:** Memastikan setidaknya satu kolom diisi. Jika tidak, pesan kesalahan akan ditampilkan dan form tidak akan disubmit.
- **Menyusun Pesan Konfirmasi:** Mengumpulkan informasi yang diisi ke dalam pesan ringkasan.
- **Konfirmasi:** Menampilkan dialog konfirmasi. Jika pengguna memilih "OK", form akan disubmit; jika tidak, akan menampilkan pesan pembatalan.

3. Penjelasan notifikasi apabila ada data yang ingin di apus.

```
39 // bagian notifikasi hapus
40 function confirmDeletion(event, userName) {
41     event.preventDefault();
42     const notification = document.getElementById('notification');
43
44     const confirmDeletion = confirm(`Are you sure you want to delete the user: ${userName}?`);
45     if (confirmDeletion) {
46         notification.innerText = `User ${userName} has been deleted.`;
47         notification.style.display = 'block';
48         window.location.href = event.target.href;
49     } else {
50         notification.innerText = 'User deletion cancelled.';
51         notification.style.display = 'block';
52     }
53 }
54 </script>
55 </head>
```

- **Mencegah Aksi Default:** Menghentikan perilaku default dari link (navigasi ke URL).
- **Konfirmasi Penghapusan:** Menampilkan dialog konfirmasi untuk menghapus pengguna. Jika pengguna setuju, akan menampilkan pesan bahwa pengguna telah dihapus dan mengarahkan ke URL penghapusan. Jika tidak, menampilkan pesan pembatalan.

4. Judul Halaman dan Tabel Pengguna

```

8      <body>
9          <h1>Daftar User/Pengguna</h1>
10         <table border="1">
11             <tr>
12                 <th>ID</th>
13                 <th>Nama</th>
14                 <th>Email</th>
15                 <th>Telephon</th>
16                 <th>Aksi</th>
17             </tr>

```

Bagian ini membuat judul halaman "Daftar User/Pengguna" dan sebuah tabel HTML untuk menampilkan daftar pengguna. Baris pertama tabel (header) menggunakan tag <th> untuk menampilkan judul kolom seperti ID, Nama, Email, Telepon, dan Aksi.

5. Rendering Data Pengguna dengan EJS

```

18      <% users.forEach(Pengguna => {%)
19          <tr>
20              <td><%= Pengguna.id %></td>
21              <td><%= Pengguna.nama %></td>
22              <td><%= Pengguna.email %></td>
23              <td><%= Pengguna.phone %></td>
24              <td>
25                  <a href="/edit/<%= Pengguna.id %>">Edit</a>
26                  <a href="/delete/<%= Pengguna.id %>">Hapus</a>
27              </td>
28          </tr>
29      <% })%>
30  </table>

```

Bagian ini adalah kode EJS yang digunakan untuk menampilkan data pengguna yang berasal dari server.

- users.forEach adalah metode yang digunakan untuk melakukan iterasi (loop) pada array users yang berisi daftar pengguna yang didapat dari database.

- Setiap kali loop, variabel Pengguna akan merujuk ke satu data pengguna, dan kita bisa mengakses properti seperti id, nama, email, dan phone menggunakan sintaks EJS (<%= ... %>).
- Pada kolom "Aksi", terdapat dua tautan:
 - Edit: Mengarahkan ke halaman /edit/<%= Pengguna.id %> untuk mengedit data pengguna berdasarkan ID.
 - Hapus: Mengarahkan ke halaman /delete/<%= Pengguna.id %> untuk menghapus data pengguna berdasarkan ID.

6. Form Tambah Pengguna

```

31
32     <h2>Tambah Pengguna Baru </h2>
33     <form action="/add" method="post">
34         <label for="nama">Nama:</label>
35         <input type="text" id="nama" name="nama" required><br>
36         <label for="email">Email:</label>
37         <input type="email" name="email" id="email"><br>
38         <label for="phone">Telephon:</label>
39         <input type="text" name="phone" id="phone"><br>
40         <button type="submit">Tambah</button>
41     </form>
42
43 </body>
44 </html>

```

Bagian ini berfungsi untuk menampilkan form yang memungkinkan pengguna menambahkan data pengguna baru.

- Form ini menggunakan metode POST dan akan mengirimkan data ke endpoint /add.
- Form berisi tiga input yaitu nama, email dan telephon

Setiap input dilengkapi dengan label dan atribut name yang digunakan untuk mengirimkan data ke server. Setelah data diisi, pengguna bisa menekan tombol Tambah untuk menambahkan data ke database.

Ini beberapa contoh dari data table sebelum di tambah

Properties DATA Diagram Monitor Log

SELECT * FROM users LIMIT 100

Search results 1 Export Cost: 15ms < 1 > Total 9

	* id int(11)	nama varchar(100)	email varchar(100)	phone varchar(15)
	Filter	Filter	Filter	Filter
>	1	Syifa	syifafauzia04@upi.edu	081322447364
>	2	Novia	Noviaramadhani@upi.edu	081322443355
>	3	Safira	Safira2345@upi.edu	08122233444
>	4	Sarah	Sarahalfira@upi.edu	081234567890
>	5	Aini	Nuraini@upi.edu	08132345678
>	6	Dinda	Dindaputri12@upi.edu	08987765467
>	7	Nadia	Nadiafarah@upi.edu	08987765467
>	8	Nabila	Nabila45612@upi.edu	081211223344
>	9	Ale	Alya3751@upi.edu	08234567232

Dan ini table setelah data di tambah di web

Properties DATA Diagram Monitor Log

SELECT * FROM users LIMIT 100

Search results 1 Export Cost: 14ms < 1 > Total 10

	* id int(11)	nama varchar(100)	email varchar(100)	phone varchar(15)
	Filter	Filter	Filter	Filter
>	1	Syifa	syifafauzia04@upi.edu	081322447364
>	2	Novia	Noviaramadhani@upi.edu	081322443355
>	3	Safira	Safira2345@upi.edu	08122233444
>	4	Sarah	Sarahalfira@upi.edu	081234567890
>	5	Aini	Nuraini@upi.edu	08132345678
>	6	Dinda	Dindaputri12@upi.edu	08987765467
>	7	Nadia	Nadiafarah@upi.edu	08987765467
>	8	Nabila	Nabila45612@upi.edu	081211223344
>	9	Ale	Alya3751@upi.edu	08234567232
>	10	Yasmin	Yasmin3764@upi.edu	0813456246

Hasil akhir sebelum di tambahkan data di web

localhost:3000

Daftar User/Pengguna

ID	Nama	Email	Telephon	Aksi
1	Syifa	syifafauzia04@upi.edu	081322447364	Edit Hapus
2	Novia	Noviaramadhani@upi.edu	081322443355	Edit Hapus
3	Safira	Safira2345@upi.edu	08122233444	Edit Hapus
4	Sarah	Sarahalfira@upi.edu	081234567890	Edit Hapus
5	Aini	Nuraini@upi.edu	08132345678	Edit Hapus
6	Dinda	Dindaputri12@upi.edu	08987765467	Edit Hapus
7	Nadia	Nadiafarah@upi.edu	08987765467	Edit Hapus
8	Nabila	Nabila45612@upi.edu	081211223344	Edit Hapus
9	Ale	Alya3751@upi.edu	08234567232	Edit Hapus

Tambah Pengguna Baru

Nama:

Email:

Telephon:

Tambah

Hasil mencoba menambahkan data di web

(5) WhatsApp x Crud Node JS - MySQL x +

localhost:3000

5	Aini	Nuraini@upi.edu	08132345678	Edit Hapus
6	Dinda	Dindaputri12@upi.edu	08987765467	Edit Hapus
7	Nadia	Nadiafarah@upi.edu	08987765467	Edit Hapus
8	Nabila	Nabila45612@upi.edu	081211223344	Edit Hapus
9	Ale	Alya3751@upi.edu	08234567232	Edit Hapus

Tambah Pengguna Baru

Nama:

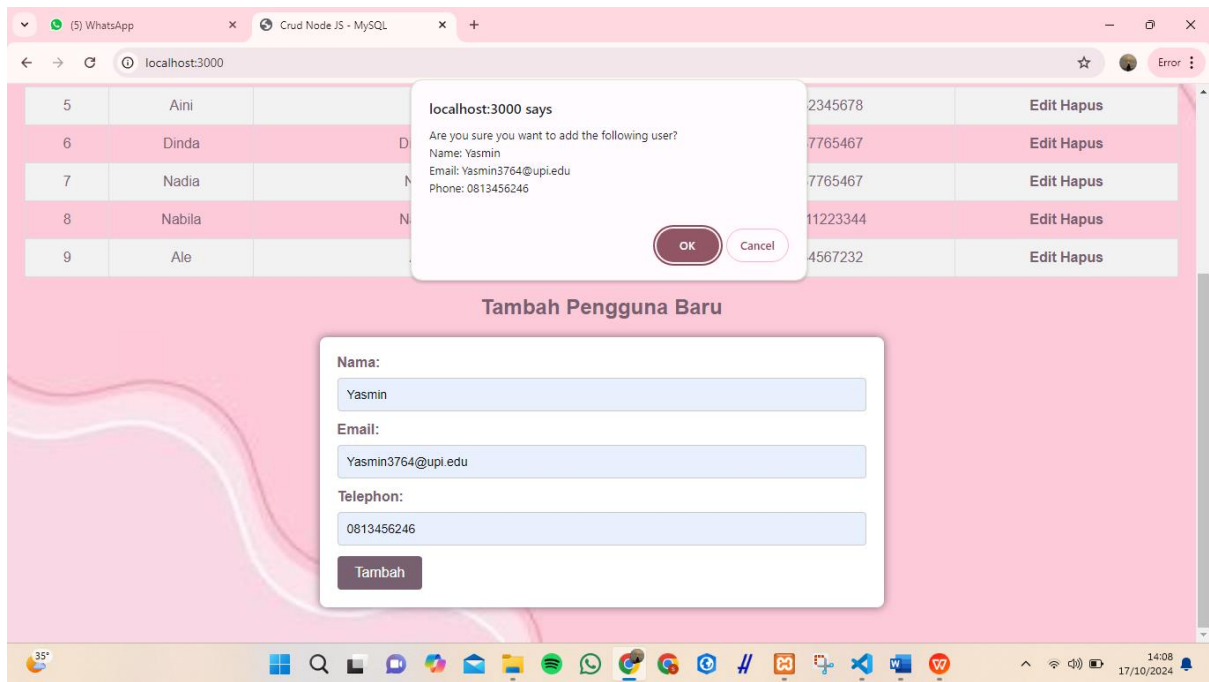
Email:

Telephon:

Tambah

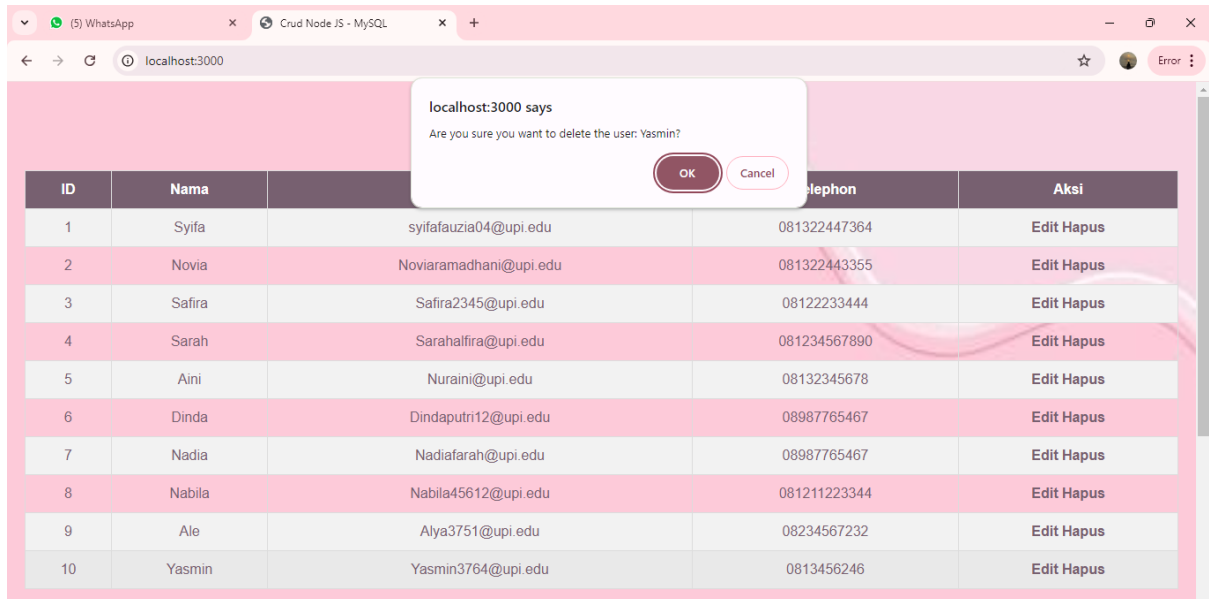
12345678987
081211223344
08987765467
081234567890
08987764356
0813456246 x

35° 14:07 17/10/2024



Hasil mencoba menghapus data di web

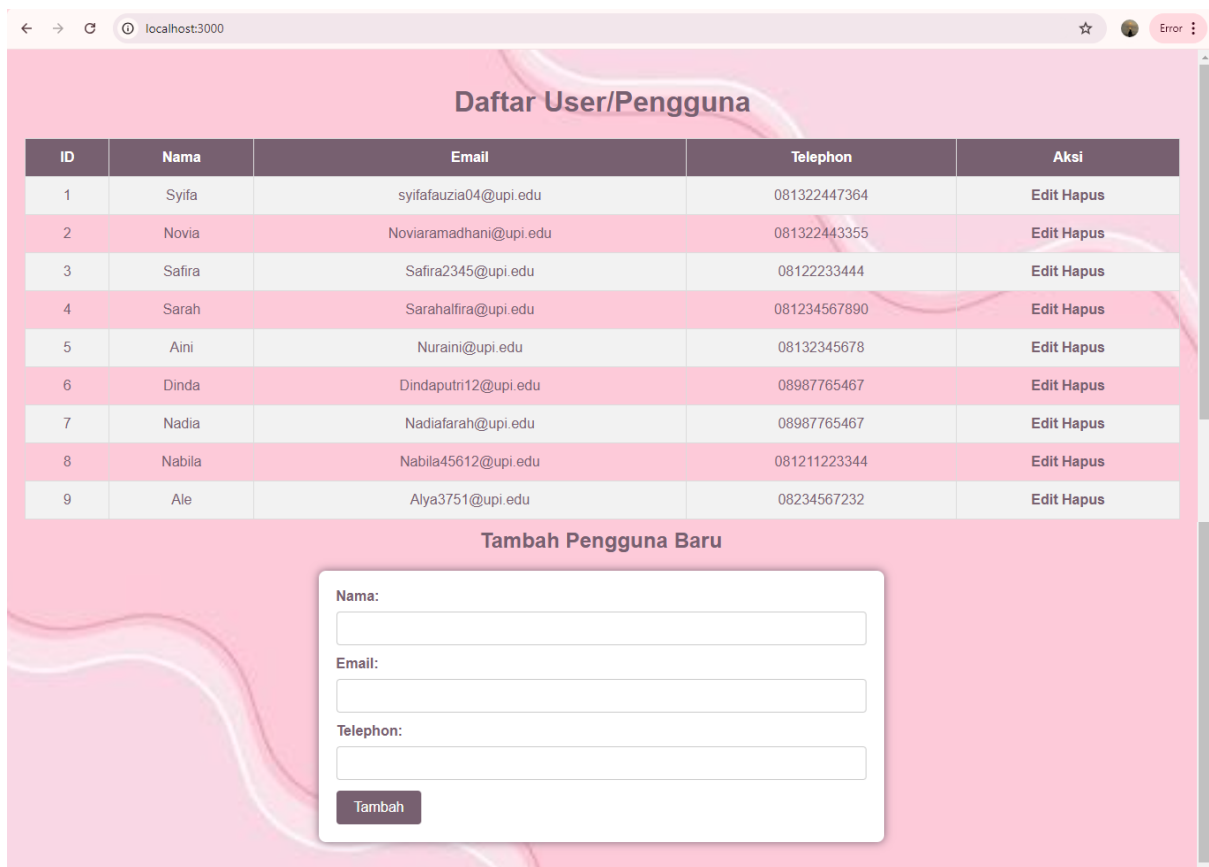
- Ingin menghapus data Yasmin yang sebelumnya di tambahkan.



localhost:3000 says
Are you sure you want to delete the user: Yasmin?

ID	Nama	Email	Telephon	Aksi
1	Syifa	syifafauzia04@upi.edu	081322447364	Edit Hapus
2	Novia	Noviaramadhani@upi.edu	081322443355	Edit Hapus
3	Safira	Safira2345@upi.edu	08122233444	Edit Hapus
4	Sarah	Sarahalfira@upi.edu	081234567890	Edit Hapus
5	Aini	Nuraini@upi.edu	08132345678	Edit Hapus
6	Dinda	Dindaputri12@upi.edu	08987765467	Edit Hapus
7	Nadia	Nadiafarah@upi.edu	08987765467	Edit Hapus
8	Nabila	Nabila45612@upi.edu	081211223344	Edit Hapus
9	Ale	Alya3751@upi.edu	08234567232	Edit Hapus
10	Yasmin	Yasmin3764@upi.edu	0813456246	Edit Hapus

Hasil akhir setelah di hapus



Daftar User/Pengguna

ID	Nama	Email	Telephon	Aksi
1	Syifa	syifafauzia04@upi.edu	081322447364	Edit Hapus
2	Novia	Noviaramadhani@upi.edu	081322443355	Edit Hapus
3	Safira	Safira2345@upi.edu	08122233444	Edit Hapus
4	Sarah	Sarahalfira@upi.edu	081234567890	Edit Hapus
5	Aini	Nuraini@upi.edu	08132345678	Edit Hapus
6	Dinda	Dindaputri12@upi.edu	08987765467	Edit Hapus
7	Nadia	Nadiafarah@upi.edu	08987765467	Edit Hapus
8	Nabila	Nabila45612@upi.edu	081211223344	Edit Hapus
9	Ale	Alya3751@upi.edu	08234567232	Edit Hapus

Tambah Pengguna Baru

Nama:

Email:

Telephon:

4. Kesimpulan

Form HTML yang tersedia untuk input dan pembaruan data memberikan pengalaman pengguna yang intuitif dalam mengelola informasi. Operasi CRUD dalam aplikasi ini berjalan sesuai harapan, mulai dari pengambilan data, penyimpanan, hingga pembaruan dan penghapusan data. Secara keseluruhan, kombinasi teknologi yang digunakan menciptakan aplikasi yang sederhana namun efektif untuk mengelola data, dan dapat menjadi dasar untuk pengembangan aplikasi yang lebih kompleks di masa depan.