

LAPORAN PRAKTIKUM
PEMROGRAMAN WEB



RIYADI TRI WALUYA BUDI

2308065

SIK A3

UNIVERSITAS PENDIDIKAN INDONESIA
SISTEM INFORMASI KELAUTAN
2024

I.PENDAHULUAN

Di era digital saat ini, pengembangan aplikasi web telah menjadi keterampilan yang sangat diperlukan.

Aspek penting dalam pengembangan aplikasi adalah kemampuan untuk melakukan operasi dasar pada data yang dikenal dengan istilah CRUD (Create, Read, Update, Delete).

Praktik tersebut bertujuan untuk memberikan pemahaman mendalam mengenai implementasi CRUD dengan menggunakan Node.js sebagai backend dan MySQL sebagai database.

Node.js adalah platform yang memungkinkan pengembang menjalankan JavaScript di sisi server, memberikan kecepatan dan efisiensi dalam pemrosesan data.

Di sisi lain, MySQL adalah sistem manajemen basis data relasional yang banyak digunakan karena stabilitas dan kemudahan penggunaannya.

Tujuan Praktik

1. Memahami konsep dasar CRUD.
2. Pelajari cara menghubungkan Node.js ke MySQL.
3. Membuat aplikasi sederhana yang melakukan operasi CRUD pada data.

Cakupan Praktik ini akan mencakup:

- Menginstal dan mengkonfigurasi Node.js dan MySQL.
- Membuat database dan tabel di MySQL.
- Melakukan operasi buat, baca, perbarui, dan hapus dengan Node.js.
- Pengujian aplikasi dan penanganan kesalahan.

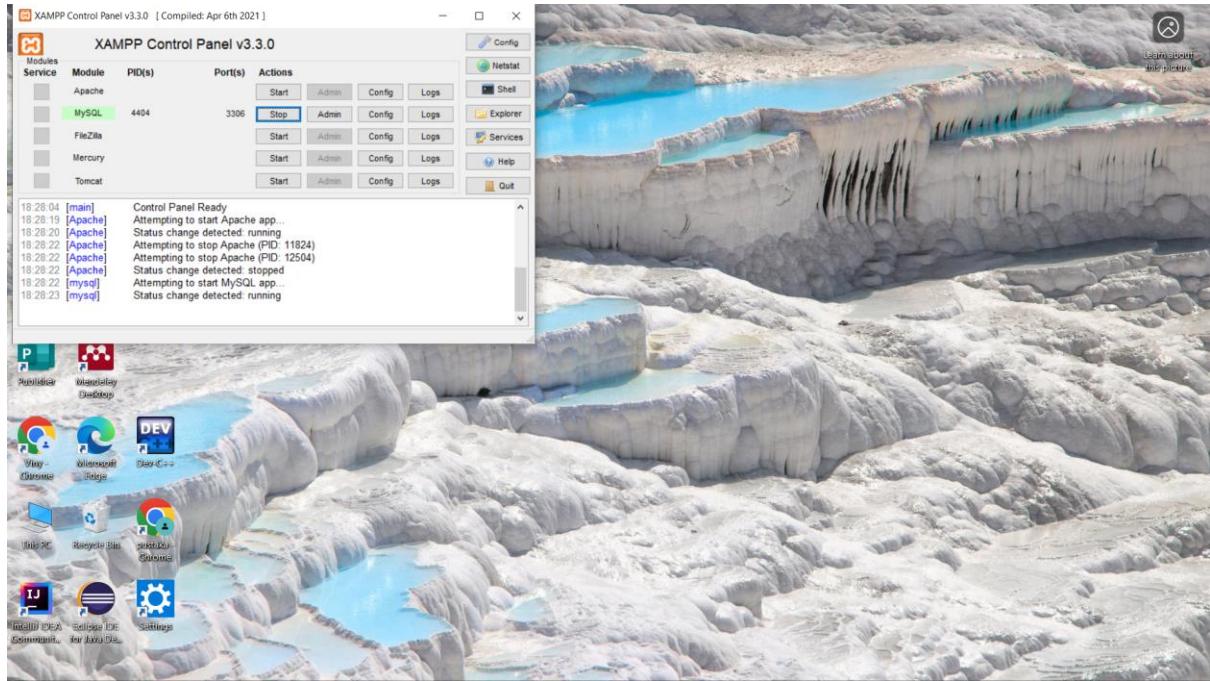
Dengan memahami dan menguasai praktik-praktik tersebut, peserta akan mampu mengembangkan aplikasi web yang interaktif dan responsif serta mampu mengelola data secara efektif.

II.ALAT DAN BAHAN

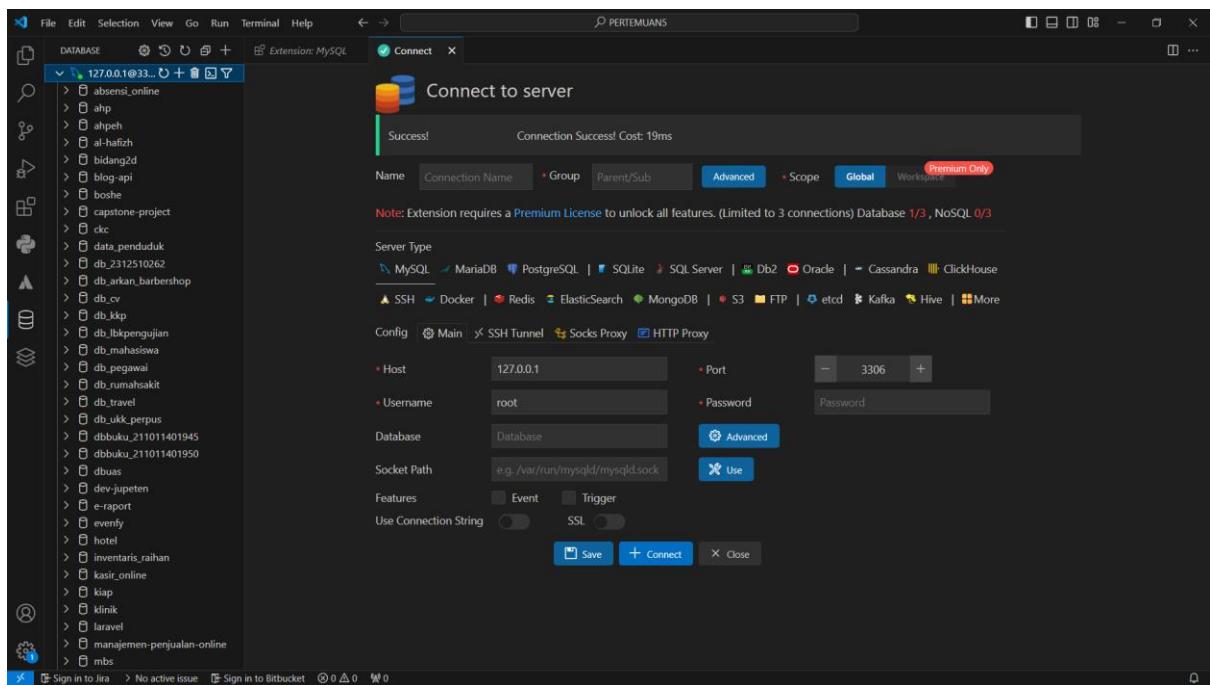
- VScode
- Node.js
- MySQL

III.LANGKAH – LANGKAH

1.Pertama, Buka Aplikasi XAMPP Kemudian Aktifkan MySQL nya



2. Kemudian buka ekstensi MySQL pada VSCode, kemudian connect pada database.



3. Kemudian tambahkan database dengan nama pertemuan5

The screenshot shows the MySQL Workbench interface. On the left, the 'DATABASE' tree view lists various databases, including 'absensi_online', 'ahp', 'al-hafizh', 'bidang2d', 'blog-api', 'boshe', 'capstone-project', 'ckc', 'data_penduduk', 'db_2312510262', 'db_arkan_barbershop', 'db_cv', 'db_kkp', 'db_lbkgungjilan', 'db_mahasiswa', 'db_pegawai', 'db_rumahsakit', 'db_travel', 'db_ukk_perpus', 'dbbuku_211011401945', 'dbbuku_211011401950', 'dbwas', 'dev-jupeten', 'e-raport', 'evenfy', 'hotel', 'inventaris_raihan', 'kasir_online', 'kiap', 'klinik', 'laravel', 'manajemen-penjualan-online', and 'mbs'. A connection to '127.0.0.1:3306' is active. In the center, the SQL editor window contains the following code:

```
1 -- Active: 1728646342594@127.0.0.1@3306 MySQL
2 CREATE DATABASE pertemuan5
3 | DEFAULT CHARACTER SET = 'utf8mb4';
```

4. Tambahkan tabel users

The screenshot shows the MySQL Workbench interface. The 'DATABASE' tree view now includes 'pertemuan5' under '127.0.0.1:3306'. The SQL editor window contains the following code:

```
1 -- Active: 1728646342594@127.0.0.1@3306@pertemuan5 MySQL
2 CREATE TABLE users(
3 | id INT AUTO_INCREMENT PRIMARY KEY,
4 | name VARCHAR(100),
5 | email VARCHAR(100),
6 | phone VARCHAR(15)
7 );
```

5. Tambahkan data dummy pada tabel users

The screenshot shows the phpMyAdmin interface connected to a MySQL database named 'PERTEMUANS'. The 'users' table is selected, displaying the following data:

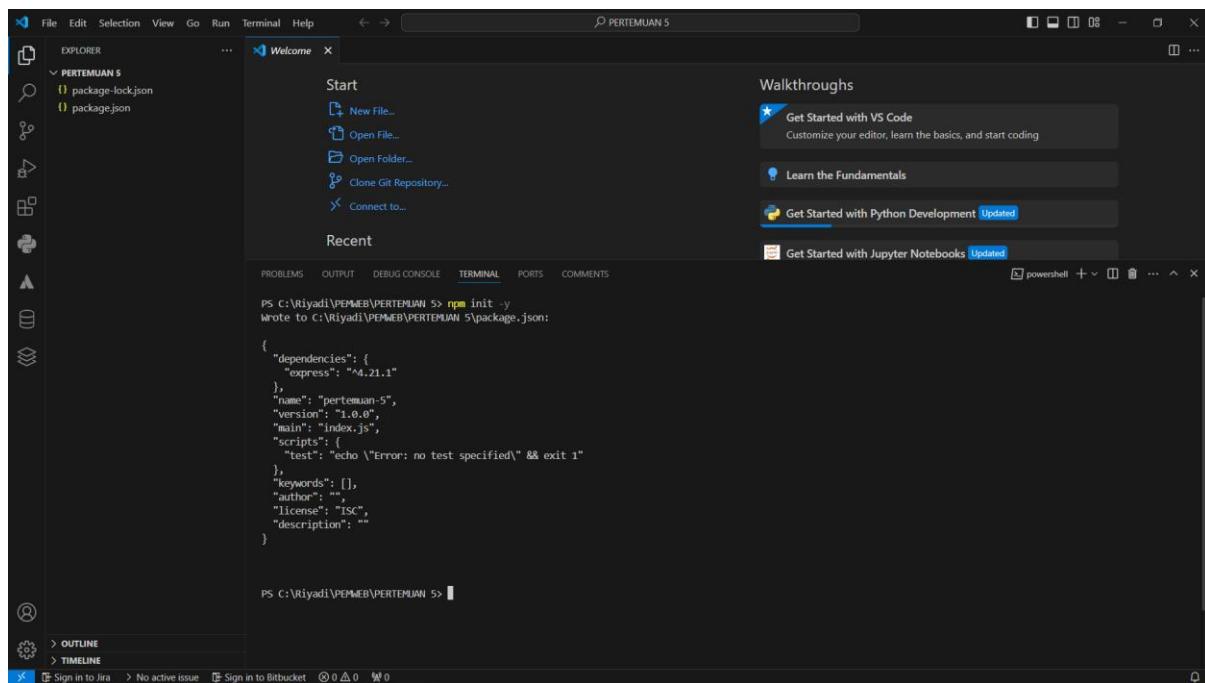
	<code>* id</code>	<code>name</code>	<code>email</code>	<code>phone</code>
1	Wildan	wildanarifin@upi.edu	123456789	
2	Habibirohim	habibirohim@upi.edu	234567890	
3	Bintang	bintang@upi.edu	345678901	
4	Angga	angga@upi.edu	456789012	

6. Kemudian, buka VScode klik open folder. Kemudian cari folder PEMWEB lalu klik dan di dalamnya Membuat Folder lagi dengan nama PERTEMUAN 5

The screenshot shows the Visual Studio Code interface with the title bar 'PERTEMUAN 5'. The left sidebar shows a folder structure under 'EXPLORER' with 'PERTEMUAN 5' expanded. The main area displays the 'Welcome' screen with the following content:

- Welcome**: A large heading 'Visual Studio Code' followed by 'Editing evolved'.
- Start**: A list of actions:
 - New File...
 - Open File...
 - Open Folder...
 - Clone Git Repository...
 - Connect to...
- Recent**: A list of recent projects:
 - riyadi-web C:\Users\Viny\Downloads
 - ukk_perpus_sekolah C:\xampp\htdocs
 - mahasiswa C:\xampp\htdocs
 - RPL C:\xampp\htdocs
 - latihan C:\xampp\htdocs
- Walkthroughs**: A section with three cards:
 - Get Started with VS Code**: Customize your editor, learn the basics, and start coding
 - Learn the Fundamentals**
 - Get Started with Python Development** [Updated]
 - Get Started with Jupyter Notebooks** [Updated]
- Bottom Status Bar**: Shows 'Show welcome page on startup' checked.

7. Buka terminal, dan inisiasi npm untuk node.js

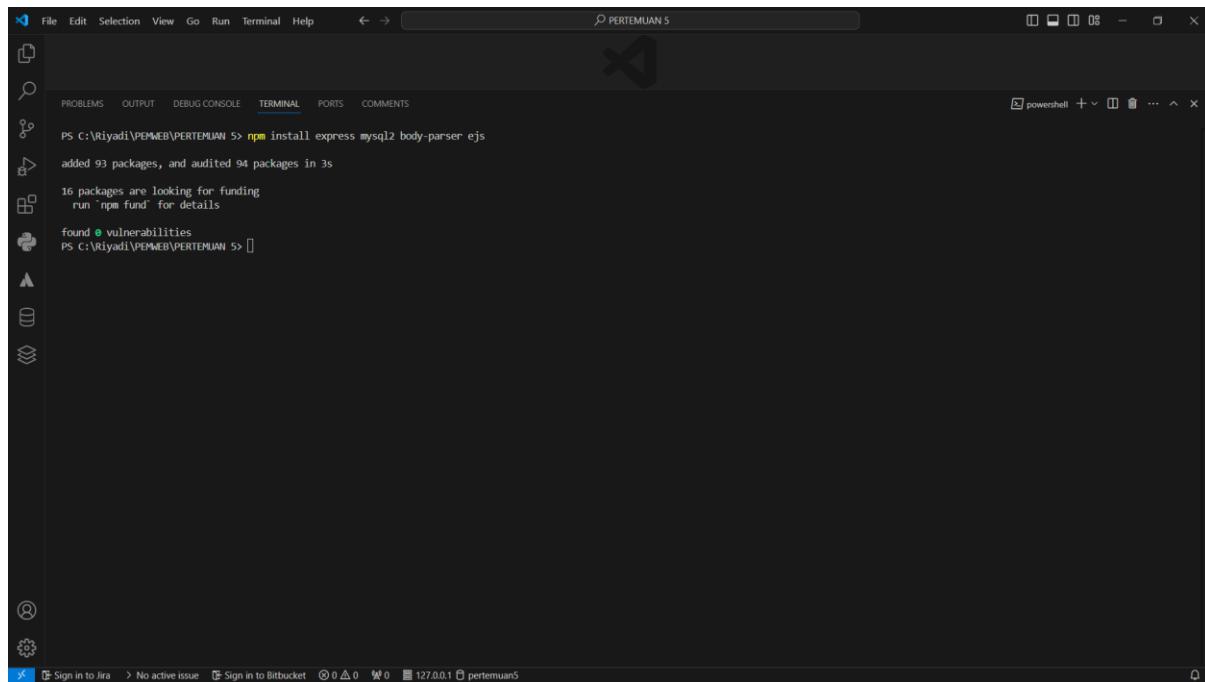


A screenshot of the Visual Studio Code interface. The left sidebar shows an 'EXPLORER' view with a folder named 'PERTEMUAN 5' containing 'package-lock.json' and 'package.json'. The main area is titled 'Welcome' and contains a 'Walkthroughs' section with links like 'Get Started with VS Code' and 'Learn the Fundamentals'. Below that is a 'Recent' section. At the bottom, there's a terminal window with the following content:

```
PS C:\Riyadi\PEMWEB\PERTEMUAN 5> npm init -y
Write to C:\Riyadi\PEMWEB\PERTEMUAN 5\package.json:
{
  "dependencies": {
    "express": "^4.21.1"
  },
  "name": "pertemuan-5",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \'Error: no test specified\' && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": ""
}

PS C:\Riyadi\PEMWEB\PERTEMUAN 5>
```

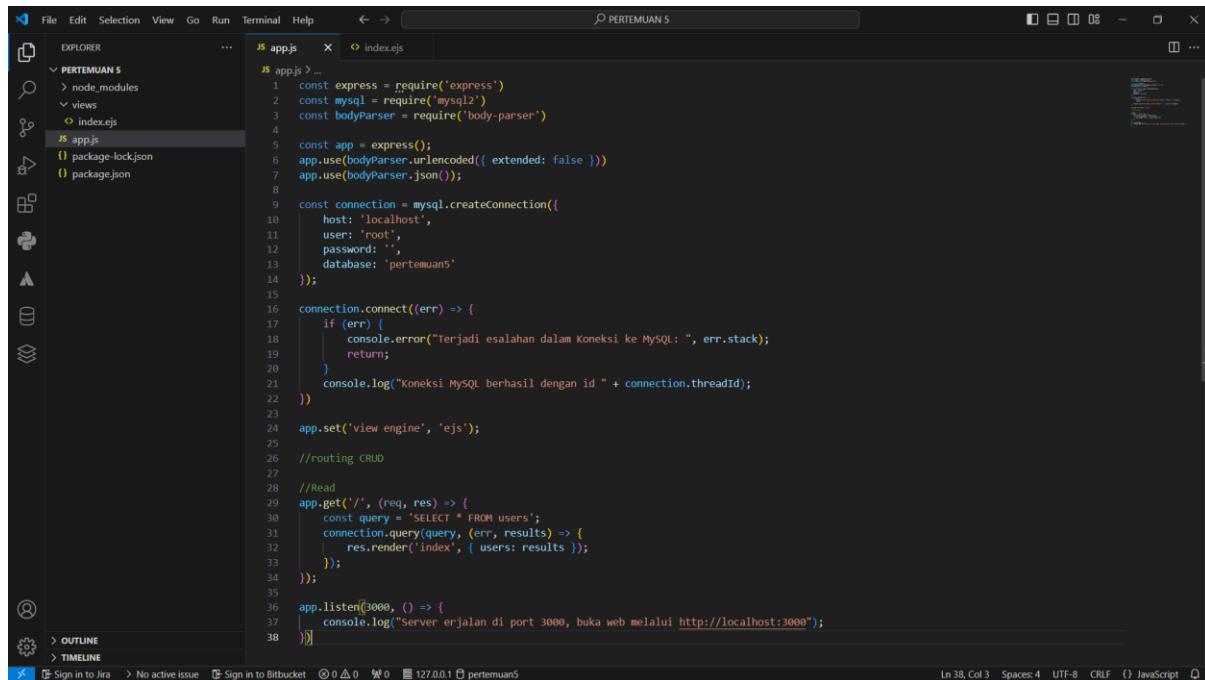
8. Install express.js, mysql2, body-parser, dan ejs



A screenshot of the Visual Studio Code interface. The left sidebar shows an 'EXPLORER' view with a folder named 'PERTEMUAN 5' containing 'package-lock.json' and 'package.json'. The main area is titled 'Welcome' and contains a 'Walkthroughs' section with links like 'Get Started with VS Code' and 'Learn the Fundamentals'. Below that is a 'Recent' section. At the bottom, there's a terminal window with the following content:

```
PS C:\Riyadi\PEMWEB\PERTEMUAN 5> npm install express mysql2 body-parser ejs
added 93 packages, and audited 94 packages in 3s
16 packages are looking for funding
  run 'npm fund' for details
found 0 vulnerabilities
PS C:\Riyadi\PEMWEB\PERTEMUAN 5>
```

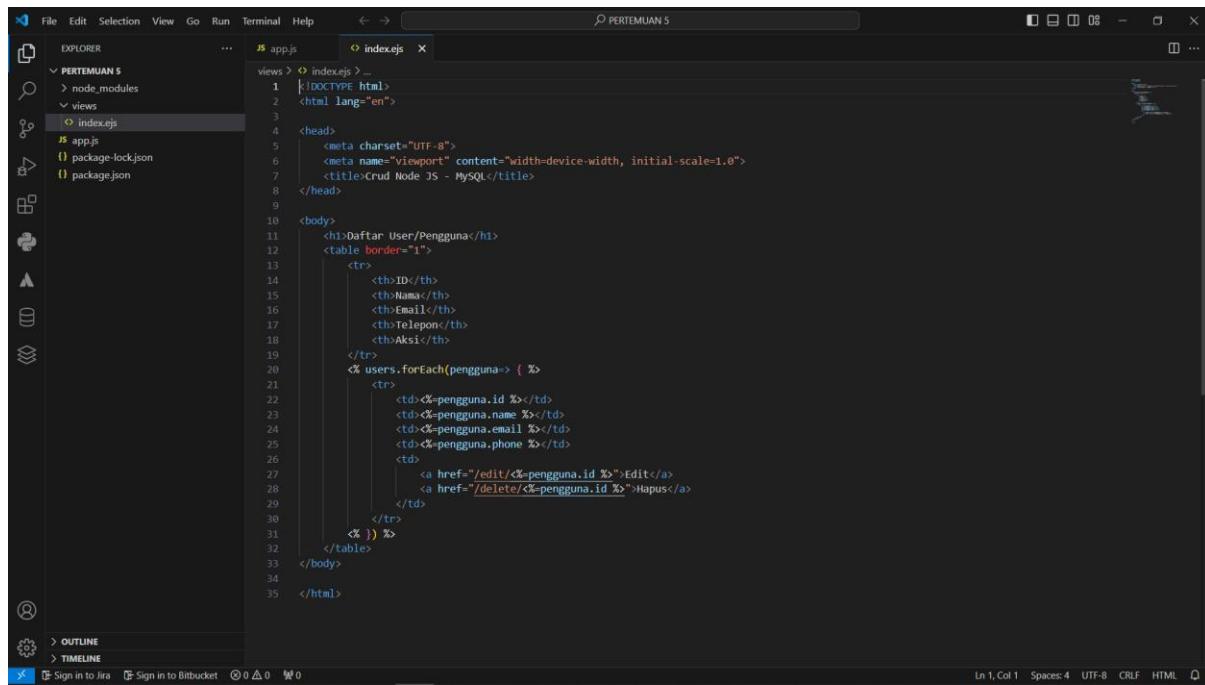
9. Buat file app.js dan isi codenya sebagai route



```
File Edit Selection View Go Run Terminal Help <- > PERTEMUAN 5
EXPLORER app.js index.js
PERTEMUAN 5
node_modules
views
index.js
app.js
package-lock.json
package.json
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
9 const connection = mysql.createConnection({
10   host: "localhost",
11   user: "root",
12   password: "",
13   database: 'pertemuan5'
14 });
15
16 connection.connect((err) => {
17   if (err) {
18     console.error("Terjadi kesalahan dalam koneksi ke MySQL: ", err.stack);
19     return;
20   }
21   console.log("Koneksi MySQL berhasil dengan id " + connection.threadId);
22 })
23
24 app.set('view engine', 'ejs');
25
26 //routing CRUD
27
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
36 app.listen(3000, () => {
37   console.log("Server berjalan di port 3000, buka web melalui http://localhost:3000");
38 })
```

Sign in to Jira Sign in to Bitbucket 0 △ 0 127.0.0.1 pertemuan5 In 38, Col 3 Spaces: 4 UTF-8 CRLF JavaScript

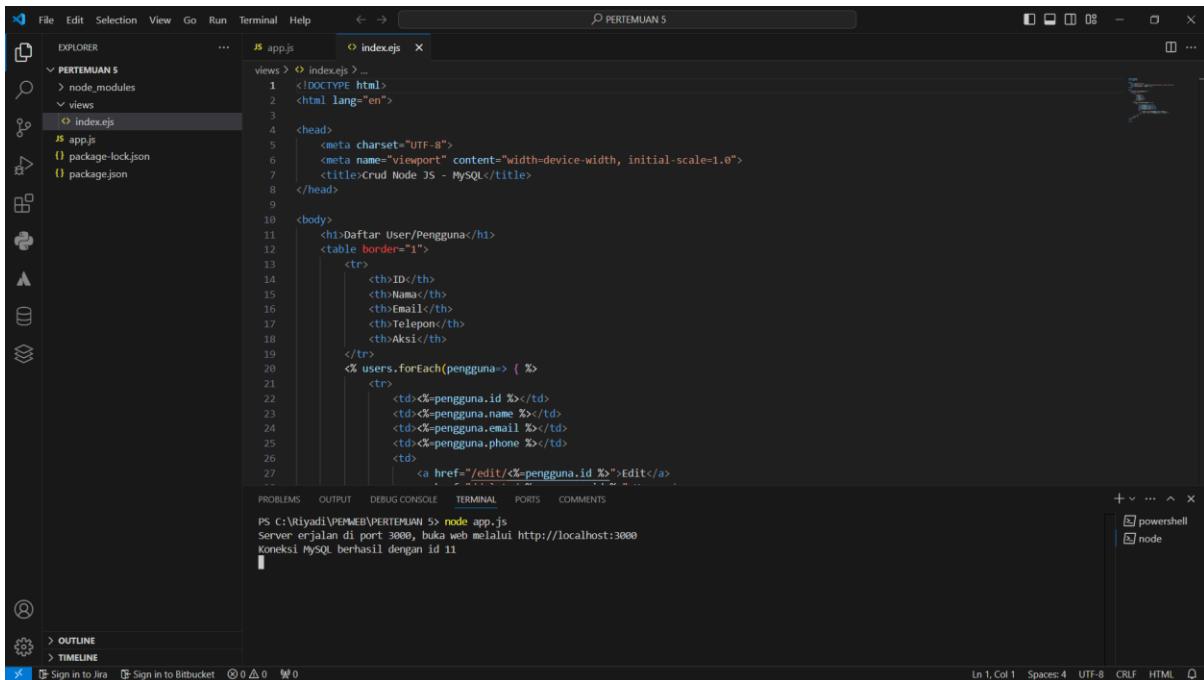
10. Tambahkan folder views dan tambahkan file index.js dengan tabel untuk menampilkan data users



```
File Edit Selection View Go Run Terminal Help <- > PERTEMUAN 5
EXPLORER views index.ejs
PERTEMUAN 5
node_modules
views
index.ejs
app.js
package-lock.json
package.json
views > index.ejs > ...
1 <!DOCTYPE html>
2 <html lang="en">
3
4   <head>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Crud Node JS - MySQL</title>
8   </head>
9
10  <body>
11    <h1>Daftar User/Pengguna</h1>
12    <table border="1">
13      <tr>
14        <th>ID</th>
15        <th>Nama</th>
16        <th>Email</th>
17        <th>Telepon</th>
18        <th>Aksi</th>
19      </tr>
20      <% users.forEach(pengguna) => { %>
21        <tr>
22          <td><%= pengguna.id %></td>
23          <td><%= pengguna.name %></td>
24          <td><%= pengguna.email %></td>
25          <td><%= pengguna.phone %></td>
26          <td>
27            <a href="/edit/<%= pengguna.id %>">Edit</a>
28            <a href="/delete/<%= pengguna.id %>">Hapus</a>
29          </td>
30        </tr>
31      <% } %>
32    </table>
33  </body>
34  </html>
```

Sign in to Jira Sign in to Bitbucket 0 △ 0 127.0.0.1 pertemuan5 In 1, Col 1 Spaces: 4 UTF-8 CRLF HTML

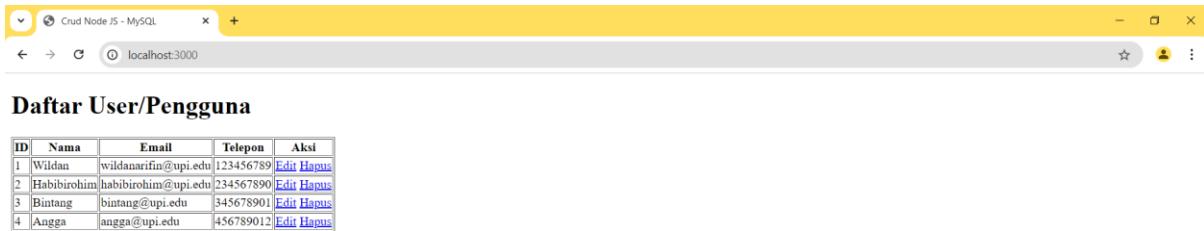
11. Jalankan node app.js



```
<!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>
</head>
<body>
    <h1>Daftar User/Pengguna</h1>
    <table border="1">
        <tr>
            <th>ID</th>
            <th>Nama</th>
            <th>Email</th>
            <th>Telepon</th>
            <th>Aksi</th>
        </tr>
        <% 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 %>">Edit</a>
                    <a href="/hapus/<% pengguna.id %>">Hapus</a>
                </td>
            </tr>
        <% } %>
    </table>
</body>
```

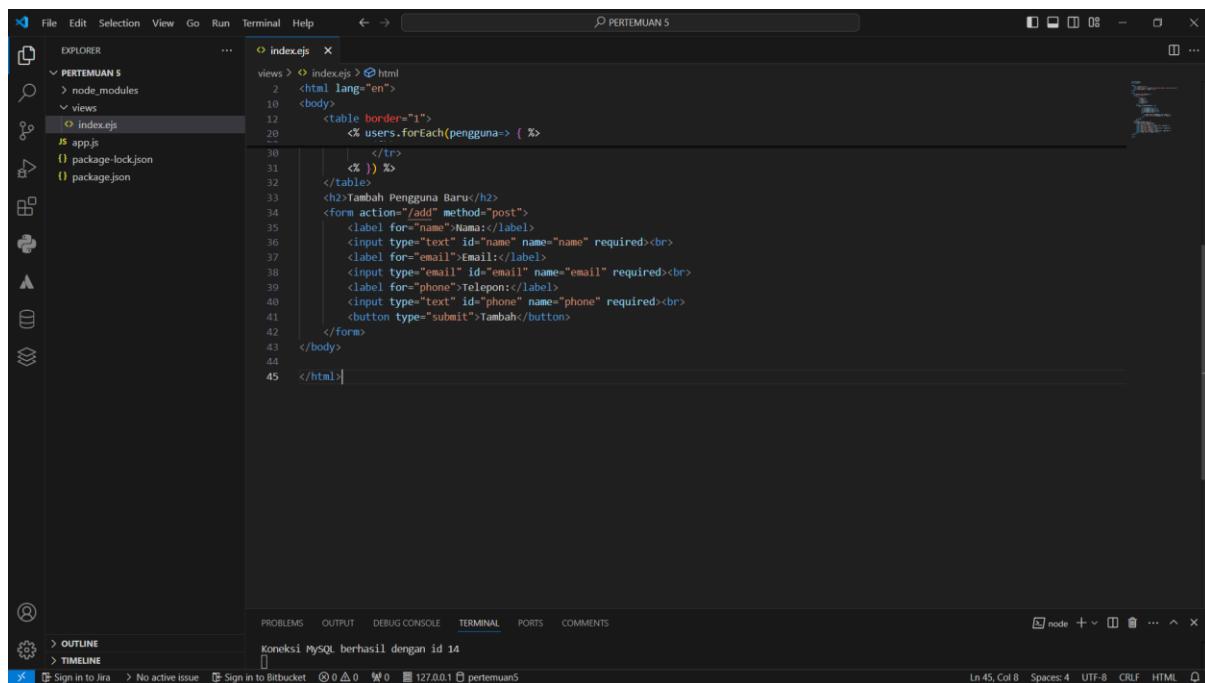
PS C:\Riyadi\PERMENB\PERTEMUAN 5> node app.js
Server berjalan di port 3000, buka web melalui http://localhost:3000
koneksi MySQL berhasil dengan id 11

12. Tampilan akan menjadi seperti ini



ID	Nama	Email	Telepon	Aksi
1	Wildan	wildanarfin@upi.edu	123456789	Edit Hapus
2	Habibirohim	habibirohim@upi.edu	234567890	Edit Hapus
3	Bintang	bintang@upi.edu	345678901	Edit Hapus
4	Angga	angga@upi.edu	456789012	Edit Hapus

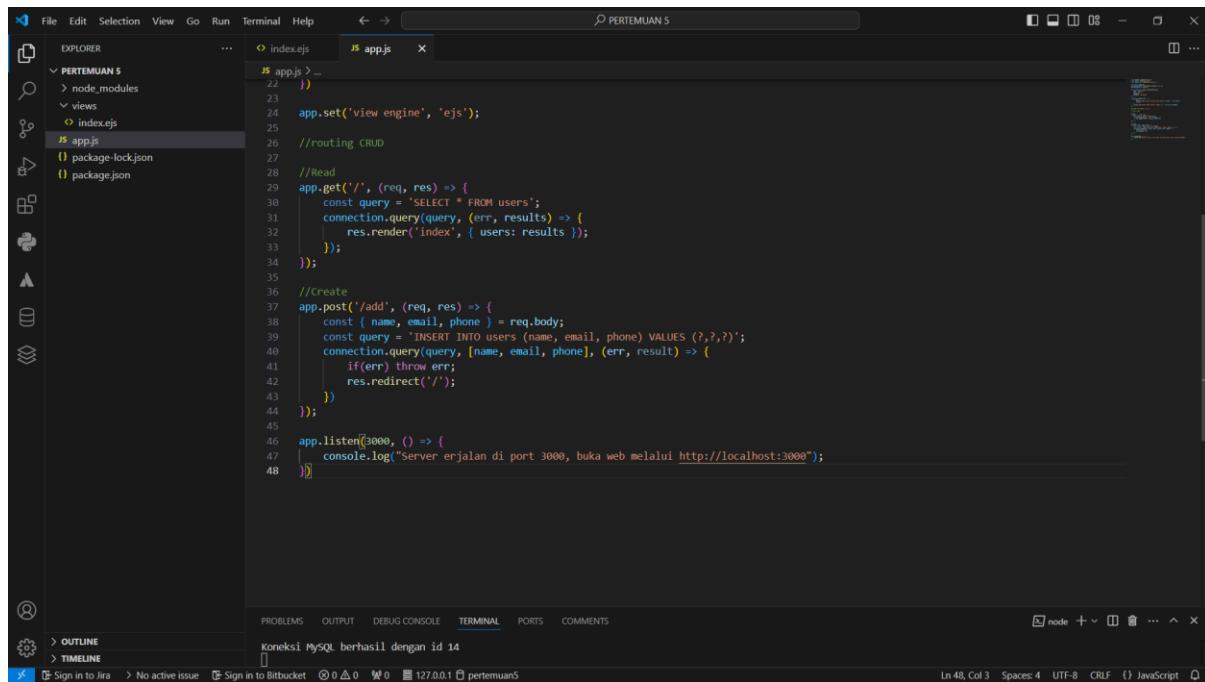
13. Tambahkan form untuk menambah user/pengguna



The screenshot shows the VS Code interface with the title bar "PERTEMUAN 5". The Explorer sidebar shows files like "node_modules", "views", and "index.ejs". The "index.ejs" file is open in the editor, displaying an HTML template with a table and a form for adding a new user. The code includes labels and inputs for name, email, and phone, and a submit button. The bottom status bar shows "Koneksi MySQL berhasil dengan id 14" and the file path "pertemuan5/index.ejs".

```
<html lang="en">
  <body>
    <table border="1">
      <% users.forEach(pengguna) { %>
        <tr>
          <td>${pengguna.name}</td>
          <td>${pengguna.email}</td>
          <td>${pengguna.phone}</td>
        </tr>
      <% } %>
    </table>
    <h2>Tambah Pengguna Baru</h2>
    <form action="/add" method="post">
      <label for="name">Nama:</label>
      <input type="text" id="name" name="name" required><br>
      <label for="email">Email:</label>
      <input type="email" id="email" name="email" required><br>
      <label for="telepon">Telepon:</label>
      <input type="text" id="phone" name="phone" required><br>
      <button type="submit">Tambah</button>
    </form>
  </body>
</html>
```

14. Tambahkan post dengan routing '/add' agar dapat menyimpan data dari form



The screenshot shows the VS Code interface with the title bar "PERTEMUAN 5". The Explorer sidebar shows files like "node_modules", "views", and "app.js". The "app.js" file is open in the editor, displaying Node.js code. It includes routes for reading users ('/'), creating users ('/add'), and listening on port 3000. The "POST /add" route uses a query to insert data into the "users" table. The bottom status bar shows "Koneksi MySQL berhasil dengan id 14" and the file path "pertemuan5/app.js".

```
const express = require('express');
const connection = require('./connection');
const app = express();

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

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

//Create
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('/');
  });
});

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

15. Form akan seperti berikut

The screenshot shows a web application titled "Crud Node JS - MySQL" running on localhost:3000. At the top, there is a table titled "Daftar User/Pengguna" with columns: ID, Nama, Email, Telepon, and Aksi. The data in the table is:

ID	Nama	Email	Telepon	Aksi
1	Wildan	wildanarfin@upi.edu	123456789	Edit Hapus
2	Habibirohim	habibirohim@upi.edu	234567890	Edit Hapus
3	Bintang	bintang@upi.edu	345678901	Edit Hapus
4	Angga	angga@upi.edu	456789012	Edit Hapus

Below the table, there is a section titled "Tambah Pengguna Baru" containing input fields for Name, Email, and Phone number, followed by a "Tambah" button.

16. Bila ditambahkan akan seperti berikut

The screenshot shows the same web application after a new user has been added. The "Daftar User/Pengguna" table now includes an additional row:

ID	Nama	Email	Telepon	Aksi
1	Wildan	wildanarfin@upi.edu	123456789	Edit Hapus
2	Habibirohim	habibirohim@upi.edu	234567890	Edit Hapus
3	Bintang	bintang@upi.edu	345678901	Edit Hapus
4	Angga	angga@upi.edu	456789012	Edit Hapus
5	Riyadi	riyaditri12@gmail.com	082134451235	Edit Hapus

The "Tambah Pengguna Baru" section remains the same, allowing for the addition of new users.

17. Buat routing untuk edit

```
File Edit Selection View Go Run Terminal Help < - > ⚡ PERTEMUAN 5
DATABASE 127.0.0.1@3006 10.4.25-MariaDB
  db_mahasiswa
  db_pegawai
  db_rumahsakit
  db_travel
  db_ukr_perpus
  dbbuku_211011401945
  dbbuku_211011401950
  dbus
  dev-jupeten
  e-raport
  evenfy
  hotel
  inventaris_raihan
  kasir_online
  klap
  klinik
  laravel
  manajemen-penjualan-online
  mbs
  metronic
  pendaftaran_pelatihan
  pertemuan5 16k
    Query
    Tables (1)
      users 5
      Views
      Functions
      Procedures
      phpmyadmin
      pokdarwis
      rental_eria
      resepdapurkuu
      shop
      siam_aulia
js app.js > ...
37   app.post('/add', (req, res) => {
38     const { name, email, phone } = req.body;
39     const query = 'INSERT INTO users (name, email, phone) VALUES (?, ?, ?)';
40     connection.query(query, [name, email, phone], (err, result) => {
41       if (err) throw err;
42       res.redirect('/');
43     });
44   });
45
46 //Update
47 app.get('/edit/:id', (req, res) => {
48   const query = 'SELECT * FROM users WHERE id = ?';
49   connection.query(query, [req.params.id], (err, results) => {
50     if (err) throw err;
51     res.render('edit', { user: results[0] });
52   });
53 });
54
55 app.listen(3000, () => {
56   console.log("Server berjalan di port 3000, buka web melalui http://localhost:3000");
57 })
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

at exports.renderFile [as engine] (C:\Riyadi\PEMMEB\PERTEMUAN 5\node_modules\ejs\lib\ejs.js:491:10)
at View.render (C:\Riyadi\PEMMEB\PERTEMUAN 5\node_modules\express\lib\view.js:135:8)
at tryRender (C:\Riyadi\PEMMEB\PERTEMUAN 5\node_modules\express\lib\application.js:657:10)
at Function.render (C:\Riyadi\PEMMEB\PERTEMUAN 5\node_modules\express\lib\application.js:609:3)
at ServerResponse.render (C:\Riyadi\PEMMEB\PERTEMUAN 5\node_modules\express\lib\response.js:1049:7)
at Query.onResult (C:\Riyadi\PEMMEB\PERTEMUAN 5\app.js:51:13)
at C:\Riyadi\PEMMEB\PERTEMUAN 5\node_modules\mysql\lib\commands\query.js:86:16

node + v 🌐 🗃 ... ^ x

Sign in to Jira > No active issue Sign in to Bitbucket ⚡ 0 0 127.0.0.1 pertemuan5

In 53, Col 4 Spaces: 4 UTF-8 CRLF { JavaScript

18. Buat tampilan untuk edit :

```
File Edit Selection View Go Run Terminal Help < - > ⚡ PERTEMUAN 5
EXPLORER PERTEMUAN 5
  node_modules
  views
    edit.ejs
    index.ejs
  app.js
  package-lock.json
  package.json
index.ejs js app.js edit.ejs
views > edit.ejs > 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 Pengguna</title>
</head>
<body>
  <h2>Edit Data Pengguna</h2>
  <form action="/update/:user.id" method="post">
    <label for="name">Nama</label>
    <input type="text" id="name" name="name" required value="<%= user.name %>"><br>
    <label for="email">Email:</label>
    <input type="email" id="email" name="email" required value="<%= user.email %>"><br>
    <label for="phone">Telepon:</label>
    <input type="text" id="phone" name="phone" required value="<%= user.phone %>"><br>
    <button type="submit">Simpan</button>
  </form>
</body>
</html>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

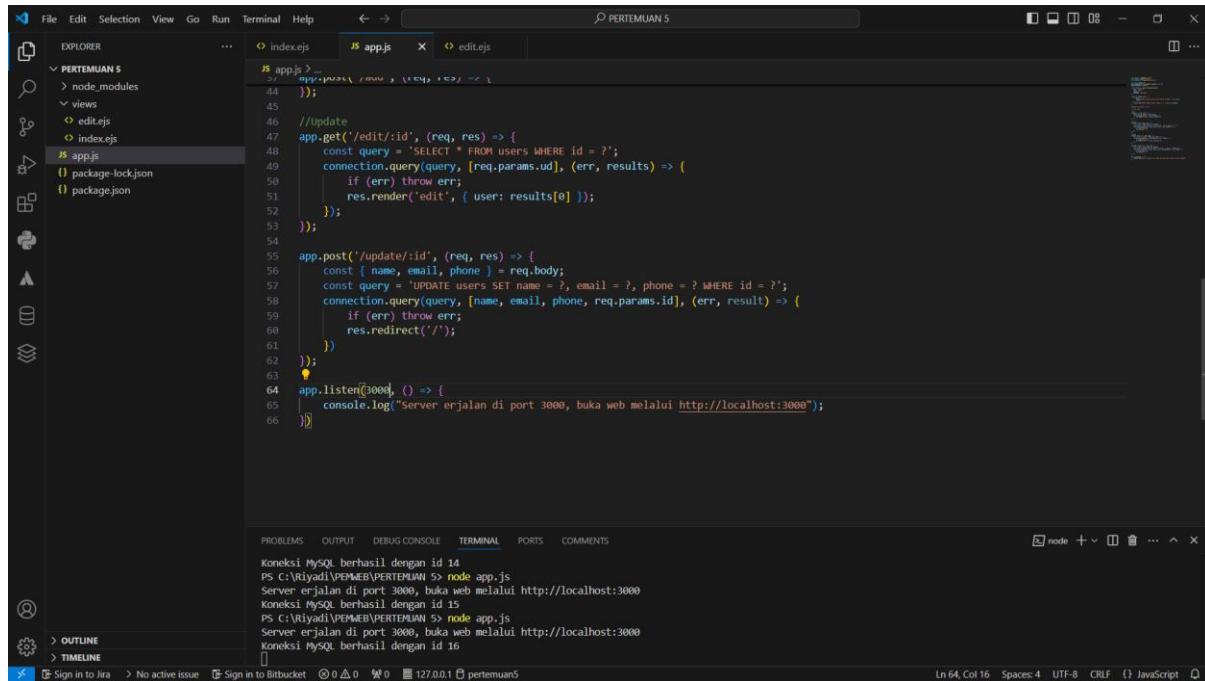
Koneksi MySQL berhasil dengan id 14
PS C:\Riyadi\PEMMEB\PERTEMUAN 5> node app.js
Server berjalan di port 3000, buka web melalui http://localhost:3000
Koneksi MySQL berhasil dengan id 15

node + v 🌐 🗃 ... ^ x

Sign in to Jira > No active issue Sign in to Bitbucket ⚡ 0 0 127.0.0.1 pertemuan5

In 23, Col 8 Spaces: 4 UTF-8 CRLF HTML

19. Tambahkan post dengan routing '/update/:id' agar dapat menyimpan perubahan data dari form



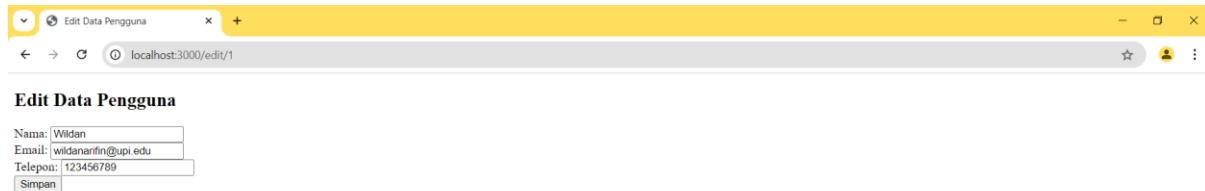
```
File Edit Selection View Go Run Terminal Help ⏪ ⏩ PERTEMUAN 5
EXPLORER JS app.js x edit.js
PERTEMUAN 5
node_modules
views
edit.js
index.js
JS app.js
package-lock.json
package.json
44 );
45
46 //update
47 app.get('/edit/:id', (req, res) => {
48   const query = 'SELECT * FROM users WHERE id = ?';
49   connection.query(query, [req.params.id], (err, results) => {
50     if (err) throw err;
51     res.render('edit', { user: results[0] });
52   });
53 );
54
55 app.post('/update/:id', (req, res) => {
56   const { name, email, phone } = req.body;
57   const query = 'UPDATE users SET name = ?, email = ?, phone = ? WHERE id = ?';
58   connection.query(query, [name, email, phone, req.params.id], (err, result) => {
59     if (err) throw err;
60     res.redirect('/');
61   });
62 );
63
64 app.listen(3000, () => {
65   console.log("Server berjalan di port 3000, buka web melalui http://localhost:3000");
66 })
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

Koneksi MySQL berhasil dengan id 14
PS C:\Riyadi\PEMMEB\PERTEMUAN> node app.js
Server berjalan di port 3000, buka web melalui http://localhost:3000
Koneksi MySQL berhasil dengan id 15
PS C:\Riyadi\PEMMEB\PERTEMUAN> node app.js
Server berjalan di port 3000, buka web melalui http://localhost:3000
koneksi MySQL berhasil dengan id 16

Ln 64, Col 16 Spaces: 4 UTF-8 CRLF {} JavaScript

20. Tampilan akan seperti ini



21. Jika mengganti data maka akan langsung direct ke index seperti ini

The screenshot shows a web application titled "Daftar User/Pengguna". It displays a table of users with columns: ID, Nama, Email, Telepon, and Aksi. The data is as follows:

ID	Nama	Email	Telepon	Aksi
1	Wildanarfin	wildanarfin@upi.edu	123456789	Edit Hapus
2	Habibirohim	habibirohim@upi.edu	234567890	Edit Hapus
3	Bintang	bintang@upi.edu	345678901	Edit Hapus
4	Angga	angga@upi.edu	456789012	Edit Hapus
5	Riyadi	riyaditri12@gmail.com	082134451235	Edit Hapus

Below the table is a form titled "Tambah Pengguna Baru" with fields for Nama, Email, and Telepon, and a "Tambah" button.

22. Tambahkan get dengan routing '/delete/:id' agar dapat menghapus data yang ada pada tabel

The screenshot shows a code editor with an open file named "app.js". The code contains the following routing logic:

```
app.get('/delete/:id', callback);
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, results) => {
    if (err) throw err;
    res.redirect('/');
  });
});
app.listen(3000, () => {
  console.log("Server berjalan di port 3000, buka web melalui http://localhost:3000");
})
```

The terminal below the code editor shows the command "node app.js" being run, and the output indicates the server is listening on port 3000.

23. Seperti ini tampilan setelah menghapus yang mana akan direct ke index

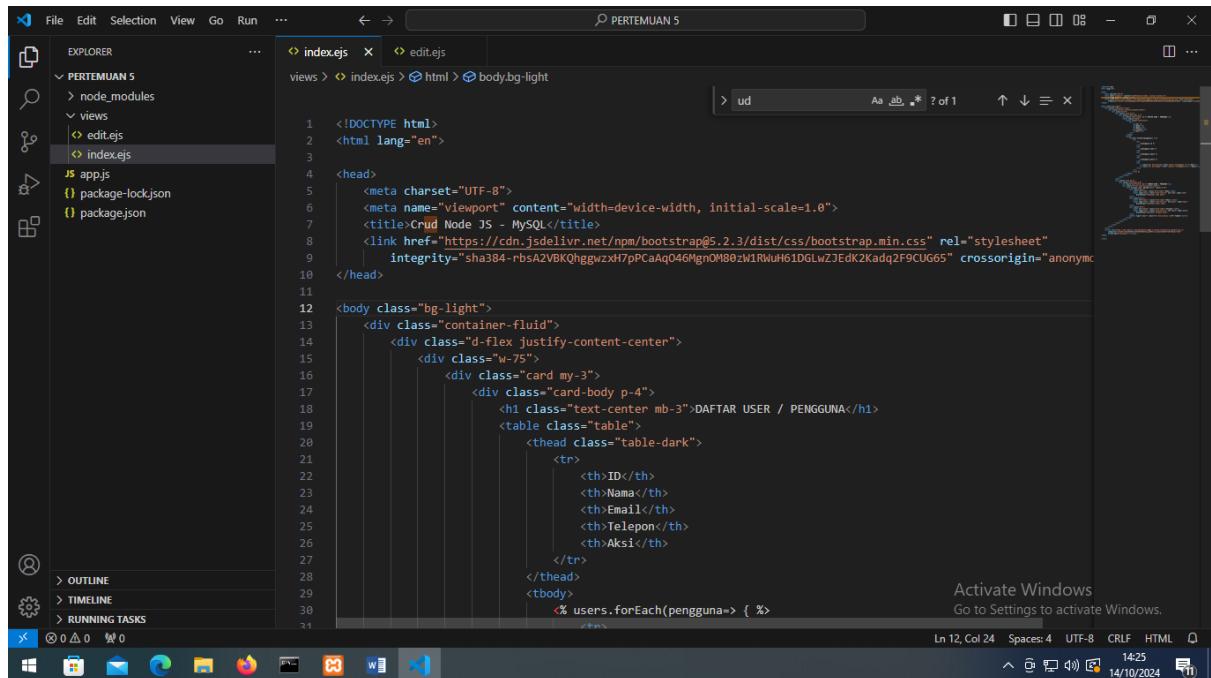
The screenshot shows a web application titled "Daftar User/Pengguna" (User List). At the top, there is a table with columns: ID, Nama, Email, Telepon, and Aksi (Action). The table contains five rows of data:

ID	Nama	Email	Telepon	Aksi
2	Habibirohim	habibirohim@upi.edu	234567890	Edit Hapus
3	Bintang	bintang@upi.edu	345678901	Edit Hapus
4	Angga	angga@upi.edu	456789012	Edit Hapus
5	Riyadi	riyaditri12@gmail.com	082134451235	Edit Hapus

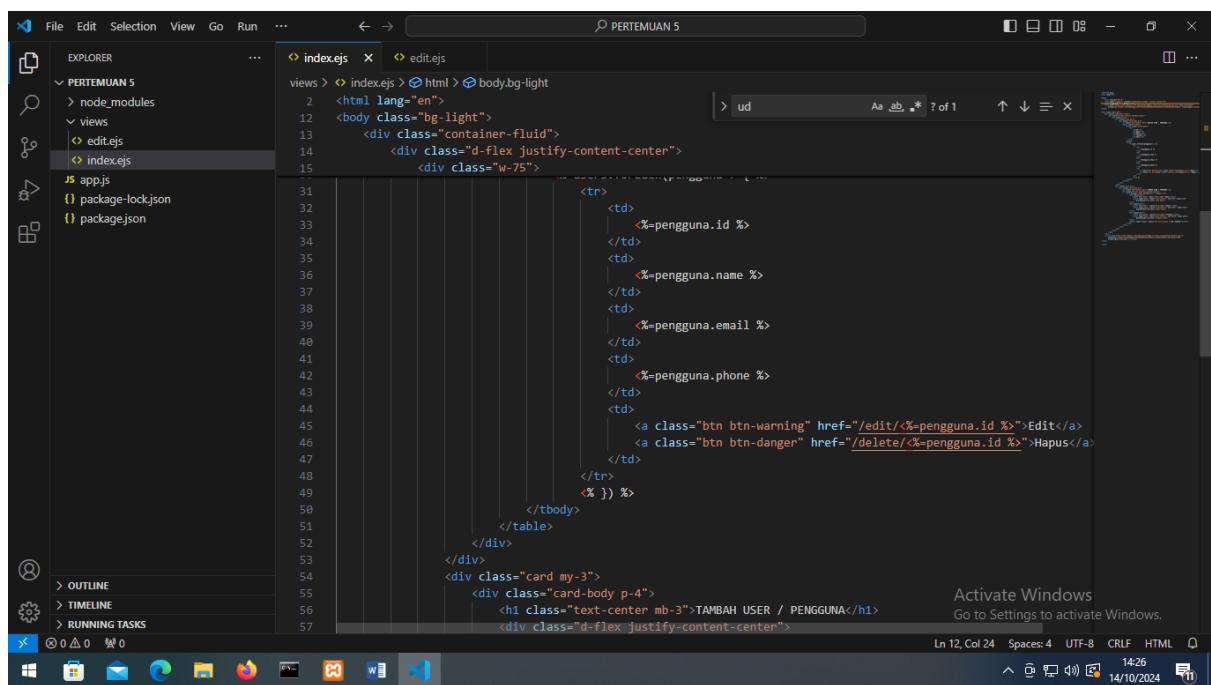
Below the table, there is a section titled "Tambah Pengguna Baru" (Add New User) with three input fields: Nama, Email, and Telepon, followed by a "Tambah" button.

*TUGAS 2

24. Agar tampilan lebih menarik, tambahkan bootstrap seperti berikut pada index dan mengubah sedikit tampilannya



```
<!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@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-rbsA2VBlQHqJnCGF1I/ce29fPzO7ZtHd8gFQDf3mWzCfKJfK=" crossorigin="anonymous">
  </head>
  <body class="bg-light">
    <div class="container-fluid">
      <div class="d-flex justify-content-center">
        <div class="w-75">
          <div class="card my-3">
            <div class="card-body p-4">
              <h1 class="text-center mb-3">DAFTAR USER / PENGGUNA</h1>
              <table class="table">
                <thead class="table-dark">
                  <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 class="btn btn-warning" href="/edit/<%= pengguna.id %>">Edit</a>
                        <a class="btn btn-danger" href="/delete/<%= pengguna.id %>">Hapus</a>
                      </td>
                    </tr>
                  <% } %>
                </tbody>
              </table>
            </div>
            <div class="card my-3">
              <div class="card-body p-4">
                <h1 class="text-center mb-3"> TAMBAH USER / PENGGUNA</h1>
                <div class="d-flex justify-content-center">
```



```
<tbody>
  <% users.forEach(pengguna) => { %>
    <tr>
      <td><%= pengguna.id %></td>
      <td><%= pengguna.name %></td>
      <td><%= pengguna.email %></td>
      <td><%= pengguna.phone %></td>
      <td>
        <a class="btn btn-warning" href="/edit/<%= pengguna.id %>">Edit</a>
        <a class="btn btn-danger" href="/delete/<%= pengguna.id %>">Hapus</a>
      </td>
    </tr>
  <% } %>
</tbody>
</table>
</div>
<div class="card my-3">
  <div class="card-body p-4">
    <h1 class="text-center mb-3"> TAMBAH USER / PENGGUNA</h1>
    <div class="d-flex justify-content-center">
```

Screenshot of VS Code showing the file structure and code for a Node.js application.

File Explorer:

- PERTEMUAN 5
- views
 - index.ejs
 - edit.ejs
- app.js
- package-lock.json
- package.json

Code Editor: edit.ejs

```

<html lang="en">
  <body class="bg-light">
    <div class="container-fluid">
      <div class="d-flex justify-content-center">
        <div class="w-75">
          <form>
            <div class="mb-3">
              <label for="name" class="form-label">Nama</label>
              <input type="text" class="form-control" id="name" name="name" placeholder="Masukkan nama anda">
            </div>
            <div class="mb-3">
              <label for="email" class="form-label">Email</label>
              <input type="text" class="form-control" id="email" name="email" placeholder="Masukkan email anda">
            </div>
            <div class="mb-3">
              <label for="phone" class="form-label">Telepon</label>
              <input type="text" class="form-control" id="phone" name="phone" placeholder="Masukkan telepon anda">
            </div>
            <button type="submit" class="btn btn-primary w-100">Tambah</button>
          </form>
        </div>
      </div>
    </div>
  </body>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ken1KFdbIE4ZVF0s0G1N5b4hcpxy09F7jl+jjXkk+Q2h455rYXK/7HAuoJl+0I4" crossorigin="anonymous"></script>

```

Status Bar: Activate Windows
Go to Settings to activate Windows.
Ln 12, Col 24 Spaces: 4 UTF-8 CRLF HTML Q
14/10/2024

25. Tampilan akan menjadi seperti berikut

Screenshot of a web browser showing the application's user interface.

Title Bar: Crud Node JS - MySQL

Address Bar: localhost:3000

DAFTAR USER / PENGGUNA

ID	Nama	Email	Telepon	Aksi
1	Wildan	wildanarifin@upi.edu	123456789	<button>Edit</button> <button>Hapus</button>
2	Habibirohim	habibirohim@upi.edu	234567891	<button>Edit</button> <button>Hapus</button>
3	Bintang	bintang@upi.edu	345678912	<button>Edit</button> <button>Hapus</button>
4	Angga	angga@upi.edu	456789123	<button>Edit</button> <button>Hapus</button>
5	Test	test@gmail.com	08123456789	<button>Edit</button> <button>Hapus</button>

TAMBAH USER / PENGGUNA

Nama
Masukkan nama anda

Status Bar: Activate Windows
Go to Settings to activate Windows.
14/10/2024

Crud Node JS - MySQL

localhost:3000

	Nama	Email	Telepon	Edit	Hapus
3	Bintang	bintang@upi.edu	345678912	Edit	Hapus
4	Angga	angga@upi.edu	456789123	Edit	Hapus
5	Test	test@gmail.com	08123456789	Edit	Hapus

TAMBAH USER / PENGGUNA

Nama

Email

Telepon

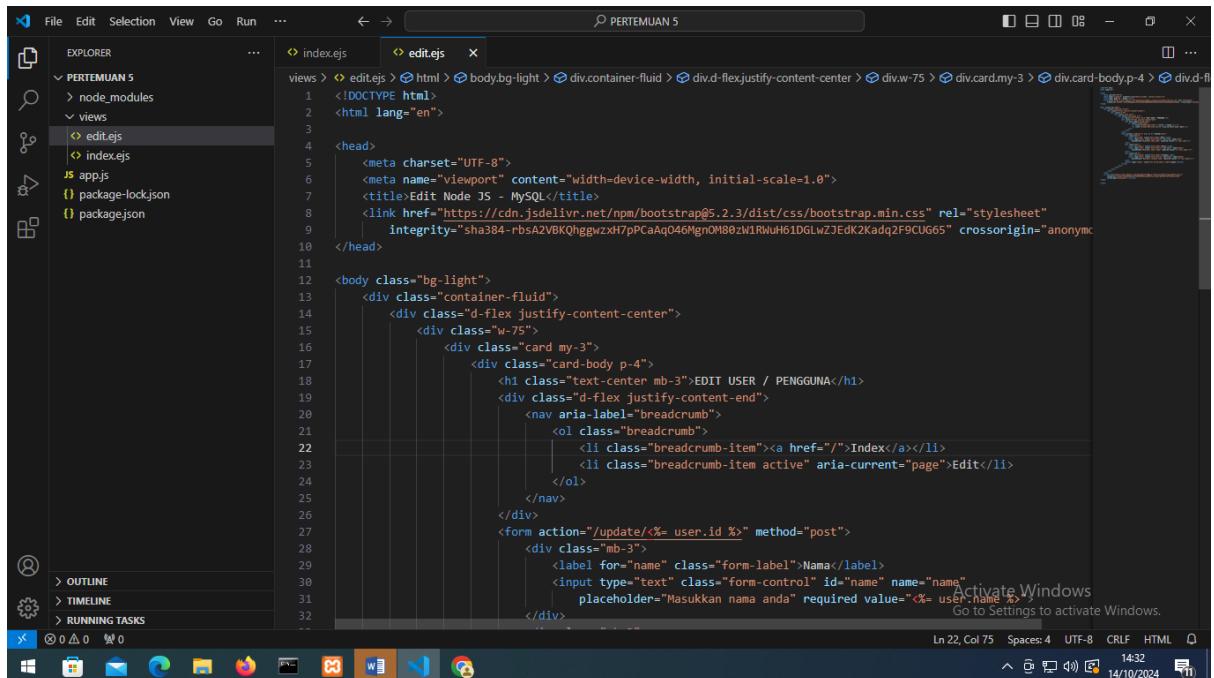
[Tambah](#)

Activate Windows
Go to Settings to activate Windows.

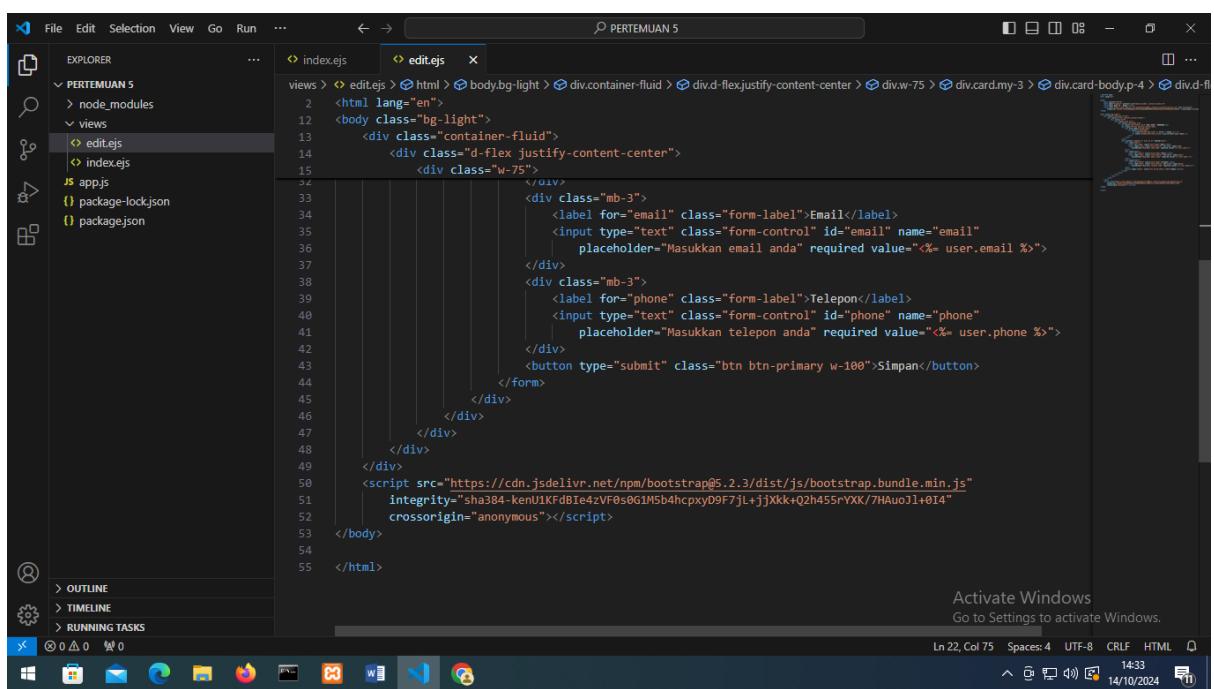


14:29
14/10/2024

26. Tambahkan juga bootstrap pada edit dan ubah sedikit tampilannya agar lebih menarik

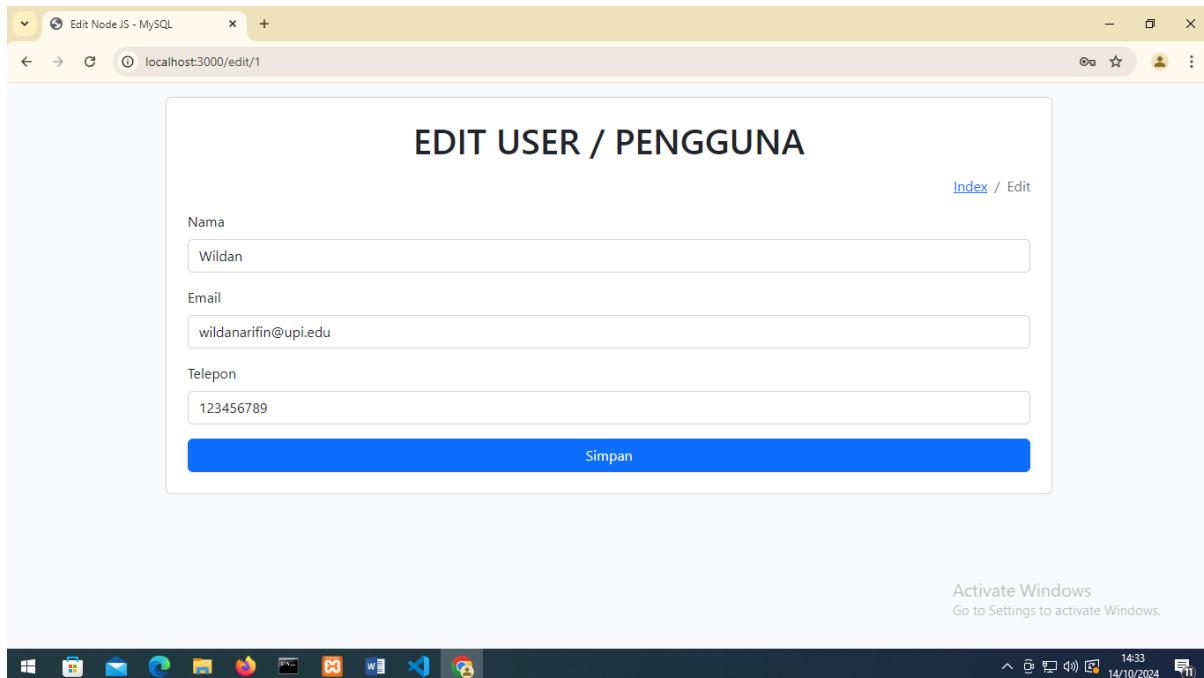


```
<?> index.ejs <?> edit.ejs <?> PERTEMUAN 5
views > edit.ejs > <!DOCTYPE html>
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Edit Node JS - MySQL</title>
8   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet"
9        integrity="sha384-rbsA2VbQZJJDWzH7pCgAqO46HgnOM8zW1RWuH61D6lwZJEdK2Kadq2F9CU65" crossorigin="anonymous">
10 </head>
11
12 <body class="bg-light">
13   <div class="container-fluid">
14     <div class="d-flex justify-content-center">
15       <div class="w-75">
16         <div class="card my-3">
17           <div class="card-body p-4">
18             <h1 class="text-center mb-3">EDIT USER / PENGGUNA</h1>
19             <div class="d-flex justify-content-end">
20               <nav aria-label="breadcrumb">
21                 <ol class="breadcrumb">
22                   <li class="breadcrumb-item"><a href="/">Index</a></li>
23                   <li class="breadcrumb-item active" aria-current="page">Edit</li>
24                 </ol>
25               </nav>
26             </div>
27             <form action="/update/<%= user.id %>" method="post">
28               <div class="mb-3">
29                 <label for="name" class="form-label">Nama</label>
30                 <input type="text" class="form-control" id="name" name="name"
31                   placeholder="Masukkan nama anda" required value="<%= user.name %>">
32               </div>
33             </form>
34           </div>
35         </div>
36       </div>
37     </div>
38   </div>
39   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"
40        integrity="sha384-kenU1KfdIe4zVf0s0G1M5b4hcpxyD9F7jl+jjXkk+Q2h455rYXK/7HAuoJl+014"
41        crossorigin="anonymous"></script>
42 </body>
43 </html>
```

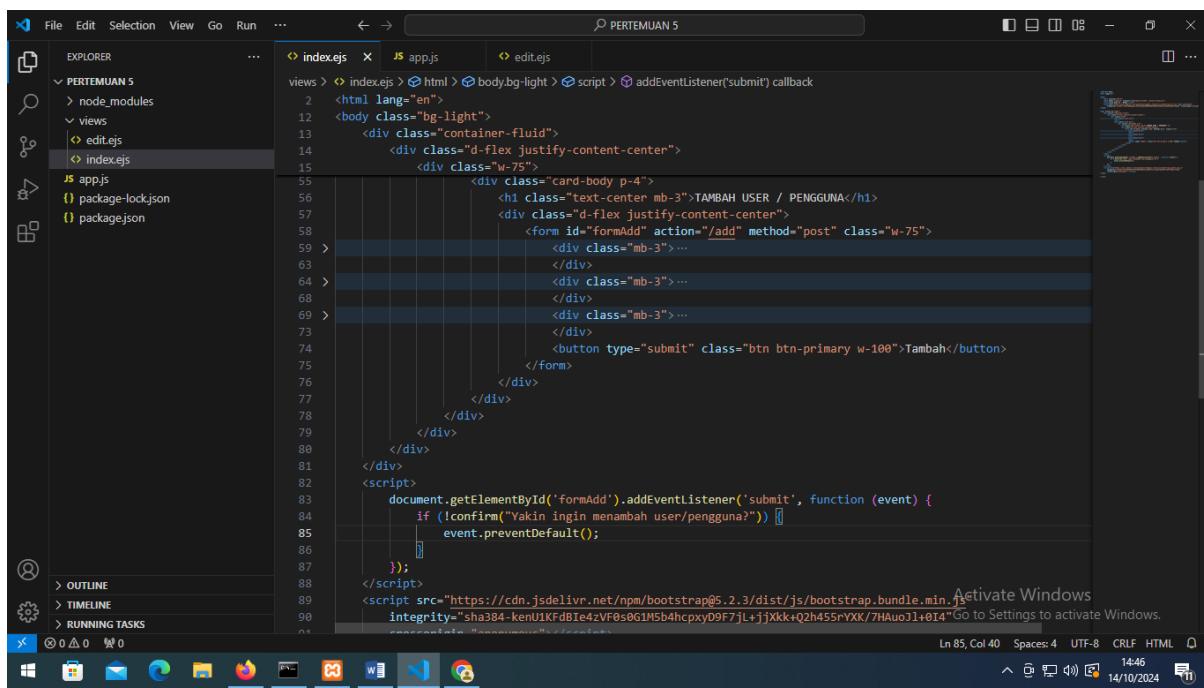


```
<?> index.ejs <?> edit.ejs <?> PERTEMUAN 5
views > edit.ejs > <!DOCTYPE html>
1 <!DOCTYPE html>
2 <html lang="en">
3 <body class="bg-light">
4   <div class="container-fluid">
5     <div class="d-flex justify-content-center">
6       <div class="w-75">
7         <div>
8           <div class="mb-3">
9             <label for="email" class="form-label">Email</label>
10            <input type="text" class="form-control" id="email" name="email"
11              placeholder="Masukkan email anda" required value="<%= user.email %>">
12          </div>
13          <div class="mb-3">
14            <label for="phone" class="form-label">Telepon</label>
15            <input type="text" class="form-control" id="phone" name="phone"
16              placeholder="Masukkan telepon anda" required value="<%= user.phone %>">
17          </div>
18          <button type="submit" class="btn btn-primary w-100">Simpan</button>
19        </div>
20      </div>
21    </div>
22  </div>
23  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"
24        integrity="sha384-kenU1KfdIe4zVf0s0G1M5b4hcpxyD9F7jl+jjXkk+Q2h455rYXK/7HAuoJl+014"
25        crossorigin="anonymous"></script>
26 </body>
27 </html>
```

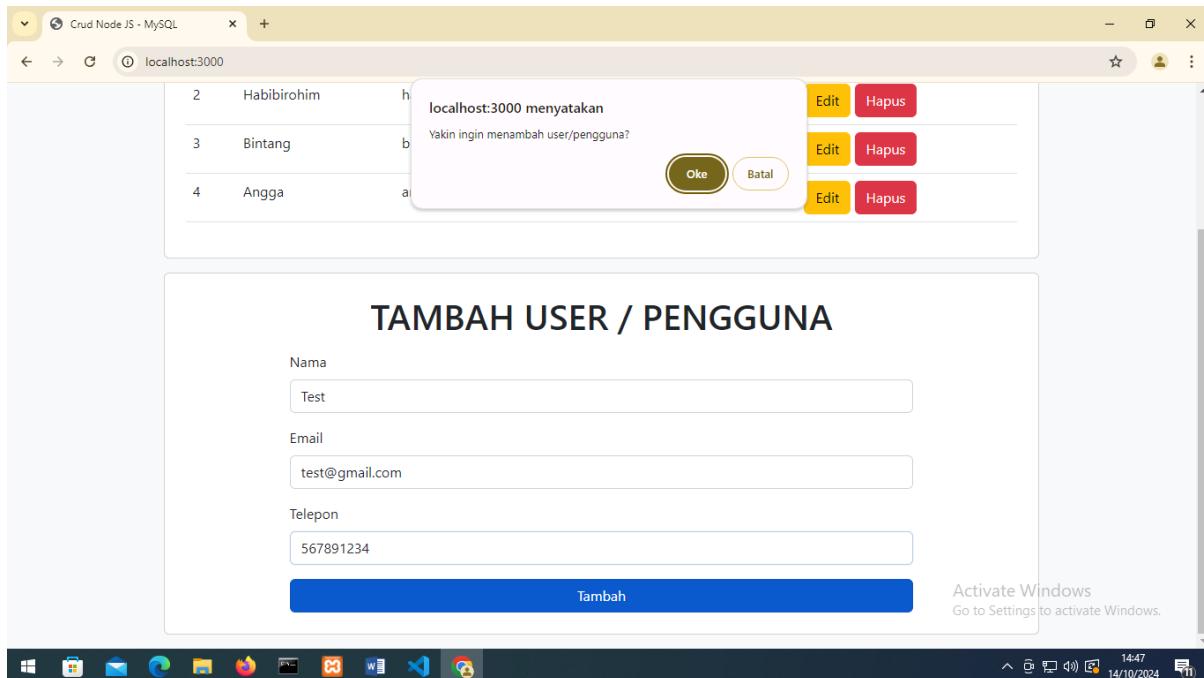
27. Tampilan edit akan menjadi seperti ini



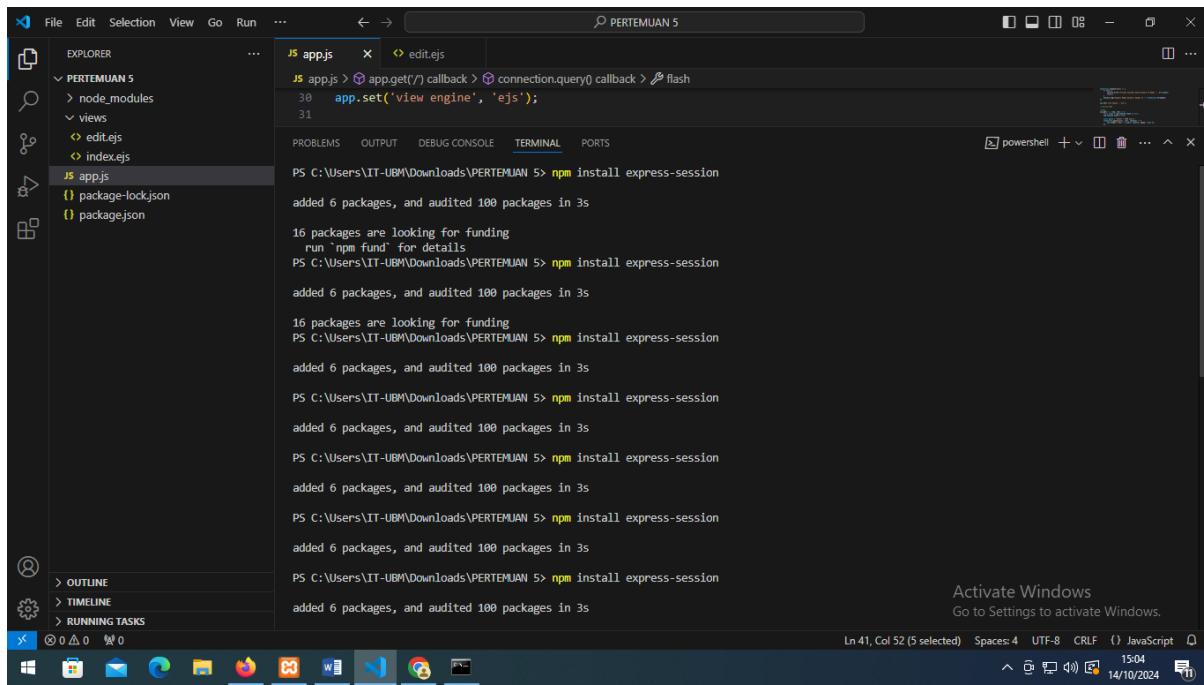
28. Tambahkan konfirmasi saat sebelum melakukan penambahan.



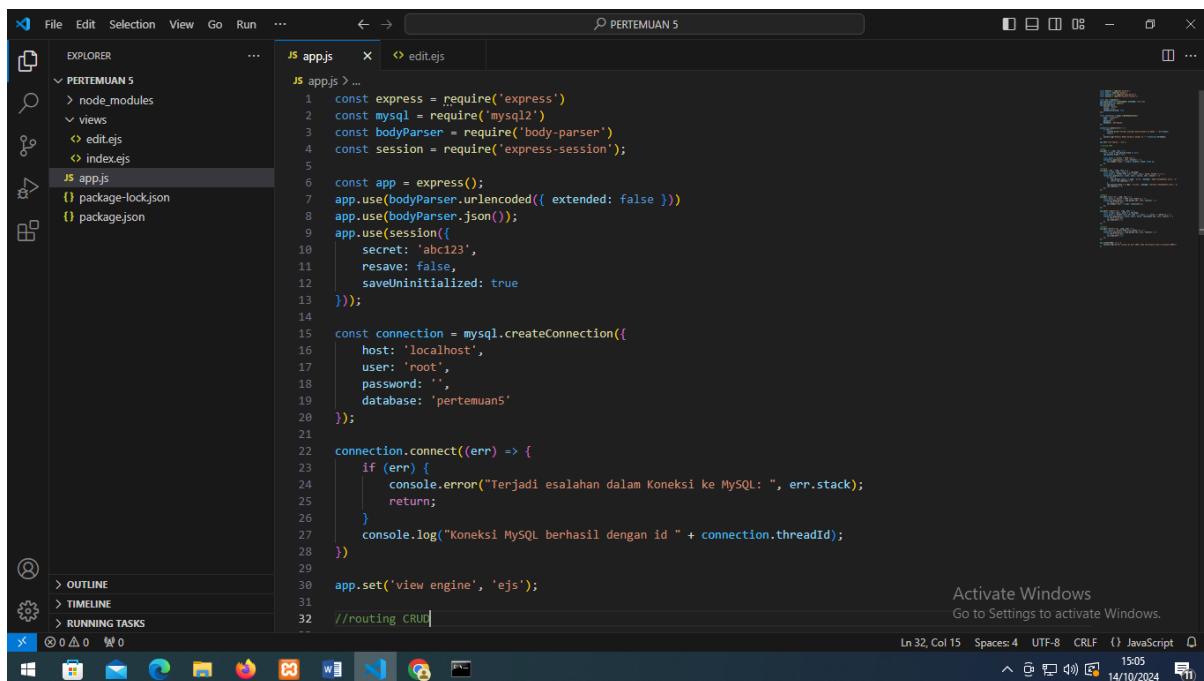
29. Tampilan saat tambah akan seperti ini.



30. Install express-session untuk sesi sebagai flash message untuk notifikasi setelah melakukan penambahan.



31. Set session dan tambahkan session saat berhasil menambah



The screenshot shows the VS Code interface with the file `app.js` open. The code sets up an Express application, connects to a MySQL database, and configures session middleware. A comment indicates the addition of session handling.

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

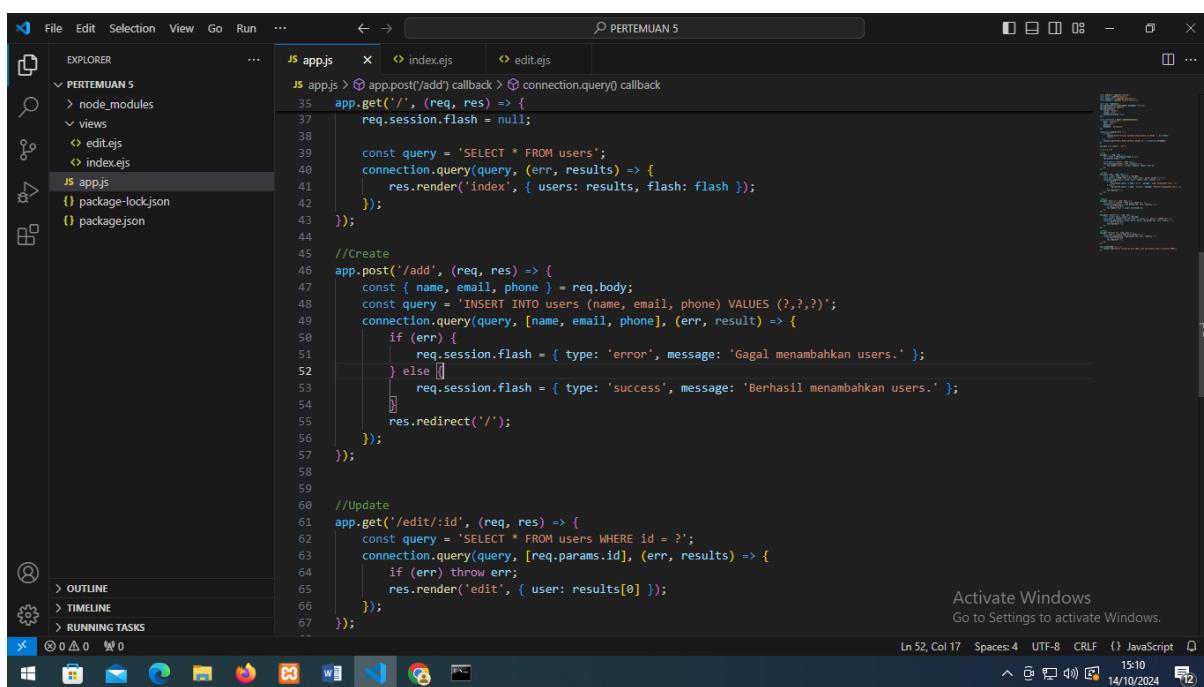
const app = express();
app.use(bodyParser.urlencoded({ extended: false }));
app.use(bodyParser.json());
app.use(session({
  secret: 'abc123',
  resave: false,
  saveUninitialized: true
}));

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

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

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

//routing CRUD
```



The screenshot shows the VS Code interface with the file `app.js` open. It includes routes for adding, updating, and deleting users. For the add route, it performs a query to get all users, then inserts a new user into the database. It uses the `req.session.flash` object to store messages for the user.

```
app.get('/', (req, res) => {
  req.session.flash = null;

  const query = 'SELECT * FROM users';
  connection.query(query, (err, results) => {
    res.render('index', { users: results, flash: flash });
  });
});

//Create
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) {
      req.session.flash = { type: 'error', message: 'Gagal menambahkan users.' };
    } else {
      req.session.flash = { type: 'success', message: 'Berhasil menambahkan users.' };
    }
    res.redirect('/');
  });
});

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

32. Tampilkan pada tampilan yang ada

33. Tampilan akan seperti berikut

The screenshot shows a web browser window titled "Crud Node JS - MySQL". The URL is "localhost:3000". The main content is a table titled "DAFTAR USER / PENGGUNA" listing five users:

ID	Nama	Email	Telepon	Aksi
1	Wildan	wildanarifin@upi.edu	123456789	<button>Edit</button> <button>Hapus</button>
2	Habibirohim	habibirohim@upi.edu	234567891	<button>Edit</button> <button>Hapus</button>
3	Bintang	bintang@upi.edu	345678912	<button>Edit</button> <button>Hapus</button>
4	Angga	angga@upi.edu	456789123	<button>Edit</button> <button>Hapus</button>
13	Test	test@gmail.com	567891234	<button>Edit</button> <button>Hapus</button>

Below the table is a form titled "TAMBAH USER / PENGGUNA". A watermark in the bottom right corner says "Activate Windows Go to Settings to activate Windows." The taskbar at the bottom shows various application icons.

34. Tambahkan konfirmasi sebelum mengubah data pengguna

The screenshot shows a Microsoft Visual Studio Code interface. The top bar includes File, Edit, Selection, View, Go, Run, and a search bar for 'PERTEMUAN 5'. The left sidebar has sections for EXPLORER, PERTEMUAN 5 (node_modules, views), and app.js (edit.ejs, index.ejs). The main area displays the content of edit.ejs:

```
2  <html lang="en">
3      <body class="bg-light">
4          <div class="container-fluid">
5              <div class="d-flex justify-content-center">
6                  <div class="w-75">
7                      <nav>
8                          <ul class="list-group list-group-flush">
9                              <li class="list-item"><a href="#">Home</a></li>
10                         <li class="list-item"><a href="#">About</a></li>
11                         <li class="list-item"><a href="#">Contact</a></li>
12                     </ul>
13                 </div>
14             </div>
15         </div>
16     </body>
17 </html>
```

The code continues with a form section and a script block containing JavaScript for handling the form submission:

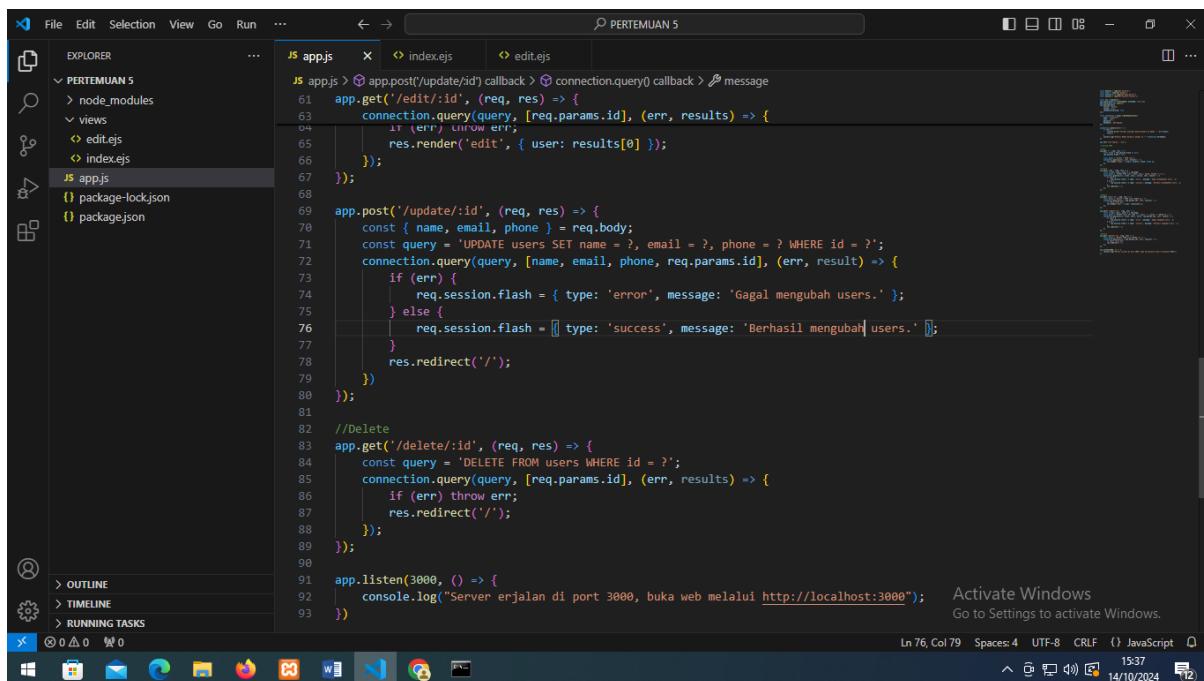
```
27     <form id="formEdit" action="/update/<%= user.id %>" method="post">
28         <div class="mb-3">...
29             <label for="name">Name</label>
30             <input type="text" id="name" name="name" value="...
31         </div>
32         <div class="mb-3">...
33             <label for="email">Email</label>
34             <input type="email" id="email" name="email" value="...
35         </div>
36         <div class="mb-3">...
37             <label for="password">Password</label>
38             <input type="password" id="password" name="password" value="...
39         </div>
40         <button type="submit" class="btn btn-primary w-100">Simpan</button>
41     </form>
42 
43 </div>
44 </div>
45 </div>
46 </div>
47 </div>
48 </div>
49 </div>
50 </script>
51     document.getElementById('formEdit').addEventListener('submit', function (event) {
52         if (!confirm("Yakin ingin mengubah data user/pengguna?")) {
53             event.preventDefault();
54         }
55     });
56 </script>
57 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"
58         integrity="sha384-ken1kFBl4zVf0s0G1M5b4hpcxyD9F7JL+jXkk+Q2h455rYXK/7HAuoJl+0I4.">
59             crossorigin="anonymous"</script>
60 </body>
```

The status bar at the bottom shows 'Ln 27, Col 43' and other standard status indicators.

35. Berikut tampilan saat ingin mengubah

A screenshot of a web browser window titled "Edit Node JS - MySQL". The URL is "localhost:3000/edit/1". A modal dialog box is centered on the page, containing the text "localhost:3000 menyatakan" and "Yakin ingin mengubah data user/pengguna?". Below the text are two buttons: "Oke" (in a dark green rounded rectangle) and "Batal" (in a light orange rounded rectangle). To the right of the modal, the text "Index / Edit" is visible. Below the modal, there are three input fields: "Nama" with value "Wildan", "Email" with value "wildanarifin@upi.edu", and "Telepon" with value "123456789". At the bottom of the form is a large blue "Simpan" button. The browser's taskbar at the bottom shows various open tabs and system icons.

36. Tambahkan notifikasi saat sudah mengubah



```
PERTEMUAN 5
views
JS app.js
  app.get('/edit/:id', (req, res) => {
    connection.query(query, [req.params.id], (err, results) => {
      if (err) throw err;
      res.render('edit', { user: results[0] });
    });
  });

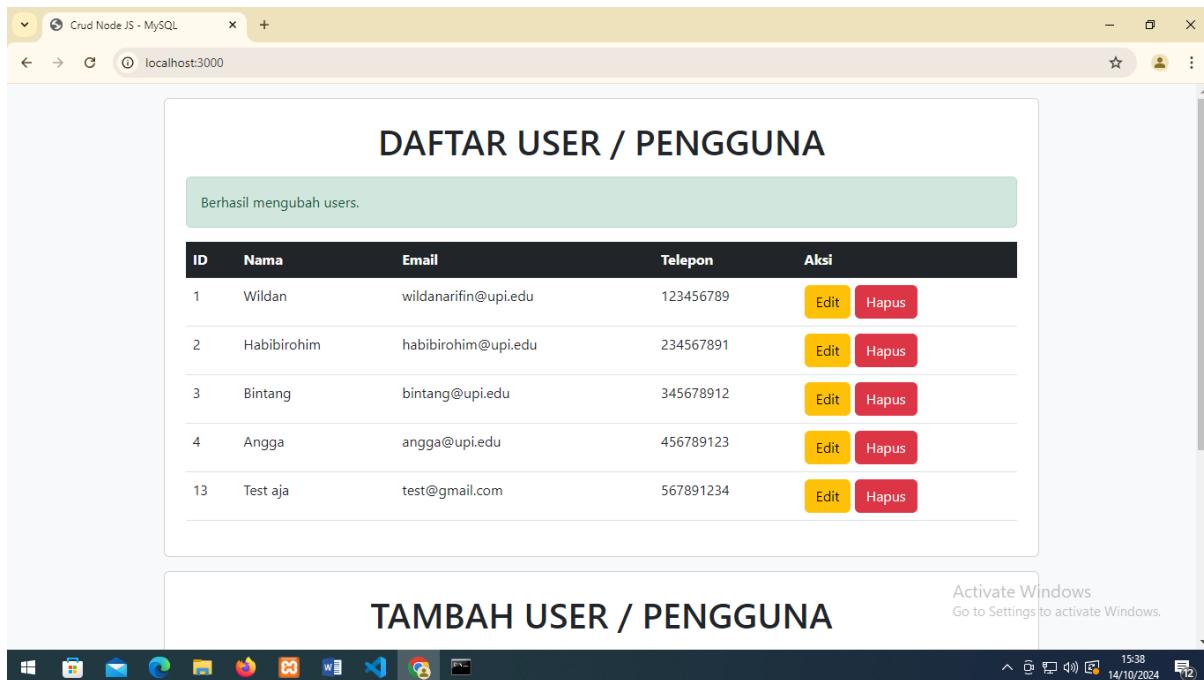
  app.post('/update/:id', (req, res) => {
    const { name, email, phone } = req.body;
    const query = 'UPDATE users SET name = ?, email = ?, phone = ? WHERE id = ?';
    connection.query(query, [name, email, phone, req.params.id], (err, result) => {
      if (err) {
        req.session.flash = { type: 'error', message: 'Gagal mengubah users.' };
      } else {
        req.session.flash = { type: 'success', message: 'Berhasil mengubah users.' };
      }
      res.redirect('/');
    });
  });

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

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

```

37. Tampilan akan seperti berikut



DAFTAR USER / PENGGUNA

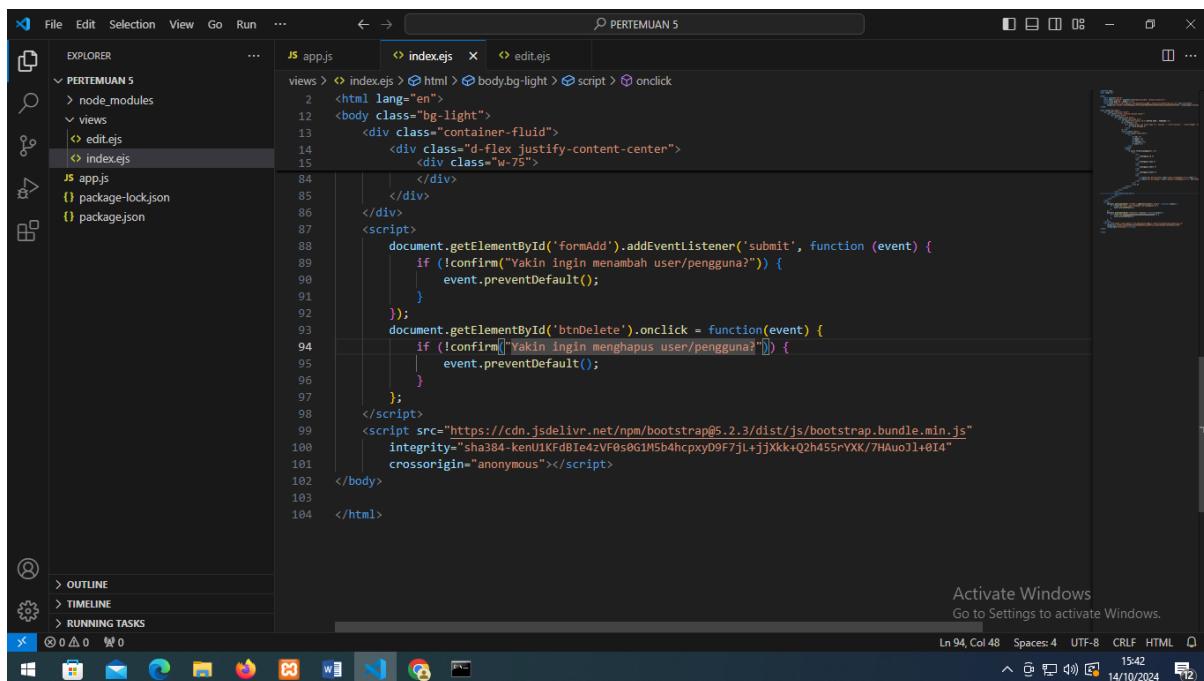
Berhasil mengubah users.

ID	Nama	Email	Telepon	Aksi
1	Wildan	wildanarifin@upi.edu	123456789	<button>Edit</button> <button>Hapus</button>
2	Habibirohim	habibirohim@upi.edu	234567891	<button>Edit</button> <button>Hapus</button>
3	Bintang	bintang@upi.edu	345678912	<button>Edit</button> <button>Hapus</button>
4	Angga	angga@upi.edu	456789123	<button>Edit</button> <button>Hapus</button>
13	Test aja	test@gmail.com	567891234	<button>Edit</button> <button>Hapus</button>

TAMBAH USER / PENGGUNA

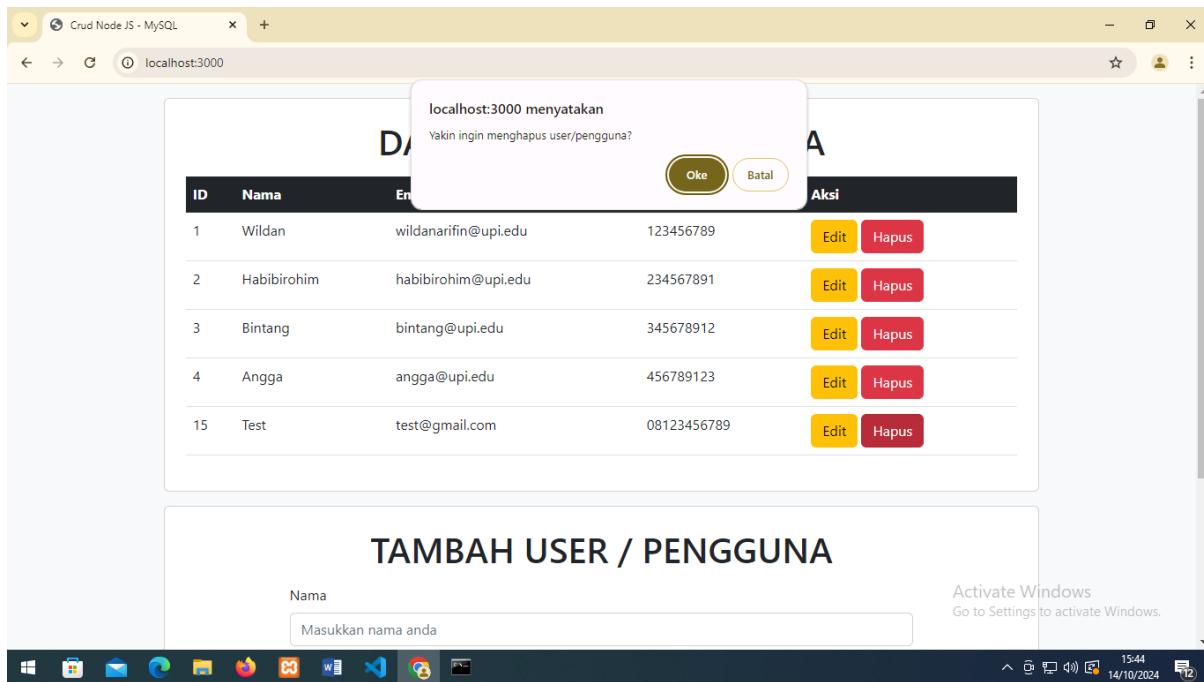
Activate Windows
Go to Settings to activate Windows.

38. Tambahkan konfirmasi sebelum menghapus data pengguna



```
views > index.ejs > html > body.bg-light > script > onclick
  2  <html lang="en">
  3  <body class="bg-light">
  4  <div class="container-fluid">
  5  <div class="d-flex justify-content-center">
  6  <div class="w-75">
  7  </div>
  8  </div>
  9  </div>
 10 <script>
 11   document.getElementById('formAdd').addEventListener('submit', function (event) {
 12     if (!confirm("Yakin ingin menambah user/pengguna?")) {
 13       event.preventDefault();
 14     }
 15   });
 16   document.getElementById('btnDelete').onclick = function(event) {
 17     if (!confirm("Yakin ingin menghapus user/pengguna?")) {
 18       event.preventDefault();
 19     }
 20   };
 21 </script>
 22 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"
 23        integrity="sha384-ken0lKfdIe4zVf0s0G1M5b4hcpxy09F7jL+jjXkk+Q2h45SrYXK/7HuoJl+6I4"
 24        crossorigin="anonymous"></script>
 25 </body>
 26 </html>
```

39. Tampilan akan seperti berikut

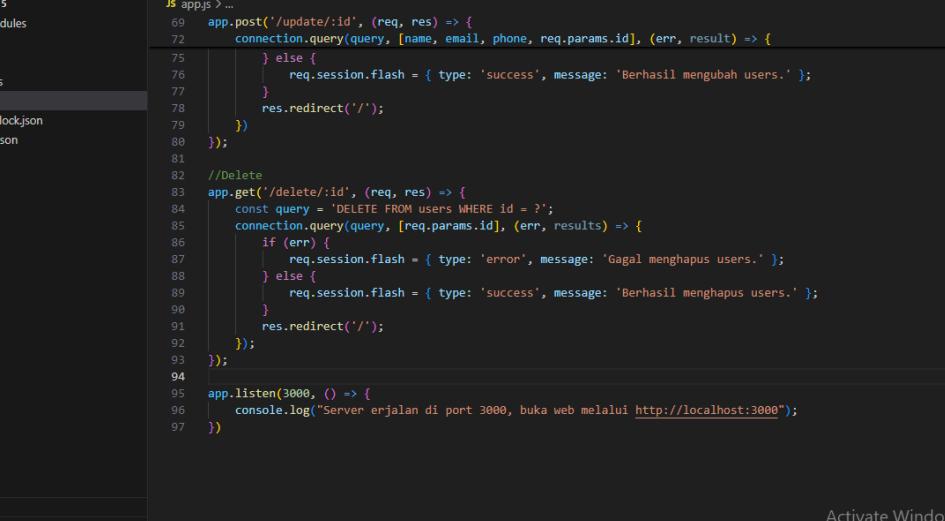


The screenshot shows a web application interface. At the top, there is a navigation bar with a yellow logo and the text "Crud Node JS - MySQL". Below it, a browser address bar shows "localhost:3000". A modal dialog box is centered on the screen, displaying the message "localhost:3000 menyatakan" and "Yakin ingin menghapus user/pengguna?" with "Oke" and "Batal" buttons. In the background, a table lists users with columns "ID", "Nama", "Email", and "Aksi". The table data is:

ID	Nama	Email	Aksi
1	Wildanarifin	wildanarifin@upi.edu	Edit Hapus
2	Habibirohim	habibirohim@upi.edu	Edit Hapus
3	Bintang	bintang@upi.edu	Edit Hapus
4	Angga	angga@upi.edu	Edit Hapus
15	Test	test@gmail.com	Edit Hapus

Below the table, a separate form titled "TAMBAH USER / PENGGUNA" is visible, with a "Nama" input field containing "Masukkan nama anda".

40. Tambahkan notifikasi saat sudah menghapus



The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The title bar reads "PERTEMUAN 5". The left sidebar (Explorer) shows a file tree with "PERTEMUAN 5" expanded, containing "node_modules", "views", "edit.ejs", and "index.ejs". Below these are "app.js", "package-lock.json", and "package.json". The main editor area displays the "app.js" file. The code in "app.js" handles POST and GET requests for updating and deleting users from a database. It uses Express.js routes and the MySQL module to interact with a database named "connection". The code includes session management using req.session.flash to show success or error messages. The file ends with a call to app.listen(3000). The bottom status bar shows "Activate Windows" with a link to settings, and the footer indicates the code is at line 94, column 1, with 1545 characters, in UTF-8 encoding, and using CRLF line endings. The status bar also shows tabs for "JavaScript" and "HTML/CSS".

```
File Edit Selection View Go Run ... ← → PERTEMUAN 5 Explorer JS app.js index.ejs edit.ejs PERTEMUAN 5 node_modules views edit.ejs index.ejs app.js package-lock.json package.json OUTLINE TIMELINE RUNNING TASKS 1545 14/10/2024 JavaScript HTML/CSS
```

```
JS app.js > ...
69 app.post('/update/:id', (req, res) => {
70   connection.query(query, [name, email, phone, req.params.id], (err, result) => {
71     if (err) {
72       req.session.flash = { type: 'error', message: 'Terjadi kesalahan saat mengubah users.' };
73     } else {
74       req.session.flash = { type: 'success', message: 'Berhasil mengubah users.' };
75     }
76     res.redirect('/');
77   });
78 });
79 });
80 });
81 });
82 //Delete
83 app.get('/delete/:id', (req, res) => {
84   const query = 'DELETE FROM users WHERE id = ?';
85   connection.query(query, [req.params.id], (err, results) => {
86     if (err) {
87       req.session.flash = { type: 'error', message: 'Gagal menghapus users.' };
88     } else {
89       req.session.flash = { type: 'success', message: 'Berhasil menghapus users.' };
90     }
91     res.redirect('/');
92   });
93 });
94 });
95 app.listen(3000, () => {
96   console.log("Server berjalan di port 3000, buka web melalui http://localhost:3000");
97 })
```

41. Tampilan akan seperti berikut

DAFTAR USER / PENGGUNA

Berhasil menghapus users.

ID	Nama	Email	Telepon	Aksi
1	Wildan	wildanarifin@upi.edu	123456789	<button>Edit</button> <button>Hapus</button>
2	Habibirohim	habibirohim@upi.edu	234567891	<button>Edit</button> <button>Hapus</button>
3	Bintang	bintang@upi.edu	345678912	<button>Edit</button> <button>Hapus</button>
4	Angga	angga@upi.edu	456789123	<button>Edit</button> <button>Hapus</button>

TAMBAH USER / PENGGUNA

Nama

Masukkan nama anda

Activate Windows
Go to Settings to activate Windows.

IV.KESIMPULAN

- Peningkatan pengalaman pengguna: Menggunakan CSS dalam perintah/pemberitahuan memberikan umpan balik visual yang jelas kepada pengguna, meningkatkan keterlibatan, dan memastikan bahwa pengguna mengetahui status tindakan yang mereka ambil.

- Fungsi CRUD kini berjalan efektif.

Implementasi fungsi buat, baca, perbarui, dan hapus berjalan dengan baik.

Menambahkan validasi membantu menghilangkan kesalahan pengguna seperti:

B.Untuk mengurangi perilaku yang tidak disengaja.

- Responsif dan Cantik: Desain responsif dan indah menggunakan CSS membuat aplikasi Anda terlihat lebih menarik dan membuat situs web Anda lebih menarik bagi pengguna.

- Pengembangan Lebih Lanjut: Praktik ini memberikan peluang pengembangan lebih lanjut antara lain: Contohnya termasuk menambahkan animasi ke perintah, mengintegrasikan notifikasi berbasis waktu, dan menerapkan modalitas konfirmasi tindakan.

Oleh karena itu, menerapkan CSS di situs CRUD Anda tidak hanya akan meningkatkan fungsinya tetapi juga tampilannya, sehingga lebih ramah pengguna.