

PRAKTIKUM SISTEM BASIS DATA

Nama	Muhammad Abrisam Cahyo Juhartono	No. Modul	9
NPM	2206026050	Tipe	Case Study

1. Website

Kode :

AddNoteForm.js

```
import React, { useState } from 'react';
import axios from 'axios';

const AddNoteForm = ({ onAdd }) => {
  const [title, setTitle] = useState('');
  const [content, setContent] = useState('');

  const handleSubmit = (e) => {
    e.preventDefault();
    axios.post('/api/notes', { title, content })
      .then(response => {
        onAdd(response.data);
        setTitle('');
        setContent('');
      })
      .catch(error => {
        console.error('Error adding note', error);
      });
  };

  return (
    <div>
      <h2>Add Note</h2>
      <form onSubmit={handleSubmit}>
        <input type="text" placeholder="Title" value={title} onChange={(e) =>
```

```
setTitle(e.target.value)} required />
      <textarea placeholder="Content" value={content} onChange={(e) =>
setContent(e.target.value)} required></textarea>
      <button type="submit">Add</button>
    </form>
  </div>
);
};

export default AddNoteForm;
```

app.js

```
import React from 'react';
import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
import NoteList from './NoteList';
import AddNoteForm from './AddNoteForm';
import { Link } from 'react-router-dom';

const App = () => {
  return (
    <Router>
      <div>
        <nav>
          <ul>
            <li><Link to="/">Home</Link></li>
            <li><Link to="/add">Add Note</Link></li>
          </ul>
        </nav>

        <Routes>
          <Route path="/" element={<NoteList />} />
          <Route path="/add" element={<AddNoteForm />} />
        </Routes>
      </div>
    </Router>
  );
};
```

```
export default App;
```

NoteList.js

```
import React, { useState, useEffect } from 'react';
import axios from 'axios';

const NoteList = () => {
  const [notes, setNotes] = useState([]);

  useEffect(() => {
    axios.get('/api/notes')
      .then(response => {
        setNotes(response.data);
      })
      .catch(error => {
        console.error('Error fetching notes', error);
      });
  }, []);

  const handleDelete = (id) => {
    axios.delete(`/api/notes/${id}`)
      .then(() => {
        setNotes(notes.filter(note => note.id !== id));
      })
      .catch(error => {
        console.error('Error deleting note', error);
      });
  };

  return (
    <div>
      <h1>My Notes</h1>
      <ul>
        {notes.map(note => (
          <li key={note.id}>
            <h3>{note.title}</h3>
            <p>{note.content}</p>
            <button onClick={() => handleDelete(note.id)}>Delete</button>
          </li>
        ))}
      </ul>
    </div>
  );
};
```

```
    )})  
  </ul>  
</div>  
  );  
};  
  
export default NoteList;
```

Server.js

```
const express = require('express');  
const { Pool } = require('pg');  
  
const app = express();  
const port = 5432; // Menggunakan port 3000 sebagai contoh  
  
// Connect to PostgreSQL  
const pool = new Pool({  
  user: 'your_username',  
  host: 'localhost',  
  database: 'your_database',  
  password: 'your_password',  
  port: 5432,  
});  
  
// Middleware  
app.use(express.json());  
  
// Routes  
app.get('/api/notes', async (req, res) => {  
  try {  
    const result = await pool.query('SELECT * FROM notes');  
    res.json(result.rows);  
  } catch (error) {  
    console.error('Error executing query', error);  
    res.status(500).json({ message: 'Internal Server Error' });  
  }  
});  
  
app.post('/api/notes', async (req, res) => {
```

```
const { title, content } = req.body;
try {
  const result = await pool.query('INSERT INTO notes (title, content)
VALUES ($1, $2) RETURNING *', [title, content]);
  res.json(result.rows[0]);
} catch (error) {
  console.error('Error executing query', error);
  res.status(500).json({ message: 'Internal Server Error' });
}
});

app.delete('/api/notes/:id', async (req, res) => {
  const id = req.params.id;
  try {
    await pool.query('DELETE FROM notes WHERE id = $1', [id]);
    res.json({ message: 'Note deleted successfully' });
  } catch (error) {
    console.error('Error executing query', error);
    res.status(500).json({ message: 'Internal Server Error' });
  }
});

app.listen(port, () => {
  console.log(`Server is running on port ${port}`);
});
```

```
Local:          http://localhost:3000
On Your Network: http://192.168.159.1:3000
```

Note that the development build is not optimized.
To create a production build, use `npm run build`.

```
webpack compiled successfully
```

```
PS D:\SBD\Tutam Modul 9\my-notes-app> node server.js
Server is running on port 5432
```

A screenshot of a web browser window. The address bar shows "localhost:3000". The page title is "Harga Memory". There are two links: "Home" and "Add Note". Below the links is a heading "My Notes". Further down, there is another set of links: "Home" and "Add Note". Below these links is a heading "Add Note". Under "Add Note", there is a form with two input fields: "Title" and "Content". To the right of the "Content" field is an "Add" button.

Query :

```
CREATE TABLE notes (  
    id SERIAL PRIMARY KEY,  
    title VARCHAR(255) NOT NULL,  
    content TEXT NOT NULL  
);
```

2. Link Youtube \Rightarrow <https://youtu.be/GxpGrnHg9Uk>

3. Link github

https://github.com/AbrisamYuhartono/Tutam_RD_Muhammad-Abrisam-Cahyo-Juhartono_2206026050_SBD9