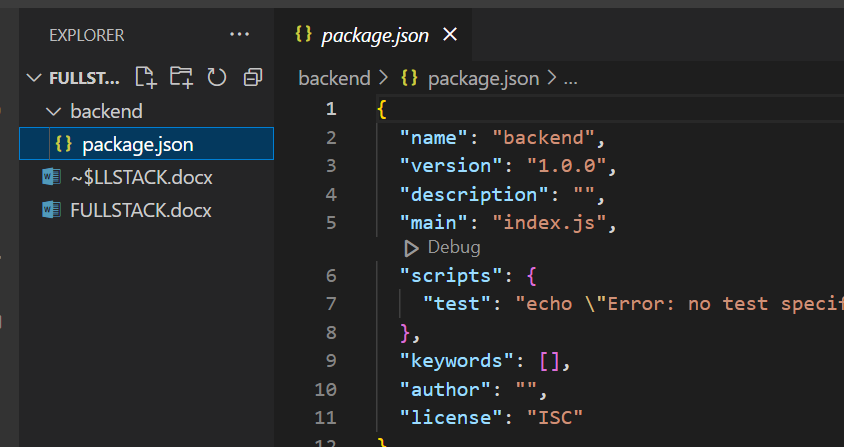
# FULLSTACK

# BACKEND

## Buat package.json dengan ketikkan perintah

|  |
| --- |
| npm init -y |



## Install dependensi yang dibutuhkan

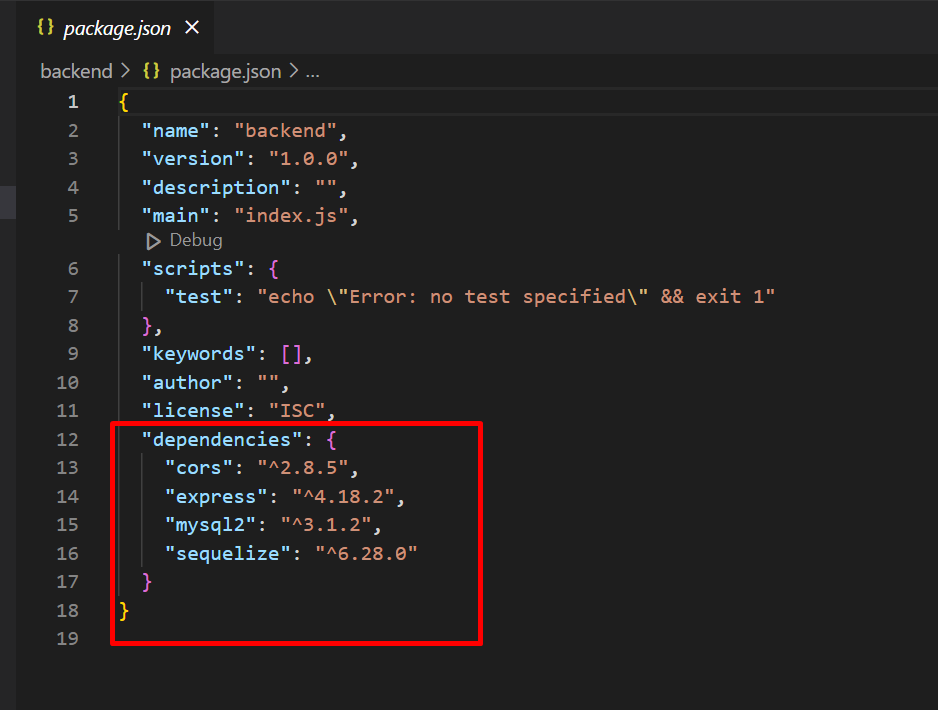
1. Express
2. Mysql klien
3. Sequelize

**promise-based ORM untuk nodejs**. ORM (Object Relational Maping) adalah sebuah tool yang kita gunakan sebagai cara kita melakukan interaksi dengan database piliha

1. Cors

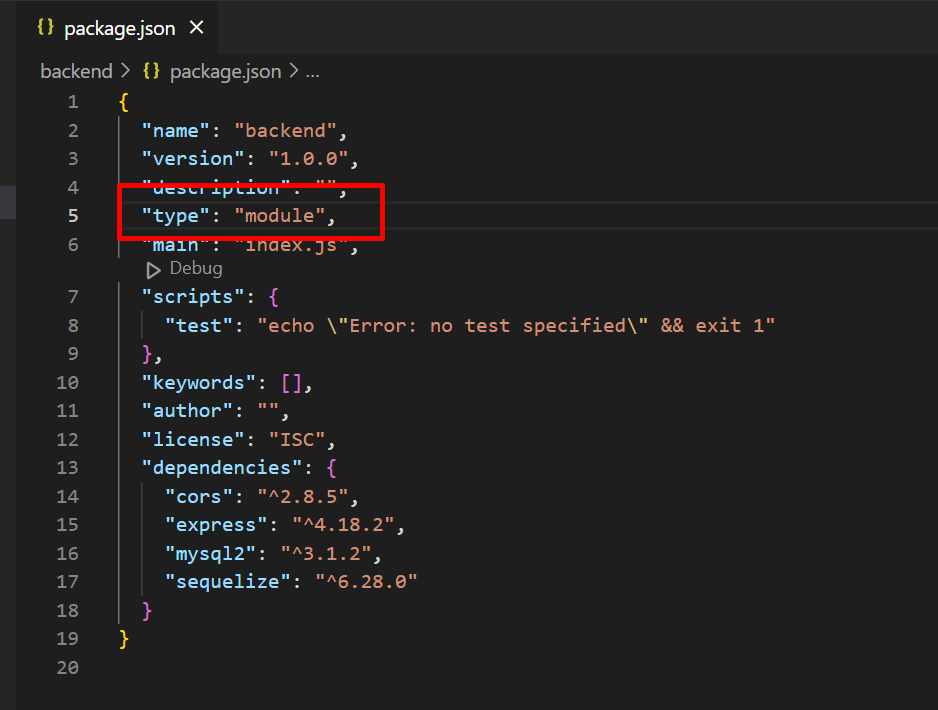
CORS (**Cross-Origin Resource Sharing**) adalah protokol yang dibangun di atas HTTP untuk memungkinkan Javascript pada halaman yang berasal dari satu situs untuk mengakses metode di situs lain.

|  |
| --- |
| Npm i express mysql2 sequelize cors |



## Tambahkan type module pada package.json

Fungsi : agar bisa menggunakan jsx syntax untuk import dan export module



## Add file Index.js di folder backend

import express from 'express';

import cors from 'cors';

const app = express();

// tambahkan middleware

app.use(cors());

app.use(express.json()); //agar bisa menerima response berupa json

app.listen(5000, () => console.log('Server up and running'));

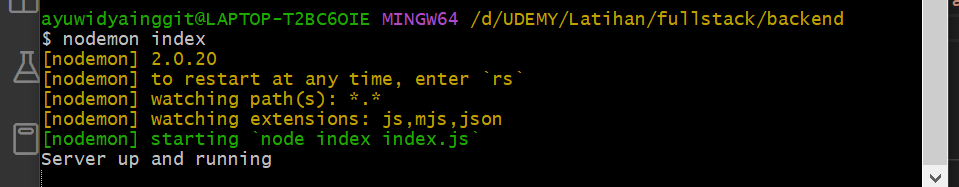
untuk memastika aplikasinya berjalan atau tdk jalan kan menggunakan nodemon

## Install nodemon

|  |
| --- |
| Npm I -g nodemon |

## Jalankan projek

|  |
| --- |
| Nodemon index |



Jika sudah muncul server up and running maka aplikasi sudah berjalan dengan baik

## Tambahkan koneksi

Tambahkan folder config di backend dan buat file baru Database.js

import { Sequelize } from 'sequelize';

const db = new Sequelize('crud\_db', 'root', '', {

  host: 'localhost',

  dialect: 'mysql',

});

export default db;

## Buat database

Buat database crud\_db di phpMyAdmin

## Buat struktur table

Buat folder models/UserModel.js

File ini berisi struktur table di mana data kita disimpan nantinya ke db

import { Sequelize } from 'sequelize';

import db from '../config/Database';

const { DataTypes } = Sequelize;

//DataTypes adalah fungsi dari sequelize

const User = db.define(

  'users',

  {

    name: DataTypes.STRING,

    email: DataTypes.STRING,

    gender: DataTypes.STRING,

  },

  {

    freezeTableName: true,

  }

);

export default user;

// membuat function (asyncronus) untuk generate table user jika user tdk terdapat di db

(async () => {

  await db.sync();

})();

## Buat folder controller/UserController.js

import User from '../models/UserModel';

export const getUsers = async (req, res) => {

  try {

    const response = await User.findAll();

    res.status(200).json(response);

  } catch (error) {

    console.log(error.message);

  }

};

## Buat folder routes/UserRoutes.js

import express from 'express';

import { getUsers } from '../controller/UserController';

const router = express.Router();

// buat endpoint baru

router.get('/users', getUsers);

export default router;

## Panggil router di index.js (untuk eksekusi create table user)

import express from 'express';

import cors from 'cors';

import UserRoute from './routes/UserRoute.js';

const app = express();

// tambahkan middleware

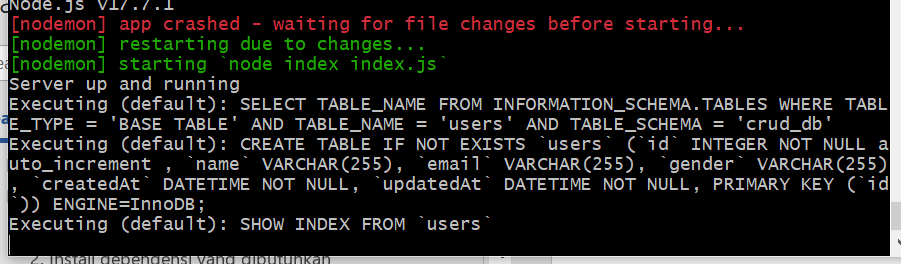
app.use(cors());

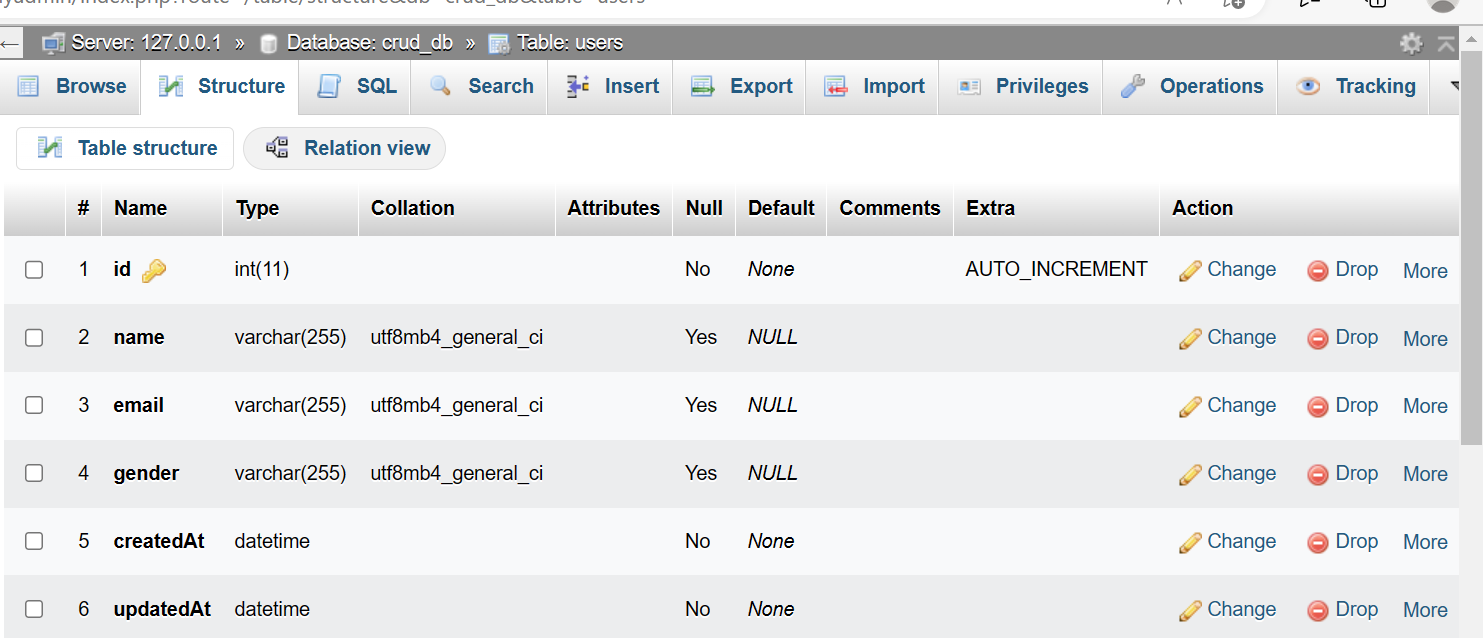
app.use(express.json()); //agar bisa menerima response berupa json

app.use(UserRoute); // panggil sebagai middleware

app.listen(5000, () => console.log('Server up and running'));

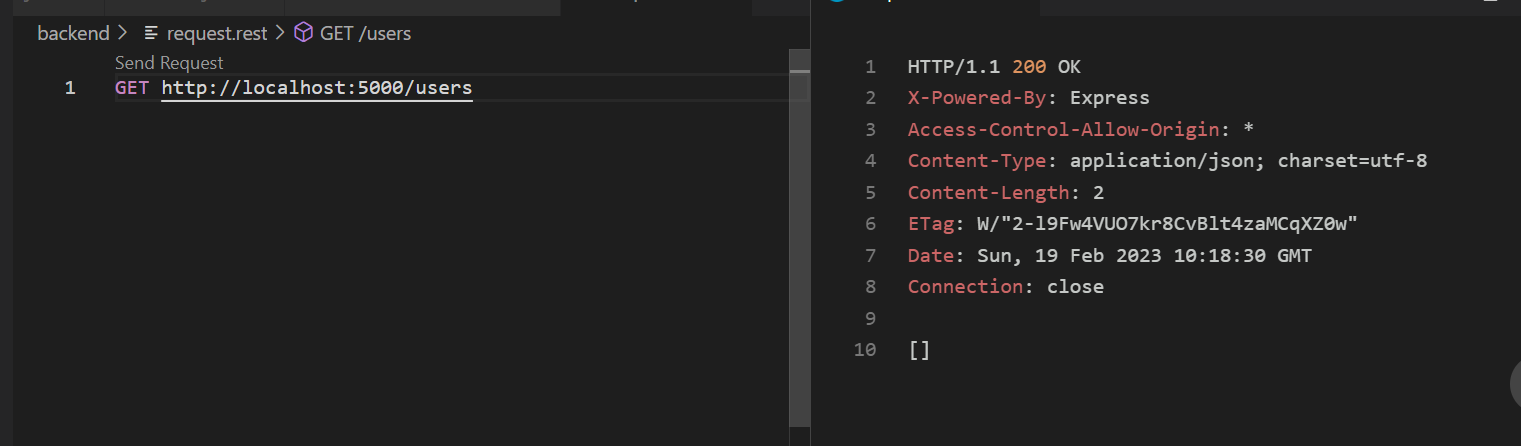
jika disimpan terdapat query yang di eksekusi untuk membuat table user





## Melakukan pengujian API apakah berjalan dengan baik

Install extention rest client



## Membuat method untuk get data byid

userController.js

import User from '../models/UserModel.js';

export const getUsers = async (req, res) => {

  try {

    const response = await User.findAll();

    res.status(200).json(response);

  } catch (error) {

    console.log(error.message);

  }

};

export const getUserById = async (req, res) => {

  try {

    const response = await User.findOne({

      where: {

        id: req.params.id,

      },

    });

    res.status(200).json(response);

  } catch (error) {

    console.log(error.message);

  }

};

UserRoute.js

import express from 'express';

import { getUsers, getUserById } from '../controller/UserController.js';

const router = express.Router();

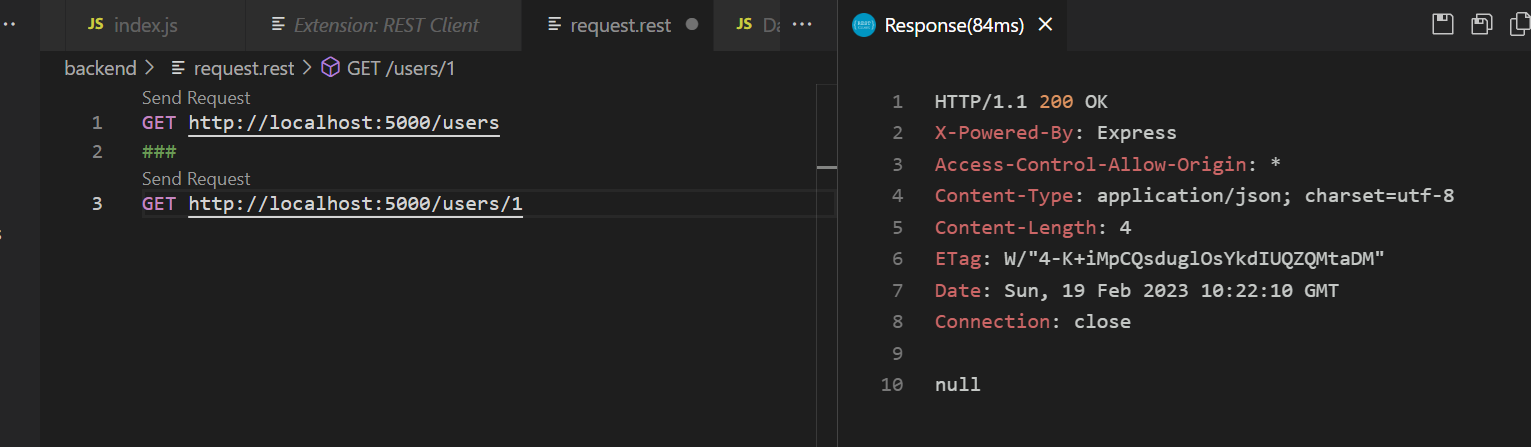
// buat endpoint baru

router.get('/users', getUsers);

router.get('/users/:id', getUserById);

export default router;

coba di request.rest



## Membuat method untuk create data

UserController.js

import User from '../models/UserModel.js';

export const getUsers = async (req, res) => {

  try {

    const response = await User.findAll();

    res.status(200).json(response);

  } catch (error) {

    console.log(error.message);

  }

};

export const getUserById = async (req, res) => {

  try {

    const response = await User.findOne({

      where: {

        id: req.params.id,

      },

    });

    res.status(200).json(response);

  } catch (error) {

    console.log(error.message);

  }

};

export const createUser = async (req, res) => {

  try {

    await User.create(req.body);

    res.status(201).json({ msg: 'User Created' });

  } catch (error) {

    console.log(error.message);

  }

};

User.Route.js

import express from 'express';

import { getUsers, getUserById, createUser } from '../controller/UserController.js';

const router = express.Router();

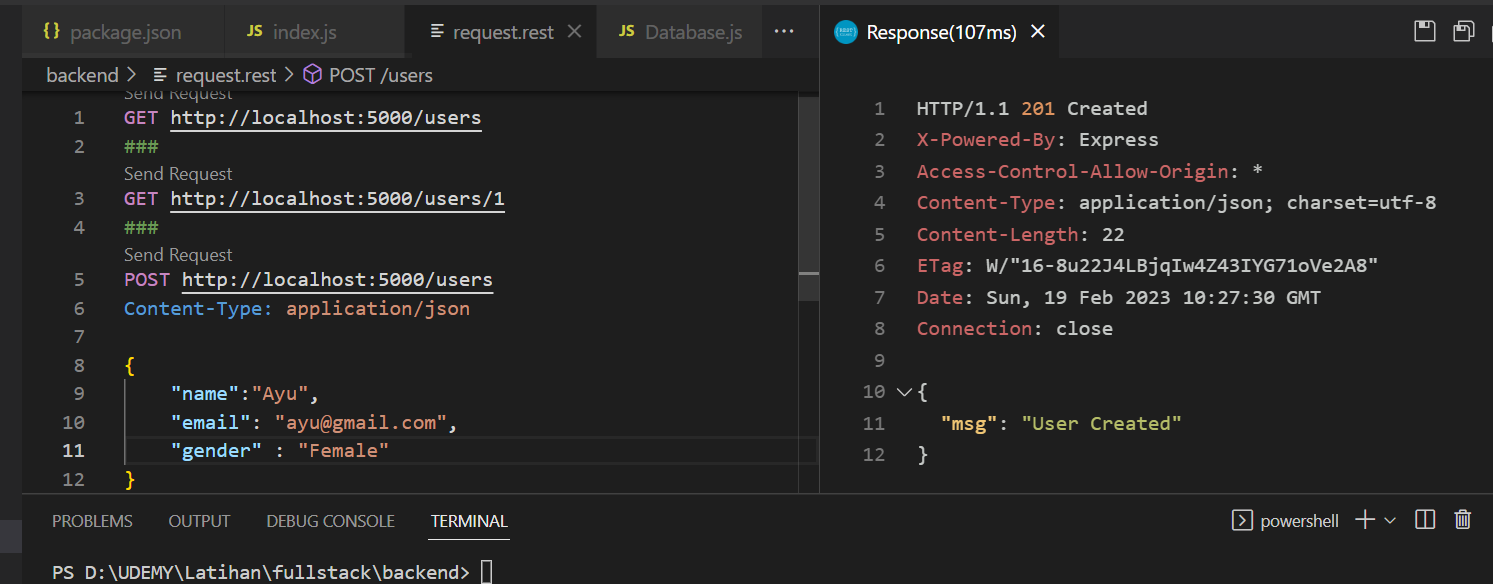
// buat endpoint baru

router.get('/users', getUsers);

router.get('/users/:id', getUserById);

router.post('/users', createUser);

export default router;



## Membuat method untuk Update

UserController.js

import User from '../models/UserModel.js';

export const getUsers = async (req, res) => {

  try {

    const response = await User.findAll();

    res.status(200).json(response);

  } catch (error) {

    console.log(error.message);

  }

};

export const getUserById = async (req, res) => {

  try {

    const response = await User.findOne({

      where: {

        id: req.params.id,

      },

    });

    res.status(200).json(response);

  } catch (error) {

    console.log(error.message);

  }

};

export const createUser = async (req, res) => {

  try {

    await User.create(req.body);

    res.status(201).json({ msg: 'User Created' });

  } catch (error) {

    console.log(error.message);

  }

};

export const updateUser = async (req, res) => {

  try {

    await User.update(req.body, {

      where: {

        id: req.params.id,

      },

    });

    res.status(200).json({ msg: 'User Updated' });

  } catch (error) {

    console.log(error.message);

  }

};

export const deleteUser = async (req, res) => {

  try {

    await User.destroy({

      where: {

        id: req.params.id,

      },

    });

    res.status(200).json({ msg: 'User Deleted' });

  } catch (error) {

    console.log(error.message);

  }

};

UserRoute.js

import express from 'express';

import { getUsers, getUserById, createUser, updateUser, deleteUser } from '../controller/UserController.js';

const router = express.Router();

// buat endpoint baru

router.get('/users', getUsers);

router.get('/users/:id', getUserById);

router.post('/users', createUser);

router.patch('/users/:id', updateUser);

router.delete('/users/:id', deleteUser);

export default router;

# FRONTEND

## Create new project di folder fullstack

Npx create-react-app frontend

Npm I react-router-dom axios bulma

Class component = ketik (rafce)

## Buat folder component

### Buat file UserList.js

import React from 'react';

const UserList = () => {

  return (

    <div className="columns mt-5">

      <div className="column is-half">

        <h1>Lorem ipsum dolor sit amet.</h1>

      </div>

    </div>

  );

};

export default UserList;

import di App.js

import UserList from './components/UserList';

function App() {

  return (

    <div>

      <UserList />

    </div>

  );

}

export default App;

UserList.js

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

import axios from 'axios';

const UserList = () => {

  const [users, setUser] = useState([]);

  useEffect(() => {

    getUsers();

  }, []);

  const getUsers = async () => {

    const response = await axios.get('http://localhost:5000/users');

    setUser(response.data);

  };

  return (

    <div className="columns mt-5 is-centered">

      <div className="column is-half">

        <table className="table is-striped is-fullwidth">

          <thead>

            <tr>

              <th>No</th>

              <th>Name</th>

              <th>Email</th>

              <th>Gender</th>

              <th>Action</th>

            </tr>

          </thead>

          <tbody>

            {users.map((user, index) => (

              <tr key={user.id}>

                <td>{index + 1}</td>

                <td>{user.name}</td>

                <td>{user.email}</td>

                <td>{user.gender}</td>

                <td>

                  <button className="button is-small is-info">Edit</button>

                  <button className="button is-small is-danger">Delete</button>

                </td>

              </tr>

            ))}

          </tbody>

        </table>

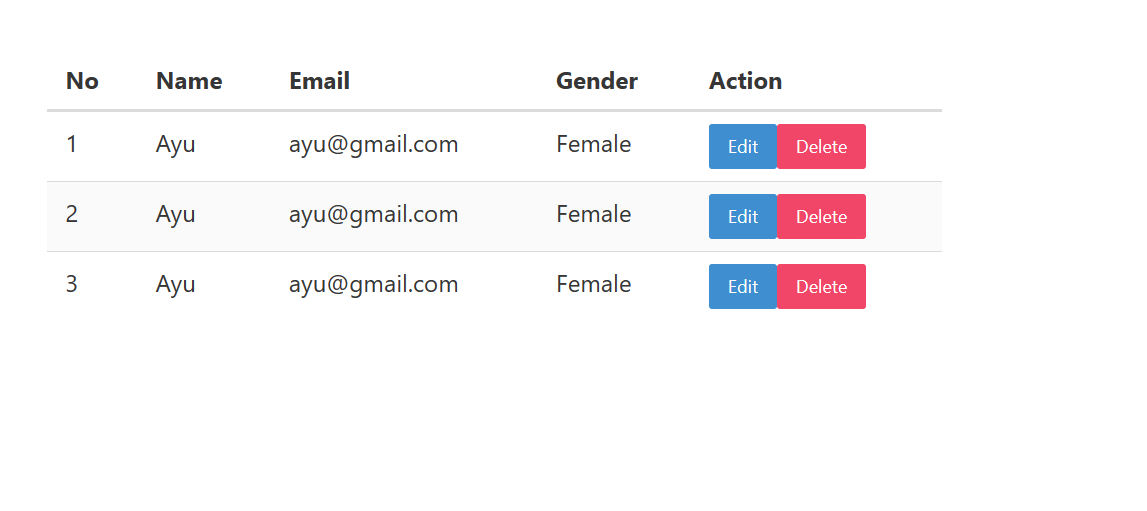
      </div>

    </div>

  );

};

export default UserList;



### Buat file AddUser.js di folder components

import React from 'react';

const AddUser = () => {

  return (

    <div className="columns mt-5 is-centered">

      <div className="column is-half">

        <form action="">

          <div className="field">

            <label className="label">Name</label>

          </div>

          <div className="control">

            <input type="text" className="input" placeholder="name" />

          </div>

          <div className="field">

            <label className="label">Email</label>

          </div>

          <div className="control">

            <input type="text" className="input" placeholder="email" />

          </div>

          <div className="field">

            <label className="label">Gender</label>

          </div>

          <div className="control">

            <div className="select is-fullwidth">

              <select name="" id="">

                <option value="Male">Male</option>

                <option value="Female">Female</option>

              </select>

            </div>

          </div>

          <div className="field">

            <button type="submit" className="button is-success">

              SAVE

            </button>

          </div>

        </form>

      </div>

    </div>

  );

};

export default AddUser;

panggil AddUser.js di App.js

import UserList from './components/UserList';

import { BrowserRouter, Routes, Route } from 'react-router-dom';

import AddUser from './components/AddUser';

function App() {

  return (

    <BrowserRouter>

      <Routes>

        <Route path="/" element={<UserList />} />

        <Route path="add" element={<AddUser />} />

      </Routes>

    </BrowserRouter>

  );

}

export default App;

### consume API di AddUser.js

import React, { useState } from 'react';

import axios from 'axios';

import { useNavigate } from 'react-router-dom'; //untuk redirect

const AddUser = () => {

  const [name, setName] = useState('');

  const [email, setEmail] = useState('');

  const [gender, setGender] = useState('Male');

  const navigate = useNavigate();

  const saveUser = async (e) => {

    e.preventDefault(); //digunakan untuk agar ketika di submit tdak reload

    try {

      await axios.post('http://localhost:5000/users', {

        name,

        email,

        gender,

      });

      navigate('/'); // ketika di simpan maka akan  redirect ke halaman home

    } catch (error) {

      console.log(error);

    }

  };

  return (

    <div className="columns mt-5 is-centered">

      <div className="column is-half">

        <form onSubmit={saveUser}>

          <div className="field">

            <label className="label">Name</label>

          </div>

          <div className="control">

            <input type="text" className="input" placeholder="name" value={name} onChange={(e) => setName(e.target.value)} />

          </div>

          <div className="field">

            <label className="label">Email</label>

          </div>

          <div className="control">

            <input type="text" className="input" placeholder="email" value={email} onChange={(e) => setEmail(e.target.value)} />

          </div>

          <div className="field">

            <label className="label">Gender</label>

          </div>

          <div className="control">

            <div className="select is-fullwidth">

              <select name="" id="" value={gender} onChange={(e) => setGender(e.target.value)}>

                <option value="Male">Male</option>

                <option value="Female">Female</option>

              </select>

            </div>

          </div>

          <div className="field">

            <button type="submit" className="button is-success">

              SAVE

            </button>

          </div>

        </form>

      </div>

    </div>

  );

};

export default AddUser;

kita sudah bisa menambahkan data

### Buat component EditUser.js

EditUser.js

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

import axios from 'axios';

import { useNavigate, useParams } from 'react-router-dom'; //useNavigate untuk redirect , useParam untuk mengambil id dr parameter

const EditUser = () => {

  const [name, setName] = useState('');

  const [email, setEmail] = useState('');

  const [gender, setGender] = useState('Male');

  const navigate = useNavigate();

  const { id } = useParams();

  //   untuk memanggil function getUserById tambahkan useEffect

  useEffect(() => {

    getUserById();

  }, []);

  const updateUser = async (e) => {

    e.preventDefault(); //digunakan untuk agar ketika di submit tdak reload

    try {

      await axios.patch(`http://localhost:5000/users/${id}`, {

        name,

        email,

        gender,

      });

      navigate('/'); // ketika di simpan maka akan  redirect ke halaman home

    } catch (error) {

      console.log(error);

    }

  };

  const getUserById = async () => {

    const response = await axios.get(`http://localhost:5000/users/${id}`);

    setName(response.data.name);

    setEmail(response.data.email);

    setGender(response.data.gender);

  };

  return (

    <div className="columns mt-5 is-centered">

      <div className="column is-half">

        <form onSubmit={updateUser}>

          <div className="field">

            <label className="label">Name</label>

          </div>

          <div className="control">

            <input type="text" className="input" placeholder="name" value={name} onChange={(e) => setName(e.target.value)} />

          </div>

          <div className="field">

            <label className="label">Email</label>

          </div>

          <div className="control">

            <input type="text" className="input" placeholder="email" value={email} onChange={(e) => setEmail(e.target.value)} />

          </div>

          <div className="field">

            <label className="label">Gender</label>

          </div>

          <div className="control">

            <div className="select is-fullwidth">

              <select name="" id="" value={gender} onChange={(e) => setGender(e.target.value)}>

                <option value="Male">Male</option>

                <option value="Female">Female</option>

              </select>

            </div>

          </div>

          <div className="field">

            <button type="submit" className="button is-success">

              Update

            </button>

          </div>

        </form>

      </div>

    </div>

  );

};

export default EditUser;

Panggil EditUser.js di App.js

import UserList from './components/UserList';

import { BrowserRouter, Routes, Route } from 'react-router-dom';

import AddUser from './components/AddUser';

import EditUser from './components/EditUser';

function App() {

  return (

    <BrowserRouter>

      <Routes>

        <Route path="/" element={<UserList />} />

        <Route path="add" element={<AddUser />} />

        <Route path="edit/:id" element={<EditUser />} />

      </Routes>

    </BrowserRouter>

  );

}

export default App;

### membuat tombol edit jika di klik akan mengarah ke halaman form EditUser.js

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

import axios from 'axios';

import { Link } from 'react-router-dom';

const UserList = () => {

  const [users, setUser] = useState([]);

  useEffect(() => {

    getUsers();

  }, []);

  const getUsers = async () => {

    const response = await axios.get('http://localhost:5000/users');

    setUser(response.data);

  };

  return (

    <div className="columns mt-5 is-centered">

      <div className="column is-half">

        <Link to={`add`} className="button is-succesed">

          Add New

        </Link>

        <table className="table is-striped is-fullwidth">

          <thead>

            <tr>

              <th>No</th>

              <th>Name</th>

              <th>Email</th>

              <th>Gender</th>

              <th>Action</th>

            </tr>

          </thead>

          <tbody>

            {users.map((user, index) => (

              <tr key={user.id}>

                <td>{index + 1}</td>

                <td>{user.name}</td>

                <td>{user.email}</td>

                <td>{user.gender}</td>

                <td>

                  <Link to={`edit/${user.id}`} className="button is-small is-info">

                    Edit

                  </Link>

                  <Link to={`delete/${user.id}`} className="button is-small is-danger">

                    Delete

                  </Link>

                </td>

              </tr>

            ))}

          </tbody>

        </table>

      </div>

    </div>

  );

};

export default UserList;

### Membuat fungsi delete

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

import axios from 'axios';

import { Link } from 'react-router-dom';

const UserList = () => {

  const [users, setUser] = useState([]);

  useEffect(() => {

    getUsers();

  }, []);

  const getUsers = async () => {

    const response = await axios.get('http://localhost:5000/users');

    setUser(response.data);

  };

  const deleteUser = async (id) => {

    try {

      alert('apakah ingin menghapus?');

      await axios.delete(`http://localhost:5000/users/${id}`);

      getUsers();

    } catch (error) {

      console.log(error);

    }

  };

  return (

    <div className="columns mt-5 is-centered">

      <div className="column is-half">

        <Link to={`add`} className="button is-succesed">

          Add New

        </Link>

        <table className="table is-striped is-fullwidth">

          <thead>

            <tr>

              <th>No</th>

              <th>Name</th>

              <th>Email</th>

              <th>Gender</th>

              <th>Action</th>

            </tr>

          </thead>

          <tbody>

            {users.map((user, index) => (

              <tr key={user.id}>

                <td>{index + 1}</td>

                <td>{user.name}</td>

                <td>{user.email}</td>

                <td>{user.gender}</td>

                <td>

                  <Link to={`edit/${user.id}`} className="button is-small is-info">

                    Edit

                  </Link>

                  <button onClick={() => deleteUser(user.id)} className="button is-small is-danger">

                    Delete

                  </button>

                </td>

              </tr>

            ))}

          </tbody>

        </table>

      </div>

    </div>

  );

};

export default UserList;