DOKUMENTASI APLIKASI

”USER MANAGEMENT”

A logo with text on it

Description automatically generated

Disusun oleh:

Yohanes Dimas Pratama

A11.2021.13254

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS DIAN NUSWANTORO

Daftar Isi

[Bab 1 - Demo Aplikasi 3](#_Toc171453552)

[1.1 Link Deployment 3](#_Toc171453553)

[1.2 Link GitHub 3](#_Toc171453554)

[1.3 Halaman Home 3](#_Toc171453555)

[1.4 Halaman Data User 3](#_Toc171453556)

[Bab 2 - Test Fitur 4](#_Toc171453557)

[2.1 Tampil Data 4](#_Toc171453558)

[2.2 Tambah Data 5](#_Toc171453559)

[2.3 Edit Data 6](#_Toc171453560)

[2.4 Hapus Data 8](#_Toc171453561)

[Bab 3 - Source Code 9](#_Toc171453562)

[3.1 App.js 9](#_Toc171453563)

[3.2 Home.js 9](#_Toc171453564)

[3.3 Users.js 11](#_Toc171453565)

[3.4 UserContext.js 14](#_Toc171453566)

[3.5 Actions.js 15](#_Toc171453567)

[3.6 Reducers.js 16](#_Toc171453568)

[3.7 Store.js 16](#_Toc171453569)

[3.8 AddUserForm.js 17](#_Toc171453570)

[3.9 UserCard.jss 19](#_Toc171453571)

[3.10 Navbar.js 20](#_Toc171453572)

[3.11 CustomCard.js 21](#_Toc171453573)

# Bab 1 - Demo Aplikasi

## 1.1 Link Deployment

<https://user-management-app-kappa.vercel.app/>

## 1.2 Link GitHub

<https://github.com/weztcy/User-Management-App>

## 1.3 Halaman Home

A screenshot of a computer

Description automatically generated

## 1.4 Halaman Data User

A screenshot of a computer

Description automatically generated

# Bab 2 - Test Fitur

## 2.1 Tampil Data

Melihat data user dengan menekan tombol lihat data pengguna pada card di halaman home atau menekan menu users pada navbar.

A screenshot of a computer

Description automatically generated

Sistem menampilkan data users pada halaman users. Data users diambil dengan cara melakukan fetch API. API yang digunakan adalah <https://jsonplaceholder.typicode.com/> .

A screenshot of a computer

Description automatically generated

## 2.2 Tambah Data

Mengisi form tambah data users sesuai dengan keinginan, lalu menekan tombol add user.

A screenshot of a computer

Description automatically generated

Data users yang baru berhasil ditambahkan.

A screenshot of a computer

Description automatically generated

## 2.3 Edit Data

Mengganti data users dari Leanne Graham dengan menekan tombol edit data.

A screenshot of a computer

Description automatically generated

Menampilkan form edit data users.

A screenshot of a computer

Description automatically generated

Mengganti isi form dengan data baru, lalu klik save changes.

A screenshot of a computer

Description automatically generated

Data users dari Leanne Graham berhasil diganti dengan data baru dari Joseph Albert.

A screenshot of a computer

Description automatically generated

## 2.4 Hapus Data

Menghapus data users dari Nicholas Runolfsdottir V dengan menekan tombol delete.

A screenshot of a computer

Description automatically generated

Data users dari Nicholas Runolfsdottir V berhasil dihapus.

A screenshot of a computer

Description automatically generated

# Bab 3 - Source Code

## 3.1 App.js

import React from "react";

import { BrowserRouter as Router, Route, Routes } from "react-router-dom";

import { Provider } from "react-redux";

import Navigation from "./components/Navbar";

import Home from "./pages/Home";

import Users from "./pages/Users";

import { UserProvider } from "./context/UserContext";

import store from "./redux/store";

const App = () => {

  return (

    <Provider store={store}>

      <UserProvider>

        <Router>

          <div className="App bg-dark" style={{ minHeight: "100vh" }}>

            <Navigation />

            <Routes>

              <Route exact path="/" element={<Home />} />

              <Route path="/users" element={<Users />} />

            </Routes>

          </div>

        </Router>

      </UserProvider>

    </Provider>

  );

};

export default App;

## 3.2 Home.js

import React from "react";

import { Container, Row, Col, Button, Card } from "react-bootstrap";

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

import heroImage from "../assets/hero-image.jpg";

import mhsPhoto from "../assets/mhs.jpg";

import CustomCard from "../components/CustomCard";

const Home = () => {

  return (

    <div className="container mt-5">

      <Container>

        <Row className="align-items-center">

          <Col md={6}>

            <h1 style={{ color: "white" }}>Welcome to My App!!</h1>

            <br />

            <CustomCard

              title="Yohanes Dimas Pratama"

              textNIM="A11.2021.13254"

              textKampus="Universitas Dian Nuswantoro"

              imageSrc={mhsPhoto}

              style={{ margin: "auto" }}

            />

          </Col>

          <Col md={6} className="mt-4 mt-md-0">

            <Card>

              <Card.Img variant="top" src={heroImage} alt="Hero Image" />

              <Card.Body>

                <Card.Title className="text-center">

                  What is User Management App?

                </Card.Title>

                <br />

                <Card.Text className="text-center mx-5">

                  This application is designed to provide an intuitive and

                  enjoyable experience in user data management. Using React and

                  React Bootstrap technology, this application not only makes it

                  easier for users to add user data, but also provides an

                  interactive experience in displaying user information

                  efficiently and attractively.

                </Card.Text>

                <br />

                <div className="text-end">

                  <Button

                    as={Link}

                    to="/users"

                    variant="info"

                    className="float-end"

                  >

                    Lihat Data Pengguna

                  </Button>

                </div>

              </Card.Body>

            </Card>

          </Col>

        </Row>

      </Container>

    </div>

  );

};

export default Home;

## 3.3 Users.js

import React, { useContext, useEffect, useState } from "react";

import { useDispatch, useSelector } from "react-redux";

import UserCard from "../components/UserCard";

import { UserContext } from "../context/UserContext";

import {

  setUsers,

  deleteUser,

  editUser as editUserAction,

} from "../redux/actions";

import AddUserForm from "../components/AddUserForm";

import {

  Container,

  Row,

  Col,

  Modal,

  Form,

  Button,

  Pagination,

  Card,

} from "react-bootstrap";

const Users = () => {

  const { users } = useContext(UserContext);

  const dispatch = useDispatch();

  const reduxUsers = useSelector((state) => state.users);

  const [editUser, setEditUser] = useState(null);

  const [currentPage, setCurrentPage] = useState(1);

  const usersPerPage = 6;

  useEffect(() => {

    dispatch(setUsers(users));

  }, [users, dispatch]);

  const handleDeleteUser = (userId) => {

    dispatch(deleteUser(userId));

  };

  const handleEditUser = (user) => {

    setEditUser(user);

  };

  const handleCloseEditModal = () => {

    setEditUser(null);

  };

  const handleSaveEditedUser = () => {

    dispatch(editUserAction(editUser));

    handleCloseEditModal();

  };

  const indexOfLastUser = currentPage \* usersPerPage;

  const indexOfFirstUser = indexOfLastUser - usersPerPage;

  const currentUsers = reduxUsers.slice(indexOfFirstUser, indexOfLastUser);

  const totalPages = Math.ceil(reduxUsers.length / usersPerPage);

  const paginate = (pageNumber) => setCurrentPage(pageNumber);

  const handleNextPage = () => {

    if (currentPage < totalPages) {

      setCurrentPage(currentPage + 1);

    }

  };

  const handlePrevPage = () => {

    if (currentPage > 1) {

      setCurrentPage(currentPage - 1);

    }

  };

  return (

    <Container>

      <Row style={{ marginTop: "100px" }}>

        <Col md={4}>

          <AddUserForm />

        </Col>

        <Col md={8}>

          <Row>

            <h1

              style={{ color: "white", textAlign: "center", fontSize: "30px" }}

            >

              List Data User

            </h1>

            {currentUsers.map((user) => (

              <Col key={user.id} sm={12} md={6} lg={4}>

                <UserCard

                  user={user}

                  onDelete={handleDeleteUser}

                  onEdit={handleEditUser}

                />

              </Col>

            ))}

          </Row>

          <Pagination className="justify-content-center mt-2">

            <Pagination.Prev

              onClick={handlePrevPage}

              disabled={currentPage === 1}

              style={{ marginRight: "10px" }}

            >

              Sebelumnya

            </Pagination.Prev>

            {[...Array(totalPages)].map((\_, index) => (

              <Pagination.Item

                key={index + 1}

                active={index + 1 === currentPage}

                onClick={() => paginate(index + 1)}

              >

                {index + 1}

              </Pagination.Item>

            ))}

            <Pagination.Next

              onClick={handleNextPage}

              disabled={currentPage === totalPages}

              style={{ marginLeft: "10px" }}

            >

              Selanjutnya

            </Pagination.Next>

          </Pagination>

        </Col>

      </Row>

      <Modal show={!!editUser} onHide={handleCloseEditModal}>

        <Modal.Header closeButton>

          <Modal.Title>Edit User</Modal.Title>

        </Modal.Header>

        <Modal.Body>

          <Form>

            <Form.Group className="mb-3" controlId="formName">

              <Form.Label>Name</Form.Label>

              <Form.Control

                type="text"

                placeholder="Enter name"

                value={editUser?.name || ""}

                onChange={(e) =>

                  setEditUser({ ...editUser, name: e.target.value })

                }

                required

              />

            </Form.Group>

            <Form.Group className="mb-3" controlId="formEmail">

              <Form.Label>Email address</Form.Label>

              <Form.Control

                type="email"

                placeholder="Enter email"

                value={editUser?.email || ""}

                onChange={(e) =>

                  setEditUser({ ...editUser, email: e.target.value })

                }

                required

              />

            </Form.Group>

            <Form.Group className="mb-3" controlId="formCity">

              <Form.Label>City</Form.Label>

              <Form.Control

                type="text"

                placeholder="Enter city"

                value={editUser?.address.city || ""}

                onChange={(e) =>

                  setEditUser({

                    ...editUser,

                    address: { ...editUser.address, city: e.target.value },

                  })

                }

                required

              />

            </Form.Group>

            <Button variant="primary" onClick={handleSaveEditedUser}>

              Save Changes

            </Button>

          </Form>

        </Modal.Body>

      </Modal>

    </Container>

  );

};

export default Users;

## 3.4 UserContext.js

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

export const UserContext = createContext();

export const UserProvider = ({ children }) => {

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

  useEffect(() => {

    const fetchUsers = async () => {

      const response = await fetch('https://jsonplaceholder.typicode.com/users');

      const data = await response.json();

      setUsers(data);

    };

    fetchUsers();

  }, []);

  return (

    <UserContext.Provider value={{ users, setUsers }}>

      {children}

    </UserContext.Provider>

  );

};

## 3.5 Actions.js

export const SET\_USERS = "SET\_USERS";

export const ADD\_USER = "ADD\_USER";

export const DELETE\_USER = "DELETE\_USER";

export const EDIT\_USER = "EDIT\_USER";

export const setUsers = (users) => ({

  type: SET\_USERS,

  payload: users,

});

export const addUser = (user) => ({

  type: ADD\_USER,

  payload: user,

});

export const deleteUser = (userId) => ({

  type: DELETE\_USER,

  payload: userId,

});

export const editUser = (user) => ({

  type: EDIT\_USER,

  payload: user,

});

## 3.6 Reducers.js

import { SET\_USERS, ADD\_USER, DELETE\_USER, EDIT\_USER } from "./actions";

const initialState = {

  users: [],

};

const rootReducer = (state = initialState, action) => {

  switch (action.type) {

    case SET\_USERS:

      return {

        ...state,

        users: action.payload,

      };

    case ADD\_USER:

      return {

        ...state,

        users: [...state.users, action.payload],

      };

    case DELETE\_USER:

      return {

        ...state,

        users: state.users.filter((user) => user.id !== action.payload),

      };

    case EDIT\_USER:

      return {

        ...state,

        users: state.users.map((user) =>

          user.id === action.payload.id ? { ...user, ...action.payload } : user

        ),

      };

    default:

      return state;

  }

};

export default rootReducer;

## 3.7 Store.js

import { createStore, applyMiddleware } from "redux";

import { thunk } from "redux-thunk";

import rootReducer from "./reducers";

const store = createStore(rootReducer, applyMiddleware(thunk));

export default store;

## 3.8 AddUserForm.js

import React, { useState, useContext } from "react";

import { useDispatch } from "react-redux";

import { addUser } from "../redux/actions";

import { UserContext } from "../context/UserContext";

import { Form, Button, Card } from "react-bootstrap";

const AddUserForm = () => {

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

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

  const [city, setCity] = useState("");

  const [phone, setPhone] = useState("");

  const [website, setWebsite] = useState("");

  const { users, setUsers } = useContext(UserContext);

  const dispatch = useDispatch();

  const handleSubmit = (e) => {

    e.preventDefault();

    const newUser = {

      id: users.length + 1,

      name,

      email,

      address: {

        city,

      },

      phone,

      website,

    };

    setUsers([...users, newUser]);

    dispatch(addUser(newUser));

    setName("");

    setEmail("");

    setCity("");

    setPhone("");

    setWebsite("");

  };

  return (

    <Card className="my-3 add-user-form">

      <Card.Body>

        <Card.Title

          primary

          style={{ color: "#0d6efd", textAlign: "center", fontSize: "30px" }}

        >

          Add Data User

        </Card.Title>

        <Form

          onSubmit={handleSubmit}

          style={{ color: "black", marginTop: "10px" }}

        >

          <Form.Group className="mb-3" controlId="formName">

            <Form.Label>Name</Form.Label>

            <Form.Control

              type="text"

              placeholder="Enter name"

              value={name}

              onChange={(e) => setName(e.target.value)}

              required

            />

          </Form.Group>

          <Form.Group className="mb-3" controlId="formEmail">

            <Form.Label>Email address</Form.Label>

            <Form.Control

              type="email"

              placeholder="Enter email"

              value={email}

              onChange={(e) => setEmail(e.target.value)}

              required

            />

          </Form.Group>

          <Form.Group className="mb-3" controlId="formCity">

            <Form.Label>City</Form.Label>

            <Form.Control

              type="text"

              placeholder="Enter city"

              value={city}

              onChange={(e) => setCity(e.target.value)}

              required

            />

          </Form.Group>

          <Form.Group className="mb-3" controlId="formPhone">

            <Form.Label>Phone</Form.Label>

            <Form.Control

              type="text"

              placeholder="Enter phone"

              value={phone}

              onChange={(e) => setPhone(e.target.value)}

              required

            />

          </Form.Group>

          <Form.Group className="mb-3" controlId="formWebsite">

            <Form.Label>Website</Form.Label>

            <Form.Control

              type="text"

              placeholder="Enter website"

              value={website}

              onChange={(e) => setWebsite(e.target.value)}

            />

          </Form.Group>

          <Button

            variant="success"

            type="submit"

            style={{ marginTop: "10px", width: "100%" }}

          >

            Add User

          </Button>

        </Form>

      </Card.Body>

    </Card>

  );

};

export default AddUserForm;

## 3.9 UserCard.jss

import React from "react";

import { Card, Button } from "react-bootstrap";

import "./UserCard.css";

const UserCard = ({ user, onDelete, onEdit }) => {

  if (!user) {

    return null;

  }

  const handleEditClick = () => {

    onEdit(user);

  };

  return (

    <Card className="my-3">

      <Card.Body>

        <Card.Title>{user.name}</Card.Title>

        <Card.Subtitle className="mb-2 text-muted">{user.email}</Card.Subtitle>

        <Card.Text>

          <strong>City:</strong> {user.address.city}

          <br />

          <strong>Phone:</strong> {user.phone}

          <br />

          <strong>Website:</strong> {user.website}

        </Card.Text>

        <div className="button-container">

          <Button

            className="edit-button"

            variant="primary"

            onClick={handleEditClick}

          >

            Edit

          </Button>

          <Button

            className="delete-button"

            variant="danger"

            onClick={() => onDelete(user.id)}

          >

            Delete

          </Button>

        </div>

      </Card.Body>

    </Card>

  );

};

export default UserCard;

## 3.10 Navbar.js

import Container from "react-bootstrap/Container";

import Nav from "react-bootstrap/Nav";

import Navbar from "react-bootstrap/Navbar";

import NavDropdown from "react-bootstrap/NavDropdown";

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

function Navigation() {

  return (

    <Navbar expand="lg" className="bg-body-tertiary">

      <Container>

        <Navbar.Brand as={Link} to="/" className="mx-auto">

          User Management App

        </Navbar.Brand>

        <Navbar.Toggle aria-controls="basic-navbar-nav" />

        <Navbar.Collapse id="basic-navbar-nav">

          <Nav className="mx-auto">

            <Nav.Link as={Link} to="/" className="mx-2">

              Home

            </Nav.Link>

            <Nav.Link as={Link} to="/users" className="mx-2">

              Users

            </Nav.Link>

            <NavDropdown

              title="Another Menu"

              id="basic-nav-dropdown"

              className="mx-2"

            >

              <NavDropdown.Item href="#action/3.1">

                Coming Soon

              </NavDropdown.Item>

              <NavDropdown.Divider />

              <NavDropdown.Item href="#action/3.2">

                Coming Soon

              </NavDropdown.Item>

              <NavDropdown.Divider />

              <NavDropdown.Item href="#action/3.3">

                Coming Soon

              </NavDropdown.Item>

              <NavDropdown.Divider />

              <NavDropdown.Item href="#action/3.4">

                Coming Soon

              </NavDropdown.Item>

            </NavDropdown>

          </Nav>

          <Navbar.Text>

            Mata Kuliah: <a href="#login">Pemrograman Sisi Client</a>

          </Navbar.Text>

        </Navbar.Collapse>

      </Container>

    </Navbar>

  );

}

export default Navigation;

## 3.11 CustomCard.js

import React from "react";

import Card from "react-bootstrap/Card";

function CustomCard({ title, textNIM, textKampus, imageSrc }) {

  return (

    <Card style={{ width: "18rem" }}>

      <Card.Img variant="top" src={imageSrc} />

      <Card.Body>

        <Card.Title className="text-center" style={{ fontSize: "23px" }}>

          {title}

        </Card.Title>

        <br />

        <Card.Text className="text-center" style={{ fontSize: "16px" }}>

          {textNIM}

        </Card.Text>

        <Card.Text

          className="text-center"

          style={{ fontSize: "16px", marginTop: "-10px" }}

        >

          {textKampus}

        </Card.Text>

      </Card.Body>

    </Card>

  );

}

export default CustomCard;