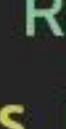


✓	src	
✓	Components	
JS	Button.js	U
JS	Card.js	U
JS	Footer.js	U
JS	Form.js	U
JS	Heading.js	U
JS	Input.js	U
JS	Logout.js	U
JS	Modal.js	U
JS	ProgressBar.js	U
JS	Sidebar.js	U
✓	Data	
JS	Dummy.js	U
JS	DummyModul.js	U
✓	Layouts	
JS	AdminLayout.js	U
JS	AuthLayout.js	U
✓	Pages	
✓	Admin	
✓	Modul	
JS	AccordionItem.js	U
JS	ModalTanya.js	U
JS	Modul.js	U
JS	Dashboard.js	U
✓	Auth	
	 gambar.png	U
JS	Login.js	U
✓	Routes	
JS	Router.js	U
✓	Utils/Helpers	
JS	SwalHelpers.js	U
JS	ToatsHelpers.js	U
JS	App.js	M
#	index.css	M
JS	index.js	M



Belajar Pintar

Platform pembelajaran digital untuk mahasiswa

Email

 evafadhillahulia22@gmail.com

Password



Masuk

Belajar Pintar

localhost:3000/admin/dashboard

Apply Activity | Me... Digital Talent Scho... (154) Belajar Java... Home | AWS Back... Telegram Evafadhillahulia at... Apple Developer iOS Academy - Yo... WhatsApp Learn Swift Progra... All Bookmarks

Logout

Belajar Pintar

Dashboard

Modul

Login Berhasil!

Selamat datang, Eva Fadhillah Ulia!

evafadhillahulia22@gmail.com

Progress Belajar 0%

Lanjutkan Belajar

© 2025 Admin Dashboard. All rights reserved.

 Dashboard

 Modul

Selamat datang, Eva Fadhillah Ulia!

evafadhillahulia22@gmail.com

Progress Belajar

0%

Lanjutkan Belajar

 Dashboard Modul

Modul Pembelajaran

Progress Belajar

0%

Pengenalan React

Dasar-dasar React dan Komponen

[Tandai Selesai](#)[Tanya Dosen](#)

State dan Props

Penjelasan tentang State, Props dan One Way Data Flow

[Tandai Selesai](#)[Tanya Dosen](#)

React Router DOM

Navigasi antar halaman menggunakan Routing

[Tandai Selesai](#)[Tanya Dosen](#)

Dashboard

Modul

Modul Pembelajaran

Progress Belajar

67%

Pengenalan React

Dasar-dasar React dan Komponen

SelesaiTanya Dosen

State dan Props



Sukses

Modul berhasil ditandai selesai!

OK

React Router DOM

Navigasi antar halaman menggunakan Routing

Tandai SelesaiTanya Dosen

 Dashboard

 Modul

Selamat datang, Eva Fadhillah Ulia!

evafadhillahulia22@gmail.com

Progress Belajar

67%

Lanjutkan Belajar

Dashboard

Modul

Modul Pembelajaran

Progress Belajar

67%

Pengenalan React

Dasar-dasar React dan Komponen

SelesaiTanya Dosen

State dan Props

Tanya Dosen



Tanyakan pertanyaanmu...

BatalKirim

One Way Data

React Router DOM

Navigasi antar halaman menggunakan Routing

Tandai SelesaiTanya Dosen

Belajar Pintar

localhost:3000/admin/modul

Apply Activity | Me... Digital Talent Scho... (154) Belajar Java... Home | AWS Back... Telegram Evafadhillahulia at... Apple Developer iOS Academy - Yo... WhatsApp Learn Swift Progra... All Bookmarks

Logout

Belajar Pintar

Dashboard

Modul

Modul Pembelajaran

Progress Belajar 67%

Pengenalan React

Dasar-dasar React dan Komponen

Selesai Tanya Dosen

State dan Props

One Way Data

React Router DOM

Navigasi antar halaman menggunakan Routing

Tandai Selesai Tanya Dosen

Tanya Dosen

Tanyakan pertanyaanmu...

Batal Kirim

© 2025 Admin Dashboard. All rights reserved.

Dashboard

Modul

Modul Pembelajaran

Progress Belajar

67%

Pengenalan React

Dasar-dasar React dan Komponen

SelesaiTanya Dosen

State dan Props

Tanya Dosen

Apa perbedaan utama antara komponen fungsi (functional component) dan komponen kelas (class component) di React?

BatalKirim

Belajar Pintar

localhost:3000/admin/modul

Apply Activity | Me... Digital Talent Scho... (154) Belajar Java... Home | AWS Back... Telegram Evafadhillahulia at... Apple Developer iOS Academy - Yo... WhatsApp Learn Swift Progra... All Bookmarks

Logout

Belajar Pintar

Dashboard

Modul

Pertanyaan berhasil dikirim

Modul Pembelajaran

Progress Belajar 67%

Pengenalan React
Dasar-dasar React dan Komponen
[Selesai](#) [Tanya Dosen](#)

State dan Props
Penjelasan tentang State, Props dan One Way Data Flow
[Selesai](#) [Tanya Dosen](#)

React Router DOM
Navigasi antar halaman menggunakan Routing
[Tandai Selesai](#) [Tanya Dosen](#)

© 2025 Admin Dashboard. All rights reserved.

 Dashboard Modul

Modul Pembelajaran

Progress Belajar

67%

Pengenalan React

Dasar-dasar React dan Komponen

[Selesai](#)[Tanya Dosen](#)

State dan Props

Belajar tentang State dan Props dalam React



Apakah Anda yakin ingin logout?

Anda akan keluar dari akun!

[Ya, Logout!](#)[Cancel](#)

React Router DOM

Navigasi antar halaman menggunakan Routing

[Tandai Selesai](#)[Tanya Dosen](#)



```
// src/Components/Button.js

import React from "react";
import PropTypes from "prop-types";

const Button = ({
  children,
  type = "button",
  onClick,
  className = "",
  loading = false,
  disabled = false,
}) => (
  <button
    type={type}
    onClick={onClick}
    disabled={disabled || loading}
    className={`w-full py-3 rounded-lg text-white font-medium
      ${loading ? "bg-gray-400 cursor-not-allowed" : "bg-gradient-to-r from-blue-500 to-indigo-500
      hover:from-indigo-500 hover:to-blue-500"}
      ${disabled && !loading ? "bg-gray-500 cursor-not-allowed" : ""}
      ${className}`}
  >
    {loading ? "Loading..." : children}
  </button>
);

Button.propTypes = {
  children: PropTypes.node.isRequired,
  type: PropTypes.oneOf(["button", "submit", "reset"]),
  onClick: PropTypes.func,
  className: PropTypes.string,
  loading: PropTypes.bool,
  disabled: PropTypes.bool,
};

export default Button;
```



```
// src/Components/Card.js

import React from "react";

const Card = ({ title, description, onClickComplete, onClickAsk, isCompleted }) => {
  return (
    <div className="p-4 border rounded-lg shadow-md">
      <h3 className="text-xl font-semibold mb-2">{title}</h3>
      <p className="text-gray-600 mb-4">{description}</p>

      <div className="flex gap-4">
        <button
          onClick={onClickComplete}
          disabled={isCompleted}
          className={`px-4 py-2 text-white ${isCompleted ? 'bg-gray-400 cursor-not-allowed' : 'bg-blue-600 hover:bg-blue-700'} rounded-lg`} // rounded-lg added here
        >
          {isCompleted ? "Selesai" : "Tandai Selesai"}
        </button>

        <button
          onClick={onClickAsk}
          className="px-4 py-2 text-white bg-green-600 hover:bg-green-700 rounded-lg" // rounded-lg
          added here
        >
          Tanya Dosen
        </button>
      </div>
    </div>
  );
};

export default Card;
```



```
// src/Components/Footer.js

import React from "react";

const Footer = () => {
  return (
    <footer className="bg-white text-center py-4 shadow-inner">
      <p className="text-sm text-gray-600">© 2025 Admin Dashboard. All rights reserved.
    </p></footer>
  );
};

export default Footer;
```



```
// src/Components/Form.js

import React from "react";
import Label from "../Components/Label";
import Input from "../Components/Input";
import Button from "../Components/Button";

const Form = ({ onSubmit }) => {
  const handleSubmit = (event) => {
    event.preventDefault();
    if (onSubmit) {
      onSubmit();
    }
  };
  return (
    <form onSubmit={handleSubmit} className="space-y-4">
      <div>
        <Label htmlFor="email">Email</Label>
        <Input type="email" id="email" placeholder="Masukkan email" />
      </div>
      <div>
        <Label htmlFor="password">Password</Label>
        <Input type="password" id="password" placeholder="Masukkan password" />
      </div>
      <Button type="submit" className="w-full bg-blue-600 hover:bg-blue-700">
        Login
      </Button>
    </form>
  );
};

export default Form;
```



```
// src/Components/Heading.js

import React from "react";
import Logout from "./Logout";

const Header = () => {
  return (
    <header className="bg-gradient-to-r from-blue-800 to-blue-600 text-white shadow-md"> <div className="flex justify-end items-center px-6 py-4">
      <Logout />
    </div>
  </header>
);
};

export default Header;
```



```
// src/Components/Input.js

import React from "react";

const Input = ({  
  label,  
  type = "text",  
  id,  
  placeholder,  
  value,  
  onChange,  
  required,  
  error,  
  helperText,  
  icon,  
  ...props  
}) => (  
  <div className="space-y-1 relative">  
    {label && (  
      <label htmlFor={id} className="block text-sm font-medium text-gray-700">  
        {label}  
      </label>  
    )}  
  
    <div className="relative">  
      <input  
        type={type}  
        id={id}  
        placeholder={placeholder}  
        value={value}  
        onChange={onChange}  
        required={required}  
        className={`w-full px-4 py-2 pl-10 border rounded-lg focus:outline-none focus:ring-2  
focus:ring-blue-500 ${  
          error ? "border-red-500" : "border-gray-300"  
        }`}  
        {...props}  
      />  
      {icon && (  
        <div className="absolute left-3 top-1/2 transform -translate-y-1/2 text-gray-500">  
          {icon}  
        </div>  
      )}  
    </div>  
  
    {error ? (  
      <p className="text-sm text-red-500">{error}</p>  
    ) : (  
      helperText && <p className="text-sm text-gray-500">{helperText}</p>  
    )}  
  </div>  
);  
  
export default Input;
```



```
// src/Components/Logout.js

import React from "react";
import { FaSignOutAlt } from "react-icons/fa";
import Swal from "sweetalert2";

const LogoutButton = () => {
  const handleLogout = () => {
    Swal.fire({
      title: 'Apakah Anda yakin ingin logout?',
      text: "Anda akan keluar dari akun!",
      icon: 'warning',
      showCancelButton: true,
      confirmButtonColor: '#3085d6',
      cancelButtonColor: '#d33',
      confirmButtonText: 'Ya, Logout!'
    }).then(result => {
      if (result.isConfirmed) {
        localStorage.removeItem("user");
        localStorage.removeItem("isLoggedIn");
        localStorage.removeItem("modulList");
        window.location.href = "/login";
        Swal.fire('Logged Out!', 'Anda berhasil keluar.', 'success');
      }
    });
  };

  return (
    <button
      onClick={handleLogout}
      className="flex items-center gap-2 text-white hover:text-gray-200 transition"
      >
      <FaSignOutAlt size={24} /> Logout
    </button>
  );
};

export default LogoutButton;
```



```
// src/Components/Modal.js

import React from "react";

const Modal = ({ isOpen, onClose, children, title }) => {
  if (!isOpen) return null; // Modal tidak ditampilkan jika isOpen false

  return (
    <div className="fixed inset-0 bg-black bg-opacity-60 flex items-center justify-center z-50">
      <div className="bg-white p-8 rounded-lg shadow-xl w-full sm:w-96 max-w-md transform transition-all scale-95 hover:scale-100">
        <div className="flex justify-between items-center mb-6">
          <h2 className="text-3xl font-bold text-indigo-600">{title}</h2>
          <button
            onClick={onClose}
            className="text-gray-600 hover:text-red-500 text-xl"
          >
            &times;
          </button>
        </div>
        {children}
      </div>
    </div>
  );
};

export default Modal;
```



```
// src/Components/ProgressBar.js

import React from "react";

const ProgressBar = ({ percent, label }) => (
  <div className="mb-6">
    {label && (
      <div className="flex justify-between mb-1">
        <span className="text-sm font-medium text-gray-700">{label}</span>
        <span className="text-sm font-medium text-gray-700">{percent}%</span>
      </div>
    )}
    <div className="w-full bg-gray-200 rounded-full h-3">
      <div
        className="bg-blue-600 h-3 rounded-full transition-all"
        style={{ width: `${percent}%` }}
      />
    </div>
  </div>
);

export default ProgressBar;
```



```
// src/Components/Sidebar.js

import React from "react";
import { NavLink } from "react-router-dom";
import { FaHome, FaBook } from "react-icons/fa";

const Sidebar = () => {
  return (
    <aside className="w-64 bg-blue-800 text-white p-4 h-screen">
      <h2 className="text-2xl font-bold mb-8 text-white">Belajar Pintar</h2>
      <nav className="flex flex-col gap-2">
        <NavLink
          to="/admin/dashboard"
          className={({ isActive }) =>
            isActive
              ? "flex items-center gap-2 bg-blue-600 text-white px-4 py-2 rounded transition duration-300"
              : "flex items-center gap-2 hover:bg-blue-500 hover:text-white px-4 py-2 rounded transition duration-300"
          }
        >
          <FaHome className="text-xl" />
          Dashboard
        </NavLink>

        <NavLink
          to="/admin/modul"
          className={({ isActive }) =>
            isActive
              ? "flex items-center gap-2 bg-blue-600 text-white px-4 py-2 rounded transition duration-300"
              : "flex items-center gap-2 hover:bg-blue-500 hover:text-white px-4 py-2 rounded transition duration-300"
          }
        >
          <FaBook className="text-xl" />
          Modul
        </NavLink>
      </nav>
    </aside>
  );
};

export default Sidebar;
```



```
// src/Data/Dummy.js
```

```
export const Dummy = {  
  name: "Eva Fadhillah Ulia",  
  email: "evafadhillahulia22@gmail.com",  
  password: "eva123"  
};
```

```
export default Dummy; // bisa digabungkan dengan  
dummymodul
```



```
// src/Data/DummyModul.js

export const modulList = [
  {
    id: "modul-1",
    title: "Pengenalan React",
    description: "Dasar-dasar React dan Komponen",
    completed: false,
  },
  {
    id: "modul-2",
    title: "State dan Props",
    description: "Penjelasan tentang State, Props dan One Way Data Flow",
    completed: false,
  },
  {
    id: "modul-3",
    title: "React Router DOM",
    description: "Navigasi antar halaman menggunakan Routing",
    completed: false,
  },
];
```



```
// src/Layouts/AdminLayout.js

import React from "react";
import { Outlet } from "react-router-dom";
import Sidebar from "../Components/Sidebar";
import Header from "../Components/Heading";

const AdminLayout = () => {
  return (
    <div className="flex h-screen">
      <Sidebar />
      <div className="flex-1 flex flex-col">
        <Header />
        <main className="flex-1 p-6 overflow-y-auto bg-gray-100">
          <Outlet />
        </main>
        <footer className="bg-white text-center py-4 shadow-inner">
          <p className="text-sm text-gray-600">
            © 2025 Admin Dashboard. All rights reserved.
          </p>
        </footer>
      </div>
    </div>
  );
};

export default AdminLayout;
```



```
import React from "react";
import { Outlet } from "react-router-dom";

const AuthLayout = () => (
  <div className="h-screen bg-gradient-to-br from-gray-100 to-gray-200 flex items-center justify-center px-4">
    <div className="w-full max-w-screen-sm bg-white rounded-2xl shadow-lg p-8 space-y-6">
      <Outlet />
    </div>
  </div>
);

export default AuthLayout;
```

```
// src/Pages/Admin/Modul/AccordionItem.js

import React from "react";

const AccordionItem = ({  
    id,  
    title,  
    description,  
    completed,  
    isOpen,  
    onToggle,  
    onComplete,  
    onAsk,  
}) => (  
    <div className="mb-4">  
        <div  
            className={`cursor-pointer p-4 rounded text-white ${  
                completed ? "bg-green-600" : "bg-blue-600"  
            }`}  
            onClick={() => onToggle(id)}  
        >  
            <h3 className="font-bold">{title}</h3>  
            <p className="text-sm opacity-75">{description}</p>  
            <p className="mt-1 font-medium">  
                Status: {completed ? "Selesai" : "Belum Selesai"}  
            </p>  
        </div>  
  
        {isOpen && (  
            <div className="p-4 border border-blue-200 rounded mt-2 bg-white">  
                <h4 className="font-semibold mb-2">Detail Modul</h4>  
                <p className="mb-4">{description}</p>  
                <div className="flex gap-4">  
                    <button  
                        className="px-4 py-2 bg-green-500 text-white rounded"  
                        onClick={() => onComplete(id)}  
                        disabled={completed}  
                    >  
                        Tandai Selesai  
                    </button>  
                    <button  
                        className="px-4 py-2 bg-yellow-500 text-white rounded"  
                        onClick={() => onAsk(id)}  
                    >  
                        Tanya Dosen  
                    </button>  
                </div>  
            </div>  
        )}  
    </div>  
);  
  
export default AccordionItem;
```



```
// src/Pages/Admin/Modul/ModalTanya.js

import React, { useRef } from "react";
import { toast } from "react-toastify";
import Modal from "../../Components/Modal";

const ModalTanya = ({ question, setQuestion, submitQuestion, setAskId }) => {
  const isSubmitting = useRef(false);

  const handleSubmit = () => {
    if (question.trim() && !isSubmitting.current) {
      isSubmitting.current = true;
      submitQuestion();
    } else if (!question.trim()) {
      toast.error("Tolong isi pertanyaan terlebih dahulu");
    }
  };

  return (
    <Modal
      isOpen={!!setAskId}
      onClose={() => setAskId(null)}
      title="Tanya Dosen"
    >
      <textarea
        className="w-full p-4 border border-gray-300 rounded-lg focus:outline-none focus:ring-2
        focus:ring-indigo-600 focus:border-indigo-600 text-lg placeholder-gray-500"
        rows="6"
        value={question}
        onChange={e => setQuestion(e.target.value)}
        placeholder="Tanyakan pertanyaanmu..."
      />
      <div className="flex justify-end gap-6 mt-6">
        <button
          className="px-6 py-2 bg-gray-400 text-white rounded-lg shadow-md hover:bg-gray-500
          focus:outline-none transition-all duration-200 transform hover:scale-105"
          onClick={() => setAskId(null)} // Close modal
        >
          Batal
        </button>
        <button
          className="px-6 py-2 bg-indigo-600 text-white rounded-lg shadow-md hover:bg-indigo-700
          focus:outline-none transition-all duration-200 transform hover:scale-105"
          onClick={handleSubmit}
        >
          Kirim
        </button>
      </div>
    </Modal>
  );
};

export default ModalTanya;
```



```
// src/Pages/Admin/Modul/Modul.js

import React, { useState } from "react";
import { ToastContainer } from "react-toastify";
import "react-toastify/dist/ReactToastify.css";
import ModalTanya from "../../../../../Pages/Admin/Modul/ModalTanya";
import Card from "../../../../../Components/Card";
import ProgressBar from "../../../../../Components/ProgressBar";
import { modulList as initialList } from "../../../../../Data/DummyModul";
import { showToastSuccess, showToastError } from "../../../../../Utils/Helpers/ToastsHelpers"; // Import helper untuk toast success/error
import { showSuccess } from "../../../../../Utils/Helpers/SwalHelpers"; // Import helper untuk SweetAlert success

const Modul = () => {
    /// Baca daftar modul yang tersimpan di localStorage
    const saved = JSON.parse(localStorage.getItem("modulList"));
    const [modulList, setModulList] = useState(
        Array.isArray(saved) && saved.length > 0 ? saved : initialList
    );

    const [askId, setAskId] = useState(null); // State untuk menyimpan ID modul yang sedang ditanya
    const [question, setQuestion] = useState(""); // State untuk menyimpan teks pertanyaan
    const [activeIndex, setActiveIndex] = useState(null); // State untuk menyimpan modul mana yang accordion-nya terbuka

    // Hitung total modul dan berapa yang sudah selesai
    const total = modulList.length;
    const completedCount = modulList.filter(m => m.completed).length;
    const progressPercent = total > 0
        ? Math.round((completedCount / total) * 100)
        : 0;

    // Fungsi untuk menandai modul selesai berdasarkan ID
    const handleComplete = id => {
        // Update properti completed pada modul yang sesuai
        const updated = modulList.map(m =>
            m.id === id ? { ...m, completed: true } : m
        );
        setModulList(updated); // Simpan state baru
        localStorage.setItem("modulList", JSON.stringify(updated)); // Simpan ke localStorage
        showSuccess("Sukses", "Modul berhasil ditandai selesai!");
    };

    // Fungsi untuk toggle buka/tutup accordion berdasarkan ID
    const handleToggle = (id) => {
        setActiveIndex(activeIndex === id ? null : id);
    };

    // Fungsi untuk buka modal tanya, set ID modul yang akan ditanya
    const handleAsk = id => setAskId(id);

    // Fungsi untuk kirim pertanyaan
    const submitQuestion = () => {
        if (question.trim()) { // Cek jika pertanyaan tidak kosong
            showToastSuccess("Pertanyaan berhasil dikirim"); // Tampilkan toast success
            setAskId(null); // Tutup modal
            setQuestion(""); // Reset input pertanyaan
        } else {
            showToastError("Tolong isi pertanyaan terlebih dahulu"); // Tampilkan toast error jika kosong
        }
    };

    return (
        <div className="container mx-auto p-6">
            <h1 className="text-3xl font-bold text-blue-600 mb-6">
                Modul Pembelajaran
            </h1>

            {/* Progress Bar Section */}
            <div className="mb-8">
                <ProgressBar
                    percent={progressPercent} // Persentase progress
                    label="Progress Belajar"
                />
            </div>

            {/* Modul List */}
            <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-6">
                {modulList.map((m) => (
                    <Card
                        key={m.id}
                        title={m.title}
                        description={m.description}
                        onClickComplete={() => handleComplete(m.id)}
                        onClickAsk={() => handleAsk(m.id)}
                        isCompleted={m.completed} // Prop untuk menandai tampilan jika sudah selesai
                        isOpen={activeIndex === m.id} // Prop untuk buka/tutup accordion
                        onToggle={() => handleToggle(m.id)} // Handler toggle accordion
                    />
                )));
            </div>

            {/* Modal for Asking Question */}
            {askId && (
                <ModalTanya
                    question={question} // Input question text
                    setQuestion={setQuestion} // Setter untuk question
                    submitQuestion={submitQuestion} // Handler submit pertanyaan
                    setAskId={setAskId} // Setter untuk tutup modal
                />
            )}

            {/* Container untuk Toast Notifications */}
            <ToastContainer
                position="top-center"
                autoClose={5000}
                hideProgressBar={false}
                newestOnTop
                closeOnClick
                rtl={false}
                pauseOnFocusLoss
                draggable
                pauseOnHover
                className="custom-toast-container"
            />
        </div>
    );
};

export default Modul;
```



```
// src/Pages/Admin/Dashboard.js

import React, { useEffect, useState } from "react";
import { useNavigate } from "react-router-dom";
import Dummy from "../../Data/Dummy";

const Dashboard = () => {
  const [progressPercent, setProgressPercent] = useState(0); // State untuk persentase progress belajar
  const navigate = useNavigate(); // Inisialisasi fungsi navigate

  useEffect(() => {
    // Ambil data modul dari localStorage, default ke array kosong
    const savedModul = JSON.parse(localStorage.getItem("modulList") || "null") || [];
    const total = savedModul.length; // Jumlah total modul
    const completedCount = savedModul.filter(m => m.completed).length; // Hitung modul yang selesai
    const percent = total > 0 ? Math.round((completedCount / total) * 100) : 0;
    setProgressPercent(percent); // Set state persentase
  }, []);

  // Fungsi untuk navigasi ke halaman modul
  const navigateToModul = () => {
    navigate("/admin/modul");
  };

  return (
    <div className="p-6 bg-blue-600 min-h-screen text-white">
      <div className="container mx-auto">
        {/* Title */}
        <h1 className="text-5xl font-extrabold text-center mb-6 md:mb-8 animate__animated animate__fadeIn">
          Selamat datang, {Dummy.name}! {/* Menampilkan nama pengguna dari Dummy */}
        </h1>

        {/* User Email */}
        <p className="text-lg text-center mb-6 md:mb-8 animate__animated animate__fadeIn animate__delay-1s">
          {Dummy.email} {/* Menampilkan email pengguna dari Dummy */}
        </p>

        {/* Progress Card */}
        <div className="mb-8 bg-white p-6 rounded-xl shadow-lg transform transition-all hover:scale-105 hover:shadow-2xl">
          <div className="flex justify-between mb-4">
            <span className="text-lg font-semibold text-gray-800">Progress Belajar</span>
            <span className="text-lg font-semibold text-gray-800">{progressPercent}%</span>
          </div>
          <div className="w-full bg-gray-300 rounded-full h-4">
            <div
              className="bg-blue-500 h-4 rounded-full transition-all ease-in-out duration-300"
              style={{ width: `${progressPercent}%` }} // Atur lebar bar sesuai persentase
            ></div>
          </div>
        </div>

        {/* Button */}
        <div className="text-center mt-6 md:mt-8">
          <button
            onClick={navigateToModul} // Panggil fungsi navigasi saat klik
            className="px-6 py-3 bg-green-500 text-white font-semibold rounded-lg shadow-lg hover:bg-blue-900 transform transition-all hover:scale-105 hover:shadow-2xl duration-300"
          >
            Lanjutkan Belajar
          </button>
        </div>
      </div>
    );
};

export default Dashboard;
```



```
// src/Pages/Auth/Login.js

import React, { useState } from "react";
import { useNavigate } from "react-router-dom";
import { toast } from "react-toastify";
import { FaCheckCircle, FaTimesCircle, FaEnvelope, FaLock } from "react-icons/fa"; // Import ikon untuk
email, password, success, dan error
import Input from "../../Components/Input";
import Button from "../../Components/Button";
import { Dummy } from "../../Data/Dummy";
import illustration from "../Auth/gambar.png"; // Import gambar ilustrasi untuk login page

const Login = () => {
  const navigate = useNavigate(); // Hook untuk navigasi antar halaman
  const [email, setEmail] = useState(""); // State untuk menangani email input
  const [password, setPassword] = useState(""); // State untuk menangani password input

  const handleLogin = (e) => { // Fungsi untuk menangani proses login
    e.preventDefault(); // Mencegah form submit secara default
    const { email, password } = e.target; // Ambil email dan password dari form

    // Cek apakah email dan password sesuai dengan Dummy data
    if (email.value === Dummy.email && password.value === Dummy.password) {
      // Simpan data user di localStorage dan tandai pengguna sudah login
      localStorage.setItem("user", JSON.stringify({ email: Dummy.email, loggedIn: true }));
      localStorage.setItem("isLoggedIn", "true");
      // Tampilkan notifikasi sukses login
      toast.success(
        <div className="flex items-center gap-2">
          <FaCheckCircle className="text-blue-400 text-xl animate-pulse" /> {/* Ikon sukses */}
          <span>Login Berhasil!</span>
        </div>,
        {
          position: "top-center",
          autoClose: 3000,
          theme: "light",
          className: "animate__animated animate__fadeInDown", // Animasi masuk
          style: { background: "linear-gradient(135deg, #6c75c1, #5ab1a2)", color: "#fff" }, // Styling
        }
      );
      navigate("/admin/dashboard"); // Arahkan pengguna ke halaman dashboard admin setelah login sukses
    } else {
      // Tampilkan notifikasi error jika email atau password salah
      toast.error(
        <div className="flex items-center gap-2">
          <FaTimesCircle className="text-red-400 text-xl animate-shake" /> {/* Ikon error */}
          <span>Email atau password salah!</span>
        </div>,
        {
          position: "top-center",
          autoClose: 3000,
          theme: "light",
          className: "animate__animated animate__shakeX", // Animasi shake saat error
          style: { background: "linear-gradient(135deg, #f44336, #ffeb3b)", color: "#fff" }, // Styling
        }
      );
    }
  };

  return (
    <>
      {/* Image Wrapper as a Rectangle */}
      <div className="flex justify-center mb-8">
        <div className="w-full sm:w-[130%] lg:w-[140%] h-60 sm:h-48 shadow-lg overflow-hidden rounded-xl">
          <img
            src={illustration} // Gambar ilustrasi login
            alt="Belajar Pintar"
            className="w-full h-full object-cover rounded-xl" // Styling gambar
          />
        </div>
      </div>

      <h1 className="text-3xl font-extrabold text-center text-blue-600">Belajar Pintar</h1> {/* Judul
halaman */}

      <p className="text-center text-gray-500 mb-6"> /* Deskripsi platform */
        Platform pembelajaran digital untuk mahasiswa
      </p>

      <form onSubmit={handleLogin} className="space-y-4"> /* Form untuk login */
        {/* Email Input */}
        <Input
          label="Email"
          type="email"
          id="email"
          name="email"
          placeholder="you@example.com"
          required // Input wajib diisi
          value={email} // Value diikat ke state email
          onChange={(e) => setEmail(e.target.value)} // Update state email ketika user mengetik
          icon={<FaEnvelope />} // Ikon email di input
        />

        {/* Password Input */}
        <Input
          label="Password"
          type="password" // Tipe input password
          id="password"
          name="password"
          placeholder="••••••••"
          required
          value={password} // Value diikat ke state password
          onChange={(e) => setPassword(e.target.value)} // Update state password ketika user mengetik
          icon={<FaLock />} // Ikon kunci di input
        />

        <Button type="submit">Masuk</Button> /* Tombol submit untuk login */
      </form>
    </>
  );
};

export default Login;
```



```
// src/Routes/Router.js

import React from "react";
import { Navigate } from "react-router-dom";

const ProtectedRoute = ({ children }) => { // Komponen ProtectedRoute yang menerima props 'children'
  const isLoggedIn = localStorage.getItem("isLoggedIn") === "true"; // Cek status login dari
  localStorage

  // Jika pengguna tidak login (isLoggedIn === false), arahkan ke halaman login
  if (!isLoggedIn) {
    return <Navigate to="/login" replace />; // Menggunakan Navigate untuk mengalihkan pengguna ke
    halaman login
  }

  // Jika pengguna sudah login, tampilkan konten yang dibungkus oleh ProtectedRoute (children)
  return children;
};

export default ProtectedRoute;
```



```
// src/Utils/Helpers/SwalHelpers.js

import Swal from 'sweetalert2'; // Mengimpor SweetAlert2 untuk menampilkan pop-up notifikasi yang interaktif

// Fungsi untuk menampilkan pop-up sukses
export const showSuccess = (title, text) => {
  Swal.fire({
    icon: 'success', // Menampilkan ikon sukses (warna hijau)
    title: title, // Menampilkan judul yang disesuaikan dengan parameter
    text: text, // Menampilkan teks yang disesuaikan dengan parameter
    confirmButtonText: 'OK', // Menampilkan tombol konfirmasi dengan teks "OK"
  });
};

// Fungsi untuk menampilkan pop-up error
export const showError = (title, text) => {
  Swal.fire({
    icon: 'error', // Menampilkan ikon error (warna merah)
    title: title, // Menampilkan judul yang disesuaikan dengan parameter
    text: text, // Menampilkan teks yang disesuaikan dengan parameter
    confirmButtonText: 'OK', // Menampilkan tombol konfirmasi dengan teks "OK"
  });
};

// Fungsi untuk menampilkan pop-up konfirmasi (Yes/No)
export const showConfirm = (title, text) => {
  return Swal.fire({
    title: title, // Menampilkan judul yang disesuaikan dengan parameter
    text: text, // Menampilkan teks yang disesuaikan dengan parameter
    icon: 'warning', // Menampilkan ikon peringatan (warna kuning)
    showCancelButton: true, // Menampilkan tombol Cancel (Batal)
    confirmButtonText: 'Yes', // Teks tombol konfirmasi (Yes)
    cancelButtonText: 'No', // Teks tombol pembatalan (No)
  });
};

// Fungsi untuk menampilkan pop-up input teks
export const showInput = () => {
  return Swal.fire({
    title: 'Input your question',
    input: 'text',
    inputPlaceholder: 'Enter question',
    showCancelButton: true,
  });
};

export const showLoading = () => {
  return Swal.fire({
    title: 'Loading...',
    text: 'Please wait',
    didOpen: () => {
      Swal.showLoading();
    },
  });
};
```



```
// src/Utils/Helpers/ToastHelpers.js

import { toast } from 'react-toastify';

// Fungsi untuk menampilkan toast success
export const showToastSuccess = (message) =>
{ toast.success(message, {
    position: "top-center",
    autoClose: 5000,
    hideProgressBar: false,
    closeOnClick: true,
    pauseOnHover: true,
    draggable: true,
    progress: undefined,
    className: "custom-toast-success",
    bodyClassName: "custom-toast-body",
  });
};

// Fungsi untuk menampilkan toast error
export const showToastError = (message) => {
  toast.error(message, {
    position: "top-center",
    autoClose: 5000,
    hideProgressBar: false,
    closeOnClick: true,
    pauseOnHover: true,
    draggable: true,
    progress: undefined,
    className: "custom-toast-error",
  });
};

// Fungsi untuk menampilkan toast info
export const showToastInfo = (message) => {
  toast.info(message, {
    position: "top-center",
    autoClose: 5000,
    hideProgressBar: false,
    closeOnClick: true,
    pauseOnHover: true,
    draggable: true,
    progress: undefined,
  });
};

// Fungsi untuk menampilkan toast warning
export const showToastWarning = (message) =>
{ toast.warning(message, {
    position: "top-center",
    autoClose: 5000,
    hideProgressBar: false,
    closeOnClick: true,
    pauseOnHover: true,
    draggable: true,
    progress: undefined,
  });
};
```



```
// src/App.js

import React from "react";
import { Routes, Route, Navigate } from "react-router-dom";
import { ToastContainer } from "react-toastify";
import "react-toastify/dist/ReactToastify.css";

import Login from "./Pages/Auth/Login";
import Dashboard from "./Pages/Admin/Dashboard";
import AuthLayout from "./Layouts/AuthLayout";
import AdminLayout from "./Layouts/AdminLayout";
import ProtectedRoute from "./Routes/Router";
import Modul from "./Pages/Admin/Modul/Modul";

const App = () => {
  return (
    <>
      <ToastContainer
        position="top-right"
        autoClose={3000}
        hideProgressBar={false}
        newestOnTop={false}
        closeOnClick
        rtl={false}
        pauseOnFocusLoss
        draggable
        pauseOnHover
      />

      <Routes>
        {/* Area Auth (Login) */}
        <Route element={<AuthLayout />}>
          <Route path="/login" element={<Login />} />
        </Route>

        {/* Area Admin – dibungkus ProtectedRoute */}
        <Route
          path="/admin"
          element={
            <ProtectedRoute>
              <AdminLayout />
            </ProtectedRoute>
          }
        >
          {/* /admin → redirect ke /admin/dashboard */}
          <Route index element={<Navigate to="/dashboard" replace />} />
        </Route>
        {/* Dashboard */}
        <Route path="dashboard" element={<Dashboard />} />

        {/* Modul */}
        <Route path="modul" element={<Modul />} />
      </Routes>
    </>
  );
};

export default App;
```



```
@tailwind base;  
@tailwind components;  
@tailwind utilities;  
  
/* Styling global */  
body {  
    @apply bg-gray-100 text-gray-800;  
}  
  
h1, h2, h3, h4, h5, h6 {  
    @apply font-semibold text-gray-900;  
}  
  
button {  
    @apply transition duration-300 ease-in-out;
```



```
//src/index.js

import React from "react";
import ReactDOM from "react-dom/client"; // Renderer React untuk DOM
import { BrowserRouter } from "react-router-dom"; // Router untuk navigasi
// Import App from "./App"; // Komponen App utama
import "./index.css";

// Buat root dan render aplikasi React
const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(
  <React.StrictMode> /* Mode ketat untuk mendeteksi potensi masalah */
    <BrowserRouter> /* Bungkus aplikasi dengan router */
      <App /> /* Render komponen utama App */
    </BrowserRouter>
  </React.StrictMode>
);
```