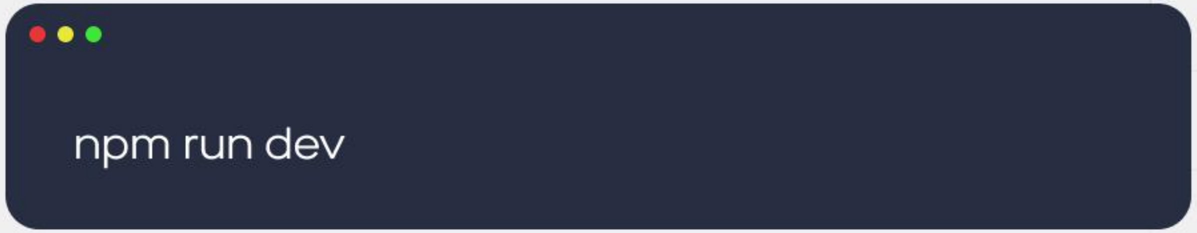
ReactJS

API & Authentification

Day 3

Quick Review

1. Layouting, Pages, Sections
2. useState for open navigation & Event Handler
3. React router dom



```
npm run dev
```

Simple Flow

1

User Login

1. Login (fake auth)
2. dapat token

2

Local Token

1. (Context + localStorage)

3

Consum API

1. API from / weather

4

Axios Get

1. temperature
2. humidity
3. wind

OpenWeather

REQRES

What's New ?

Create .env

Create lib, context, routes folders

cmd

```
src/
  lib/axios.js
  context/AuthContext.jsx
  routes/ProtectedRoute.jsx
  pages/Login.jsx
  pages/Weather.jsx
  App.jsx
.env
```

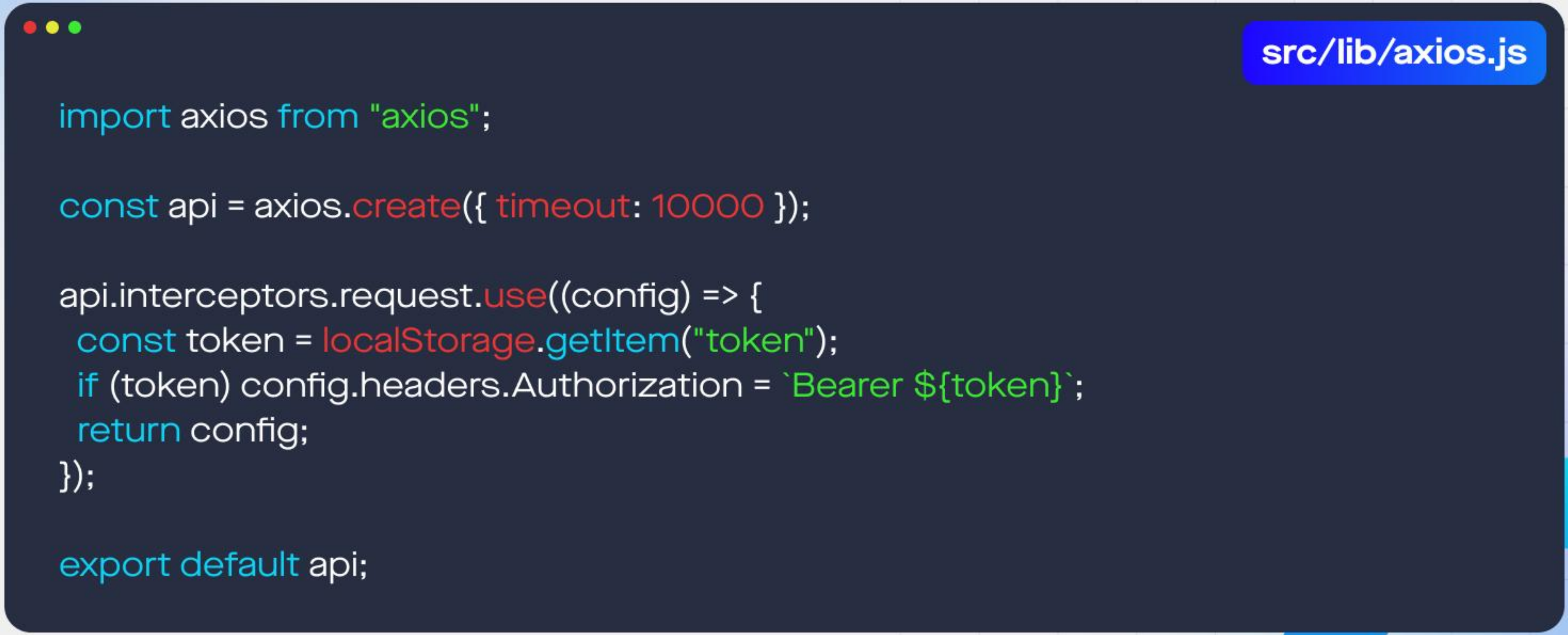
cmd

```
npm i axios
```

.env

```
VITE_OWM_KEY=ISI_API_KEY_OPENWEATHER
VITE_REQRES_KEY=ISI_API_KEY_REQRES
```

Axios Consum API Instance

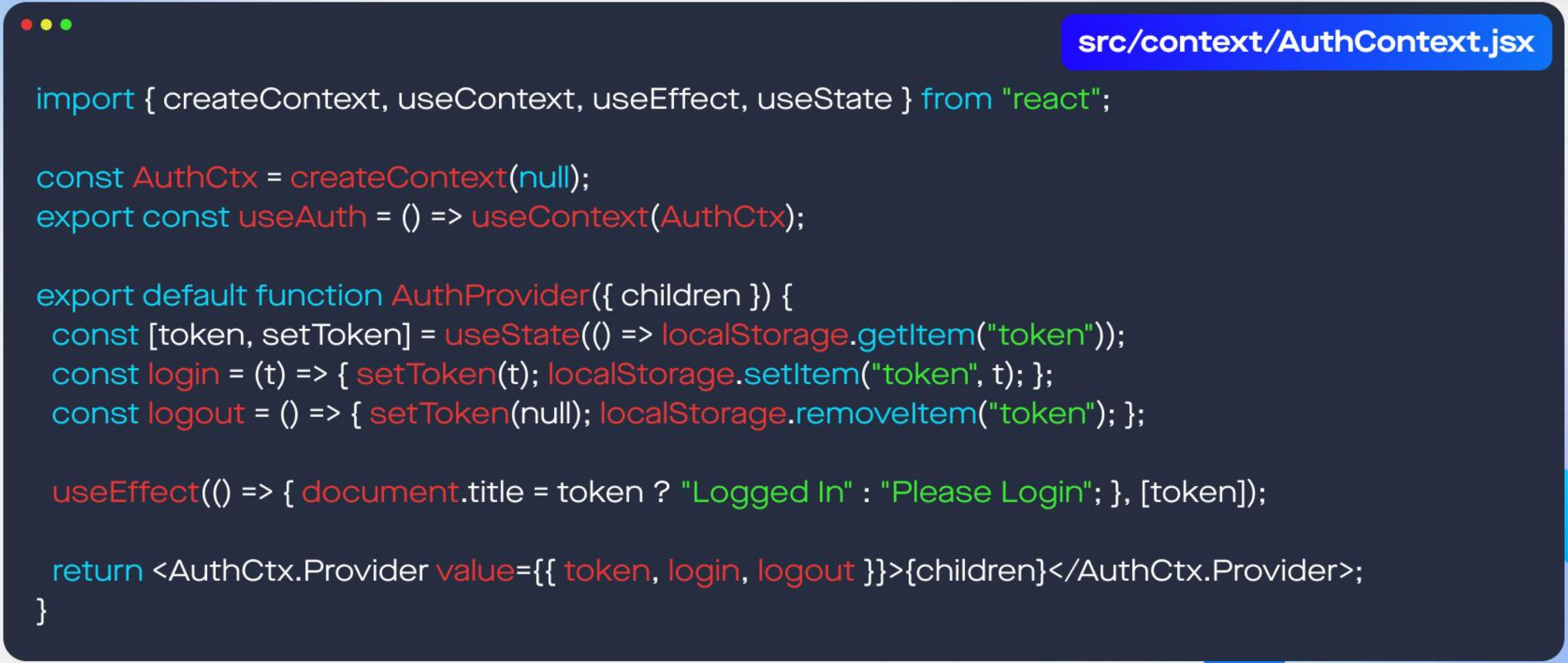


src/lib/axios.js

```
import axios from "axios";

const api = axios.create({ timeout: 10000 });

api.interceptors.request.use((config) => {
  const token = localStorage.getItem("token");
  if (token) config.headers.Authorization = `Bearer ${token}`;
  return config;
});

export default api;
```

+ header & error handling konsisten.

Context Auth Sederhana

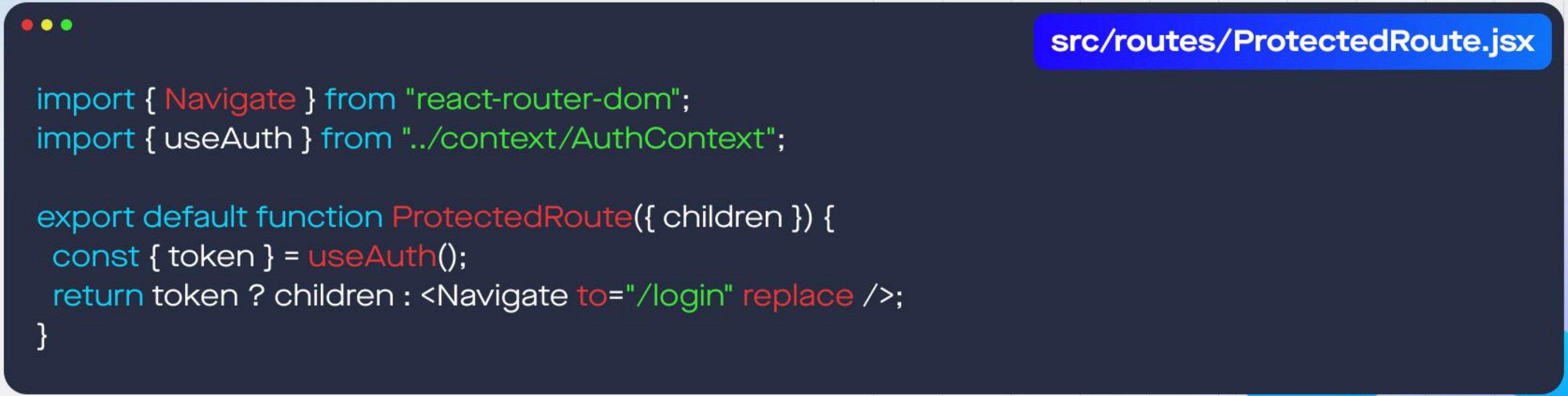


src/context/AuthContext.jsx

```
import { createContext, useContext, useEffect, useState } from "react";

const AuthCtx = createContext(null);
export const useAuth = () => useContext(AuthCtx);

export default function AuthProvider({ children }) {
  const [token, setToken] = useState(() => localStorage.getItem("token"));
  const login = (t) => { setToken(t); localStorage.setItem("token", t); };
  const logout = () => { setToken(null); localStorage.removeItem("token"); };

  useEffect(() => { document.title = token ? "Logged In" : "Please Login"; }, [token]);

  return <AuthCtx.Provider value={{ token, login, logout }}>{children}</AuthCtx.Provider>;
}
```

Tujuan: ekspose login/logout/token ke seluruh app.

Protected Route



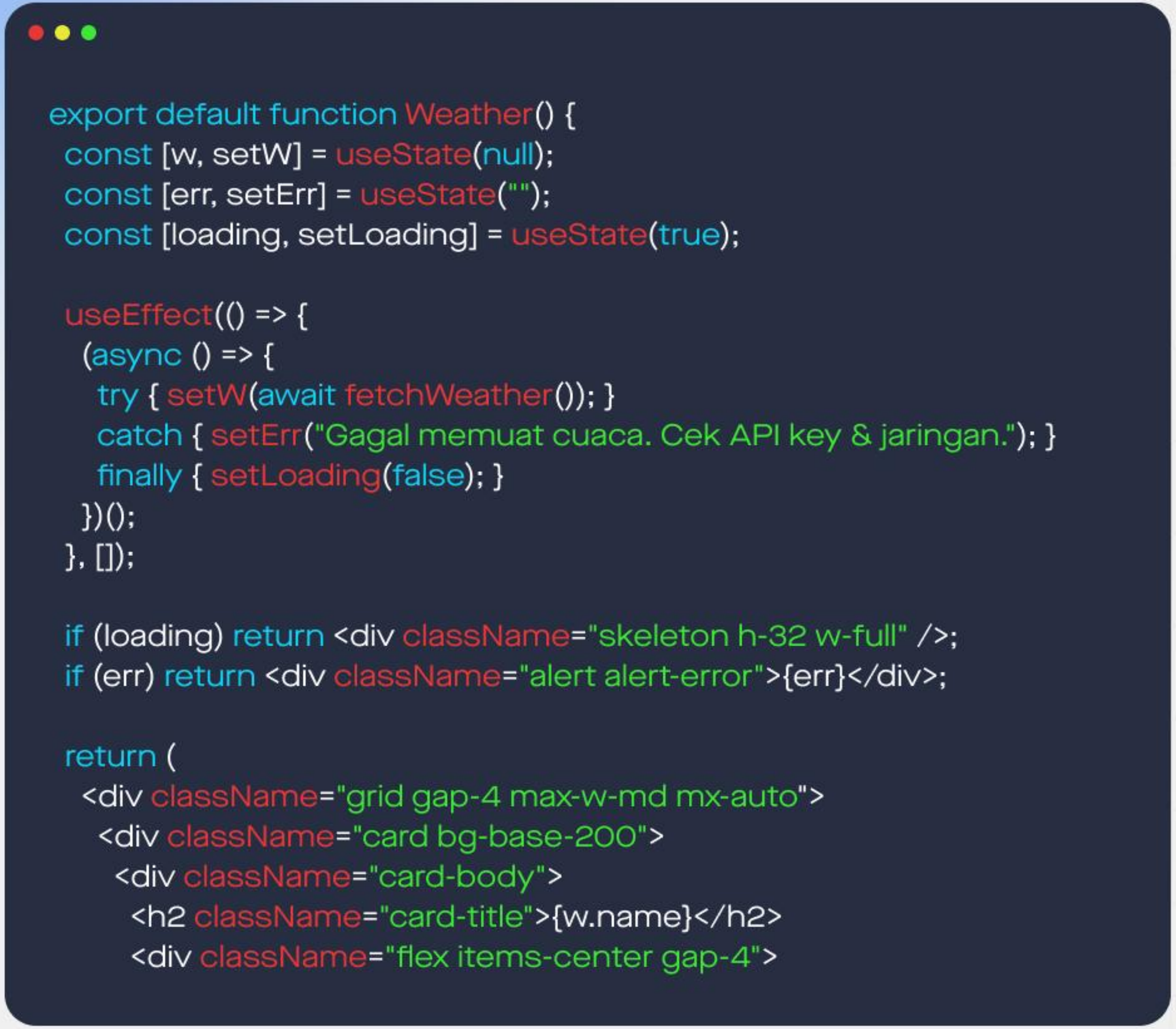
src/routes/ProtectedRoute.jsx

```jsx
import { Navigate } from "react-router-dom";
import { useAuth } from "../context/AuthContext";

export default function ProtectedRoute({ children }) {
  const { token } = useAuth();
  return token ? children : <Navigate to="/login" replace />;
}
```

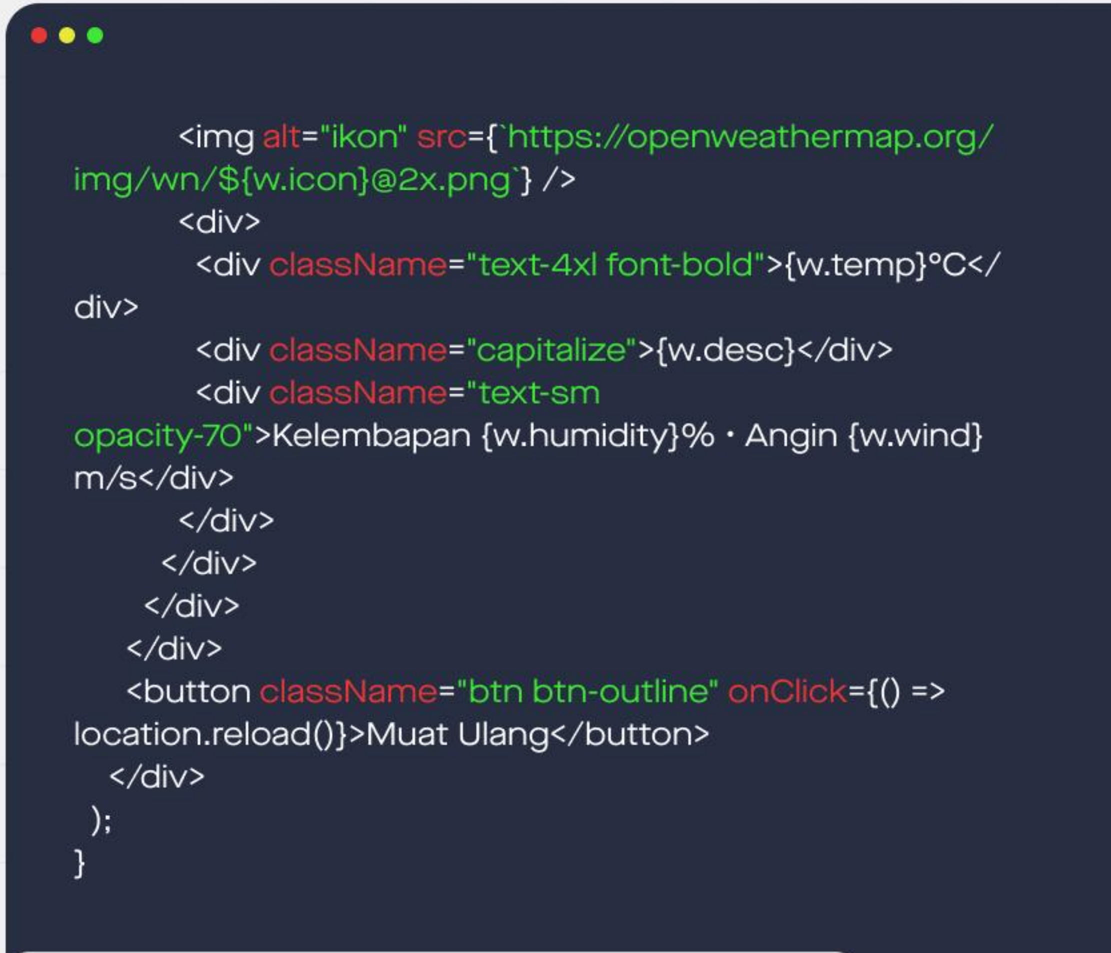
Tujuan: ekspose login/logout/token ke seluruh app.

Halaman Login (Fake Auth)

src/pages/Login.jsx



```
export default function Weather() {
  const [w, setW] = useState(null);
  const [err, setErr] = useState("");
  const [loading, setLoading] = useState(true);

  useEffect(() => {
    (async () => {
      try { setW(await fetchWeather()); }
      catch { setErr("Gagal memuat cuaca. Cek API key & jaringan."); }
      finally { setLoading(false); }
    })();
  }, []);

  if (loading) return <div className="skeleton h-32 w-full" />;
  if (err) return <div className="alert alert-error">{err}</div>;

  return (
    <div className="grid gap-4 max-w-md mx-auto">
      <div className="card bg-base-200">
        <div className="card-body">
          <h2 className="card-title">{w.name}</h2>
          <div className="flex items-center gap-4">
```

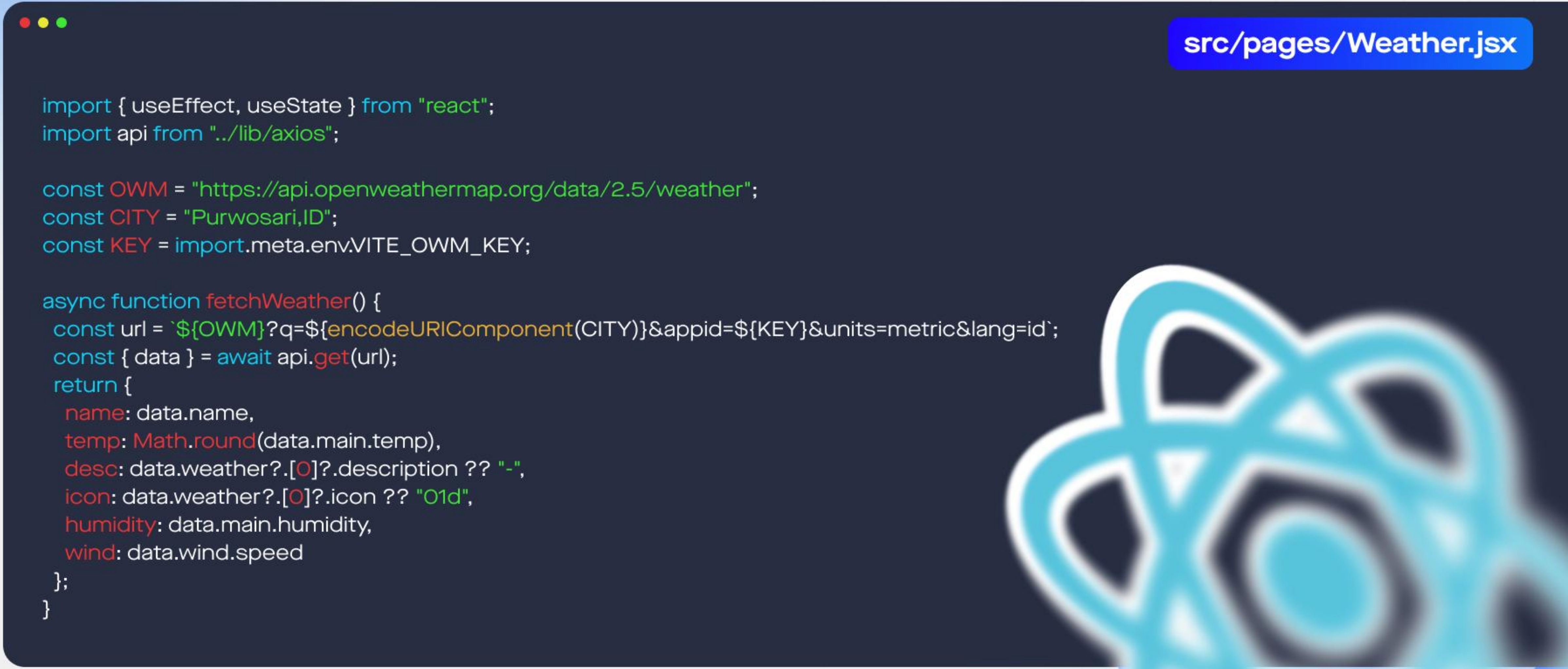


```
            <img alt="ikon" src={`https://openweathermap.org/
img/wn/${w.icon}@2x.png`} />
            <div>
             <div className="text-4xl font-bold">{w.temp}°C</
div>
             <div className="capitalize">{w.desc}</div>
             <div className="text-sm
opacity-70">Kelembapan {w.humidity}% · Angin {w.wind}
m/s</div>
            </div>
          </div>
        </div>
      </div>
      <button className="btn btn-outline" onClick={() =>
location.reload()}>Muat Ulang</button>
    </div>
  );
}
```

DaisyUI

card, skeleton, alert, btn

Service Cuaca (OpenWeather)

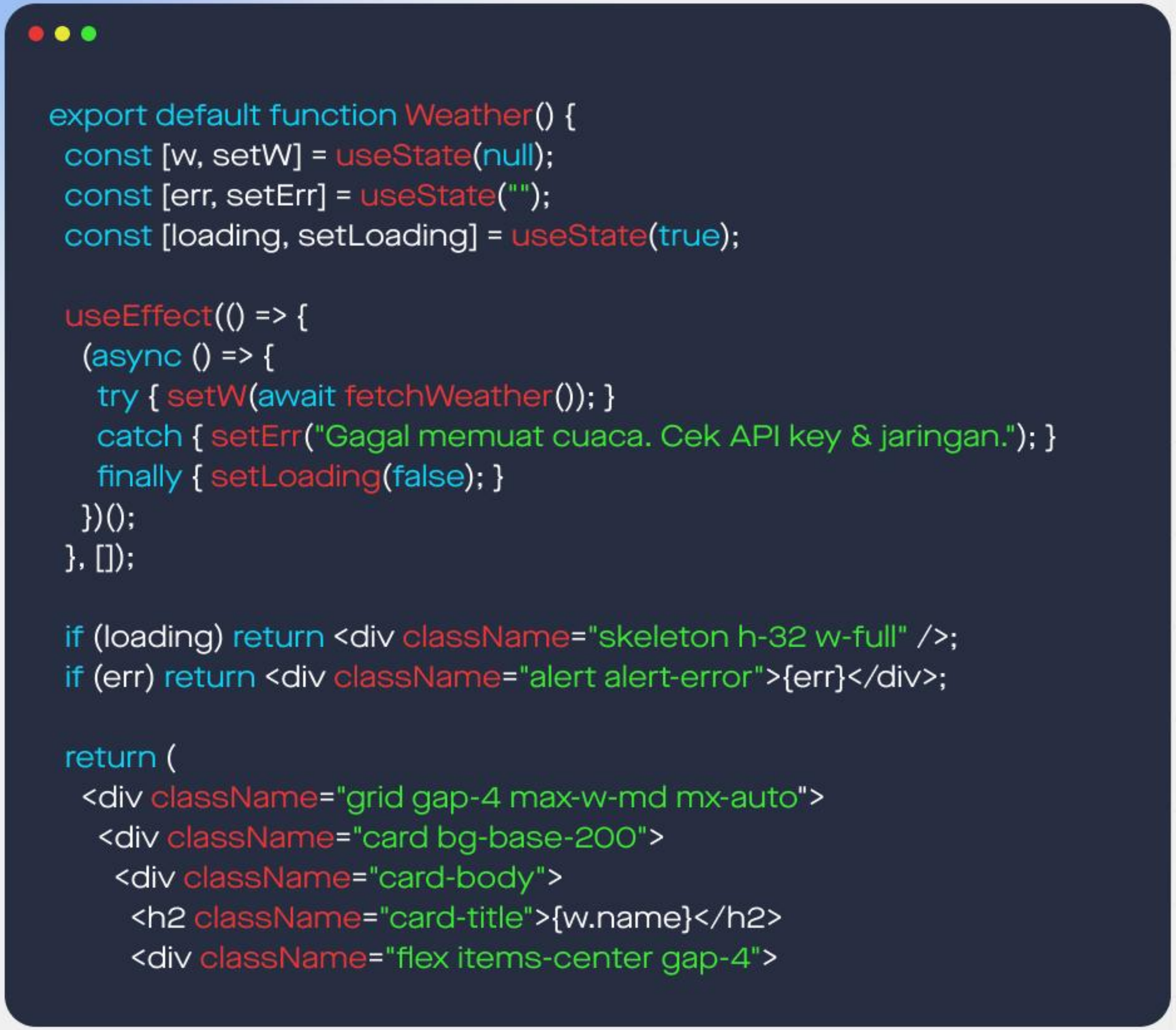


src/pages/Weather.jsx

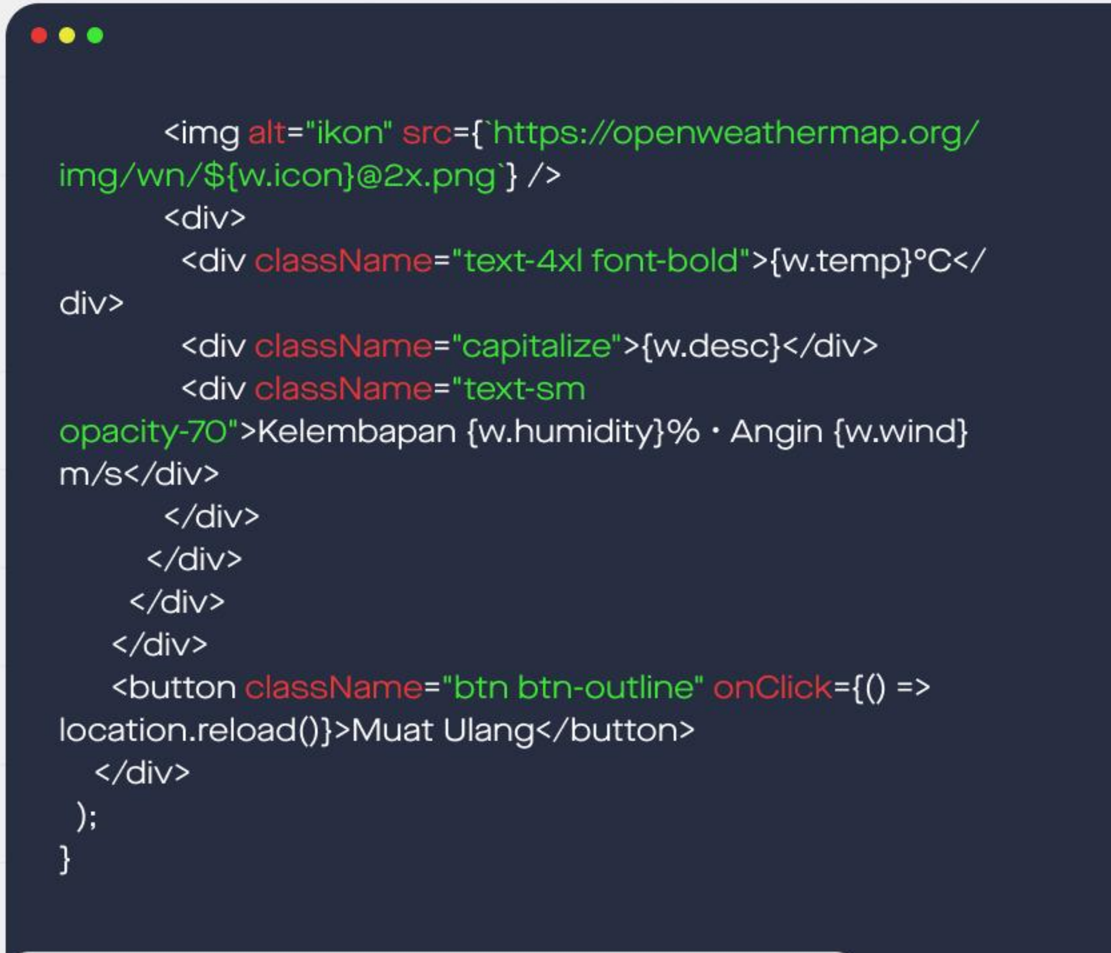
```
import { useEffect, useState } from "react";
import api from "../lib/axios";

const OWM = "https://api.openweathermap.org/data/2.5/weather";
const CITY = "Purwosari,ID";
const KEY = import.meta.env.VITE_OWM_KEY;

async function fetchWeather() {
 const url = `${OWM}?q=${encodeURIComponent(CITY)}&appid=${KEY}&units=metric&lang=id`;
 const { data } = await api.get(url);
 return {
  name: data.name,
  temp: Math.round(data.main.temp),
  desc: data.weather?.[0]?.description ?? "-",
  icon: data.weather?.[0]?.icon ?? "01d",
  humidity: data.main.humidity,
  wind: data.wind.speed
 };
}
```

Halaman Login (Fake Auth)

src/pages/Login.jsx



```jsx
export default function Weather() {
  const [w, setW] = useState(null);
  const [err, setErr] = useState("");
  const [loading, setLoading] = useState(true);

  useEffect(() => {
    (async () => {
      try { setW(await fetchWeather()); }
      catch { setErr("Gagal memuat cuaca. Cek API key & jaringan."); }
      finally { setLoading(false); }
    })();
  }, []);

  if (loading) return <div className="skeleton h-32 w-full" />;
  if (err) return <div className="alert alert-error">{err}</div>;

  return (
    <div className="grid gap-4 max-w-md mx-auto">
      <div className="card bg-base-200">
        <div className="card-body">
          <h2 className="card-title">{w.name}</h2>
          <div className="flex items-center gap-4">
```



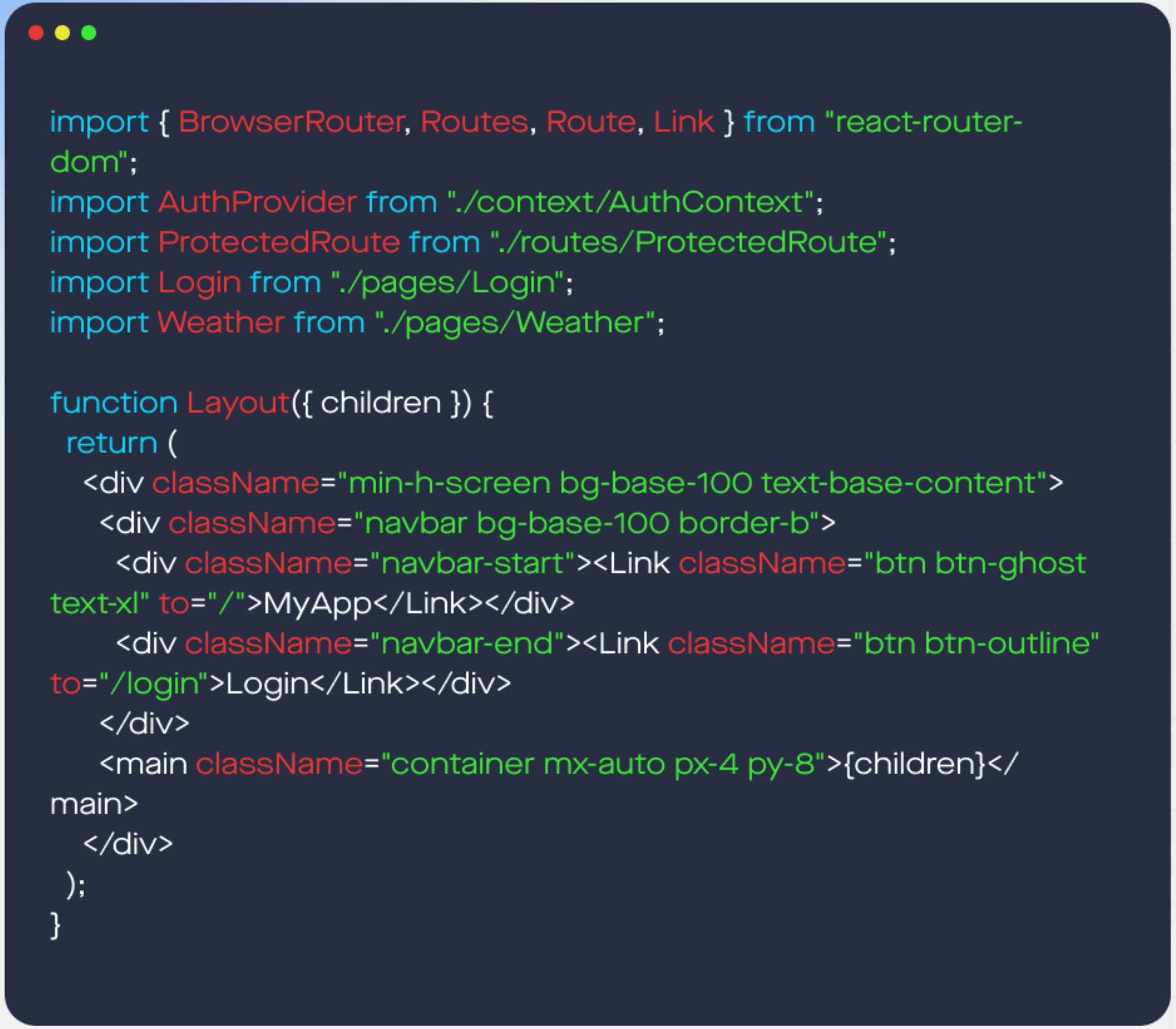
```jsx
            <img alt="ikon" src={`https://openweathermap.org/
img/wn/${w.icon}@2x.png`} />
            <div>
              <div className="text-4xl font-bold">{w.temp}°C</
div>
              <div className="capitalize">{w.desc}</div>
              <div className="text-sm
opacity-70">Kelembapan {w.humidity}% · Angin {w.wind}
m/s</div>
            </div>
          </div>
        </div>
      </div>
      <button className="btn btn-outline" onClick={() =>
location.reload()}>Muat Ulang</button>
    </div>
  );
}
```

DaisyUI

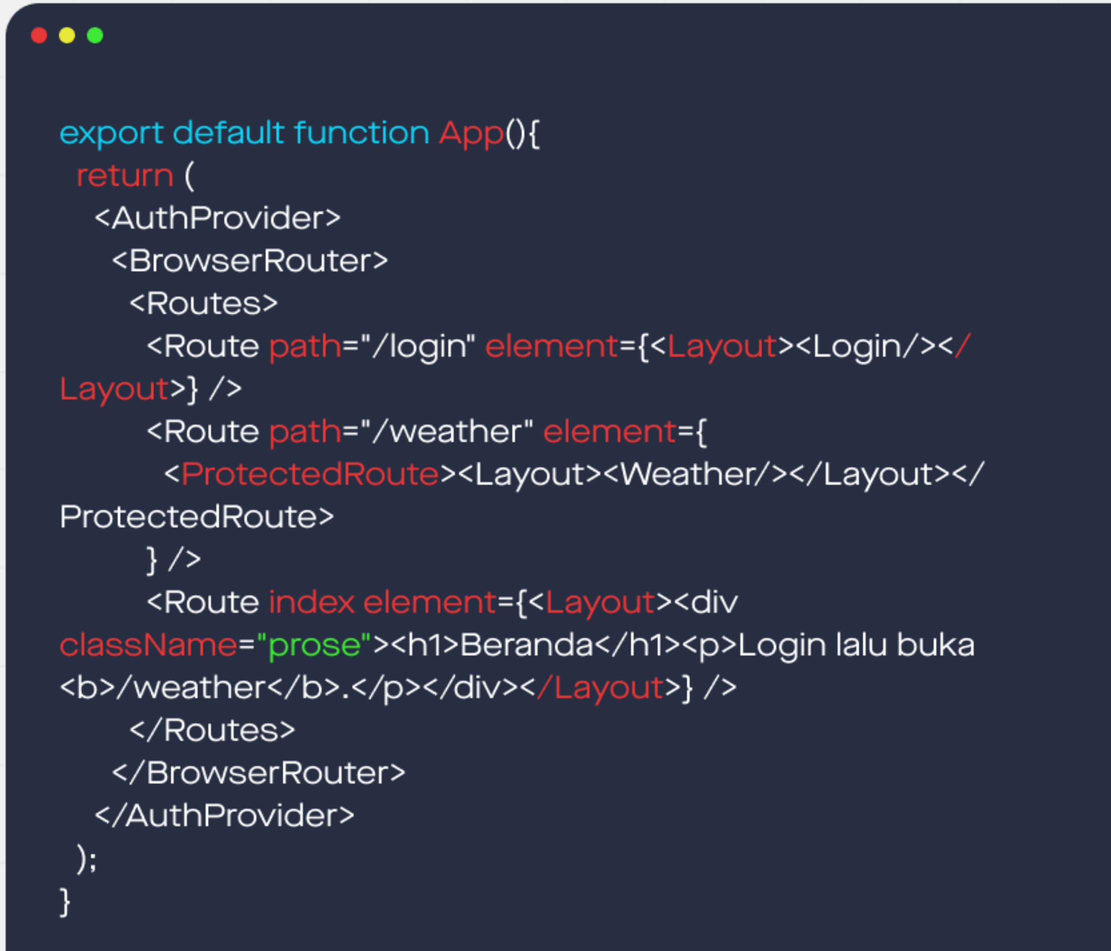
card, skeleton, alert, btn

Routing & Guard

src/App.jsx



```jsx
import { BrowserRouter, Routes, Route, Link } from "react-router-dom";
import AuthProvider from "./context/AuthContext";
import ProtectedRoute from "./routes/ProtectedRoute";
import Login from "./pages/Login";
import Weather from "./pages/Weather";

function Layout({ children }) {
 return (
  <div className="min-h-screen bg-base-100 text-base-content">
   <div className="navbar bg-base-100 border-b">
    <div className="navbar-start"><Link className="btn btn-ghost text-xl" to="/">MyApp</Link></div>
    <div className="navbar-end"><Link className="btn btn-outline" to="/login">Login</Link></div>
   </div>
   <main className="container mx-auto px-4 py-8">{children}</main>
  </div>
 );
}
```



```jsx
export default function App(){
 return (
  <AuthProvider>
   <BrowserRouter>
    <Routes>
     <Route path="/login" element={<Layout><Login/></Layout>} />
     <Route path="/weather" element={
      <ProtectedRoute><Layout><Weather/></Layout></ProtectedRoute>
     } />
     <Route index element={<Layout><div className="prose"><h1>Beranda</h1><p>Login lalu buka <b>/weather</b>.</p></div></Layout>} />
    </Routes>
   </BrowserRouter>
  </AuthProvider>
 );
}
```

Sederhana

satu layout minimal.

Context Folder

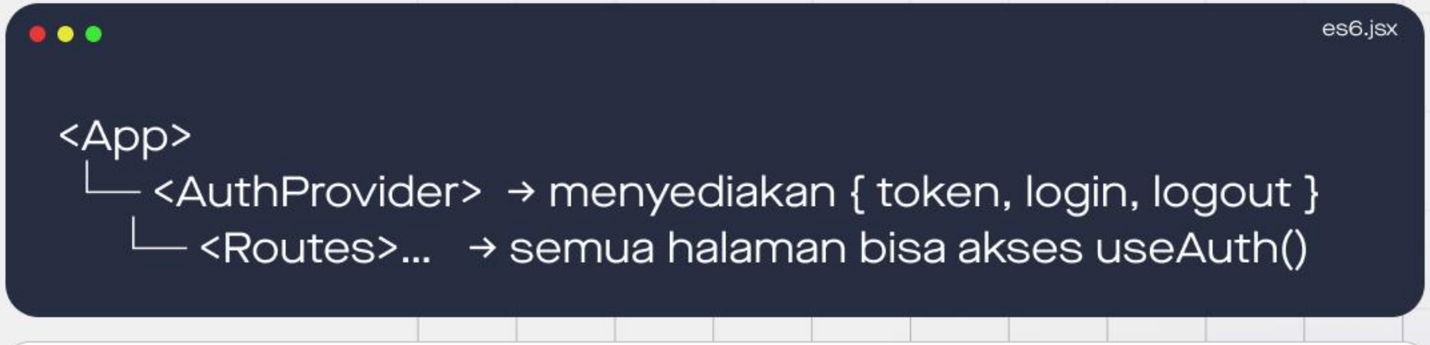
auth, tema, user prefs

Menyimpan state global yang dipakai lintas-halaman

Menghindari **prop drilling**

ThemeContext.jsx

AppContext.jsx

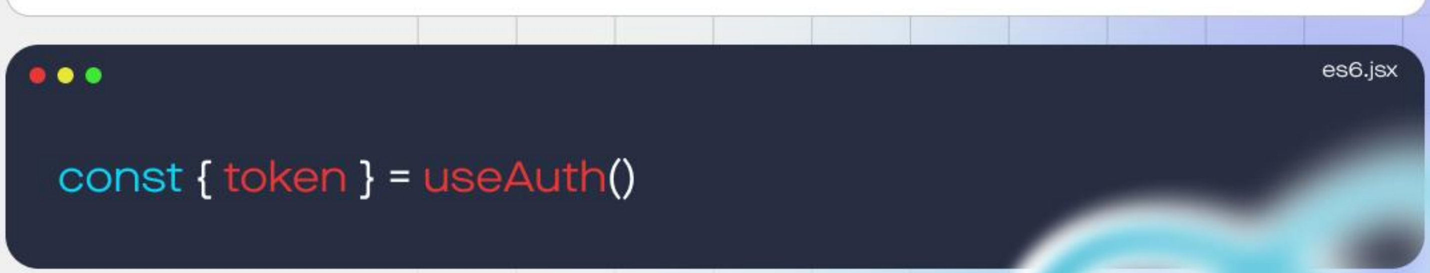


es6.jsx

```
<App>
 └── <AuthProvider>  → menyediakan { token, login, logout }
      └── <Routes>...   → semua halaman bisa akses useAuth()
```

AuthContext.jsx

1. Satu-satunya sumber data untuk token.
2. Simpan token ke localStorage agar login bertahan setelah refresh.



es6.jsx

```
const { token } = useAuth()
```

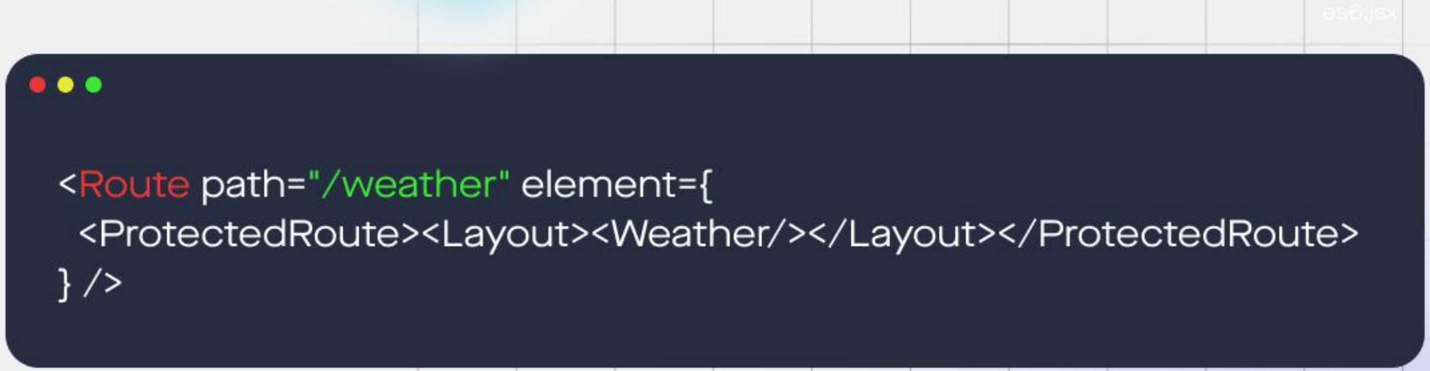
Routes Folder

auth, tema, user prefs

Menyimpan komponen pengaman rute & helper routing.

Keuntungan

Logika guard terpusat, rapi, mudah di-reuse.



```
<Route path="/weather" element={
  <ProtectedRoute><Layout><Weather/></Layout></ProtectedRoute>
} />
```

ProtectedRoute.jsx:

1. Cek status login
2. kalau tidak ada token → redirect ke /login

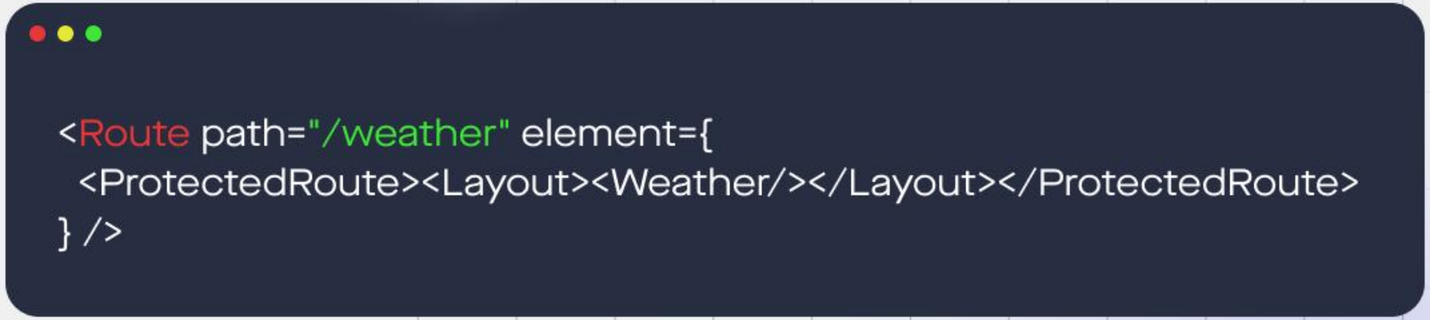
Routes Folder

auth, tema, user prefs

Menyimpan komponen pengaman rute & helper routing.

Keuntungan

Logika guard terpusat, rapi, mudah di-reuse.



```
<Route path="/weather" element={
  <ProtectedRoute><Layout><Weather/></Layout></ProtectedRoute>
} />
```

ProtectedRoute.jsx:

1. Cek status login
2. kalau tidak ada token → redirect ke /login

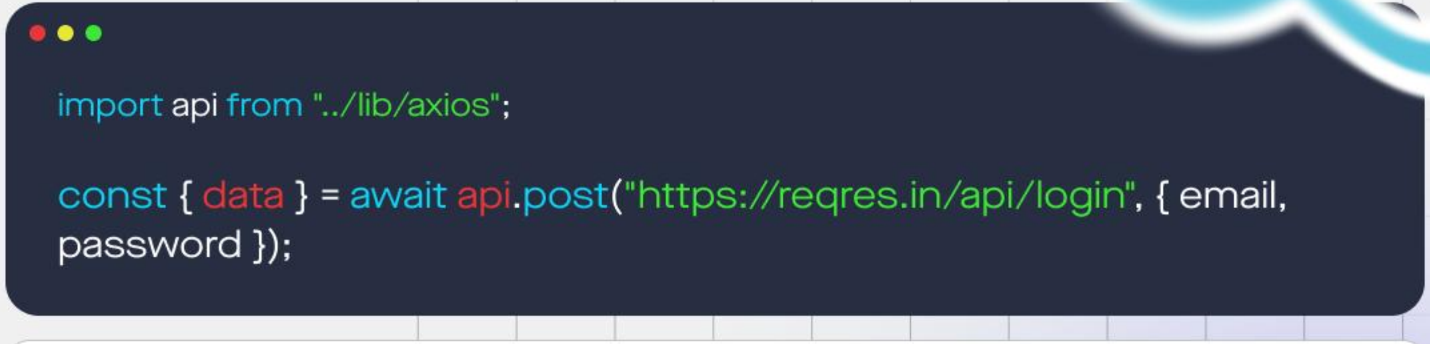
Lib Folder

lib/

Menaruh utilitas yang dapat dipakai di banyak tempat (axios instance, formatter, logger).

instance + interceptor

global setting, global initialization

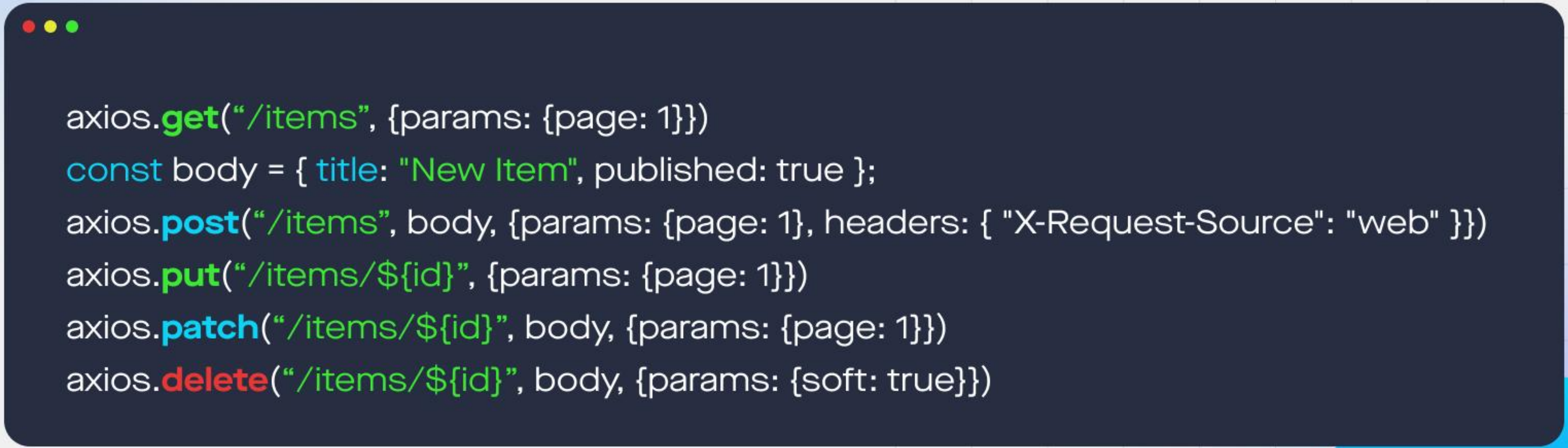


```
import api from "../lib/axios";

const { data } = await api.post("https://reqres.in/api/login", { email,
password });
```

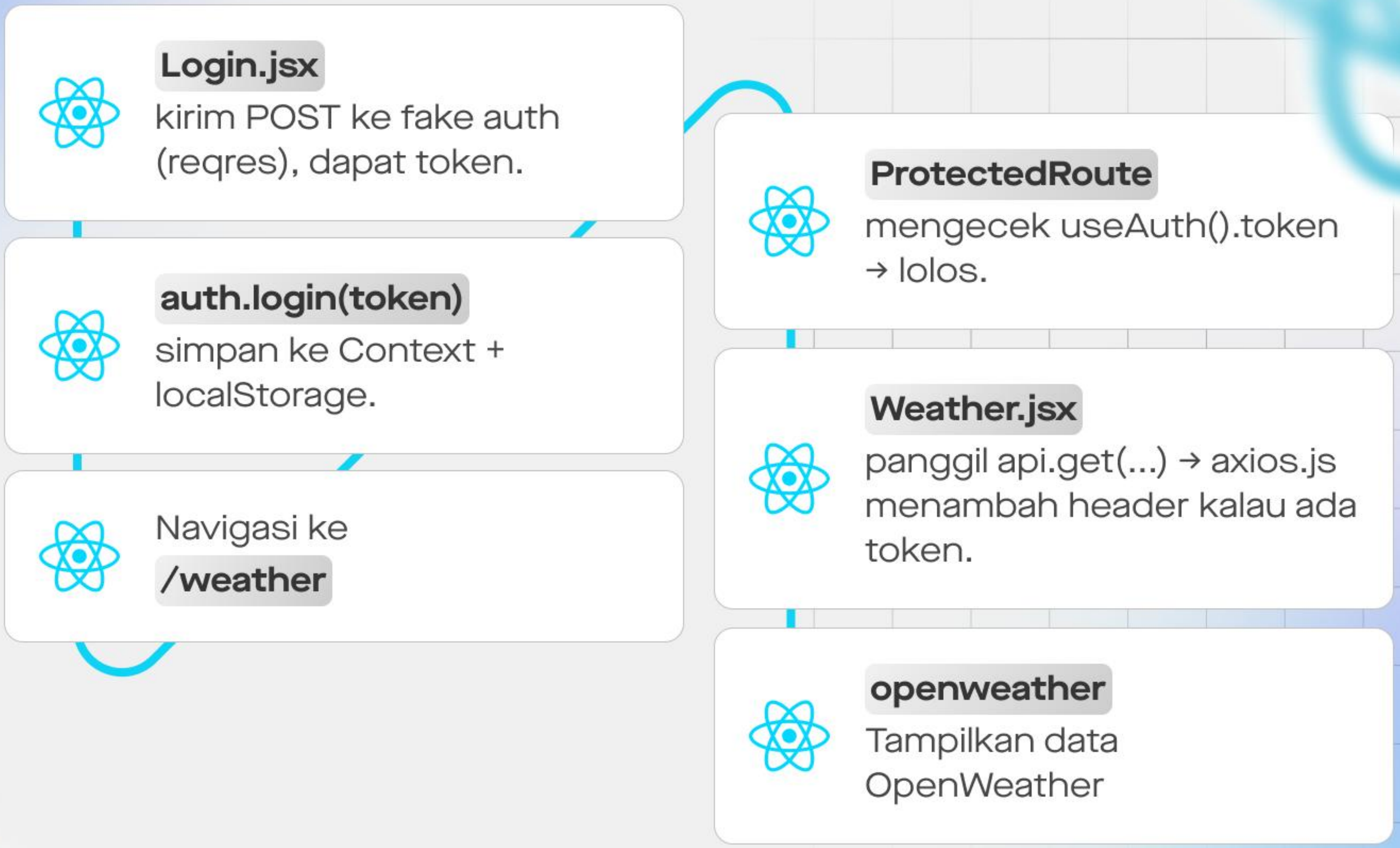
ProtectedRoute.jsx:

1. Header Authorization otomatis.
2. Satu tempat untuk timeout, baseURL, dan penanganan error.

Axios Params



```
axios.get(“/items”, {params: {page: 1}})
const body = { title: "New Item", published: true };
axios.post(“/items”, body, {params: {page: 1}, headers: { "X-Request-Source": "web" }})
axios.put(“/items/${id}”, {params: {page: 1}})
axios.patch(“/items/${id}”, body, {params: {page: 1}})
axios.delete(“/items/${id}”, body, {params: {soft: true}})
```



Login.jsx
kirim POST ke fake auth (reqres), dapat token.

auth.login(token)
simpan ke Context + localStorage.

Navigasi ke **/weather**

ProtectedRoute
mengecek useAuth().token → lolos.

Weather.jsx
panggil api.get(...) → axios.js menambah header kalau ada token.

openweather
Tampilkan data OpenWeather

Ringkasan

github.com/khulqu15/reactjs-learn

rules.jsx

1. Consum API
2. Authentification
3. Lib, Routes Guard, Context

Next

1. Search, Caching, Favorites
2. Nested Routes