



Breakdown Harian - Minggu 8: Auth Flow & Routing Terproteksi



Tujuan Minggu Ini

- Memahami cara menyimpan dan mengelola token autentikasi di frontend.
 - Mengimplementasikan kondisi routing berdasarkan status autentikasi (user login/logout).
 - Membuat halaman atau rute yang hanya bisa diakses oleh user yang sudah terautentikasi (Protected Routes).
 - Mendapatkan pemahaman dasar tentang state management global untuk mengelola status autentikasi di seluruh aplikasi.
-



Hari 1 (Senin) - Menyimpan & Mengelola Token Autentikasi

Materi Inti (2 Jam)

- Review Proses Login
- Pentingnya Token di Frontend
- Opsi Penyimpanan Token di Frontend: `localStorage`, `sessionStorage`, Cookies
- Memilih `localStorage` untuk Latihan
- Implementasi Penyimpanan Token
- Mengambil Token
- Menghapus Token (Logout)

Praktik Mandiri (8 Jam)

- Lanjutkan proyek e-commerce dari minggu 7.
 - Modifikasi handler login untuk menyimpan token di `localStorage`.
 - Buat tombol Logout dan implementasikan fungsinya.
 - Verifikasi penyimpanan dan penghapusan token di browser developer tools.
-



Hari 2 (Selasa) - Kondisi Routing Login/Logout

Materi Inti (2 Jam)

- Masalah: Akses halaman Login/Register setelah login.
- Mengecek Status Autentikasi.
- Conditional Rendering pada Routing.
- Redirect Berdasarkan Status (sudah login -> redirect dari login/register; belum login -> redirect ke login).
- Menggunakan `useNavigate` dan `useEffect` untuk Redirect.

Praktik Mandiri (8 Jam)

- Tambahkan logika `useEffect` di `LoginPage.jsx` dan `RegisterPage.jsx` untuk redirect jika sudah login.
 - Buat komponen `DashboardPage.jsx` sederhana.
 - Konfigurasi route `/dashboard`.
 - Uji coba alur redirect setelah login dan saat mencoba akses login/register.
-



Hari 3 (Rabu) - Protected Routes

Materi Inti (2 Jam)

- Konsep Protected Route.
- Membuat Komponen `ProtectedRoute`.
- Integrasi dengan React Router v6 (`<Outlet>`).

Praktik Mandiri (8 Jam)

- Buat komponen `ProtectedRoute.jsx`.
 - Pindahkan route `/dashboard` ke dalam `<Route element={}<ProtectedRoute />>`.
 - Uji coba akses `/dashboard` saat belum login dan setelah login.
 - (Opsional) Buat halaman lain yang dilindungi.
-



Hari 4 (Kamis) - State Autentikasi Global (Context API)

Materi Inti (2 Jam)

- Masalah: Mengecek `localStorage` berulang, kebutuhan state global.
- Solusi: State Global.
- Menggunakan Context API untuk Autentikasi.
- Membuat Context dan Provider (`AuthContext`, `AuthProvider`).
- Mengintegrasikan Context dengan `ProtectedRoute`.
- Mengintegrasikan Context dengan Komponen Lain (misalnya Header).

Praktik Mandiri (8 Jam)

- Buat file `AuthContext.js` dan implementasikan Context API.
 - Tambahkan logika baca token dari `localStorage` saat init di `AuthProvider`.
 - Modifikasi login/logout handler untuk menggunakan fungsi dari `useAuth`.
 - Modifikasi `ProtectedRoute` untuk menggunakan `useAuth().isAuthenticated`.
 - Gunakan `useAuth()` di komponen UI untuk menampilkan status login.
 - Pastikan alur autentikasi berfungsi dengan Context API.
-



Hari 5 (Jum'at) - Review, State Management Lanjutan (Pengantar Redux/Zustand), & Tugas Mingguan

Materi Inti (2 Jam)

- Review Auth Flow.

- Kelebihan & Kekurangan Context API untuk State Global.
- Pengantar State Management Libraries (Lanjutan): Redux (Toolkit), Zustand.
- Memilih Library.
- Persiapan Tugas Mingguan.

Tugas Mingguan (Waktu Pengerjaan: Sisa Minggu + Akhir Pekan)

Tema Tugas: Menyempurnakan Alur Autentikasi dan Routing di Aplikasi E-commerce.

Persyaratan Fungsional:

1. Implementasi Auth Flow Lengkap.
2. Conditional Routing.
3. Protected Routes.
4. State Autentikasi Global (Context API).
5. Integrasi UI dengan Auth State.
6. Refactor Kode.
7. Dokumentasi ([README.md](#)).

Fitur Opsional (Nilai Tambah):

- Interceptor [axios](#) untuk header [Authorization](#).
- Menangani refresh token.
- Menggunakan Zustand atau Redux Toolkit.
- Menampilkan pesan error login/register yang informatif.

Output Tugas:

- Kode sumber aplikasi React yang sudah disempurnakan.
- File [README.md](#) yang diperbarui.
- Push ke repositori GitHub.



Sesi Presentasi (Waktu disesuaikan)

Fokus Presentasi:

- Demo alur login, akses halaman terproteksi, logout.
- Penjelasan implementasi [AuthContext](#) dan [ProtectedRoute](#).
- Bagaimana UI beradaptasi dengan status autentikasi.
- Tantangan dan solusi.



Tips Belajar Tambahan

- Pahami Alur Data.
- Gunakan React DevTools.
- Simulasikan API Respons.
- Fokus pada Error Handling.



Referensi Tambahan

- [React Documentation - Context](#)

- [React Router Documentation - Auth Examples](#)
 - [MDN Web Docs - Using the Web Storage API](#)
 - [MDN Web Docs - Cookies](#)
 - [JWT Introduction](#)
-

Minggu ini kita akan fokus pada aspek krusial dari aplikasi web modern: autentikasi dan otorisasi di sisi frontend. Membangun alur login/logout yang aman dan mengelola akses ke rute terproteksi adalah keterampilan fundamental. Pahami bagaimana state autentikasi mengalir di aplikasi Anda, baik menggunakan `localStorage` sederhana maupun state global dengan Context API.