Materi Bahan Ajar: Alur Kerja Aplikasi Frontend–Backend (React + Redux & ExpressJS + Session)

# Tujuan Pembelajaran

Setelah mempelajari materi ini, mahasiswa/siswa diharapkan dapat:  
1. Memahami bagaimana frontend dan backend berkomunikasi.  
2. Menjelaskan alur login dengan session di backend dan state management di frontend.  
3. Mampu mengimplementasikan login sederhana menggunakan React, Redux, dan ExpressJS.

# 1. Komponen Utama Aplikasi

## Frontend

- Dibangun dengan React.  
- Menggunakan Redux untuk menyimpan data pengguna (user state).  
- Menggunakan Axios untuk berkomunikasi dengan backend.

## Backend

- Dibangun dengan ExpressJS.  
- Menggunakan express-session untuk menyimpan data sesi pengguna.  
- Merespon permintaan (request) dari frontend dan mengatur autentikasi.

# 2. Alur Kerja Aplikasi

## 1. Pengguna membuka aplikasi (React)

Aplikasi React ditampilkan di browser. Redux state awal masih kosong (belum login).

## 2. Pengguna mengisi form login (email & password)

User mengetik data login di input form React. Data disimpan di state lokal React.

## 3. Klik tombol Login

Data dikirim ke backend menggunakan axios.post("/login").

## 4. Backend menerima dan memproses login

Jika valid, backend menyimpan data user ke dalam session dan mengirimkan respon ke frontend.

## 5. Frontend menyimpan data user ke Redux

Data user digunakan untuk mengubah state Redux dan menampilkan nama user.

## 6. Saat halaman di-refresh

Redux state hilang, maka frontend mengecek session di backend dan mengisi ulang Redux jika masih login.

# 3. Keamanan dan Session

- Backend menyimpan session di server, bukan di frontend.  
- Data session dihubungkan dengan browser melalui cookie session ID.  
- Cookie ini dikirim otomatis oleh browser saat request selanjutnya.

# 4. Ringkasan Data yang Disimpan

|  |  |  |
| --- | --- | --- |
| Lokasi Penyimpanan | Tujuan | Contoh |
| Redux (Frontend) | Menyimpan user login untuk tampilan UI | state.user.name = "Bagas" |
| Session (Backend) | Menyimpan identitas login yang valid di server | req.session.user = { name: "Bagas" } |
| Cookie (Browser) | Menyimpan session ID | connect.sid=xyz123... |

# 5. Diagram Alur Sederhana

[User Browser]  
 |  
 v  
[React App] --(axios POST)--> [Express Backend]  
 | |  
 |<------(user JSON)---------|  
 |  
 [Redux menyimpan user]  
 |  
(Refresh halaman)  
 |  
[React App] --(axios GET)--> /check-session  
 | |  
 |<------(user JSON)---------|  
 |  
 [Redux update lagi]

# Latihan Tambahan

1. Modifikasi sistem login agar menggunakan database (MySQL atau MongoDB).  
2. Simpan session ID secara aman menggunakan HTTPS dan opsi secure di cookie.  
3. Buat tombol Logout yang menghapus session dan Redux state.

# Kesimpulan

- Redux menyimpan state di sisi frontend (browser) untuk UI responsif.  
- Session menyimpan data user di server agar tetap login meskipun browser direfresh.  
- Kombinasi keduanya memungkinkan aplikasi web yang aman, cepat, dan dinamis.