

Context API

Context API

Apa itu Context API?

- Tools untuk mengelola state secara global di aplikasi React.
- Menghindari "prop drilling" (meneruskan props melalui banyak komponen).

Kapan Menggunakan Context API?

- Saat data perlu diakses oleh banyak komponen pada berbagai tingkatan.
- Contoh: tema, preferensi bahasa, data autentikasi.

Cara Kerja Context API

- **Membuat Context:** `createContext`
- **Provider:** Menyediakan data ke komponen turunannya.
- **Consumer:** Mengakses data dari Provider.

Contoh Implementasi

```
// AuthContext.jsx
import { createContext } from 'react'

const AuthContext = createContext({
  isAuthenticated: false,
  login: () => {},
  logout: () => {},
})
```

Contoh Implementasi

```
// AuthContext.jsx
const AuthProvider = ({ children }) => {
  const [isAuthenticated, setIsAuthenticated] = useState(false)

  const login = () => {
    setIsAuthenticated(true)
  }

  const logout = () => {
    setIsAuthenticated(false)
  }

  return (
    <AuthContext.Provider value={{ isAuthenticated, login, logout }}>
      {children}
    </AuthContext.Provider>
  )
}
```

Contoh Implementasi

```
// App.jsx
import { AuthProvider } from './context/AuthContext'

const App = () => {
  return (
    <AuthProvider>
      <Header />
      <Main />
      <Footer />
    </AuthProvider>
  )
}
```

Contoh Implementasi

```
// Header.jsx
import { useContext } from 'react'
import { AuthContext } from '../context/AuthContext'

const Header = () => {
  const { isAuthenticated, login, logout } = useContext(AuthContext)

  return (
    <header>
      {isAuthenticated ? (
        <button onClick={logout}>Logout</button>
      ) : (
        <button onClick={login}>Login</button>
      )}
    </header>
  )
}
```

Contoh Implementasi

```
// PrivateComponent.jsx
import { useNavigate } from 'react-router-dom'

const PrivateComponent = () => {
  const { isAuthenticated } = useContext(AuthContext)
  const navigate = useNavigate()

  useEffect(() => {
    if (!isAuthenticated) {
      navigate('/login')
    }
  }, [isAuthenticated, navigate])

  return <h1>Private Component</h1>
}
```


Keuntungan & Keterbatasan

Keuntungan:

- Memudahkan manajemen state global.
- Mengurangi prop drilling.

Keterbatasan:

- Tidak untuk manajemen state yang kompleks.
- Bisa menyebabkan render yang tidak perlu.