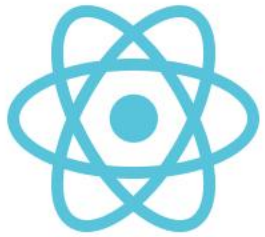
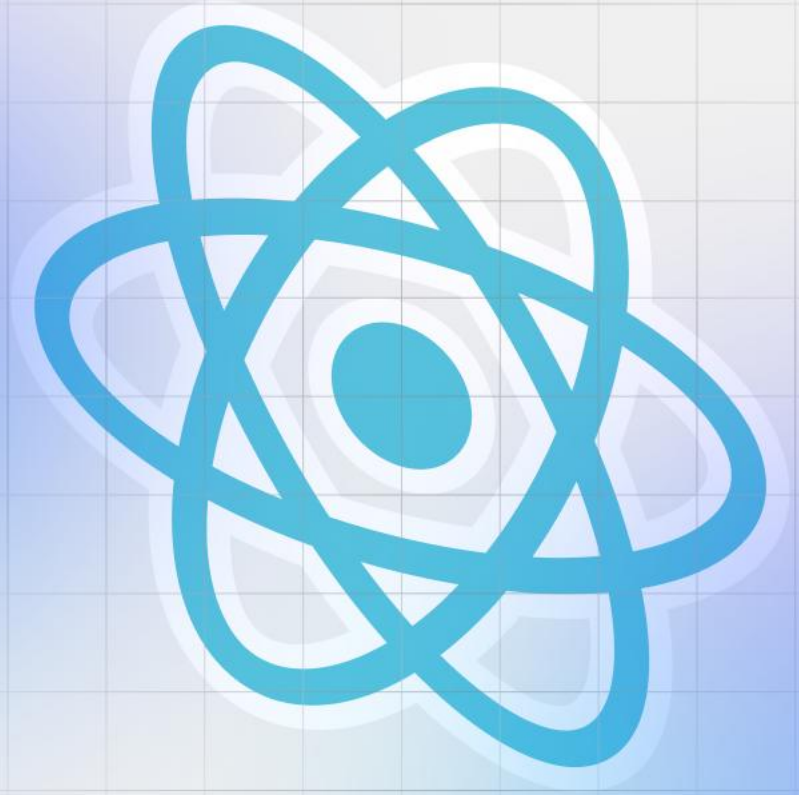


ReactJS

Search, Caching,
Favorites, Nested Routes

Day 4



Quick Review

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

```
npm run dev
```

Goals

1. List products & search (DummyJson)
2. Caching hasil agar hemat request
3. Favorites (localStorage)
4. Nested Routes

Endpoint API

List
/products

Search
/products/search?q=phone

Detail
/products/:id

```
api.get('https://dummyjson.com/products')
    .then(res => res.json())
    .then(console.log)
```

{ products, total, skip, limit }
dan product untuk detail.

UI & Alur Hari Ini

- `/shop`
  daftar + search + favorit toggle
- `/shop/:id`
  detail dalam layout yang sama (nested route)
- Cache per query + page
- Favorites tersimpan di localStorage

What’s New ?

create FavContext.jsx

create pages/shop

cmd

```
src/
  lib/axios.js
  context/FavContext.jsx
  pages/shop/ShopLayout.jsx
  pages/shop/ShopList.jsx
  pages/shop/ProductDetail.jsx
  App.jsx
```

Fav Context

src/context/FavContext.jsx

```jsx
import { createContext, useContext, useEffect, useState } from "react";
const FavCtx = createContext(null);
export const useFav = ()=> useContext(FavCtx);

export default function FavProvider({ children }){
  const [favs,setFavs] = useState(()=> JSON.parse(localStorage.getItem("FAVS")||"[]"));
  useEffect(()=> localStorage.setItem("FAVS", JSON.stringify(favs)), [favs]);

  const toggleFav = (id)=> setFavs(f=> f.includes(id) ? f.filter(x=>x!==id) : [...f,id]);
  const isFav = (id)=> favs.includes(id);

  return <FavCtx.Provider value={{favs, toggleFav, isFav}}>{children}</FavCtx.Provider>;
}
```

Manfaat: global & persisten.

Caching Sederhana

ShopList.jsx

```
const cache = {};

async function fetchProducts({ q="", page=1, limit=12 }) {
  const key = `${q}-${page}`;
  if (cache[key]) return cache[key];
  const skip = (page-1)*limit;
  const url = q
    ? `/products/search?q=${encodeURIComponent(q)}&limit=${limit}&skip=${skip}`
    : `/products?limit=${limit}&skip=${skip}`;
  const { data } = await api.get(url);
  cache[key] = data;
  return data;
}
```

Konsep: cache per kombinasi query & halaman.

Nested Routes Layout

ShopList.jsx

```jsx
import { Outlet, NavLink } from "react-router-dom";
export default function ShopLayout(){
 return (
  <div className="container mx-auto px-4 py-6">
   <nav className="mb-4 flex gap-3">
    <NavLink to="/shop" end className="btn">Katalog</NavLink>
   </nav>
   <Outlet />
  </div>
 );
}
```

Outlet untuk render ShopList / ProductDetail di tempat yang sama.

ShopList: State & Search Bar

src/pages/shop/ShopList.jsx

```
import { useEffect, useState } from "react";
import { Link, useSearchParams } from "react-router-dom";
import { api } from "../../lib/api";
import { useFav } from "../../context/FavContext";

const cache = {};

export default function ShopList() {
 const [params, setParams] = useSearchParams();
 const [q, setQ] = useState(params.get("q") || "");
 const [page, setPage] = useState(Number(params.get("page")||1));
 const [data, setData] = useState(null);
 const [loading,setLoading] = useState(true);
 const { isFav, toggleFav } = useFav();

 function syncParams(nextQ= q, nextPage= page){
  const sp = new URLSearchParams();
  if (nextQ) sp.set("q", nextQ);
  if (nextPage>1) sp.set("page", String(nextPage));
  setParams(sp);
 }
```

Catatan: search state sinkron ke URL query.

ShopList: Fetch + Render Grid

src/pages/shop/ShopList.jsx

```
  async function fetchProducts(){
    const key = `${q}-${page}`;
    if (cache[key]) { setData(cache[key]); setLoading(false); return; }
    setLoading(true);
    const limit = 12, skip = (page-1)*limit;
    const url = q
      ? `/products/search?q=${encodeURIComponent(q)}&limit=${limit}
&skip=${skip}`
      : `/products?limit=${limit}&skip=${skip}`;
    const { data } = await api.get(url);
    cache[key] = data;
    setData(data);
    setLoading(false);
  }

  useEffect(()=>{ syncParams(q, page); fetchProducts(); /* eslint-
disable-next-line */ }, [q,page]);

  return (
    <div className="space-y-4">
        <form onSubmit={e=>{e.preventDefault(); setPage(1);
          fetchProducts();}} className="join w-full">
            <input className="input input-bordered join-item w-full"
              placeholder="Cari produk (e.g. phone)"
              value={q} onChange={e=> setQ(e.target.value)} />
          <button className="btn btn-primary join-item" type="submit">Cari</
button>
        </form>

        {/* Grid */}
        {loading && <div className="skeleton h-32 w-full" />}
        {!loading && data && (
          <>
            <div className="grid gap-3 sm:grid-cols-2 md:grid-cols-3">
              {data.products.map(p=> (
                <div key={p.id} className="card bg-base-200">
                  <div className="card-body">
                    <h3 className="card-title">{p.title}</h3>
                    <p className="line-clamp-2 text-sm opacity-80">{p.description}
</p>
                    <div className="flex gap-2 mt-2">
                      <Link className="btn btn-sm btn-outline" to={`/shop/${p.id}`}
>Detail</Link>
                      <button className={`btn btn-sm ${isFav(p.id)?'btn-
secondary':'btn-ghost'}`} onClick={()=>toggleFav(p.id)}>
                        {isFav(p.id)? '★ Fav' : '☆ Fav'}
                      </button>
                    </div>
                  </div>
                </div>
              ))}
            </div>
```

ShopList: Pagination Sederhana

src/pages/shop/ShopList.jsx

```
        {/* pagination */}
        <div className="join mt-2">
          <button className="btn join-item" disabled={page<=1} onClick={()=>setPage(p=>p-1)}>Prev</button>
          <button className="btn join-item" onClick={()=>setPage(p=>p+1)}>Next</button>
        </div>
      </>
    )}
  </div>
 );
}
```

Catatan: skip=(page-1)*limit. Bisa tambahkan batas total.

ProductDetail (Nested)

src/pages/shop/ProductDetail.jsx

```jsx
import { useEffect, useState } from "react";
import { useParams, Link } from "react-router-dom";
import { api } from "../../lib/api";
import { useFav } from "../../context/FavContext";

export default function ProductDetail(){
 const { id } = useParams();
 const [p, setP] = useState(null);
 const [loading, setLoading] = useState(true);
 const { isFav, toggleFav } = useFav();

 useEffect(()=>{
  (async ()=>{
   try {
    const { data } = await api.get(`/products/${id}`);
    setP(data);
   } finally { setLoading(false); }
  })();
 },[id]);

 if (loading) return <div className="skeleton h-32 w-full" />;
 if (!p) return <div className="alert alert-error">Produk tidak
ditemukan.</div>;

 return (
  <div className="grid gap-4">
   <Link to="/shop" className="btn btn-ghost">← Kembali</Link>
   <div className="card bg-base-200">
    <div className="card-body">
     <h2 className="card-title">{p.title}</h2>
     <p>{p.description}</p>
     <div className="flex gap-2 mt-2">
      <button className={`btn ${isFav(p.id)?'btn-secondary':'btn-
outline'}`} onClick={()=>toggleFav(p.id)}>
       {isFav(p.id)? '★ Favorit' : '☆ Favorit'}
      </button>
      <span className="badge badge-outline">Price: ${p.price}</
span>
     </div>
    </div>
   </div>
  </div>
 );
}
```

Routing: Nested di App.jsx

src/App.jsx

```jsx
import { BrowserRouter, Routes, Route, Navigate } from "react-router-dom";
import ShopLayout from "./pages/shop/ShopLayout";
import ShopList from "./pages/shop/ShopList";
import ProductDetail from "./pages/shop/ProductDetail";
import FavProvider from "./context/FavContext";

export default function App(){
 return (
  <FavProvider>
   <BrowserRouter>
    <Routes>
     <Route path="/" element={<Navigate to="/shop" replace />} />
     <Route path="/shop" element={<ShopLayout />}>
      <Route index element={<ShopList />} />
      <Route path=":id" element={<ProductDetail />} />
     </Route>
    </Routes>
   </BrowserRouter>
  </FavProvider>
 );
}
```

/shop sebagai parent, children-nya index & :id.

Explore

github.com/khulqu15/reactjs-learn

1. Tambahkan page number & total dari total.
2. Tambah sort (price asc/desc), hanya client-side.
3. Simpan cache ke localStorage dengan TTL 5 menit.

See you next time

yorozuya.ninno