



Breakdown Harian - Minggu 9: Cart & Checkout



Tujuan Minggu Ini

- Memahami cara mengelola state keranjang belanja (cart) di frontend.
 - Mampu menambahkan produk ke keranjang.
 - Mampu menampilkan isi keranjang, serta mengupdate jumlah dan menghapus item.
 - Memahami proses pengiriman data checkout ke backend.
 - Melakukan simulasi transaksi sederhana.
-



Hari 1 (Senin) - Mengelola State Keranjang Belanja (Cart)

Materi Inti (2 Jam)

- Konsep state keranjang belanja di frontend.
- Memilih pendekatan state management untuk cart (Context API atau `useState` lokal).
- Struktur data untuk menyimpan item di keranjang (Array of Objects).
- Menambahkan item ke keranjang (logika cek duplikat, update kuantitas).

Praktik Mandiri (8 Jam)

- Lanjutkan proyek e-commerce.
 - Buat Context API baru (`CartContext`) atau gunakan state lokal di komponen parent untuk mengelola state keranjang.
 - Definisikan struktur state keranjang (misal: `[{ product: {...}, quantity: N }]`).
 - Buat fungsi `addToCart` yang menerima objek produk dan menambahkannya ke state keranjang. Tangani kasus jika produk sudah ada di keranjang.
 - Integrasikan fungsi `addToCart` ke tombol "Tambah ke Keranjang" di halaman detail produk atau daftar produk.
-



Hari 2 (Selasa) - Menampilkan Isi Keranjang

Materi Inti (2 Jam)

- Mengakses state keranjang dari Context API atau state lokal.
- Membuat komponen terpisah untuk menampilkan daftar item di keranjang (`CartItem`, `CartList`).
- Menampilkan detail setiap item (nama produk, harga, kuantitas, subtotal).
- Menampilkan total harga keranjang.

Praktik Mandiri (8 Jam)

- Buat halaman baru `/cart` dan komponen `CartPage.jsx`.
- Di `CartPage`, gunakan `useContext(CartContext)` atau akses state keranjang yang relevan.
- Buat komponen `CartItem.jsx` untuk menampilkan satu baris item keranjang.
- Render daftar item keranjang menggunakan `map` dari state keranjang, tampilkan detail item.
- Hitung dan tampilkan total harga semua item di keranjang.

- Tambahkan link ke halaman `/cart` di Header/Navbar.
-



Hari 3 (Rabu) - Update & Delete Item Keranjang

Materi Inti (2 Jam)

- Kebutuhan untuk mengubah kuantitas item atau menghapus item dari keranjang.
- Membuat fungsi `updateQuantity` di Context/state manager cart.
- Membuat fungsi `removeFromCart` di Context/state manager cart.
- Mengintegrasikan fungsi update/delete dengan UI di halaman keranjang.

Praktik Mandiri (8 Jam)

- Tambahkan tombol (+) dan (-) atau input number di komponen `CartItem` untuk mengubah kuantitas.
 - Implementasikan fungsi `updateQuantity` di `CartContext` (atau state manager Anda) yang mengubah kuantitas item berdasarkan ID produk.
 - Tambahkan tombol "Hapus" di komponen `CartItem`.
 - Implementasikan fungsi `removeFromCart` di `CartContext` (atau state manager Anda) yang menghapus item berdasarkan ID produk.
 - Hubungkan tombol update/delete di UI dengan fungsi yang sesuai di Context/state manager.
 - Uji coba menambah, mengubah kuantitas, dan menghapus item di keranjang.
-



Hari 4 (Kamis) - Proses Checkout (Frontend ke Backend)

Materi Inti (2 Jam)

- Memahami alur checkout: dari keranjang di frontend ke proses order di backend.
- Data apa saja yang perlu dikirim saat checkout (item keranjang, info user, alamat pengiriman, dll.).
- Membuat endpoint API di backend untuk menerima data checkout (jika ada backend nyata, jika tidak, simulasi).
- Mengirim data keranjang dari frontend ke backend menggunakan `axios` atau `fetch`.
- Menangani respons dari backend (sukses/gagal).

Praktik Mandiri (8 Jam)

- Buat tombol "Checkout" di halaman keranjang.
 - Buat fungsi `handleCheckout` di frontend.
 - Di dalam `handleCheckout`, ambil data keranjang dari state global/Context.
 - Siapkan data yang akan dikirim ke backend (sesuaikan dengan kebutuhan backend Anda).
 - Gunakan `axios.post` atau `fetch` untuk mengirim data checkout ke endpoint backend yang relevan (simulasi atau nyata).
 - Tangani respons: jika sukses, kosongkan keranjang dan arahkan user ke halaman sukses order; jika gagal, tampilkan pesan error.
 - (Simulasi) Jika tidak ada backend, simulasikan respons sukses/gagal setelah beberapa detik.
-



Hari 5 (Jum'at) - Simulasi Transaksi & Tugas Mingguan

Materi Inti (2 Jam)

- Review alur Cart & Checkout.
- Membahas simulasi transaksi sederhana di frontend (jika tidak ada backend nyata).
- Persiapan Tugas Mingguan.

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

Tema Tugas: Implementasi Fitur Cart & Checkout di Aplikasi E-commerce.

Persyaratan Fungsional:

1. **Manajemen Cart:** Implementasikan state keranjang belanja menggunakan Context API atau state management lain yang Anda pilih.
2. **Tambah ke Keranjang:** Fungsi untuk menambahkan produk ke keranjang dari halaman produk.
3. **Tampilkan Cart:** Halaman terpisah untuk menampilkan semua item di keranjang.
4. **Update Kuantitas:** Fungsi untuk mengubah kuantitas item di keranjang.
5. **Hapus Item:** Fungsi untuk menghapus item dari keranjang.
6. **Total Harga:** Tampilkan total harga semua item di keranjang.
7. **Proses Checkout:** Implementasikan logika untuk mengirim data keranjang ke backend (simulasi atau nyata).
8. **Kosongkan Cart:** Setelah checkout berhasil, kosongkan state keranjang.
9. **Dokumentasi (README.md):** Perbarui file `README.md` dengan penjelasan fitur cart & checkout.

Fitur Opsional (Nilai Tambah):

- Menyimpan state keranjang di `localStorage` agar tidak hilang saat refresh.
- Validasi stok atau ketersediaan produk saat menambah ke keranjang atau checkout.
- Menampilkan notifikasi (toast) saat item ditambahkan ke keranjang.
- Implementasi halaman sukses order atau halaman riwayat order (membutuhkan backend).

Output Tugas:

- Kode sumber aplikasi React yang sudah disempurnakan.
- File `README.md` yang diperbarui.
- Push semua perubahan ke repositori GitHub pribadi Anda.

Sesi Presentasi (Waktu disesuaikan)

Siapkan diri Anda untuk sesi presentasi singkat di awal minggu berikutnya. Fokus presentasi adalah:

- **Demo:** Tunjukkan alur menambah item ke cart, melihat/mengubah/menghapus item di cart, dan proses checkout.
- **Penjelasan Implementasi:** Jelaskan bagaimana Anda mengelola state cart (Context API atau lainnya) dan bagaimana data dikirim saat checkout.
- **Tantangan & Solusi:** Ceritakan tantangan yang Anda hadapi saat mengerjakan tugas dan bagaimana Anda menyelesaikannya.



Tips Belajar Tambahan

- **Immutability:** Ingat prinsip immutability saat mengupdate state array atau objek di React.
 - **Key Prop:** Pastikan menggunakan **key** prop yang unik saat me-render daftar item keranjang.
 - **Error Handling:** Pertimbangkan bagaimana menangani error saat mengirim data checkout ke backend.
-

Minggu ini kita akan membangun fitur inti dari aplikasi e-commerce: keranjang belanja dan proses checkout. Ini melibatkan manajemen state yang cukup dinamis di frontend dan komunikasi dengan backend. Fokus pada bagaimana data mengalir dan diubah saat user berinteraksi dengan keranjang mereka. Selamat belajar dan mengerjakan tugas!