

## **SDLC - Fitur: “Keranjang Kuning”**

### **1. Planning**

Tujuan Fitur: Menyediakan alur keranjang penuh dimana pengguna dapat menambah item, menghapus item, melihat ringkasan keranjang, lalu melakukan checkout (membuat order, memilih alamat & metode pembayaran, konfirmasi).

Value/ KPI (ukuran keberhasilan proyek):

- (keranjang → checkout) meningkat
- Bounce rate halaman produk turun
- Waktu rata-rata proses checkout < 2 menit (target)

Stakeholder: Product Owner, pengguna, dev frontend/backend, QA tester.

Scope MVP:

- Add to Cart (tombol + update qty bila sama)
- Remove from Cart (hapus item & clear cart)
- Cart view (list item, subtotal)
- Checkout flow: alamat pengiriman (pilih/masukkan), metode pembayaran, konfirmasi order → order disimpan
- Validasi stok, notifikasi error, badge jumlah di navbar.

### **2. Design**

#### **1. Desain Arsitektur Perangkat Lunak**

Pemilihan teknologi (sederhana untuk 1 fitur)

- frontend: HTML5, CSS/ React, Next.js
- backend: Node.js (Express)
- database: MySQL

Alur interaksi komponen:

1. User klik Add to Cart → Frontend kirim request:
2. Backend memvalidasi:
  - Apakah produk ada?
  - Apakah stok cukup?
3. Backend cek apakah produk sudah ada di keranjang:
  - Jika ada → update qty
  - Jika tidak → insert item baru
4. Simpan ke MySQL atau Redis.
5. Backend merespons jumlah item total
6. Frontend update ikon keranjang.

#### **2. Design Antarmuka Pengguna (UI)**

Interaksi Pengguna

Pada halaman produk pengguna melihat:

- Nama produk

- Harga
- Tombol “Tambah ke Keranjang”

Setelah mengklik:

- Notifikasi: “Produk ditambahkan ke keranjang”
- Badge jumlah item pada ikon keranjang bertambah

Hapus Item: konfirmasi

Checkout: disable tombol, spinner; show success

Error Handling: stok tidak cukup → show modal dengan opsi reduce qty atau remove.

Elemen UI yang harus ada

- Product card: gambar, nama, price, tombol Tambah ke Keranjang dan pilihan qty.
- Button “Add to Cart”
- Badge jumlah item pada navbar
- cart page: list items (gambar kecil, nama, price, qty (editable), tombol hapus per item), ringkasan (subtotal, ongkir (estimasi), kupon input, total), tombol checkout (disabled jika kosong).
- Checkout modal: pilih alamat pengiriman / input alamat baru, pilih metode pembayaran, review order & tombol confirm dan pay, loading & success/failure feedback.
- Validasi error (stok habis, qty melebihi limit)

### **3. Development**

Implementasi fitur:

1. Frontend
  - Membuat komponen tombol Add to Cart
  - Membuat fungsi call API
  - Update badge jumlah item
  - Menampilkan notifikasi sukses/gagal
2. Backend
  - Endpoint POST, PATCH, DELETE, GET
  - Validasi user login (JWT)
  - Cek stok
  - Insert/update keranjang
  - Response total item
3. Database
  - Membuat tabel
  - Index
4. Error Handling
  - 400 jika produk tidak ditemukan
  - 409 jika stok habis

### **4. Testing**

1. Pengujian Fungsional

- Unit tests backend: add, update, remove cart item logic; checkout transaction flows.
  - Berhasil menambah produk ke keranjang
  - Jika produk sudah ada, qty bertambah
  - Tidak bisa menambah lebih banyak dari stok
  - Badge jumlah item bertambah
2. Pengujian non-fungsional
    - Response API < 300ms
    - Test load 100 user: request tetap stabil
    - Validasi keamanan API
  3. regression testing
    - Pastikan fitur lain tidak rusak: Login, Halaman produk

## 5. Deployment

Deployment adalah proses memindahkan aplikasi dari tahap pengembangan sampai siap digunakan oleh pengguna. Biasanya, aplikasi melewati tiga lingkungan: development untuk tempat developer bekerja, staging untuk pengujian sebelum rilis, dan production sebagai aplikasi yang benar-benar dipakai user.

- Sebelum aplikasi dijalankan, frontend perlu *di-build* terlebih dahulu agar menjadi file siap hosting, dan backend juga perlu di-build lalu di deploy ke server.
- pengelolaan database menggunakan migration, yaitu sistem untuk mencatat perubahan struktur tabel.
- Rollback: image tagging & DB backup; menggunakan feature flags untuk mematikan fitur jika bug besar muncul.

## 6. Maintenance

- Monitor key metrics: failed checkout %, avg checkout time, cart abandonment  
Memantau metrik proses belanja, persentase gagal checkout, rata-rata waktu checkout, dan jumlah pengguna yang sudah isi keranjang.
- Automasi backup DB, test restore berkala  
Membuat cadangan database secara otomatis, dan rutin test secara berkala.
- Patch security dependencies, pentest periodik  
Memperbarui komponen yang digunakan supaya tidak ada celah keamanan, serta melakukan tes secara rutin apakah bisa diretas apa ngga.
- Triage bug & tech debt setiap sprint retrospective  
Setiap akhir sprint, tim perlu memilih dan menentukan prioritas bug mana yang harus diperbaiki dulu, serta mengevaluasi “utang teknis” pekerjaan yang ditunda untuk dirapikan agar ga numpuk.
- Exec plan untuk inventory reconciliation pada restock  
Saat barang di-restock, sistem harus memiliki rencana eksekusi untuk mengecek stok di sistem dan cek stok di gudang, dan menyamakan data stok agar akurat.

## 7. Metodologi Pengembangan (Agile)

Alasan penggunaan agile :

- Fitur e-commerce sering berubah berdasar feedback pengguna & bisnis → Agile memberi adaptabilitas.
- Iterasi pendek (2 minggu) memungkinkan rilis incremental (mis. Add & Remove dulu, lalu Checkout).
- Stakeholder bisa priorisasikan fitur (promo, pembayaran baru) dengan cepat.
- Transparansi & continuous delivery mengurangi risk besar di akhir proyek.

## 8. Scrum Framework (penerapan)

Roles (peran dalam scrum):

- Product Owner (PO)  
menentukan apa yang harus dibuat dan mana yang jadi prioritas.  
Contoh: PO menempatkan fitur checkout sebagai prioritas paling tinggi karena menghasilkan pendapatan.
- Scrum Master  
membantu tim agar bekerja tanpa hambatan, memastikan proses Scrum berjalan benar, menjaga ritme sprint, dan menghilangkan masalah yang menghalangi tim.
- Dev Team: frontend, backend, QA, devops.

Ceremonies (kegiatan dalam scrum):

- Sprint Planning (menentukan rencana kerja 1 sprint)
- Daily Standup (15 menit)
- Sprint Review (demo fitur add/remove/checkout ke PO)
- Sprint Retrospective (perbaikan proses kerja untuk sprint berikutnya)

Artifacts:

- Product Backlog, (daftar kebutuhan produk)
- Sprint Backlog, (daftar tugas yang dipilih untuk dikerjakan dalam 1 sprint)
- Sprint Increment (produk versi terbaru setelah sprint).

## 9. Product Backlog

Daftar semua kebutuhan atau fitur yang ingin dibuat dalam suatu proyek.  
keranjang kuning: Add, Remove, Checkout.

Id	User Story	Priority	Estimasi	Status
----	------------	----------	----------	--------

PB-01	Sebagai user, saya ingin menambah produk ke keranjang agar bisa berbelanja.	High	5	Backlog
PB-02	Sebagai user, jika menambah produk yang sama, qty bertambah (not duplicate).	High	3	Backlog
PB-03	Sebagai user, saya ingin melihat isi keranjang dan subtotal.	High	5	Backlog
PB-04	Sebagai user, saya ingin menghapus item dari keranjang.	High	3	Backlog
PB-05	Sebagai user, saya ingin mengedit jumlah (qty) di keranjang.	Medium	2	Backlog
PB-06	Sebagai user, saya ingin checkout (pilih alamat & metode pembayaran) sehingga pesanan dibuat.	High	8	Backlog
PB-07	Sebagai sistem, saat checkout stok harus divalidasi dalam transaksi.	High	5	Backlog
PB-08	Sebagai user, saya ingin melihat notifikasi sukses/gagal saat checkout.	Medium	2	Backlog

Keterangan: status backlog: rencana

## 11. Sprint Activities

Aktivitas harian yang dilakukan selama 1 sprint

Day	Activity
Day 1	Sprint Planning – Menentukan Sprint Goal & task
Day 2-13	Development + Daily Stand-up (15 menit)
Day 14	Sprint Review – Demo fitur <i>Keranjang Kuning</i>
Day 15	Sprint Retrospective – Evaluasi proses sprint

## 12. Sprint Backlog (Sprint#1)

Daftar tugas yang dikerjakan khusus untuk Sprint #1 (1 periode kerja). Diambil dari Product Backlog

Task ID	User Story	Deskripsi Task	SP	PIC	Status
T-01	PB-01	Membuat database migration untuk tabel carts dan cart_items	2	Backend	Todo
T-02	PB-01	Membuat API Add to Cart supaya user bisa menambahkan barang ke keranjang.	3	Backend	Todo
T-03	PB-01	FE Membuat tombol Add to Cart di frontend.	2	Frontend	Todo
T-04	PB-02	Menambahkan logika agar qty otomatis bertambah jika item yang sama ditambahkan ulang.	3	Backend	Todo
T-05	PB-03	Membuat API untuk mengambil isi keranjang beserta subtotalnya.	2	Backend	Todo
T-06	PB-03	FE Membuat tampilan halaman keranjang (Cart Page)	2	Frontend	Todo
T-07	PB-04	Membuat API untuk menghapus item dari keranjang.	2	Backend	Todo
T-08	PB-04	FE Membuat UI tombol hapus item	2	Frontend	Todo
T-09	PB-05	FE Membuat tampilan untuk mengubah jumlah (qty) item	2	Frontend	Todo
T-10	PB-06	Membuat API Checkout untuk membuat data orders dan order_items.	5	Backend	Todo
T-11	PB-06	FE Membuat UI Checkout	3	Frontend	Todo
T-12	PB-06	Membuat tes E2E untuk alur checkout (dari tambah sampai checkout).	2	QA	Todo

T-13	PB-07	Membuat validasi stok menggunakan transaksi database agar stok tidak minus	2	Backend	Todo
T-14	PB-08	Membuat notifikasi sukses atau gagal setelah checkout.	1	Frontend	Todo

### 13. Sprint Board

Tampilan papan tugas (Todo → In progress → Done) yang digunakan untuk melihat siapa mengerjakan apa dan statusnya.

Todo	In Progress	Done
T-01	-	-
T-02	-	-
T-03	-	-
T-04	-	-
T-05	-	-
T-06	-	-
T-07	-	-
T-08	-	-
T-09	-	-
T-10	-	-