**Rencana Sprint untuk Aplikasi E-Commerce**

**Sprint 1: Desain dan Pengembangan Katalog Produk**

**Durasi:** 2 Minggu (tanggal)

**Tujuan Sprint:**

* Membuat sistem login untuk Admin dan User.
* Mengembangkan Dashboard

**Tugas Utama:**

1. **Desain Antarmuka Pengguna (UI) Katalog Produk:**
   * Sidebar navigasi dengan menu tambahan "Katalog Produk".
   * Halaman utama berisi daftar produk dalam bentuk tabel atau kartu produk.
   * Tombol aksi (Tambah Produk, Edit, Hapus).
   * Mendiskusikan desain UI dengan tim untuk mendapatkan umpan balik sebelum implementasi.
2. **Implementasi Backend Katalog Produk:**
   * Membuat model database untuk Menu.  
     Membuat Tabel Menus

Id\_menu, nama\_menu, harga, deskripsi, dan category\_id

* + Mengembangkan API untuk CRUD (Create, Read, Update, Delete) produk

**Create**: Menambahkan produk baru.

**Read**: Menampilkan daftar produk.

**Update**: Mengedit produk.

**Delete**: Menghapus produk.

1. **Implementasi Frontend Katalog Produk:**
   * Menyusun komponen untuk menampilkan daftar produk.

Membuat halaman "Katalog Produk" yang menampilkan daftar produk dalam format tabel.

Menambahkan fitur pencarian, filter kategori, dan pagination untuk meningkatkan navigasi.

* + Menghubungkan frontend dengan API backend untuk data produk.

Menggunakan Axios atau Fetch API untuk mengambil dan mengirim data dari API backend.

Menampilkan data produk di halaman frontend.

1. **Integrasi dan Pengujian:**
   * Integrasi.

Menghubungkan fitur frontend dan backend untuk memastikan data katalog produk tampil dengan baik.

* + Pengujian Fungsional

Melakukan pengujian fitur CRUD:

* Tambah produk baru.
* Edit produk yang ada.
* Hapus produk.
* Tampilkan semua produk dengan filter dan pagination.
  + Uji Kegunaan

Menguji responsivitas dan kegunaan halaman katalog

1. **Dokumentasi:**
   * Dokumentasi Pengguna.

Menyusun panduan penggunaan fitur katalog produk, seperti cara menambahkan, mengedit, dan menghapus produk.

* + Dokumentasi Pengembang.

Menulis dokumentasi teknis terkait API endpoint, model database, dan struktur frontend yang digunakan.

**Kriteria Keberhasilan:**

* Halaman katalog produk dapat menampilkan daftar produk dengan benar.
* User dapat menambah, mengedit, dan menghapus Pemesanan melalui antarmuka pengguna.

**Sprint 2: Pengembangan Keranjang Belanja**

**Durasi:** 2 Minggu (tanggal)

**Tujuan Sprint:**

* Mengembangkan fitur Pemesanan supaya bisa untuk melakukan pemesanan.

**Tugas Utama:**

1. **Desain Antarmuka Pengguna (UI) Keranjang Belanja:**
   * Membuat wireframe dan mockup untuk halaman keranjang belanja.

Menambahkan menu baru "Keranjang Belanja" di sidebar navigasi (seperti menu "Menus", "Paket Hemat", dll.).

Mendesain wireframe dan mockup halaman keranjang belanja dengan:

* + Daftar produk yang ditambahkan ke keranjang.
  + Kolom jumlah produk yang dapat diperbarui.
  + Tombol "Hapus Produk" untuk menghapus item dari keranjang.
  + Ringkasan total harga produk dalam keranjang.
  + Melakukan review desain dan mendapatkan umpan balik.

1. **Implementasi Backend Keranjang Belanja:**
   * **Model Database**: Membuat tabel keranjang dengan kolom:

id (Primary Key)

id\_user (Foreign Key ke tabel users)

id\_produk (Foreign Key ke tabel produk)

jumlah (Integer)

total\_harga (Integer, dihitung berdasarkan jumlah dan harga produk).

* + **API CRUD Keranjang**: Mengembangkan endpoint untuk:

Menambahkan produk ke keranjang.

Menampilkan daftar item dalam keranjang.

Memperbarui jumlah produk.

Menghapus produk dari keranjang.

1. **Implementasi Frontend Keranjang Belanja:**
   * Menambahkan menu "Keranjang Belanja" di sidebar navigasi.
   * Membuat halaman keranjang belanja dengan komponen:

Tabel atau daftar produk dalam keranjang.

Tombol untuk memperbarui jumlah produk.

Tombol untuk menghapus produk.

Tampilan total harga keranjang.

* + Menghubungkan halaman frontend dengan API backend menggunakan Axios atau Fetch API.

1. **Integrasi dan Pengujian:**
   * Mengintegrasikan API backend dengan halaman frontend keranjang belanja.
   * Melakukan pengujian fungsional untuk memastikan:

Produk dapat ditambahkan ke keranjang.

Jumlah produk bisa diperbarui.

Produk dapat dihapus dari keranjang.

Total harga diperbarui sesuai perubahan jumlah produk.

1. **Dokumentasi:**
   * Menyusun dokumentasi teknis mengenai struktur database, API endpoint, dan implementasi frontend.
   * Menyediakan panduan pengguna untuk fitur keranjang belanja.

**Kriteria Keberhasilan:**

* User dapat menambahkan dan menghapus pemesanan dari keranjang belanja.

**Sprint 3: Implementasi Sistem Pembayaran**

**Durasi:** 2 Minggu (tanggal)

**Tujuan Sprint:**

* Mengembangkan dan mengintegrasikan sistem pembayaran termasuk pemrosesan pembayaran dan konfirmasi pesanan.

**Tugas Utama:**

1. **Desain Antarmuka Pengguna (UI) Pembayaran:**
   * Menambahkan menu "Pembayaran" di sidebar navigasi (bersebelahan dengan menu "Pemesanan" atau "Keranjang Belanja").
   * Membuat wireframe dan mockup halaman pembayaran yang meliputi:

Formulir input informasi pembayaran (nama, metode pembayaran, dll.).

Ringkasan pesanan (produk, jumlah, total harga).

Tombol "Bayar" untuk melanjutkan pembayaran.

Notifikasi status pembayaran (berhasil/gagal).

* + Melakukan review desain dan mendapatkan umpan balik.

1. **Implementasi Backend Sistem Pembayaran:**
   * Mengembangkan API untuk pemrosesan pembayaran dan konfirmasi pesanan.

Membuat endpoint untuk memproses pembayaran.

Menyimpan data pembayaran ke database, seperti:

* id\_pembayaran, id\_user, id\_pesanan, metode\_pembayaran, status\_pembayaran, total\_bayar
  + **Integrasi Gateway Pembayaran**:

Mengintegrasikan sistem dengan gateway pembayaran seperti Stripe, PayPal, atau Midtrans.

Menangani respons sukses/gagal dari gateway pembayaran.

1. **Implementasi Frontend Sistem Pembayaran:**
   * Menyusun komponen frontend untuk formulir pembayaran dan konfirmasi pesanan.

* Formulir pembayaran yang terhubung ke backend.
* Tampilan ringkasan pesanan dan total harga.
* Menampilkan status pembayaran setelah transaksi selesai (notifikasi berhasil/gagal).
  + Menghubungkan frontend dengan API backend untuk memproses pembayaran menggunakan Axios atau Fetch API.

1. **Integrasi dan Pengujian:**
   * Mengintegrasikan sistem pembayaran antara frontend, backend, dan gateway pembayaran.
   * Melakukan **pengujian end-to-end**, termasuk:

* Simulasi pembayaran sukses dan gagal.
* Verifikasi penyimpanan data pembayaran ke database.

- Konfirmasi status pesanan setelah pembayaran berhasil.

1. **Dokumentasi:**
   * Menyusun dokumentasi untuk fitur pembayaran dan instruksi integrasi.
   * Menyediakan panduan pengguna untuk melakukan pembayaran.
   * Menulis instruksi integrasi sistem pembayaran bagi pengembang.

**Kriteria Keberhasilan:**

* Sistem pembayaran dapat memproses transaksi melalui gateway pembayaran.
* Pengguna dapat melihat status pembayaran secara real-time (berhasil/gagal).
* Pesanan pengguna mendapatkan konfirmasi pembayaran yang akurat.
* Notifikasi pembayaran ditampilkan setelah transaksi selesai.

**Sprint 4: Pengembangan Manajemen Pengguna**

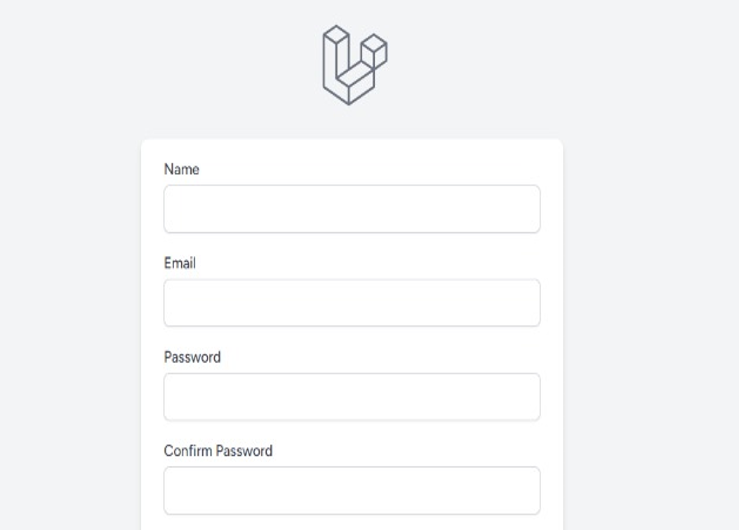
**Durasi:** 2 Minggu (tanggal)

**Tujuan Sprint:**

* Mengembangkan fitur manajemen pengguna termasuk pendaftaran, login, profil pengguna, dan manajemen akun.

**Tugas Utama:**

1. **Desain Antarmuka Pengguna (UI) Manajemen Pengguna:**
   * Halaman pendaftaran



* + Melakukan review desain dan mendapatkan umpan balik.

Melakukan review wireframe dan mockup bersama tim.

Menindaklanjuti umpan balik untuk menyempurnakan desain.

1. **Implementasi Backend Manajemen Pengguna:**
   * Membuat model database untuk pengguna.

Id, name, email, role\_id

* + API Pendaftaran.

Endpoint untuk registrasi pengguna baru dengan validasi input.

* + API Login.
* Endpoint autentikasi pengguna menggunakan email dan password.
* Menggunakan **JWT** atau Laravel Passport untuk manajemen token autentikasi.

1. **Implementasi Frontend Manajemen Pengguna:**
   * Formulir Pendaftaran.
     1. Komponen frontend untuk input data pendaftaran
     2. Menampilkan validasi input
   * Formular Login.
     1. Komponen untuk input email dan password.
     2. Menyimpan token autentikasi.
   * Autorisasi.
     1. Membuar middleware atau guard untuk membatasi akses.
2. **Integrasi dan Pengujian:**
   * Mengintegrasikan frontend dan backend.
   * Melakukan pengujian fungsional.
     1. Pengujian pendaftaran akun, autentikasi login, dan pembaruan profil.
     2. Menangani scenario error seperti email duplikat
   * Pengujian Keamanan:
     1. Pengujian keamanan autentikasi menggunakan token dan enkripsi password.
3. **Dokumentasi:**
   * Menyusun dokumentasi untuk fitur manajemen pengguna:
     1. Struktur table database pengguna.
     2. Endpoint API untuk registrasi, login, dan profil pengguna.
   * Menyediakan panduan pengguna:
     1. Cara mendaftar, login, dan mengelola profil.
   * Menambahkan instruksi teknis untuk integrasi frontend dan backend.

**Kriteria Keberhasilan:**

* Pengguna dapat mendaftar, login, dan mengelola profil mereka dengan sukses.
* Sistem autentikasi dan otorisasi berfungsi dengan baik.

**Sprint 5: Pengujian dan Perbaikan Bug**

**Durasi:** 2 Minggu (tanggal)

**Tujuan Sprint:**

* Melakukan pengujian menyeluruh untuk aplikasi, memperbaiki bug, dan meningkatkan kualitas aplikasi.

**Tugas Utama:**

1. **Pengujian Fungsional:**
   * Menguji setiap fitur secara individual untuk memastikan fungsionalitas bekerja dengan baik.
   * Contoh:
     1. Menambahkan produk ke keranjang.
     2. Proses pembayaran hingga konfirmasi pesanan.
     3. Manajemen profil pengguna (edit data, avatar, dll).
   * Pengujian Manual:
     1. Memastikan setiap halaman dan komponen dapat diakses dan bekerja sesuai dengan tujuan desain.
     2. Menjalankan skenario penggunaan seperti pengguna biasa (mendaftar, login, transaksi).
2. **Pengujian Integrasi:**
   * Pengujian Hubungan Antarfungsi:
     1. Menguji integrasi antara keranjang belanja dengan sistem pembayaran.
     2. Menguji bagaimana data pengguna berpindah antar fitur (misalnya, profil pengguna dan proses pemesanan).
   * Contoh Pengujian:
     1. Menambahkan produk ke keranjang → Melakukan pembayaran → Konfirmasi pesanan → Validasi data yang disimpan.
3. **Pengujian Kinerja:**
   * Load testing:
     1. Mengukur performa aplikasi ketika banyak pengguna melakukan transaksi secara bersamaan.
   * Responsivitas Aplikasi:
     1. Memastikan aplikasi bekerja dengan cepat dan tanpa log.
   * Optimasi Kode:
     1. Mengevaluasi performa query database dan API yang berjalan lambat.
     2. Meminimalkan ukuran aset frontend (CSS, JS) jika diperlukan.
4. **Perbaikan Bug:**
   * Identifikasi dan Prioritas Bug:
     1. Menyusun daftar bug berdasarkan tingkat keparahan:
        1. **Kritis**: Bug yang mencegah pengguna menyelesaikan transaksi atau menggunakan fitur penting.
        2. **Sedang**: Bug yang menyebabkan ketidaknyamanan tetapi tidak menghentikan fungsi utama.
        3. **Ringan**: Bug minor seperti tampilan yang kurang rapi atau error kecil.
   * Perbaikan dan Pengujian Ulang:
     1. Memperbaiki bug yang ditemukan selama pengujian.
     2. Melakukan pengujian regresi untuk memastikan bug yang diperbaiki tidak menimbulkan masalah baru.
5. **Dokumentasi:**
   * Laporan Pengujian:
     1. Menyusun laporan hasil pengujian yang mencakup:
        1. Fitur yang diuji.
        2. Bug yang ditemukan dan status perbaikannya.
        3. Skenario pengujian yang berhasil dan gagal

**Kriteria Keberhasilan:**

* Semua fitur berfungsi sesuai dengan yang diharapkan tanpa bug kritis.
* Aplikasi memiliki performa yang baik dan responsif.
* Aplikasi memiliki **kinerja yang optimal** dan responsif di berbagai kondisi penggunaan.
* Laporan pengujian dan perbaikan bug **tersusun rapi** dan terdokumentasi.