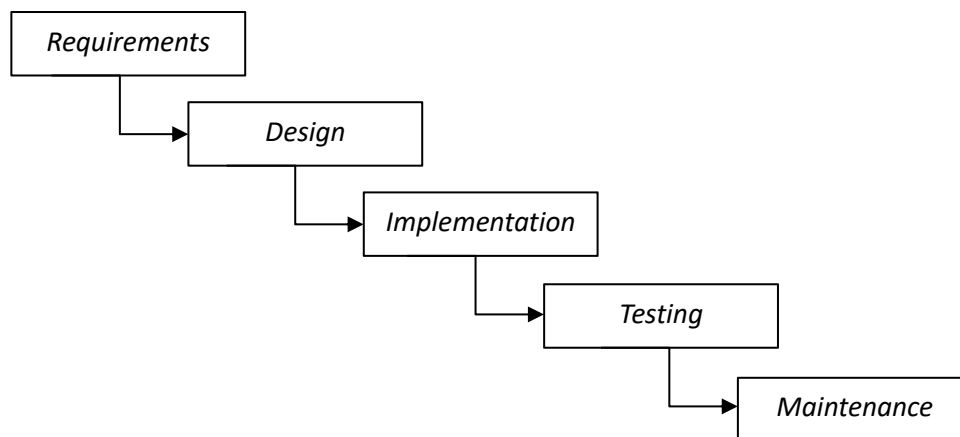


## BAB III

### METODOLOGI PENELITIAN

Data kebutuhan sistem diperoleh melalui wawancara semi-terstruktur dengan pelaku usaha tembakau yang beroperasi di wilayah Temanggung dan sekitarnya, yang dipilih secara purposive berdasarkan keterlibatan aktif mereka dalam proses produksi dan distribusi. Metodologi *Waterfall* dipilih dalam penelitian ini sebagai pendekatan rekayasa perangkat lunak yang sistematis dan terstruktur [7]. Metodologi *Waterfall* dipilih karena menawarkan kerangka pengembangan yang jelas, berurutan, dan mudah diimplementasikan dalam tahap-tahap yang saling berkesinambungan [8]. Tahapan-tahapan dalam metode ini dilakukan secara linear dari atas ke bawah (seperti air terjun), sehingga tahapan pengembangan dilakukan secara berurutan, di mana setiap tahap harus terselesaikan sebelum melanjutkan ke tahap selanjutnya [9].



Gambar 1. Penelitian Metode *Waterfall*

#### 3.1 *Requirements* (Analisis Kebutuhan)

Sebelum proses pengembangan dimulai, analisis terhadap kebutuhan pengguna harus dilakukan guna memastikan perangkat lunak yang dibangun dapat memenuhi fungsi yang diinginkan. Tahap identifikasi informasi melibatkan berbagai pendekatan, di antaranya mengumpulkan dan mendokumentasikan semua kebutuhan dari pengguna. Misalnya: fitur transaksi, laporan penjualan, dan manajemen stok. Informasi yang telah dikumpulkan kemudian dianalisis guna memperoleh data yang menyeluruh mengenai kebutuhan spesifik pengguna terhadap perangkat lunak yang dikembangkan [10].

#### 3.2 *Design* (Perancangan Sistem)

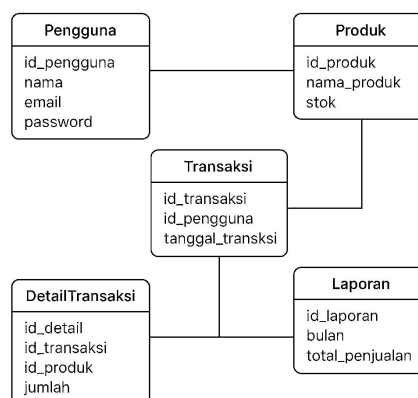
Tahap perancangan sistem memiliki peran krusial dalam memastikan bahwa kebutuhan pengguna diterjemahkan secara optimal ke dalam struktur perangkat lunak yang dibangun. Perancangan sistem merupakan fase esensial yang menentukan arah teknis dan fungsional dari pengembangan perangkat lunak. sebagai tahapan kunci, perancangan sistem berfungsi

menyusun fondasi logis dan teknis sebelum implementasi dilakukan [11]. Dalam penelitian ini, pendekatan perancangan dilakukan dengan menyusun model konseptual dan logis yang merepresentasikan alur informasi dan interaksi antar entitas dalam sistem penjualan tembakau.

Tabel 1. *Entity Relationship Diagram (ERD)*

Entitas	Atribut Utama	Relasi
<b>Pengguna</b>	id_pengguna (PK), nama, email, password	1:N ke Transaksi
<b>Produk</b>	id_produk (PK), nama_produk, stok, harga	1:N ke Transaksi
<b>Transaksi</b>	id_transaksi (PK), id_pengguna (FK), tanggal_transaksi	N:1 ke Pengguna, N:M ke Produk (via detail)
<b>Detail Transaksi</b>	id_detail (PK), id_transaksi (FK), id_produk (FK), jumlah	N:1 ke Transaksi dan Produk
<b>Laporan</b>	id_laporan (PK), bulan, total_penjualan	1:1 ke Transaksi (agregasi periodik)

Sebagai langkah awal dalam proses desain sistem, ERD dimanfaatkan untuk menggambarkan relasi antar entitas penting seperti *user*, produk, proses transaksi, dan penyusunan laporan. Tujuan penggunaan ERD adalah agar perancangan basis data lebih terstruktur dan mendukung integritas data [12]. Setiap entitas memiliki atribut kunci yang membantu proses normalisasi database, sehingga mengurangi redundansi data dan meningkatkan efisiensi pencarian informasi.



Gambar 2. Diagram ERD (*Entity Relationship Diagram*)

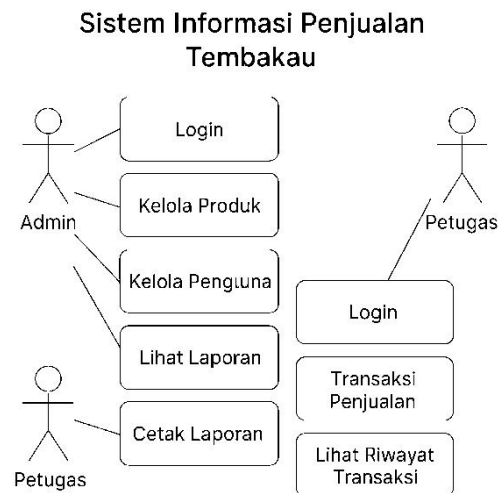
Selanjutnya, Diagram use case ini digunakan untuk mendefinisikan keterlibatan pengguna dalam berbagai proses sistem, serta memastikan bahwa setiap fitur dirancang berdasarkan

kebutuhan aktual serta membantu menguraikan kebutuhan fungsional sistem berdasarkan skenario penggunaan dari sisi aktor utama seperti admin dan *user*.

Tabel 2. Diagram ERD

Aktor	Use Case
Admin	Login, Kelola Produk, Kelola Pengguna, Lihat Laporan, Cetak Laporan
Petugas	Login, Transaksi Penjualan, Lihat Riwayat Transaksi
Sistem	Validasi Login, Proses Stok Otomatis, Hasilkan Laporan Penjualan

Tujuan dari perancangan diagram ini adalah untuk mengidentifikasi skenario-skenario penggunaan yang menjadi dasar pengembangan fitur dalam sistem informasi. Melalui representasi grafis, diagram ini menggambarkan alur interaksi antara pengguna dan komponen sistem yang relevan, sehingga membantu dalam merancang fungsi-fungsi utama.



Gambar 3. Use Case Diagram

Selain itu, antarmuka pengguna (*User Interface/UI*) dirancang menggunakan prinsip keterbacaan, kejelasan navigasi, dan konsistensi visual. Perangkat bantu seperti Figma digunakan untuk membuat mockup halaman, agar dapat divalidasi sejak awal oleh calon pengguna sistem. Desain antarmuka berorientasi pada Interaksi dan persepsi pengguna terhadap sistem (UX), dirancang agar memudahkan navigasi serta mempercepat proses operasional transaksi dan pelaporan penjualan.

Tabel 3. Sketsa Desain UI (*Mockup* – Deskripsi Tekstual)

Halaman	Elemen Utama
Dashboard	Tampilan halaman depan berisi Informasi Tobacco
Halaman Produk	Tabel CRUD produk, tombol tambah/ubah/hapus
Transaksi	Form pencarian produk, daftar keranjang, tombol simpan transaksi
Laporan Penjualan	Pilihan filter tanggal/bulan, tabel ringkasan, ekspor ke PDF/Excel
Login	Input email & password, tombol <i>login</i> , notifikasi kesalahan <i>login</i>

## 2.1 Implementation (Pembuatan Coding)

Tahap implementasi dilakukan setelah proses perancangan sistem selesai dan divalidasi. Pada penelitian ini, *framework Laravel* digunakan sebagai fondasi utama pengembangan sistem informasi penjualan tembakau. *Laravel* dipilih karena memiliki struktur pendekatan MVC memungkinkan pemisahan tugas antara layer tampilan, kontrol interaksi, dan representasi data, sehingga meningkatkan modularitas dan kemudahan pemeliharaan sistem [13]. Pengembangan sistem dimulai dengan konfigurasi environment menggunakan Composer dan *Laravel Installer*. Setiap modul fungsional seperti autentikasi pengguna, manajemen produk, transaksi penjualan, dan pelaporan diimplementasikan menggunakan fitur bawaan *Laravel* seperti Eloquent ORM, middleware, serta blade templating engine untuk tampilan antarmuka. Selain itu, database MySQL digunakan untuk menyimpan data transaksi, produk, dan laporan secara terstruktur, dengan memanfaatkan fitur migration dan seeder untuk efisiensi pembangunan database [14].

## 2.2 Testing (Pengujian Sistem)

Metode *black-box* testing digunakan dalam tahap uji coba guna memastikan bahwa setiap komponen sistem memberikan output yang sesuai berdasarkan input yang diberikan, tanpa mempertimbangkan struktur internal kode. Data *dummy* digunakan untuk mensimulasikan transaksi dan mengevaluasi kinerja sistem. Pengujian dilakukan seperti proses verifikasi sistem dan dilakukan secara berkelanjutan melalui pengujian unit, integrasi, serta penerimaan pengguna, guna memastikan setiap komponen berfungsi sesuai spesifikasi [15].

## 2.3 Maintenance (Pemeliharaan & Evaluasi)

Setelah sistem diuji, dilakukan evaluasi berdasarkan hasil uji coba dan umpan balik dari pengguna. Tahap ini juga mencakup dokumentasi teknis dan panduan penggunaan sistem. Tahap pemeliharaan (*maintenance*) sepanjang tahapan pengembangan sistem perangkat lunak bertujuan guna menjamin bahwa sistem beroperasi sesuai dengan fungsi yang telah dirancang dan tetap berjalan optimal sehingga relevan terhadap kebutuhan pengguna seiring waktu. Pemeliharaan mencakup perbaikan kesalahan yang tidak terdeteksi saat pengujian, adaptasi terhadap perubahan lingkungan, serta peningkatan fungsionalitas

berdasarkan masukan dari pengguna. Evaluasi dilakukan dengan melibatkan pengguna akhir dalam proses uji coba sistem (*user acceptance testing*) untuk mengidentifikasi tingkat keberfungsian dan kegunaan (*usability*) sistem. Masukan yang diperoleh kemudian dianalisis untuk menentukan apakah perlu dilakukan perbaikan atau pengembangan lebih lanjut. Hasil evaluasi juga menjadi dasar dalam menentukan kelayakan sistem untuk implementasi skala penuh dalam lingkungan operasional nyata. Proses pemeliharaan dalam konteks sistem informasi penjualan tembakau ini bersifat iterative dan responsif terhadap kebutuhan pengguna UMKM, mengingat dinamika bisnis mereka yang dapat berubah seiring waktu. Dokumentasi teknis dan manual pengguna disiapkan untuk mendukung proses pelatihan dan pemeliharaan jangka panjang.