

BAB II

KAJIAN LITERATUR

2.1 Sistem Informasi

Perkembangan teknologi informasi yang semakin pesat telah membawa perubahan besar dalam cara organisasi maupun pelaku usaha menjalankan kegiatan operasionalnya. Aktivitas yang sebelumnya dilakukan secara manual kini banyak dialihkan ke dalam sistem berbasis *digital* agar proses kerja menjadi lebih cepat, terstruktur, dan efisien. Pengelolaan data secara manual umumnya memiliki berbagai keterbatasan, seperti risiko kesalahan pencatatan, data yang tersebar di banyak tempat, keterlambatan penyampaian informasi, serta kesulitan dalam melakukan pengawasan terhadap aktivitas usaha secara menyeluruh. Oleh karena itu, dibutuhkan suatu sistem yang mampu mengelola data secara terintegrasi sehingga informasi yang dihasilkan dapat mendukung kebutuhan operasional maupun pengambilan keputusan [6, 7].

Sistem informasi merupakan suatu kesatuan yang menggabungkan manusia, teknologi, data, dan prosedur kerja untuk mengumpulkan, mengolah, menyimpan, serta menyajikan informasi yang dibutuhkan pengguna. Keberadaan sistem informasi tidak hanya berfungsi sebagai alat bantu pencatatan data, tetapi juga sebagai sarana untuk mendukung proses operasional, pengawasan, koordinasi, dan pengambilan keputusan dalam suatu organisasi maupun usaha. Dalam konteks usaha di bidang penjualan dan pengelolaan stok, sistem informasi memiliki peran yang sangat penting karena aktivitas operasional melibatkan banyak data yang saling berkaitan. Data menu, data bahan baku, data stok, data transaksi penjualan, hingga data laporan perlu dikelola secara konsisten agar tidak menimbulkan selisih informasi di lapangan. Sistem informasi memungkinkan seluruh data tersebut berada dalam satu *platform* yang saling terhubung, sehingga perubahan pada satu bagian dapat langsung memengaruhi data lain yang terkait [6 – 8].

Secara umum, alur kerja sistem informasi dimulai dari proses pemasukan data (*input*) oleh pengguna melalui antarmuka sistem. Data yang masuk kemudian disimpan ke dalam basis data dan diproses sesuai aturan bisnis yang telah ditetapkan. Proses tersebut dapat berupa validasi data, perhitungan transaksi, pembaruan data, maupun pengolahan laporan. Hasil pengolahan selanjutnya disajikan kembali dalam bentuk informasi yang dapat diakses pengguna sesuai kebutuhan dan hak akses yang dimiliki. Dengan mekanisme tersebut, sistem informasi mampu mengubah data mentah menjadi informasi yang bernilai guna bagi aktivitas operasional maupun manajerial.

Perkembangan teknologi jaringan juga mendorong banyak sistem informasi diimplementasikan dalam bentuk berbasis *web*, yang memungkinkan pengguna mengakses sistem melalui peramban tanpa perlu melakukan instalasi aplikasi khusus pada setiap perangkat. Data yang tersimpan pada *server* terpusat dapat diperbarui secara lebih cepat dan digunakan bersama oleh banyak pengguna dalam waktu yang sama, memberikan keunggulan berupa kemudahan akses, konsistensi data, dan fleksibilitas penggunaan [6, 7].

Dalam konteks penelitian ini, konsep sistem informasi diterapkan dalam perancangan sistem manajemen stok dan penjualan pada Codel Coffee. Sistem yang dibangun diharapkan mampu membantu proses pengelolaan menu, pencatatan transaksi, pemantauan persediaan, serta penyusunan laporan secara lebih cepat dan akurat dibandingkan proses manual. Dengan tersedianya sistem informasi berbasis *web* yang terintegrasi, *owner* maupun *staff* dapat mengakses data dan informasi sesuai peran masing-masing kapan saja dan dari mana saja, sehingga pengelolaan usaha Codel Coffee dapat berjalan lebih tertata, efektif, dan berbasis data [6, 8].

2.1.1 Sistem

Sistem merupakan sekumpulan elemen atau komponen yang saling berhubungan dan bekerja sama untuk mencapai tujuan tertentu. Setiap komponen di dalam sistem memiliki fungsi masing-masing, namun seluruh komponen tersebut harus dapat berinteraksi secara terkoordinasi agar tujuan sistem dapat tercapai secara efektif. Suatu sistem tidak dapat dipahami hanya dari satu bagian saja, melainkan dari hubungan antarbagian yang membentuk satu kesatuan kerja. Oleh karena itu, keberhasilan suatu sistem sangat dipengaruhi oleh kemampuan setiap komponen dalam menjalankan perannya serta keterpaduan antar komponen yang ada [7].

Secara umum, sistem bekerja melalui suatu alur dasar yang terdiri dari masukan (*input*), proses (*process*), dan keluaran (*output*). *Input* merupakan data, sumber daya, atau perintah yang masuk ke dalam sistem untuk diolah. Tahap proses adalah bagian inti sistem, yaitu ketika data yang diterima diolah berdasarkan aturan, prosedur, atau logika tertentu. Hasil dari pengolahan tersebut kemudian menjadi *output*, yaitu keluaran yang dapat dimanfaatkan oleh pengguna dalam bentuk informasi, tindakan, atau keputusan. Hubungan antara *input*, *process*, dan *output* menunjukkan bahwa sistem bekerja secara dinamis dan memiliki alur yang terstruktur [7].

Selain ketiga komponen utama tersebut, sistem juga memiliki mekanisme umpan balik (*feedback*) dan pengendalian (*control*). *Feedback* berfungsi untuk menilai apakah hasil keluaran

yang dihasilkan sudah sesuai dengan tujuan yang diharapkan. Jika ditemukan ketidaksesuaian, maka sistem dapat diperbaiki atau disesuaikan agar kinerjanya kembali optimal. Sementara itu, *control* digunakan untuk memastikan bahwa seluruh proses berjalan sesuai aturan yang telah ditetapkan. Adanya *feedback* dan *control* menjadikan sistem tidak bersifat statis, tetapi mampu beradaptasi terhadap perubahan maupun permasalahan yang muncul selama proses berjalan [7, 9].

Dalam konteks penelitian ini, sistem dapat dilihat pada proses operasional Codel Coffee yang terdiri dari berbagai komponen yang saling berkaitan, seperti pengguna sistem, data menu, data stok, transaksi penjualan, dan laporan. Ketika staf melakukan transaksi penjualan, data transaksi masuk sebagai *input* yang kemudian diproses oleh sistem untuk menghasilkan informasi penjualan dan laporan yang dibutuhkan. Selain itu, pengelolaan stok dilakukan melalui proses stok awal, restok, pengeluaran stok, dan penyesuaian stok yang saling terintegrasi dalam sistem. Jika ditemukan kesalahan data atau ketidaksesuaian jumlah stok, maka proses evaluasi dan koreksi dapat dilakukan melalui fitur penyesuaian stok yang tersedia. Hal tersebut menunjukkan bahwa operasional Codel Coffee pada dasarnya merupakan suatu sistem yang terdiri dari berbagai komponen yang saling berinteraksi dan dapat ditingkatkan efektivitasnya melalui dukungan teknologi informasi.

2.1.2 Informasi

Informasi merupakan hasil dari pengolahan data yang telah diproses sehingga memiliki makna, nilai guna, dan relevansi bagi penggunaannya. Data pada dasarnya masih berupa fakta mentah yang belum dapat langsung digunakan sebagai dasar pengambilan keputusan. Oleh karena itu, data perlu melalui serangkaian proses tertentu agar berubah menjadi informasi yang mampu memberikan pemahaman terhadap kondisi yang sedang terjadi. Dengan kata lain, perbedaan utama antara data dan informasi terletak pada tingkat pengolahan serta manfaat yang dihasilkan bagi pengguna [7].

Proses pembentukan informasi yang berkualitas dimulai dari pengumpulan data yang relevan dari berbagai aktivitas operasional. Data yang terkumpul kemudian melewati tahapan pengolahan, seperti pengelompokan, perhitungan, peringkasan, atau perbandingan, agar lebih mudah dipahami dan dianalisis. Hasil dari proses ini kemudian disajikan dalam bentuk informasi yang siap digunakan oleh pengguna, baik dalam bentuk tampilan data pada layar maupun laporan terstruktur. Agar informasi yang dihasilkan benar-benar bermanfaat, informasi tersebut harus

memenuhi beberapa kriteria kualitas, yaitu akurat, relevan, tepat waktu, dan lengkap. Informasi yang akurat mencerminkan kondisi yang sebenarnya, informasi yang relevan sesuai dengan kebutuhan pengguna, informasi yang tepat waktu tersedia saat dibutuhkan, sedangkan informasi yang lengkap tidak menimbulkan kekurangan yang dapat menyebabkan kesalahan interpretasi [7].

Dalam sistem manajemen stok dan penjualan, kualitas informasi menjadi faktor yang sangat penting karena setiap keputusan operasional bergantung pada informasi yang tersedia. Informasi stok yang akurat membantu pengguna mengetahui jumlah persediaan aktual sehingga pengadaan bahan baku dapat dilakukan secara tepat. Informasi penjualan yang lengkap dan tepat waktu membantu pemilik usaha mengevaluasi performa penjualan, melihat menu yang paling diminati pelanggan, serta menilai perkembangan pendapatan dalam periode tertentu [7, 8].

Pada penelitian ini, sistem yang dirancang untuk Codel Coffee diharapkan mampu mengubah data transaksi dan data stok menjadi informasi yang bernilai bagi pemilik usaha maupun staf operasional. Dengan tersedianya informasi yang cepat, akurat, dan mudah dipahami, proses pengambilan keputusan dapat dilakukan secara lebih efektif sehingga pengelolaan usaha dapat berjalan lebih terarah dan efisien.

2.2 Sistem Informasi Manajemen

Sistem Informasi Manajemen (SIM) merupakan pengembangan lanjutan dari konsep sistem informasi yang secara khusus dirancang untuk mendukung kegiatan manajerial dalam suatu organisasi atau perusahaan. Jika sistem informasi umum berfokus pada pencatatan, pengolahan, dan penyimpanan data operasional, maka SIM menempatkan hasil pengolahan data tersebut sebagai dasar dalam proses manajemen. Oleh karena itu, SIM berperan dalam menyediakan informasi yang dibutuhkan untuk perencanaan, pengendalian, koordinasi, evaluasi, serta pengambilan keputusan secara lebih terarah. Informasi yang dihasilkan melalui SIM umumnya bersifat ringkas, relevan, dan mudah dipahami sehingga dapat digunakan oleh pihak manajemen sesuai kebutuhan organisasi [7].

Dalam Sistem Informasi Manajemen, data yang berasal dari berbagai aktivitas operasional dikumpulkan dari beberapa bagian yang saling berkaitan, kemudian diproses menjadi informasi yang memiliki nilai guna lebih tinggi. Informasi tersebut tidak hanya menunjukkan kondisi yang sedang terjadi, tetapi juga membantu manajemen memahami pola kerja, mengevaluasi hasil yang telah dicapai, serta mengidentifikasi masalah yang memerlukan tindakan lanjutan. Dengan kata

lain, SIM berfungsi sebagai jembatan antara aktivitas operasional sehari-hari dengan kebutuhan informasi di tingkat manajemen, sehingga keputusan yang diambil tidak hanya berdasarkan intuisi, tetapi berdasarkan data yang telah diolah secara sistematis [7, 9].

Pemanfaatan teknologi komputer dalam SIM memberikan banyak keuntungan dibandingkan sistem manual. Proses pengolahan data dapat dilakukan lebih cepat, hasil yang diperoleh lebih konsisten, serta risiko kesalahan pencatatan dapat dikurangi. Data yang tersimpan dalam basis data terpusat juga dapat diolah menjadi berbagai bentuk informasi, seperti laporan periodik, tabel ringkasan, maupun grafik perkembangan kinerja sesuai kebutuhan pengguna. Selain itu, sistem yang terkomputerisasi memungkinkan informasi disampaikan dengan lebih efisien karena pengguna tidak perlu melakukan rekapitulasi data secara manual yang cenderung memakan waktu dan rentan menimbulkan ketidaksesuaian data [6, 7].

Secara umum, manfaat utama Sistem Informasi Manajemen dapat dijelaskan sebagai berikut [7]:

1. Mendukung Perencanaan: Menyediakan informasi yang dibutuhkan dalam penyusunan target, strategi, dan rencana kerja.
2. Mendukung Pengendalian: Membantu manajemen memantau jalannya kegiatan operasional agar tetap sesuai tujuan.
3. Mendukung Evaluasi: Menyediakan dasar penilaian terhadap hasil kerja dan kinerja organisasi.
4. Mendukung Pengambilan Keputusan: Menjadi landasan dalam menentukan kebijakan atau tindakan yang tepat berdasarkan data.
5. Meningkatkan Efisiensi Pengelolaan: Mengurangi pekerjaan manual dan mempercepat penyediaan informasi yang dibutuhkan.

Dalam konteks sistem manajemen stok dan penjualan pada Codel Coffee, SIM berfungsi untuk mengolah data stok, data transaksi penjualan, dan data pendukung lainnya menjadi informasi yang berguna bagi pemilik usaha dalam menjalankan fungsi manajerial. Informasi seperti laporan stok harian, laporan penjualan harian, tren penjualan, serta kondisi persediaan barang membantu pemilik usaha dalam memantau jalannya operasional, mengevaluasi performa usaha, dan menentukan kebijakan pengadaan bahan baku secara lebih tepat. Penerapan SIM berbasis web juga memberikan fleksibilitas karena informasi dapat diakses kapan saja dan dari mana saja sesuai kebutuhan pemilik usaha. Dengan demikian, pengelolaan usaha di Codel Coffee tidak hanya

berjalan berdasarkan kebiasaan, tetapi didukung oleh informasi yang sistematis, aktual, dan dapat dipertanggungjawabkan [6 – 8].

2.3 Sistem Manajemen Stok dan Penjualan

Sistem manajemen stok dan penjualan merupakan bagian dari penerapan sistem informasi yang berfokus pada pengelolaan persediaan barang serta aktivitas penjualan dalam satu mekanisme yang saling terintegrasi. Pada banyak usaha, stok dan penjualan merupakan dua aktivitas yang tidak dapat dipisahkan karena setiap transaksi penjualan akan memengaruhi jumlah persediaan yang tersedia. Oleh sebab itu, kedua proses tersebut perlu dikelola dalam satu sistem agar data yang dihasilkan tetap konsisten, akurat, dan mudah dipantau. Melalui sistem yang terintegrasi, pelaku usaha dapat menjalankan kegiatan operasional secara lebih tertata serta mengurangi risiko kesalahan pencatatan yang sering terjadi pada proses manual [7, 8].

Dalam praktiknya, sistem manajemen stok dan penjualan mengelola berbagai jenis data operasional, seperti data produk atau menu, data bahan baku, data persediaan, data transaksi penjualan, data pelanggan, serta data laporan. Setiap data tersebut saling berkaitan dan harus diperbarui secara berkelanjutan sesuai aktivitas yang terjadi. Ketika transaksi penjualan dilakukan, sistem tidak hanya mencatat nilai transaksi, tetapi juga dapat memperbarui jumlah stok, menyimpan riwayat transaksi, dan menyiapkan data untuk kebutuhan pelaporan. Dengan demikian, sistem berfungsi bukan hanya sebagai alat pencatat transaksi, tetapi juga sebagai pusat pengendalian operasional usaha [8].

Tujuan utama penerapan sistem manajemen stok dan penjualan dapat dijelaskan sebagai berikut [7, 8]:

1. **Menjaga Akurasi Data Persediaan:** Membantu pengguna mengetahui jumlah stok yang tersedia secara tepat.
2. **Mempermudah Pencatatan Penjualan:** Setiap transaksi dicatat secara sistematis dan tersimpan sebagai riwayat data.
3. **Mendukung Pengawasan Operasional:** Pemilik usaha dapat memantau aktivitas penjualan dan kondisi stok secara berkala.
4. **Menyediakan Laporan yang Dibutuhkan:** Sistem menghasilkan laporan penjualan maupun laporan stok sebagai bahan evaluasi.

5. Meningkatkan Efisiensi Kerja: Proses pencatatan dan pelaporan menjadi lebih cepat dibandingkan metode manual.

Dalam konteks penelitian ini, sistem manajemen stok dan penjualan diterapkan pada operasional Codel Coffee yang memiliki kebutuhan untuk mengelola data menu, bahan baku, transaksi penjualan, serta laporan usaha secara terintegrasi. Dengan sistem yang dibangun, staf dapat melakukan pencatatan transaksi dan pembaruan data stok dengan lebih mudah, sedangkan pemilik usaha dapat memperoleh informasi mengenai kondisi persediaan dan performa penjualan secara lebih cepat. Penerapan sistem ini diharapkan mampu meningkatkan ketertiban administrasi, mempercepat alur kerja, serta membantu pengambilan keputusan yang lebih tepat berdasarkan data yang tersedia.

2.3.1 Sistem Manajemen Stok

Sistem manajemen stok merupakan sistem yang digunakan untuk mengelola persediaan barang agar ketersediaannya tetap terjaga sesuai kebutuhan operasional. Pengelolaan stok menjadi hal yang penting karena persediaan berhubungan langsung dengan kelancaran aktivitas usaha. Jika jumlah stok terlalu sedikit, proses pelayanan dapat terganggu karena bahan atau barang yang dibutuhkan tidak tersedia. Sebaliknya, jika stok terlalu banyak, usaha berisiko mengalami penumpukan barang, pemborosan biaya penyimpanan, maupun kerusakan barang akibat terlalu lama disimpan. Oleh karena itu, sistem manajemen stok dibutuhkan untuk menjaga keseimbangan antara ketersediaan dan kebutuhan persediaan [7, 8].

Dalam praktiknya, sistem manajemen stok mengelola berbagai informasi yang berkaitan dengan barang atau bahan yang digunakan dalam kegiatan operasional. Informasi tersebut dapat berupa nama barang, kategori, jumlah stok, satuan, harga, pemasok, tanggal masuk, hingga riwayat perubahan stok. Data tersebut harus diperbarui secara berkala agar kondisi persediaan yang tercatat di dalam sistem sesuai dengan kondisi aktual di lapangan. Dengan data yang akurat, pengguna dapat mengetahui kapan stok perlu ditambah, barang mana yang paling sering digunakan, serta item mana yang perlu diawasi karena jumlahnya mulai menipis [8].

Fungsi utama sistem manajemen stok dapat dijelaskan sebagai berikut [7, 8]:

1. Mencatat Data Persediaan: Menyimpan informasi barang atau bahan baku secara terstruktur.
2. Memantau Jumlah Stok: Menunjukkan kondisi persediaan yang tersedia saat ini.

3. Mengelola Penambahan dan Pengurangan: Memperbarui stok berdasarkan pembelian, pemakaian, atau penjualan.
4. Menyimpan Riwayat Perubahan: Mencatat setiap pergerakan stok untuk kebutuhan pelacakan.
5. Mendukung Perencanaan Pengadaan: Membantu menentukan kapan dan berapa jumlah stok yang perlu ditambah.
6. Mengurangi Risiko Kehabisan atau Penumpukan: Menjaga keseimbangan persediaan agar lebih efisien.

Dalam konteks penelitian ini, sistem manajemen stok diterapkan pada operasional Codel Coffee untuk mengelola bahan baku yang digunakan dalam penjualan menu, seperti kopi, susu, gula, dan bahan pendukung lainnya. Sistem yang dibangun membantu staf dalam mencatat penambahan stok, memantau jumlah persediaan, serta mengetahui bahan yang mulai menipis. Selain itu, data stok yang tersimpan juga dapat digunakan oleh pemilik usaha sebagai dasar dalam menentukan kebijakan pengadaan bahan baku agar operasional usaha tetap berjalan lancar. Dengan adanya sistem manajemen stok yang terintegrasi, proses pengelolaan persediaan di Codel Coffee diharapkan menjadi lebih tertib, akurat, dan efisien.

2.3.2 Sistem Manajemen Penjualan

Sistem manajemen penjualan merupakan suatu sistem yang dirancang untuk mengelola seluruh aktivitas penjualan secara terstruktur, mulai dari pencatatan transaksi, pengelolaan data produk, perhitungan pembayaran, hingga penyusunan laporan penjualan. Sistem ini berperan penting dalam membantu perusahaan atau usaha dalam mengontrol proses penjualan agar berjalan lebih efektif, efisien, dan akurat. Dengan adanya sistem yang terkomputerisasi, proses pencatatan transaksi tidak lagi dilakukan secara manual sehingga dapat meminimalkan kesalahan perhitungan maupun kehilangan data [7, 8].

Dalam kegiatan operasional bisnis, penjualan merupakan salah satu proses utama karena berhubungan langsung dengan pendapatan usaha. Oleh sebab itu, pengelolaan data penjualan harus dilakukan secara cepat dan tepat agar informasi yang dihasilkan dapat digunakan sebagai dasar pengambilan keputusan. Sistem manajemen penjualan mampu menyediakan informasi mengenai jumlah transaksi, produk yang paling sering terjual, pendapatan harian, hingga riwayat transaksi yang telah terjadi. Informasi tersebut sangat berguna bagi pemilik usaha dalam mengevaluasi kinerja bisnis dan menyusun strategi penjualan berikutnya [7, 8].

Secara umum, sistem manajemen penjualan memiliki beberapa fungsi utama, yaitu [7, 8]:

1. Pencatatan Transaksi Penjualan: Sistem digunakan untuk mencatat setiap transaksi yang terjadi, seperti produk yang dibeli, jumlah item, harga, diskon, dan total pembayaran.
2. Pengelolaan Data Produk/Menu: Sistem membantu menampilkan daftar produk atau menu yang tersedia beserta harga dan status ketersediaannya.
3. Perhitungan Otomatis: Sistem melakukan perhitungan subtotal, pajak, diskon, dan total pembayaran secara otomatis sehingga mengurangi kesalahan manual.
4. Penyimpanan Riwayat Transaksi: Data transaksi yang telah dilakukan disimpan ke dalam basis data sehingga dapat dilihat kembali sewaktu-waktu.
5. Penyusunan Laporan Penjualan: Sistem dapat menghasilkan laporan penjualan harian, mingguan, maupun bulanan sebagai bahan evaluasi usaha.

Pada usaha seperti Codel Coffee, sistem manajemen penjualan dibutuhkan untuk mempercepat proses transaksi, meningkatkan akurasi pencatatan, serta menghasilkan laporan penjualan secara otomatis. Selain itu, sistem penjualan yang terintegrasi dengan manajemen stok juga dapat mengurangi jumlah stok secara otomatis setiap kali transaksi terjadi, sehingga data persediaan selalu diperbarui secara real-time. Dengan demikian, operasional usaha menjadi lebih tertata dan pengambilan keputusan dapat dilakukan berdasarkan data yang akurat.

2.3.3 Metode Pengeluaran Barang

Metode pengeluaran barang merupakan pendekatan yang digunakan dalam sistem manajemen stok untuk menentukan urutan barang yang dikeluarkan dari persediaan ketika terjadi penjualan, pemakaian, atau distribusi barang. Metode ini penting karena berpengaruh terhadap ketepatan pencatatan stok, nilai persediaan akhir, serta efisiensi pengelolaan barang di dalam usaha. Pemilihan metode yang tepat membantu organisasi menjaga ketersediaan barang, mengurangi pemborosan, dan memastikan alur keluar masuk persediaan dapat dikendalikan dengan baik [9].

Dalam praktiknya, terdapat beberapa metode pengeluaran barang yang umum digunakan dalam pengelolaan persediaan. Setiap metode memiliki karakteristik, kelebihan, dan penerapan yang berbeda sesuai jenis barang maupun kebutuhan usaha. Metode yang paling sering dibahas dalam manajemen persediaan antara lain FIFO (*First In First Out*), LIFO (*Last In First Out*), dan

Average Cost. Selain itu, pada beberapa jenis usaha juga dikenal pendekatan *Just In Time* (JIT) yang menekankan ketersediaan barang sesuai kebutuhan agar persediaan tidak berlebihan [9]

Secara umum, beberapa metode pengeluaran barang dapat dijelaskan sebagai berikut [9]:

1. FIFO (*First In First Out*): Barang yang pertama masuk ke persediaan akan menjadi barang yang pertama dikeluarkan.
2. LIFO (*Last In First Out*): Barang yang terakhir masuk akan menjadi barang yang pertama dikeluarkan.
3. *Average Cost*: Nilai persediaan dihitung berdasarkan rata-rata harga seluruh barang yang tersedia.
4. *Just In Time* (JIT): Pengadaan barang dilakukan sedekat mungkin dengan waktu kebutuhan sehingga jumlah persediaan disimpan seminimal mungkin.

Di antara beberapa metode tersebut, FIFO merupakan metode yang paling relevan untuk usaha yang menggunakan barang dengan masa simpan terbatas atau bahan baku yang harus dijaga kualitasnya. Dengan metode FIFO, barang yang lebih dahulu masuk akan digunakan lebih dahulu sehingga risiko barang rusak, kedaluwarsa, atau menurun kualitasnya dapat dikurangi. Selain itu, metode ini mempermudah pelacakan stok karena urutan keluar barang mengikuti waktu penerimaan barang [9].

Pada sistem manajemen stok Codel Coffee, penerapan FIFO dilakukan melalui pencatatan setiap penambahan stok berdasarkan tanggal masuk barang. Setiap bahan baku yang dibeli dan dimasukkan ke dalam sistem akan memiliki data jumlah, tanggal masuk, serta informasi batch atau urutan penerimaan. Ketika terjadi transaksi penjualan, sistem akan membaca komposisi bahan baku pada menu yang terjual, kemudian secara otomatis mengurangi stok dari batch yang paling lama masuk terlebih dahulu. Jika jumlah pada batch pertama tidak mencukupi, maka sistem akan melanjutkan pengurangan ke batch berikutnya sesuai urutan tanggal masuk. Mekanisme ini membuat proses pengeluaran barang berjalan konsisten tanpa harus dipilih secara manual oleh pengguna.

Sebagai contoh, apabila stok susu terdiri dari 5 liter yang masuk pada tanggal 1 dan 7 liter yang masuk pada tanggal 3, lalu transaksi penjualan membutuhkan 6 liter, maka sistem akan mengurangi terlebih dahulu 5 liter dari stok tanggal 1 dan sisa 1 liter dari stok tanggal 3. Dengan cara tersebut, bahan yang lebih lama tersimpan akan digunakan terlebih dahulu. Penerapan FIFO pada sistem ini membantu menjaga kualitas bahan baku, meminimalkan risiko pemborosan akibat

barang rusak, serta memastikan data persediaan yang tampil pada sistem selalu sesuai dengan kondisi aktual di lapangan.

2.4 Basis Data dan Sistem Manajemen Basis Data

Basis data dan sistem manajemen basis data merupakan komponen yang sangat penting dalam pembangunan sistem informasi modern. Hampir seluruh aplikasi yang digunakan saat ini memerlukan media penyimpanan data yang terstruktur agar informasi dapat dikelola, diperbarui, dan ditampilkan kembali sesuai kebutuhan pengguna. Tanpa pengelolaan data yang baik, sistem akan sulit menjaga konsistensi informasi, menelusuri riwayat data, maupun menghasilkan laporan yang akurat. Oleh karena itu, keberadaan basis data menjadi fondasi utama dalam mendukung proses operasional maupun pengambilan keputusan pada suatu organisasi atau usaha [10, 11].

Secara umum, basis data berfungsi sebagai tempat penyimpanan terpusat yang menampung berbagai data penting dalam sistem, sedangkan sistem manajemen basis data atau *Database Management System* (DBMS) berfungsi sebagai perangkat lunak yang mengatur cara data tersebut disimpan, diakses, diubah, dan diamankan. Hubungan antara keduanya tidak dapat dipisahkan, karena basis data membutuhkan DBMS agar dapat digunakan secara efektif, sementara DBMS memerlukan basis data sebagai objek utama yang dikelola. Kombinasi keduanya memungkinkan data diolah secara lebih cepat, konsisten, dan aman dibandingkan pencatatan manual atau penyimpanan data yang terpisah-pisah [10, 11].

Dalam konteks sistem manajemen stok dan penjualan, basis data digunakan untuk menyimpan berbagai informasi seperti data pengguna, data menu, data stok bahan baku, data transaksi penjualan, data resep, serta data laporan. Seluruh data tersebut saling berkaitan dan harus dikelola dalam satu sistem yang terintegrasi agar perubahan pada satu bagian dapat langsung tercermin pada bagian lainnya. Sebagai contoh, ketika transaksi penjualan terjadi, data transaksi disimpan ke dalam basis data dan pada saat yang sama jumlah stok bahan baku diperbarui sesuai komposisi menu yang terjual. Proses seperti ini hanya dapat berjalan dengan baik apabila sistem didukung oleh rancangan basis data yang terstruktur dan DBMS yang mampu mengelola relasi data secara konsisten [10, 11].

Penerapan basis data yang baik juga memberikan manfaat dalam aspek pengawasan dan pengembangan sistem. Data historis yang tersimpan dapat digunakan untuk audit, evaluasi kinerja, analisis tren penjualan, maupun perencanaan pengadaan stok di masa mendatang. Selain itu,

struktur basis data yang terorganisasi memudahkan proses pengembangan fitur baru, integrasi dengan modul lain, serta pemeliharaan sistem dalam jangka panjang. Dengan demikian, basis data dan DBMS tidak hanya berfungsi sebagai media penyimpanan data, tetapi juga sebagai infrastruktur penting yang menentukan kualitas dan keberlanjutan sistem informasi yang dibangun.

2.4.1 Basis Data

Basis data (*database*) merupakan kumpulan data yang disusun secara terstruktur dan saling berhubungan sehingga dapat disimpan, dikelola, dicari kembali, serta digunakan untuk menghasilkan informasi yang dibutuhkan pengguna. Data di dalam basis data tidak disimpan secara acak, tetapi diorganisasi berdasarkan aturan tertentu agar mudah diakses dan diproses oleh sistem. Dengan struktur yang baik, basis data mampu membantu organisasi menjaga ketertiban data, mengurangi duplikasi, serta meningkatkan ketepatan informasi yang dihasilkan [10, 11].

Dalam sistem informasi, basis data berfungsi sebagai pusat penyimpanan seluruh data yang dibutuhkan oleh aplikasi. Berbagai data dari aktivitas operasional dikumpulkan ke dalam satu tempat sehingga dapat digunakan bersama oleh banyak proses maupun pengguna sesuai hak aksesnya. Pendekatan ini lebih baik dibandingkan penyimpanan data secara terpisah, karena perubahan data dapat diperbarui pada satu sumber yang sama dan langsung digunakan oleh seluruh bagian sistem. Dengan demikian, konsistensi dan integritas data dapat lebih mudah dijaga [10, 11].

Secara umum, basis data memiliki beberapa fungsi utama, yaitu [10, 11]:

1. Menyimpan Data Secara Terpusat: Seluruh data disimpan dalam satu media yang terorganisasi.
2. Mempermudah Pencarian Data: Data dapat dicari kembali dengan cepat sesuai kebutuhan.
3. Mengurangi Redundansi Data: Duplikasi data dapat diminimalkan karena tersimpan dalam satu sistem.
4. Menjaga Konsistensi Data: Perubahan data dapat diterapkan secara seragam pada seluruh sistem.
5. Mendukung Keamanan Data: Data dapat diatur hak aksesnya sesuai pengguna yang berwenang.
6. Menyediakan Dasar Informasi: Data yang tersimpan dapat diolah menjadi laporan dan informasi manajerial.

Dalam penelitian ini, basis data digunakan sebagai tempat penyimpanan seluruh data pada sistem manajemen stok dan penjualan Codel Coffee, seperti data pengguna, menu, stok bahan baku, resep, transaksi penjualan, dan laporan. Dengan basis data yang terintegrasi, setiap transaksi yang terjadi dapat langsung memengaruhi data terkait lainnya, misalnya pengurangan stok bahan baku setelah penjualan dilakukan. Hal ini menjadikan proses operasional lebih tertib, akurat, dan mudah dipantau.

2.4.2 Sistem Manajemen Basis Data (DBMS)

Sistem Manajemen Basis Data atau *Database Management System* (DBMS) merupakan perangkat lunak yang digunakan untuk membuat, mengelola, mengakses, dan mengamankan basis data. DBMS menjadi penghubung antara pengguna atau aplikasi dengan data yang tersimpan di dalam basis data. Melalui DBMS, pengguna tidak perlu berinteraksi langsung dengan struktur penyimpanan fisik data, karena seluruh proses pengolahan data dilakukan melalui mekanisme yang telah disediakan oleh sistem [10, 11].

Keberadaan DBMS sangat penting karena pengelolaan data dalam sistem modern membutuhkan proses yang cepat, akurat, dan aman. DBMS memungkinkan data ditambah, diubah, dihapus, maupun ditampilkan kembali secara efisien. Selain itu, DBMS juga berfungsi menjaga integritas data ketika banyak pengguna mengakses sistem secara bersamaan, serta membantu melindungi data dari kerusakan maupun akses yang tidak sah. Dengan kata lain, DBMS berperan besar dalam memastikan data tetap tersedia dan dapat dipercaya untuk mendukung operasional organisasi [10, 11].

Secara umum, fungsi utama DBMS dapat dijelaskan sebagai berikut [10, 11]:

1. *Data Definition*: Membuat dan mengatur struktur tabel, atribut, serta relasi data.
2. *Data Manipulation*: Menambah, mengubah, menghapus, dan menampilkan data sesuai kebutuhan.
3. *Security Management*: Mengatur hak akses agar data hanya digunakan pihak berwenang.
4. *Integrity Control*: Menjaga agar data tetap valid dan konsisten.
5. *Concurrency Control*: Mengelola akses banyak pengguna secara bersamaan tanpa konflik data.
6. *Backup and Recovery*: Membantu pencadangan dan pemulihan data ketika terjadi gangguan sistem.

Dalam penelitian ini, DBMS digunakan untuk mengelola seluruh data pada sistem manajemen stok dan penjualan Codel Coffee. Setiap data yang dimasukkan melalui aplikasi akan diproses dan disimpan melalui DBMS, kemudian ditampilkan kembali dalam bentuk informasi sesuai kebutuhan pengguna. Sebagai contoh, ketika transaksi penjualan dicatat, DBMS membantu menyimpan data transaksi sekaligus memperbarui stok bahan baku yang terkait. Dengan dukungan DBMS, sistem dapat berjalan secara lebih *stabil*, terintegrasi, dan efisien.

2.4.3 Basis Data dalam Sistem Manajemen Stok dan Penjualan

Dalam sistem manajemen stok dan penjualan, basis data berfungsi sebagai pusat penyimpanan dan pengelolaan seluruh data operasional yang dibutuhkan oleh sistem. Seluruh aktivitas seperti pencatatan data barang, pengelolaan menu, transaksi penjualan, perubahan stok, hingga penyusunan laporan bergantung pada data yang tersimpan di dalam basis data. Tanpa basis data yang terstruktur, setiap proses akan berjalan secara terpisah sehingga berpotensi menimbulkan duplikasi data, ketidaksesuaian informasi, serta keterlambatan dalam penyajian laporan. Oleh karena itu, penggunaan basis data menjadi elemen penting agar sistem dapat bekerja secara terintegrasi, cepat, dan konsisten [10, 11].

Pada sistem ini, basis data menyimpan berbagai entitas utama yang saling berhubungan. Data tersebut meliputi data pengguna (*user*), data menu, data stok bahan baku, data satuan, data resep atau komposisi menu, data transaksi penjualan, data detail transaksi, data riwayat stok, serta data laporan. Masing-masing entitas memiliki fungsi tersendiri, namun seluruhnya dihubungkan melalui relasi data agar sistem dapat memproses informasi secara otomatis. Hubungan antar tabel inilah yang memungkinkan data pada satu bagian memengaruhi bagian lain sesuai proses bisnis yang berjalan [10, 11].

Sebagai contoh, ketika transaksi penjualan dilakukan, sistem akan menyimpan data transaksi utama dan detail produk yang terjual ke dalam basis data. Setelah itu, sistem membaca data resep untuk mengetahui jumlah bahan baku yang digunakan pada setiap menu, kemudian mengurangi stok bahan baku secara otomatis sesuai jumlah penjualan. Seluruh perubahan tersebut juga dapat dicatat ke dalam tabel riwayat stok sehingga setiap pergerakan persediaan dapat ditelusuri kembali. Mekanisme ini menunjukkan bahwa basis data tidak hanya berfungsi sebagai tempat penyimpanan, tetapi juga sebagai penghubung utama antar proses di dalam sistem [10].

Selain mendukung proses operasional, basis data juga berperan penting dalam penyajian informasi manajerial. Data transaksi yang tersimpan dapat diolah menjadi laporan penjualan harian, mingguan, maupun bulanan. Data stok yang tersedia dapat digunakan untuk menampilkan kondisi persediaan terkini, bahan baku yang hampir habis, maupun riwayat penggunaan barang. Informasi tersebut membantu pemilik usaha dalam melakukan pengawasan, evaluasi kinerja, serta menentukan keputusan pengadaan stok dan strategi penjualan secara lebih tepat [10, 11].

Dalam konteks penelitian ini, perancangan basis data yang baik menjadi faktor penting karena kualitas sistem sangat bergantung pada ketepatan struktur data yang dibangun. Jika relasi antar tabel dirancang dengan benar, maka proses transaksi, pembaruan stok, pencarian data, dan pembuatan laporan dapat berjalan secara efisien. Sebaliknya, rancangan basis data yang kurang tepat dapat menimbulkan kesalahan data dan menghambat operasional sistem. Dengan demikian, basis data dalam sistem manajemen stok dan penjualan bukan hanya komponen pendukung, tetapi merupakan fondasi utama yang menentukan keberhasilan implementasi sistem di Codel Coffee.

2.4.4 MySQL

MySQL merupakan salah satu *Relational Database Management System* (RDBMS) yang banyak digunakan dalam pengembangan aplikasi berbasis web. MySQL dirancang untuk mengelola data dalam bentuk tabel yang saling berhubungan melalui relasi tertentu, sehingga sangat sesuai digunakan pada sistem informasi yang membutuhkan penyimpanan data terstruktur dan konsisten. Sebagai DBMS relasional, MySQL menggunakan *Structured Query Language* (SQL) sebagai bahasa utama untuk membuat struktur basis data, mengelola isi data, serta menampilkan kembali informasi sesuai kebutuhan pengguna [10, 12].

Popularitas MySQL didukung oleh berbagai keunggulan yang menjadikannya banyak dipilih oleh pengembang sistem informasi. MySQL dikenal memiliki performa yang baik dalam memproses query, mendukung banyak pengguna secara bersamaan, serta mampu menangani kebutuhan aplikasi skala kecil hingga menengah dengan stabil. Selain itu, MySQL juga menyediakan dukungan terhadap *primary key*, *foreign key*, indeks, transaksi, serta mekanisme keamanan data yang penting dalam menjaga integritas informasi yang tersimpan di dalam sistem [12].

Secara umum, beberapa kemampuan utama MySQL dapat dijelaskan sebagai berikut [10, 12]:

1. Pengelolaan Tabel Relasional: Data disimpan dalam tabel yang dapat dihubungkan melalui relasi antar entitas.
2. Query Data dengan SQL: Pengguna dapat menambah, mengubah, menghapus, dan menampilkan data menggunakan perintah SQL.
3. Integritas Data: Dukungan *primary key* dan *foreign key* membantu menjaga konsistensi hubungan antar tabel.
4. Manajemen Transaksi: Mendukung proses transaksi agar perubahan data berjalan aman dan konsisten.
5. Keamanan Data: Menyediakan pengaturan akun pengguna dan hak akses sesuai kebutuhan sistem.
6. Backup dan Pemulihan: Mendukung proses pencadangan data untuk mengurangi risiko kehilangan informasi.

Dalam penelitian ini, MySQL digunakan sebagai basis data utama pada sistem manajemen stok dan penjualan Codel Coffee. Seluruh data seperti pengguna, menu, bahan baku, resep, transaksi penjualan, dan laporan disimpan di dalam MySQL. Pemilihan MySQL didasarkan pada kemampuannya dalam mengelola relasi data secara baik, kemudahan integrasi dengan Laravel sebagai *backend framework*, serta kestabilannya dalam mendukung aplikasi berbasis web. Dengan menggunakan MySQL, sistem yang dibangun diharapkan mampu menjalankan proses penyimpanan data, pembaruan stok, dan penyajian laporan secara cepat, aman, dan efisien [10, 12].

2.5 Teknologi Web

Teknologi *web* merupakan sekumpulan teknologi yang digunakan untuk membangun, menjalankan, dan mengakses aplikasi melalui jaringan internet maupun intranet dengan menggunakan peramban (*browser*). Perkembangan teknologi *web* telah mendorong banyak organisasi dan pelaku usaha untuk beralih dari sistem konvensional menuju sistem digital yang lebih fleksibel, karena aplikasi dapat diakses dari berbagai perangkat tanpa harus melakukan instalasi khusus pada sisi pengguna. Dengan pendekatan ini, distribusi informasi menjadi lebih cepat, jangkauan penggunaan lebih luas, serta proses pemeliharaan sistem dapat dilakukan secara lebih terpusat [13, 14].

Pada dasarnya, teknologi *web* bekerja dengan model *client-server*, yaitu pengguna mengakses aplikasi melalui *browser* sebagai pihak *client*, sedangkan proses pengolahan data dan logika aplikasi dijalankan pada *server*. Ketika pengguna melakukan permintaan, seperti membuka halaman, mengirim data, atau menampilkan laporan, *server* akan memproses permintaan tersebut lalu mengirimkan respons kembali ke *client*. Mekanisme ini memungkinkan banyak pengguna menggunakan sistem yang sama secara bersamaan dengan data yang tetap terpusat dan konsisten [13].

Teknologi *web* memiliki beberapa keunggulan yang menjadikannya banyak digunakan dalam pengembangan sistem informasi, yaitu [13, 14]:

1. Akses Fleksibel: Sistem dapat digunakan melalui komputer, laptop, *tablet*, maupun ponsel selama terhubung ke jaringan.
2. Tanpa Instalasi di Sisi Pengguna: Pengguna cukup menggunakan *browser* tanpa memasang aplikasi tambahan.
3. Pemeliharaan Terpusat: Pembaruan sistem dilakukan pada *server* sehingga langsung berlaku bagi seluruh pengguna.
4. Data Terintegrasi: Seluruh data tersimpan dalam satu sistem yang sama.
5. Kolaborasi Multiuser: Banyak pengguna dapat mengakses sistem secara bersamaan sesuai hak akses masing-masing.
6. Efisiensi Distribusi Informasi: Informasi dapat diperbarui dan disampaikan dengan lebih cepat.

Dalam konteks penelitian ini, teknologi *web* dipilih sebagai dasar pengembangan sistem manajemen stok dan penjualan Codel Coffee karena sesuai dengan kebutuhan usaha yang memerlukan akses data secara cepat dan terpusat. Melalui sistem berbasis *web*, staf dapat melakukan pencatatan transaksi maupun pengelolaan stok secara langsung, sedangkan pemilik usaha dapat memantau laporan penjualan dan kondisi persediaan dari lokasi yang berbeda. Selain itu, sistem berbasis *web* memudahkan proses pengembangan dan pemeliharaan di masa mendatang karena seluruh komponen aplikasi dikelola pada satu lingkungan *server* yang terintegrasi [6, 13, 14].

2.5.1 Arsitektur *Client-Server*

Arsitektur *client-server* merupakan model arsitektur sistem yang membagi proses kerja aplikasi ke dalam dua bagian utama, yaitu *client* dan *server*. Pada model ini, *client* berfungsi

sebagai pihak yang mengirimkan permintaan layanan, sedangkan *server* berfungsi sebagai pihak yang menerima, memproses, dan memberikan respons terhadap permintaan tersebut. Pembagian tugas ini menjadikan pengelolaan sistem lebih terstruktur karena antarmuka pengguna dan proses pengolahan data dapat dipisahkan sesuai perannya masing-masing [13, 14].

Dalam implementasi aplikasi berbasis *web*, *client* umumnya berupa perangkat pengguna yang menjalankan *browser*, seperti komputer, laptop, *tablet*, atau telepon pintar. Melalui *browser*, pengguna dapat mengakses halaman sistem, mengisi formulir, menekan tombol aksi, maupun melihat laporan yang disediakan aplikasi. Sementara itu, *server* merupakan komputer atau layanan yang menjalankan aplikasi utama, memproses logika bisnis, berkomunikasi dengan basis data, serta mengirimkan hasil proses kembali kepada pengguna [13].

Alur kerja arsitektur *client-server* secara umum dapat dijelaskan sebagai berikut [13, 14]:

1. Permintaan (*Request*): Pengguna melakukan aksi melalui *client*, misalnya membuka halaman atau menyimpan data.
2. Pengiriman ke Server: Permintaan dikirim melalui jaringan menggunakan protokol seperti HTTP atau HTTPS.
3. Pemrosesan: *Server* menjalankan logika aplikasi, memvalidasi data, dan mengakses basis data bila diperlukan.
4. Respons (*Response*): Hasil proses dikirim kembali ke *client* dalam bentuk halaman, data, atau pesan status.
5. Tampilan ke Pengguna: *Client* menampilkan hasil tersebut melalui antarmuka aplikasi.

Model *client-server* memiliki beberapa keunggulan, seperti pengelolaan data yang terpusat, kemudahan pemeliharaan sistem, peningkatan keamanan karena data utama berada di sisi *server*, serta kemampuan melayani banyak pengguna secara bersamaan. Selain itu, pemisahan antara sisi pengguna dan sisi server memudahkan pengembangan sistem karena perubahan pada tampilan tidak selalu memerlukan perubahan pada logika inti aplikasi [13, 14].

Dalam penelitian ini, arsitektur *client-server* diterapkan pada sistem manajemen stok dan penjualan Codel Coffee. Antarmuka pengguna dibangun agar dapat diakses melalui *browser* oleh staf maupun pemilik usaha sebagai *client*. Di sisi lain, aplikasi utama berjalan pada *server* yang menangani proses autentikasi, pengelolaan transaksi, pembaruan stok, penyimpanan data, dan penyajian laporan. Dengan arsitektur ini, sistem dapat digunakan secara terpusat, efisien, dan mendukung kebutuhan operasional usaha secara *real-time*.

2.5.2 Jenis – Jenis Website

Website merupakan kumpulan halaman digital yang saling terhubung dan dapat diakses melalui jaringan internet menggunakan alamat tertentu. Dalam perkembangannya, *website* tidak hanya digunakan sebagai media penyampaian informasi, tetapi juga sebagai sarana interaksi, transaksi, dan pengelolaan data secara daring. Oleh karena itu, *website* dapat dibedakan ke dalam beberapa jenis berdasarkan cara kerja, tingkat interaksi, maupun tujuan penggunaannya [13, 14].

Berdasarkan cara penyajian kontennya, *website* umumnya dibedakan menjadi dua jenis utama, yaitu *website statis* dan *website dinamis* [13].

1. **Website Statis:** Merupakan website yang isi halamannya bersifat tetap dan jarang berubah. Konten pada website statis biasanya dibuat langsung di dalam file halaman dan tidak bergantung pada basis data. Perubahan informasi harus dilakukan dengan mengubah kode atau file halaman secara manual. Website jenis ini umumnya digunakan untuk profil sederhana, portofolio, atau halaman informasi yang tidak sering diperbarui.
2. **Website Dinamis:** Merupakan website yang kontennya dapat berubah secara otomatis sesuai data yang tersimpan di dalam sistem. Website dinamis biasanya terhubung dengan basis data dan memiliki fitur interaksi, seperti *login*, formulir input, pencarian data, transaksi, serta pembuatan laporan. Jenis website ini banyak digunakan pada sistem informasi, toko daring, portal berita, dan aplikasi bisnis.

Selain berdasarkan sifat kontennya, *website* juga dapat dibedakan berdasarkan tujuan penggunaannya, yaitu sebagai berikut [13, 14]:

1. **Website Informatif:** Digunakan untuk menyampaikan informasi kepada pengunjung, seperti profil perusahaan, berita, atau artikel.
2. **Website Transaksional:** Digunakan untuk mendukung kegiatan transaksi, seperti pemesanan produk, pembayaran, atau layanan digital.
3. **Website Sistem Informasi:** Digunakan untuk mengelola data, proses operasional, dan penyajian informasi dalam suatu organisasi atau usaha.
4. **Website Komunikasi dan Layanan:** Digunakan untuk interaksi antara pengguna dan penyedia layanan, seperti forum, media sosial, atau layanan pelanggan.

Dalam penelitian ini, sistem yang dikembangkan termasuk ke dalam kategori *website dinamis* sekaligus *website sistem* informasi. Hal ini karena sistem tidak hanya menampilkan

informasi, tetapi juga mengelola data pengguna, menu, stok bahan baku, transaksi penjualan, serta menghasilkan laporan secara otomatis berdasarkan aktivitas yang terjadi. Selain itu, sistem juga mendukung interaksi pengguna melalui proses *login*, input data, transaksi, dan pemantauan laporan. Dengan karakteristik tersebut, penggunaan teknologi *website* dinamis menjadi pilihan yang tepat untuk mendukung kebutuhan operasional Codel Coffee secara terintegrasi dan *real-time* [6, 13, 14].

2.6 Framework Pengembangan Web

Framework pengembangan *web* merupakan kerangka kerja perangkat lunak yang menyediakan struktur dasar, aturan pengembangan, serta komponen siap pakai untuk membantu proses pembuatan aplikasi *web*. Dengan menggunakan *framework*, pengembang tidak perlu membangun seluruh fungsi aplikasi dari awal, karena berbagai kebutuhan umum seperti pengaturan *routing*, koneksi basis data, autentikasi pengguna, validasi data, hingga pengelolaan tampilan telah disediakan dalam bentuk modul atau pustaka yang dapat digunakan kembali. Pendekatan ini membuat proses pengembangan menjadi lebih cepat, terstruktur, dan efisien. Dalam pengembangan sistem informasi modern, penggunaan *framework* menjadi sangat penting karena aplikasi yang dibangun umumnya memiliki kompleksitas yang lebih tinggi dibandingkan aplikasi sederhana. Sistem tidak hanya membutuhkan tampilan antarmuka, tetapi juga logika bisnis, pengelolaan data, keamanan, serta kemampuan pemeliharaan jangka panjang. Tanpa *framework*, pengembangan sistem berisiko menghasilkan struktur kode yang tidak konsisten, sulit dikembangkan, dan lebih rentan terhadap kesalahan. Oleh sebab itu, *framework* digunakan untuk membantu menjaga standar pengembangan dan kualitas aplikasi secara keseluruhan [15].

Secara umum, penggunaan *framework* memberikan beberapa manfaat utama, yaitu [15, 16]:

1. Mempercepat Pengembangan: Banyak fungsi dasar telah tersedia sehingga waktu pembuatan sistem lebih singkat.
2. Struktur Kode Lebih Rapi: Framework menyediakan pola kerja yang membantu kode tersusun secara terorganisasi.
3. Meningkatkan Keamanan: Banyak framework menyediakan fitur keamanan bawaan, seperti proteksi autentikasi dan validasi data.

4. Mudah Dipelihara: Struktur yang konsisten memudahkan proses perbaikan dan pengembangan fitur baru.
5. Mendukung Kolaborasi Tim: Standar penulisan dan struktur proyek memudahkan kerja bersama antar pengembang.
6. Integrasi Lebih Mudah: Framework umumnya mendukung integrasi dengan basis data, API, dan teknologi lain.

Dalam penelitian ini, pengembangan sistem manajemen stok dan penjualan Codel Coffee menggunakan Laravel sebagai *backend framework* dan React.js sebagai teknologi *frontend*. Kombinasi tersebut dipilih karena mampu memisahkan proses pengolahan data di sisi *server* dan penyajian antarmuka di sisi pengguna. Laravel digunakan untuk mengelola logika bisnis, autentikasi, API, serta koneksi basis data, sedangkan React.js digunakan untuk membangun tampilan antarmuka yang responsif dan interaktif. Dengan pendekatan ini, sistem diharapkan lebih mudah dikembangkan, dipelihara, dan mampu memberikan pengalaman penggunaan yang baik [15 – 17].

2.6.1 Laravel

Laravel merupakan *framework* aplikasi *web* berbasis PHP yang dirancang untuk membantu pengembang membangun sistem secara cepat, terstruktur, dan mudah dipelihara. Laravel banyak digunakan dalam pengembangan sistem informasi karena menyediakan sintaks yang jelas, dokumentasi yang lengkap, serta ekosistem pendukung yang luas. Dengan menggunakan Laravel, pengembang tidak perlu membangun seluruh komponen dasar aplikasi dari awal karena berbagai kebutuhan umum telah disediakan dalam bentuk fitur bawaan. Hal ini menjadikan proses pengembangan lebih efisien, konsisten, dan sesuai untuk pembangunan aplikasi modern berbasis *web* [15, 18, 19].

Laravel menerapkan berbagai konsep arsitektur dan komponen penting yang mendukung proses pengembangan aplikasi secara sistematis. Konsep-konsep tersebut membantu pengembang dalam mengatur struktur program, pengelolaan data, keamanan, hingga komunikasi antar sistem. Beberapa konsep utama dalam Laravel adalah sebagai berikut [15, 18, 19]:

1. *Model–View–Controller* (MVC)

Model–View–Controller (MVC) merupakan pola arsitektur perangkat lunak yang membagi aplikasi ke dalam tiga bagian utama, yaitu *Model*, *View*, dan *Controller*. Pemisahan ini

bertujuan agar setiap bagian memiliki tanggung jawab yang jelas sehingga struktur kode menjadi lebih rapi dan mudah dikelola. *Model* berfungsi mengelola data serta berinteraksi dengan basis data. *View* bertanggung jawab menampilkan antarmuka kepada pengguna. *Controller* berperan sebagai penghubung antara *Model* dan *View* dengan memproses permintaan pengguna serta mengatur alur logika aplikasi. Dengan penerapan MVC, pengembangan sistem dapat dilakukan secara lebih modular, memudahkan pemeliharaan, serta mempermudah penambahan fitur baru.

2. *Routing*

Routing merupakan mekanisme yang digunakan untuk menentukan bagaimana aplikasi merespons permintaan pengguna melalui protokol HTTP. Setiap URL yang diakses pengguna akan diarahkan ke fungsi atau *controller* tertentu sesuai aturan yang telah ditetapkan. Melalui *routing*, alur navigasi sistem menjadi lebih terstruktur dan mudah dikelola. Laravel juga mendukung berbagai metode HTTP seperti GET, POST, PUT, dan DELETE yang umum digunakan dalam pengembangan aplikasi *web* maupun REST API.

3. *Object Relational Mapping (ORM)*

Laravel menyediakan fitur ORM bernama *Eloquent ORM*. ORM merupakan teknik yang memetakan tabel dalam basis data menjadi objek pada bahasa pemrograman sehingga pengembang dapat berinteraksi dengan data tanpa harus selalu menulis query SQL secara langsung. Penggunaan ORM menyederhanakan proses manipulasi data seperti penyimpanan, pengambilan, pembaruan, dan penghapusan data. Dengan pendekatan ini, pengelolaan basis data menjadi lebih mudah dipahami, mempercepat pengembangan, serta meningkatkan keterbacaan kode program.

4. *Backend API*

Laravel mendukung pembuatan *Application Programming Interface (API)* yang berfungsi sebagai penghubung antara sisi *backend* dan *frontend*. Melalui API, data dari *server* dapat dikirim dan diterima dalam format terstruktur seperti JSON. Pendekatan ini sangat sesuai untuk pengembangan aplikasi modern yang memisahkan antarmuka pengguna dengan logika sistem, misalnya penggunaan React.js pada sisi *frontend* dan Laravel pada sisi *backend*. Dengan API, pertukaran data antar sistem dapat dilakukan secara cepat dan fleksibel.

5. Keamanan dan Validasi

Laravel menyediakan berbagai fitur keamanan bawaan seperti autentikasi pengguna, proteksi CSRF, enkripsi kata sandi, pembatasan akses, serta validasi data masukan. Fitur-fitur tersebut membantu menjaga keamanan aplikasi dari akses tidak sah dan mengurangi risiko kesalahan data yang disimpan ke dalam sistem.

6. Migrasi dan Manajemen Basis Data

Laravel menyediakan fitur *migration* yang memungkinkan struktur tabel basis data dibuat, diubah, dan diperbarui menggunakan kode program. Dengan fitur ini, perubahan struktur basis data dapat terdokumentasi dengan baik dan mudah dikontrol selama proses pengembangan. Fitur ini juga memudahkan kerja tim karena seluruh perubahan struktur *database* dapat disinkronkan secara konsisten antar pengembang.

Secara umum, penggunaan Laravel memberikan beberapa manfaat utama, yaitu mempercepat pembangunan aplikasi, meningkatkan keteraturan kode program, mempermudah integrasi basis data, meningkatkan keamanan, serta mendukung pengembangan sistem skala kecil maupun besar [15, 18, 19].

Dalam penelitian ini, Laravel digunakan sebagai *backend framework* pada sistem manajemen stok dan penjualan Codel Coffee. Laravel menangani autentikasi pengguna, pengelolaan data menu, stok bahan baku, transaksi penjualan, pencatatan riwayat perubahan stok, serta penyediaan API yang diakses oleh antarmuka React.js. Pemilihan Laravel didasarkan pada struktur pengembangan yang rapi, kemudahan integrasi dengan MySQL, serta kelengkapan fitur yang sesuai untuk kebutuhan aplikasi bisnis. Dengan menggunakan Laravel, sistem yang dibangun diharapkan mampu berjalan secara *stabil*, aman, dan mudah dikembangkan di masa mendatang [15, 18, 19].

2.6.2 ReactJs

React.js merupakan pustaka JavaScript yang digunakan untuk membangun antarmuka pengguna (*user interface*) pada aplikasi *web* secara interaktif, dinamis, dan responsif. React.js banyak digunakan dalam pengembangan aplikasi modern karena mampu menghasilkan tampilan yang cepat, efisien, serta mudah dikelola. Fokus utama React.js berada pada sisi *frontend*, yaitu bagian sistem yang berinteraksi langsung dengan pengguna. Oleh karena itu, React.js sangat sesuai digunakan pada aplikasi yang membutuhkan pengalaman penggunaan yang baik, navigasi yang cepat, dan tampilan yang responsif [17, 20].

Berbeda dengan *framework* yang menyediakan struktur aplikasi secara menyeluruh, React.js lebih dikenal sebagai *library* yang berfokus pada pengelolaan tampilan dan interaksi pengguna. Pendekatan ini memberikan fleksibilitas kepada pengembang untuk menggabungkan React.js dengan berbagai teknologi lain sesuai kebutuhan sistem, seperti Laravel pada sisi *backend*, layanan API, maupun pustaka tambahan lainnya. Dengan konsep tersebut, React.js sering digunakan dalam pengembangan sistem modern yang memisahkan proses pengolahan data di *server* dengan tampilan pada sisi pengguna [17].

React.js menerapkan berbagai konsep penting yang mendukung pembangunan antarmuka secara modular, efisien, dan mudah dipelihara. Beberapa konsep utama dalam React.js adalah sebagai berikut [17, 20]:

1. *Component-Based Architecture*

Antarmuka aplikasi dibangun dari komponen-komponen kecil yang berdiri sendiri dan dapat digunakan kembali. Setiap komponen dapat mewakili bagian tertentu dari tampilan, seperti tombol, formulir, tabel, kartu informasi, menu navigasi, maupun halaman tertentu. Pendekatan berbasis komponen membuat pengembangan aplikasi menjadi lebih terstruktur karena setiap bagian dapat dikembangkan, diuji, dan diperbaiki secara terpisah. Selain itu, penggunaan kembali komponen yang sama pada beberapa halaman dapat meningkatkan efisiensi pengembangan.

2. *Virtual DOM*

React.js menggunakan konsep *Virtual Document Object Model (Virtual DOM)*, yaitu representasi virtual dari struktur halaman *web*. Ketika terjadi perubahan data, React.js akan membandingkan kondisi terbaru dengan kondisi sebelumnya, lalu hanya memperbarui bagian antarmuka yang benar-benar berubah. Mekanisme ini membuat proses *rendering* menjadi lebih cepat dan efisien dibandingkan pembaruan seluruh halaman, sehingga performa aplikasi menjadi lebih baik.

3. *State Management*

State merupakan data dinamis yang dapat berubah selama aplikasi berjalan dan memengaruhi tampilan antarmuka. Contoh *state* antara lain data pengguna yang sedang *login*, isi formulir, hasil pencarian, status transaksi, atau data stok yang sedang diperbarui. Ketika nilai *state* berubah, React.js akan memperbarui tampilan yang berkaitan secara otomatis tanpa perlu memuat ulang halaman. Hal ini membuat aplikasi terasa lebih interaktif dan responsif.

4. *Props*

Props (properties) digunakan untuk mengirim data dari satu komponen ke komponen lainnya. Melalui *props*, setiap komponen dapat saling berkomunikasi dan menerima data yang dibutuhkan untuk ditampilkan. Konsep ini membantu pengelolaan data antar komponen menjadi lebih terstruktur serta mendukung pembagian fungsi antarmuka ke dalam bagian-bagian kecil yang saling terhubung.

5. *Single Page Application (SPA)*

React.js sangat cocok digunakan untuk membangun konsep *Single Page Application (SPA)*, yaitu aplikasi yang memuat satu halaman utama dan memperbarui isi halaman secara dinamis tanpa melakukan *reload* penuh ketika pengguna berpindah menu atau halaman. Pendekatan ini membuat navigasi aplikasi terasa lebih cepat, nyaman, dan efisien bagi pengguna.

6. *API Integration*

React.js dapat dihubungkan dengan sistem *backend* melalui REST API atau layanan data lainnya. Dengan cara ini, React.js dapat mengambil, menampilkan, menambah, mengubah, maupun menghapus data yang tersimpan di *server* sesuai kebutuhan aplikasi. Integrasi ini sangat penting pada sistem modern karena memungkinkan pemisahan yang jelas antara tampilan antarmuka dan logika bisnis.

Secara umum, penggunaan React.js memberikan beberapa manfaat utama, yaitu menghasilkan antarmuka yang lebih responsif, meningkatkan pengalaman pengguna, mempermudah pengembangan tampilan secara modular, mempercepat pemeliharaan kode, serta mendukung integrasi dengan berbagai teknologi *backend*. Dalam penelitian ini, React.js digunakan sebagai teknologi *frontend* pada sistem manajemen stok dan penjualan Codel Coffee. React.js digunakan untuk membangun halaman *login*, dashboard, pengelolaan menu, pengelolaan stok, transaksi penjualan, dan tampilan laporan. Melalui React.js, antarmuka sistem dapat dibuat lebih interaktif dan responsif sehingga memudahkan pengguna dalam menjalankan aktivitas operasional sehari-hari. React.js juga terintegrasi dengan Laravel melalui API sehingga data dapat ditampilkan dan diperbarui secara cepat sesuai proses yang berjalan pada sisi *server* [17, 20].

2.7 Metode Pengembangan Sistem

Metode pengembangan sistem merupakan pendekatan terstruktur yang digunakan untuk merancang, membangun, menguji, dan mengimplementasikan sistem informasi secara sistematis.

Metode ini berfungsi sebagai pedoman kerja agar proses pengembangan dapat dilakukan secara terarah, terdokumentasi, dan sesuai dengan kebutuhan pengguna. Dengan adanya metode pengembangan sistem, setiap aktivitas pengembangan memiliki tahapan yang jelas sehingga memudahkan proses pengendalian, evaluasi, serta pengukuran hasil yang dicapai [7, 9, 21].

Dalam pengembangan sistem informasi, metode pengembangan sistem tidak hanya berfokus pada pembuatan perangkat lunak, tetapi juga mencakup proses identifikasi masalah, analisis kebutuhan, perancangan solusi, implementasi teknologi, pengujian sistem, hingga pemeliharaan setelah sistem digunakan. Pendekatan ini penting agar sistem yang dibangun benar-benar mampu menyelesaikan permasalahan yang ada dan memberikan manfaat nyata bagi organisasi atau usaha yang menggunakannya [7, 9].

Pemilihan metode pengembangan sistem harus disesuaikan dengan karakteristik proyek, tingkat kompleksitas sistem, ketersediaan waktu, sumber daya, serta kejelasan kebutuhan pengguna. Setiap metode memiliki kelebihan dan keterbatasan masing-masing. Oleh karena itu, pengembang perlu memilih metode yang paling sesuai agar proses pembangunan sistem dapat berjalan efektif dan menghasilkan sistem yang berkualitas [21].

Dalam penelitian ini, metode pengembangan sistem digunakan sebagai landasan dalam membangun sistem manajemen stok dan penjualan pada Codel Coffee. Dengan metode yang tepat, proses pengembangan dapat dilakukan secara lebih terstruktur mulai dari analisis kebutuhan pengguna, perancangan sistem, implementasi fitur, pengujian fungsi, hingga sistem siap digunakan dalam kegiatan operasional.

2.7.1 Metode Waterfall

Metode Waterfall merupakan salah satu metode pengembangan sistem yang menggunakan pendekatan berurutan dan sistematis. Pada metode ini, proses pengembangan dibagi ke dalam beberapa tahapan yang dilakukan secara bertahap, di mana setiap tahap harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahap berikutnya. Model ini menekankan pentingnya perencanaan yang matang, dokumentasi yang jelas, serta pengendalian proses pengembangan pada setiap tahap [7, 21].

Metode Waterfall banyak digunakan pada proyek yang memiliki kebutuhan sistem yang relatif jelas sejak awal dan tidak sering mengalami perubahan selama proses pengembangan. Dengan kebutuhan yang telah didefinisikan secara rinci, metode ini dapat membantu

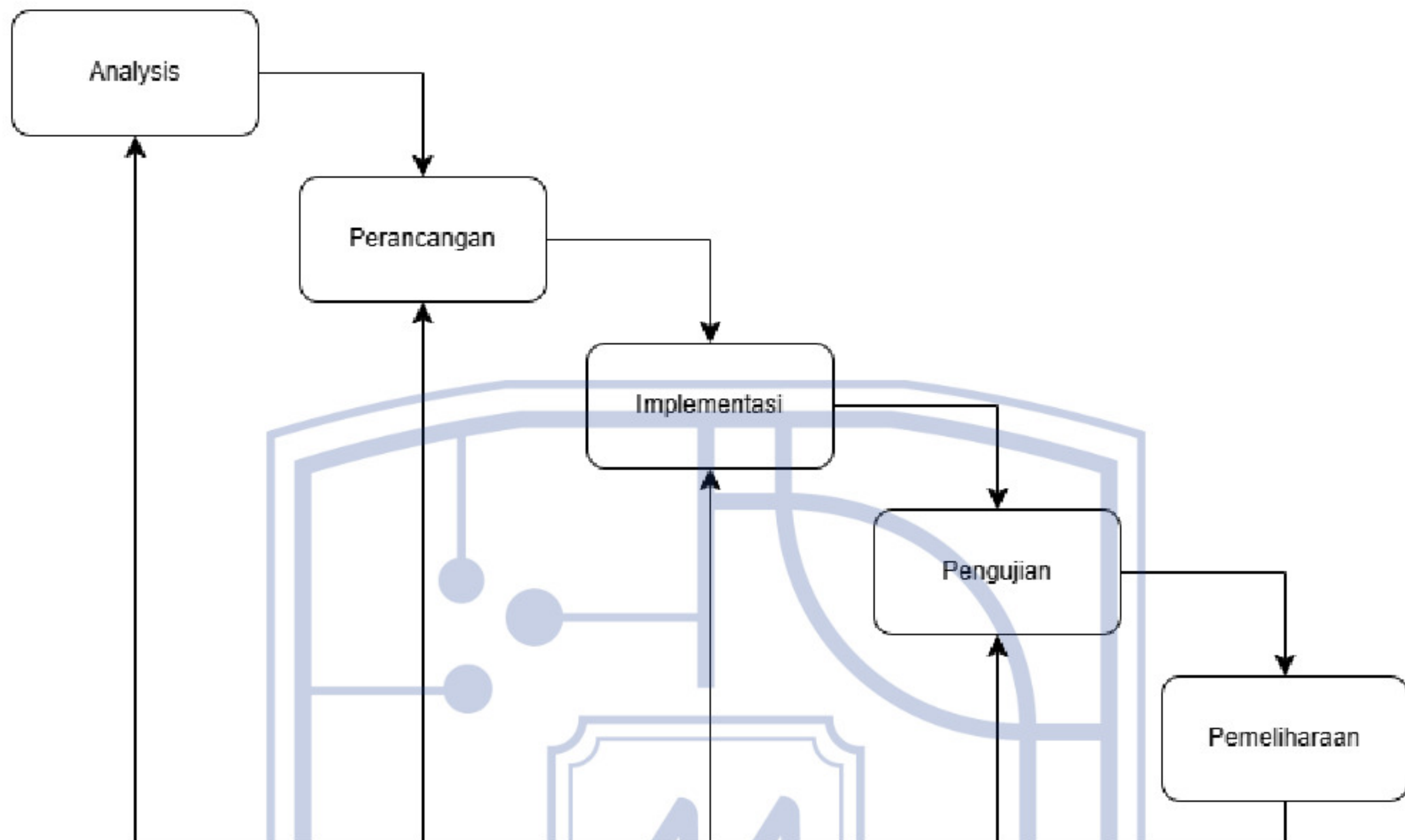
meminimalkan kesalahan, mempermudah pengawasan proyek, serta memudahkan evaluasi hasil pada setiap tahapan yang telah dilalui [9, 21].

Karakteristik utama metode Waterfall adalah alur kerja yang linear, yaitu hasil dari satu tahap menjadi dasar bagi tahap berikutnya. Oleh karena itu, kualitas keluaran dari setiap tahap sangat memengaruhi keberhasilan tahap selanjutnya. Pendekatan ini cocok digunakan pada proyek yang membutuhkan dokumentasi formal, struktur kerja yang jelas, dan proses pengembangan yang terkontrol [7, 9].

Dalam konteks penelitian ini, metode Waterfall dipilih karena pengembangan sistem manajemen stok dan penjualan Codel Coffee memiliki kebutuhan yang dapat diidentifikasi sejak awal, seperti pengelolaan data menu, pencatatan stok, transaksi penjualan, laporan, dan hak akses pengguna. Dengan pendekatan yang berurutan, proses pengembangan sistem dapat dilakukan secara lebih terstruktur mulai dari analisis kebutuhan hingga pengujian sistem, sehingga hasil implementasi lebih sesuai dengan kebutuhan pengguna. Dengan demikian, metode Waterfall merupakan metode pengembangan sistem yang menekankan tahapan kerja yang jelas, terstruktur, dan terdokumentasi untuk menghasilkan sistem informasi yang sesuai dengan kebutuhan yang telah ditentukan.

2.7.2 Tahapan Metode Waterfall

Metode Waterfall terdiri dari beberapa tahapan pengembangan sistem yang dilakukan secara berurutan. Setiap tahap memiliki tujuan, aktivitas, serta keluaran (*output*) yang menjadi dasar bagi tahap berikutnya. Karena prosesnya bersifat linear, kualitas hasil pada satu tahap akan sangat memengaruhi keberhasilan tahap selanjutnya. Oleh sebab itu, setiap fase dalam metode Waterfall perlu dilakukan secara teliti, terdokumentasi, dan sesuai kebutuhan sistem yang akan dibangun [7, 9, 21].



Gambar 2. 1 Tahapan Metode Waterfall

1. Analisis Kebutuhan Sistem

Tahap analisis kebutuhan merupakan tahap awal yang berfokus pada proses identifikasi masalah, kebutuhan pengguna, serta ruang lingkup sistem yang akan dikembangkan. Pada tahap ini, pengembang mengumpulkan informasi melalui observasi, wawancara, studi dokumentasi, maupun analisis proses bisnis yang sedang berjalan. Tujuannya adalah memahami permasalahan yang terjadi serta menentukan solusi yang benar-benar dibutuhkan pengguna. Kebutuhan sistem umumnya dibagi menjadi dua jenis, yaitu kebutuhan fungsional dan kebutuhan nonfungsional. Kebutuhan fungsional berkaitan dengan layanan atau fitur yang harus disediakan sistem, sedangkan kebutuhan nonfungsional berkaitan dengan kualitas sistem, seperti keamanan, kecepatan akses, kemudahan penggunaan, dan keandalan sistem. Hasil utama dari tahap ini adalah dokumen spesifikasi kebutuhan yang menjadi acuan pada seluruh tahapan pengembangan berikutnya [7, 9, 21].

Dalam penelitian ini, tahap analisis kebutuhan dilakukan untuk memahami kebutuhan pengelolaan stok bahan baku, pencatatan transaksi penjualan, pengelolaan data menu, pembuatan laporan, serta pengaturan hak akses pengguna pada sistem Codel Coffee.

2. Perancangan Sistem

Setelah kebutuhan sistem didefinisikan dengan jelas, tahap berikutnya adalah perancangan sistem. Tahap ini bertujuan menerjemahkan kebutuhan pengguna ke dalam bentuk desain teknis yang dapat diimplementasikan. Perancangan dilakukan agar pengembang memiliki gambaran yang jelas mengenai bagaimana sistem akan dibangun dan bagaimana setiap komponen akan saling berinteraksi. Aktivitas pada tahap ini meliputi perancangan arsitektur sistem, alur proses, struktur basis data, rancangan antarmuka pengguna, serta pembagian modul program. Dalam pengembangan sistem informasi, tahap perancangan sering menggunakan alat bantu visual seperti *use case diagram*, *activity diagram*, *class diagram*, dan rancangan basis data agar struktur sistem lebih mudah dipahami. Hasil dari tahap ini menjadi pedoman utama dalam proses implementasi [7, 9, 21].

3. Implementasi Sistem

Tahap implementasi merupakan proses penerapan hasil perancangan ke dalam bentuk program atau aplikasi yang dapat dijalankan. Pada tahap ini, seluruh desain yang telah dibuat mulai diterjemahkan ke dalam kode program menggunakan bahasa pemrograman dan teknologi yang telah dipilih. Selain itu, dilakukan pula pembuatan basis data, konfigurasi sistem, serta integrasi antar modul agar seluruh fungsi dapat berjalan sebagai satu kesatuan sistem. Pada metode Waterfall, implementasi dilakukan setelah tahap analisis dan perancangan selesai secara lengkap. Hal ini bertujuan agar proses pembangunan sistem tetap sesuai rancangan awal dan meminimalkan perubahan besar selama proses pengkodean berlangsung [7, 9, 21].

Dalam penelitian ini, implementasi sistem dilakukan menggunakan Laravel sebagai *backend framework*, React.js sebagai *frontend*, dan MySQL sebagai basis data untuk mendukung pengelolaan data stok serta transaksi penjualan.

4. Pengujian Sistem

Setelah sistem selesai diimplementasikan, tahap berikutnya adalah pengujian sistem. Tahap ini bertujuan memastikan bahwa seluruh fungsi yang dibangun telah berjalan sesuai kebutuhan dan spesifikasi yang telah ditetapkan sebelumnya. Pengujian juga dilakukan untuk menemukan kesalahan, kekurangan, atau ketidaksesuaian sistem sebelum digunakan oleh pengguna sebenarnya. Pengujian dapat mencakup pemeriksaan fungsi menu, validasi data masukan, alur proses transaksi, ketepatan keluaran laporan, hingga kestabilan sistem saat digunakan. Dengan adanya tahap pengujian, risiko kesalahan pada saat sistem diterapkan dapat diminimalkan sehingga kualitas sistem menjadi lebih baik [7, 9, 21].

Dalam penelitian ini, pengujian dilakukan terhadap fitur *login*, pengelolaan menu, transaksi penjualan, pencatatan stok, dan laporan untuk memastikan seluruh fungsi berjalan sesuai kebutuhan pengguna.

5. Pemeliharaan Sistem

Pemeliharaan merupakan tahap akhir dalam metode Waterfall yang dilakukan setelah sistem digunakan secara operasional. Tahap ini bertujuan menjaga agar sistem tetap berjalan dengan baik, memperbaiki kesalahan yang ditemukan setelah implementasi, serta menyesuaikan sistem apabila terdapat kebutuhan baru di masa mendatang. Aktivitas pemeliharaan dapat berupa perbaikan *bug*, peningkatan performa sistem, penyesuaian tampilan, pembaruan keamanan, maupun penambahan fitur baru sesuai perkembangan kebutuhan pengguna. Tahap ini sangat penting karena sistem informasi harus mampu beradaptasi dengan perubahan proses bisnis dan perkembangan teknologi agar tetap relevan digunakan dalam jangka panjang [7, 9, 21].

Dengan demikian, tahapan metode Waterfall memberikan alur pengembangan sistem yang jelas, terstruktur, dan terdokumentasi mulai dari analisis kebutuhan hingga pemeliharaan. Melalui tahapan tersebut, pembangunan sistem manajemen stok dan penjualan Codel Coffee dapat dilakukan secara sistematis, terkontrol, dan lebih terarah sesuai kebutuhan pengguna.

2.8 Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan bahasa pemodelan standar yang digunakan untuk memvisualisasikan, merancang, menganalisis, dan mendokumentasikan sistem perangkat lunak maupun sistem informasi. UML dikembangkan sebagai standar pemodelan agar pengembang, analis sistem, dan pihak terkait lainnya memiliki cara yang sama dalam menggambarkan struktur serta perilaku sistem secara jelas dan terorganisir. Dengan adanya UML, proses komunikasi antar pihak yang terlibat dalam pengembangan sistem menjadi lebih mudah karena konsep sistem dapat dijelaskan dalam bentuk model visual yang sistematis [7, 9].

Dalam proses pengembangan sistem, UML berfungsi sebagai alat bantu untuk menerjemahkan kebutuhan pengguna ke dalam rancangan sistem sebelum sistem diimplementasikan ke dalam kode program. Melalui diagram-diagram UML, pengembang dapat menggambarkan siapa saja pengguna sistem, proses bisnis yang berjalan, interaksi antar komponen, struktur data, serta alur aktivitas yang terjadi di dalam sistem. Dengan demikian,

potensi kesalahan pada tahap implementasi dapat dikurangi karena rancangan sistem telah dipahami sejak awal [7].

UML tidak hanya digunakan pada tahap perancangan, tetapi juga bermanfaat sebagai dokumentasi sistem. Dokumentasi yang baik akan memudahkan proses pengembangan lanjutan, pemeliharaan sistem, perbaikan kesalahan, maupun pengembangan fitur baru di masa mendatang. Selain itu, penggunaan UML membantu menghasilkan sistem yang lebih terstruktur karena setiap komponen dan hubungan antarbagian telah dirancang secara logis sebelum proses pembangunan dilakukan [7, 9].

Secara umum, UML menyediakan berbagai jenis diagram yang masing-masing memiliki fungsi berbeda sesuai sudut pandang sistem yang ingin dijelaskan. Beberapa diagram UML yang umum digunakan antara lain *use case diagram*, *activity diagram*, *class diagram*, dan *sequence diagram*. Selain itu, dalam perancangan basis data sering digunakan *Entity Relationship Diagram* (ERD) sebagai alat bantu pemodelan data. Pemilihan diagram disesuaikan dengan kebutuhan analisis dan kompleksitas sistem yang dikembangkan [7, 9].

Dalam penelitian ini, UML digunakan sebagai alat bantu perancangan sistem manajemen stok dan penjualan pada Codel Coffee. Penggunaan UML bertujuan agar kebutuhan sistem dapat divisualisasikan secara jelas sebelum tahap implementasi dilakukan. Diagram yang digunakan dalam penelitian ini meliputi *use case diagram* untuk menggambarkan hak akses dan fungsi pengguna, *activity diagram* untuk menunjukkan alur proses sistem, serta ERD untuk merancang struktur basis data yang digunakan oleh aplikasi. Dengan demikian, UML merupakan alat pemodelan yang penting dalam pengembangan sistem informasi karena mampu membantu proses analisis, perancangan, komunikasi, dan dokumentasi sistem secara terstruktur. Penggunaan UML membuat pembangunan sistem menjadi lebih terarah, efisien, dan sesuai dengan kebutuhan pengguna.

2.8.1 Use Case Diagram

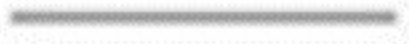
Use Case Diagram merupakan salah satu diagram dalam Unified Modeling Language (UML) yang digunakan untuk menggambarkan hubungan antara pengguna (*actor*) dengan sistem serta fungsi-fungsi yang disediakan oleh sistem. Diagram ini menampilkan layanan atau proses utama yang dapat dijalankan oleh pengguna tanpa menjelaskan detail teknis bagaimana proses

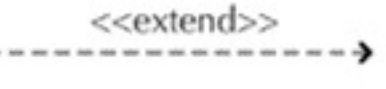
tersebut diimplementasikan. Dengan demikian, *use case diagram* berfokus pada kebutuhan pengguna dan interaksi eksternal terhadap sistem yang akan dibangun [7, 9].

Dalam tahap analisis dan perancangan sistem, *use case diagram* memiliki peran penting karena membantu pengembang memahami siapa saja pihak yang menggunakan sistem, apa saja kebutuhan masing-masing pengguna, serta fitur apa yang harus tersedia. Melalui diagram ini, ruang lingkup sistem dapat ditentukan dengan lebih jelas sehingga meminimalkan kesalahan dalam pengembangan dan mencegah fitur yang tidak sesuai dengan kebutuhan pengguna [7].

Secara umum, unsur-unsur yang terdapat pada *use case diagram* dapat dilihat pada Gambar 2.1. Unsur tersebut digunakan untuk menggambarkan hubungan antara pengguna dengan fungsi-fungsi yang tersedia di dalam sistem [7, 9].

Tabel 2. 1 Simbol Use Case Diagram [7]

Simbol	Nama	Keterangan
	<i>Actor</i>	Pihak yang berinteraksi dengan sistem. <i>Actor</i> dapat berupa manusia, organisasi, perangkat lain, maupun sistem eksternal yang menggunakan layanan sistem. Setiap <i>actor</i> memiliki peran tertentu sesuai hak akses dan kebutuhan penggunaannya.
	<i>Use Case</i>	Fungsi, layanan, atau aktivitas yang disediakan sistem untuk memenuhi kebutuhan <i>actor</i> . Setiap <i>use case</i> menggambarkan tujuan yang ingin dicapai pengguna saat berinteraksi dengan sistem, seperti <i>login</i> , mengelola data, melakukan transaksi, atau melihat laporan.
	<i>System Boundary</i>	Batas sistem yang menunjukkan fitur-fitur mana yang termasuk ke dalam sistem dan mana yang berada di luar sistem. Batas ini membantu memperjelas ruang lingkup sistem yang sedang dirancang.
	<i>Association</i>	Garis hubungan antara <i>actor</i> dan <i>use case</i> yang menunjukkan adanya interaksi atau keterlibatan pengguna terhadap fungsi tertentu di dalam sistem.

	<i>Include</i>	Hubungan <i>include</i> menunjukkan bahwa suatu <i>use case</i> selalu melibatkan <i>use case</i> lain sebagai bagian dari proses utama. Relasi ini digunakan untuk menghindari pengulangan proses yang sama pada beberapa fungsi. Pada diagram, hubungan <i>include</i> digambarkan dengan panah yang mengarah dari <i>base use case</i> menuju <i>used use case</i>.
	<i>Extend</i>	Hubungan <i>extend</i> menunjukkan adanya proses tambahan yang bersifat opsional atau kondisi tertentu terhadap <i>use case</i> utama. Relasi ini digunakan jika suatu proses hanya dijalankan dalam situasi tertentu. Pada diagram, hubungan <i>extend</i> digambarkan dengan panah yang mengarah dari <i>extension use case</i> menuju <i>base use case</i>.

Penggunaan *use case diagram* memberikan beberapa manfaat, antara lain mempermudah komunikasi antara pengembang dan pengguna, membantu proses identifikasi kebutuhan sistem, memperjelas hak akses setiap pengguna, serta menjadi dasar dalam perancangan diagram lain maupun implementasi sistem [7, 9].

Dalam penelitian ini, *use case diagram* digunakan untuk menggambarkan interaksi antara pengguna dengan sistem manajemen stok dan penjualan pada Codel Coffee. *Actor* yang terlibat meliputi administrator dan kasir. Administrator memiliki hak akses untuk mengelola data menu, data stok, pengguna, dan laporan, sedangkan kasir berfokus pada proses transaksi penjualan serta melihat data yang dibutuhkan dalam operasional harian. Dengan adanya *use case diagram*, fungsi sistem dapat divisualisasikan secara jelas sebelum tahap implementasi dilakukan. Dengan demikian, *use case diagram* merupakan alat bantu analisis dan perancangan yang penting karena mampu menggambarkan kebutuhan fungsional sistem dari sudut pandang pengguna. Melalui diagram ini, pengembangan sistem dapat dilakukan secara lebih terarah dan sesuai kebutuhan pengguna.

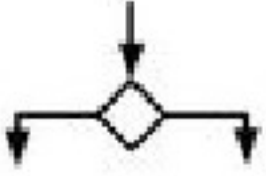
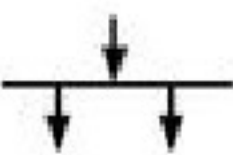
2.8.2 Activity Diagram

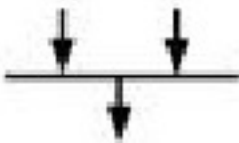
Activity Diagram merupakan diagram UML yang digunakan untuk menggambarkan alur aktivitas atau proses yang terjadi di dalam sistem. Diagram ini menunjukkan urutan aktivitas dari awal hingga akhir serta menggambarkan bagaimana suatu proses dijalankan oleh sistem [9].

Activity Diagram digunakan untuk memodelkan proses bisnis dan alur kerja sistem. Diagram ini membantu memahami bagaimana sistem merespons suatu aktivitas serta bagaimana keputusan dan alur proses terjadi di dalam sistem [7].

Dalam sistem manajemen stok dan penjualan, Activity Diagram digunakan untuk menggambarkan proses seperti pengelolaan stok, transaksi penjualan, dan alur penyajian laporan.

Tabel 2. 2 Simbol Activity Diagram [7]

Simbol	Nama	Keterangan
	<i>Initial Node</i>	Menandai titik awal dimulainya suatu alur aktivitas
	<i>Action</i>	Menunjukkan satu aksi atau langkah kerja yang dilakukan dalam sistem
	<i>Activity State</i>	Menunjukkan keadaan aktivitas yang sedang berlangsung dalam suatu proses
	<i>Control Flow</i>	Menunjukkan arah dan urutan alur aktivitas dari satu aktivitas ke aktivitas lain
	<i>Final Activity</i>	Menandai berakhirnya seluruh aktivitas dalam suatu proses
	<i>Final Flow Node</i>	Menandai akhir dari satu alur proses tanpa menghentikan keseluruhan aktivitas
	<i>Decision node</i>	Menunjukkan titik pengambilan keputusan yang menghasilkan percabangan alur
	<i>Merge node</i>	Menggabungkan kembali beberapa alur keputusan menjadi satu alur
	<i>Fork</i>	Memecah satu alur proses menjadi beberapa alur yang berjalan secara bersamaan

	<i>Join</i>	Menggabungkan kembali beberapa alur paralel menjadi satu alur proses
---	-------------	--

Dengan demikian, Activity Diagram membantu memvisualisasikan alur proses sistem secara logis dan berurutan sehingga memudahkan pemahaman proses kerja sistem.

2.8.3 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan model konseptual yang digunakan untuk menggambarkan struktur data dan hubungan antar data dalam suatu sistem. ERD digunakan sebagai alat bantu dalam perancangan basis data agar data yang akan disimpan dapat terorganisasi dengan baik serta memiliki hubungan yang jelas antar entitas [9].

ERD memiliki peranan penting dalam pengembangan sistem informasi karena membantu pengembang memahami kebutuhan data sebelum proses implementasi *database* dilakukan. Dengan menggunakan ERD, struktur data dapat divisualisasikan dalam bentuk diagram sehingga mempermudah proses analisis, perancangan, serta pengembangan basis data. Dalam perancangan sistem, ERD digunakan untuk menggambarkan hubungan antar entitas yang terdapat di dalam sistem. Setiap entitas merepresentasikan objek yang memiliki data untuk disimpan dan dikelola, sedangkan hubungan antar entitas menunjukkan keterkaitan data dalam proses bisnis sistem tersebut. Melalui ERD, pengembang dapat mengetahui alur hubungan data sehingga mengurangi kemungkinan terjadinya redundansi data dan kesalahan dalam perancangan basis data [20].

Secara umum, ERD terdiri dari beberapa komponen utama, yaitu entitas (*entity*), atribut (*attribute*), dan relasi (*relationship*) [20].

1. Entitas (*Entity*)

Entitas merupakan objek utama yang datanya disimpan di dalam sistem. Entitas dapat berupa orang, barang, transaksi, maupun objek lain yang memiliki informasi untuk dikelola. Dalam ERD, entitas biasanya digambarkan dalam bentuk persegi panjang. Setiap entitas nantinya akan direpresentasikan menjadi tabel pada basis data.

2. Atribut (*Attribute*)

Atribut merupakan karakteristik atau informasi yang dimiliki oleh suatu entitas. Atribut digunakan untuk menjelaskan data secara lebih rinci. Sebagai contoh, entitas pengguna dapat

memiliki atribut seperti id pengguna, nama, alamat, nomor telepon, dan kata sandi. Atribut biasanya digambarkan dalam bentuk oval yang terhubung dengan entitas.

3. Relasi (*Relationship*)

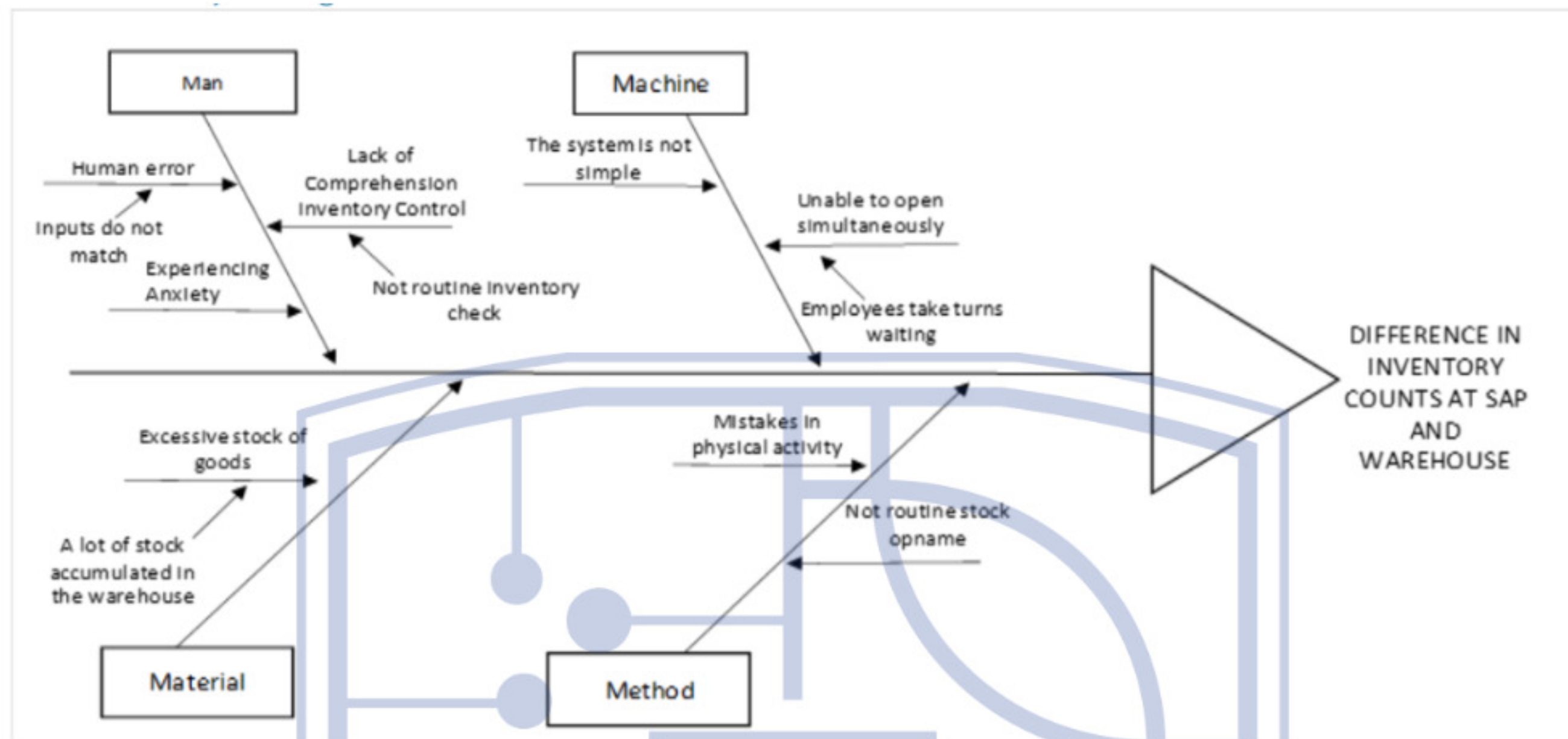
Relasi merupakan hubungan antara satu entitas dengan entitas lainnya di dalam sistem. Relasi digunakan untuk menunjukkan keterkaitan antar data sehingga proses pengelolaan data dapat berjalan secara terintegrasi. Relasi biasanya digambarkan menggunakan garis penghubung antar entitas.

Selain komponen utama tersebut, ERD juga memiliki konsep kardinalitas (*cardinality*) yang digunakan untuk menunjukkan jumlah hubungan antar entitas. Kardinalitas menjelaskan bagaimana suatu entitas berhubungan dengan entitas lain, seperti hubungan satu ke satu (*one to one*), satu ke banyak (*one to many*), maupun banyak ke banyak (*many to many*). Penggunaan kardinalitas membantu pengembang menentukan struktur relasi tabel dalam basis data agar sesuai dengan kebutuhan sistem [20].

ERD digunakan sebagai tahap awal dalam proses perancangan basis data karena mampu memberikan gambaran menyeluruh mengenai struktur data yang akan dibangun. Dengan adanya ERD, proses implementasi *database* menjadi lebih terarah karena hubungan antar tabel dan kebutuhan data telah dirancang sebelumnya [20].

Dalam penelitian ini, ERD digunakan untuk merancang struktur basis data pada sistem manajemen stok dan penjualan Codel Coffee. ERD digunakan untuk menggambarkan hubungan antar entitas seperti pengguna, menu, stok, transaksi penjualan, dan detail transaksi sehingga data yang dikelola dalam sistem dapat saling terhubung dengan baik. Perancangan ERD pada sistem ini bertujuan untuk membantu proses pengelolaan data agar lebih terstruktur, meminimalkan redundansi data, serta mendukung proses transaksi dan pengelolaan stok secara terintegrasi. Dengan demikian, ERD merupakan alat bantu penting dalam perancangan basis data karena mampu menggambarkan struktur data, atribut, serta hubungan antar entitas secara jelas dan sistematis sehingga mendukung pengembangan sistem informasi yang lebih terorganisasi dan efisien [9, 20].

2.9 Fishbone Diagram



Gambar 2. 2 Contoh Fishbone Diagram [23]

Fishbone diagram atau diagram sebab-akibat merupakan salah satu alat analisis yang digunakan untuk mengidentifikasi hubungan antara suatu permasalahan dengan berbagai faktor penyebabnya. Diagram ini dikembangkan untuk membantu proses pemecahan masalah dengan cara memetakan kemungkinan penyebab secara sistematis hingga diperoleh akar masalah yang paling berpengaruh terhadap terjadinya suatu kondisi. Bentuk diagram ini menyerupai tulang ikan, di mana bagian kepala menunjukkan masalah utama yang dianalisis, sedangkan cabang-cabang tulang menggambarkan kelompok penyebab yang memengaruhi masalah tersebut [22, 23].

Fishbone diagram dikenal juga sebagai *cause and effect diagram* atau *Ishikawa diagram*. Dalam praktik manajemen mutu dan analisis sistem, diagram ini banyak digunakan karena mampu menyajikan hubungan sebab dan akibat secara visual sehingga memudahkan proses identifikasi masalah. Melalui pendekatan ini, tim atau peneliti dapat melihat suatu masalah tidak hanya dari satu sisi, tetapi dari berbagai faktor yang saling berkaitan. Dengan demikian, proses pengambilan keputusan menjadi lebih terarah karena didasarkan pada penyebab yang nyata, bukan hanya gejala yang terlihat di permukaan [22, 23].

Fishbone diagram juga merupakan bagian dari pendekatan *Root Cause Analysis* (RCA), yaitu metode analisis yang bertujuan menemukan penyebab utama dari suatu masalah agar tindakan perbaikan yang dilakukan dapat bersifat permanen. Jika hanya menangani dampak atau gejala masalah tanpa mengetahui akar penyebabnya, maka permasalahan yang sama berpotensi

terulang kembali. Oleh karena itu, penggunaan fishbone diagram sangat membantu dalam proses evaluasi dan perbaikan berkelanjutan pada suatu organisasi maupun sistem kerja [23].

Dalam penerapannya, fishbone diagram mengelompokkan penyebab masalah ke dalam beberapa kategori tertentu agar proses analisis lebih terstruktur. Salah satu pendekatan yang umum digunakan adalah kategori **4M**, yaitu *Man*, *Machine*, *Method*, dan *Material*. Pengelompokan ini bertujuan untuk memudahkan identifikasi faktor-faktor yang memengaruhi suatu proses berdasarkan aspek sumber daya manusia, peralatan atau teknologi, prosedur kerja, serta bahan atau sumber daya yang digunakan. Penjelasan masing-masing kategori 4M adalah sebagai berikut [22].

1. *Man*

Faktor *Man* berkaitan dengan sumber daya manusia yang terlibat dalam suatu proses. Permasalahan pada faktor ini dapat berupa kurangnya keterampilan, kurangnya pengalaman, rendahnya kedisiplinan, kesalahan input data, kurang teliti dalam bekerja, maupun kurangnya koordinasi antarpegawai.

2. *Machine*

Faktor *Machine* berkaitan dengan peralatan, mesin, perangkat keras, maupun teknologi yang digunakan dalam mendukung operasional. Permasalahan dapat berupa kerusakan perangkat, sistem lambat, komputer tidak memadai, jaringan tidak stabil, atau alat kerja yang tidak berfungsi optimal.

3. *Method*

Faktor *Method* berkaitan dengan prosedur, aturan kerja, maupun metode operasional yang diterapkan. Permasalahan pada aspek ini dapat berupa tidak adanya standar prosedur kerja, alur kerja yang tidak efisien, proses pencatatan manual, atau metode pengawasan yang kurang efektif.

4. *Material*

Faktor *Material* berkaitan dengan bahan, data, atau sumber daya yang digunakan dalam proses kerja. Permasalahan pada kategori ini dapat berupa kualitas bahan yang kurang baik, data tidak lengkap, stok bahan tidak sesuai, atau keterlambatan ketersediaan sumber daya.

Pengelompokan faktor-faktor tersebut membantu proses analisis menjadi lebih sistematis karena penyebab masalah dapat ditinjau dari berbagai aspek yang saling berkaitan. Dengan demikian, solusi yang dihasilkan tidak hanya berfokus pada satu faktor, tetapi mempertimbangkan keseluruhan proses yang memengaruhi terjadinya masalah [22, 23].

Dalam penelitian ini, *fishbone diagram* digunakan untuk menganalisis berbagai faktor yang menyebabkan permasalahan pada proses pengelolaan stok dan penjualan di Codel Coffee. Analisis dilakukan dengan mengidentifikasi penyebab berdasarkan kategori *Man*, *Machine*, *Method*, dan *Material*, sehingga akar permasalahan dalam pengelolaan stok maupun proses transaksi penjualan dapat diketahui secara lebih jelas. Hasil analisis tersebut selanjutnya menjadi dasar dalam perancangan sistem manajemen stok dan penjualan yang mampu menjawab permasalahan yang ditemukan [22, 23].

2.10 Pengujian Sistem

Pengujian sistem merupakan bagian yang tidak terpisahkan dari proses pengembangan sistem informasi, khususnya untuk memastikan bahwa sistem yang telah dibangun dapat berfungsi sesuai dengan spesifikasi dan kebutuhan yang telah ditetapkan sebelumnya. Pengujian dilakukan secara terstruktur untuk menemukan kesalahan, ketidaksesuaian, maupun kegagalan fungsi sebelum sistem digunakan secara operasional, sehingga kualitas sistem dapat terjaga dan risiko kegagalan saat implementasi dapat diminimalkan [14, 24].

Pengujian fungsional diperlukan untuk memastikan bahwa setiap fitur dalam sistem berjalan sesuai dengan kebutuhan yang telah ditetapkan. Kegagalan pada fungsi-fungsi penting, seperti autentikasi, pengelolaan data, pemrosesan transaksi, dan pembuatan laporan, dapat mengganggu operasional bisnis dan menurunkan kualitas layanan yang diberikan sistem. Oleh karena itu, proses verifikasi dan validasi perangkat lunak melalui pengujian yang terstruktur menjadi bagian penting dalam pengembangan sistem informasi. Selain memastikan fungsi sistem berjalan dengan baik, aspek *usability* juga perlu dievaluasi untuk mengetahui tingkat kemudahan penggunaan dan kepuasan pengguna terhadap sistem yang dikembangkan [7, 14].

Secara umum, teknik pengujian sistem dibedakan menjadi dua berdasarkan sudut pandang penguji terhadap struktur internal sistem, yaitu *White Box Testing* yang memeriksa logika internal kode program, dan *Black Box Testing* yang berfokus pada kesesuaian fungsi sistem dari sisi pengguna. Sementara itu, evaluasi *usability* dilakukan menggunakan instrumen terstandarisasi untuk mengukur penerimaan pengguna terhadap sistem secara keseluruhan [14, 24].

Dalam penelitian ini, metode yang digunakan untuk menguji sistem adalah *Black Box Testing*, sedangkan evaluasi kemudahan penggunaan sistem dilakukan menggunakan metode *System Usability Scale (SUS)*.

2.10.1 *Black Box Testing*

Black Box Testing merupakan teknik pengujian yang berfokus pada pemeriksaan fungsi sistem berdasarkan masukan (*input*) dan keluaran (*output*) yang dihasilkan tanpa memperhatikan struktur kode program atau proses internal sistem. Pengujian dilakukan untuk memastikan bahwa setiap fungsi berjalan sesuai dengan kebutuhan dan spesifikasi yang telah ditetapkan. Teknik ini dapat digunakan untuk menguji masukan yang valid maupun tidak valid, sehingga hasil pengujian dapat merepresentasikan pengalaman pengguna dalam menggunakan sistem secara langsung. Selain itu, *Black Box Testing* dinilai efektif untuk menguji sistem informasi berbasis web yang memiliki banyak modul fungsional yang saling berinteraksi [13, 14, 24].

Pada *Black Box Testing*, penguji menilai sistem semata-mata dari sudut pandang pengguna. Penguji tidak perlu mengetahui bagaimana proses sistem ditulis di dalam kode program, melainkan hanya memperhatikan apakah sistem merespons masukan yang diberikan dengan keluaran yang sesuai harapan. Teknik ini menangani baik masukan yang valid maupun tidak valid sesuai dengan kebutuhan pengujian [24].

Karakteristik utama *Black Box Testing* adalah sebagai berikut [14, 24]:

1. Pengujian dilakukan berdasarkan fungsi sistem tanpa melihat struktur internal atau kode program.
2. Pengujian berfokus pada hubungan antara *input* dan *output* sistem.
3. Penguji tidak memerlukan pengetahuan teknis mengenai implementasi program.
4. Pengujian mengacu pada kebutuhan dan spesifikasi sistem yang telah ditetapkan.
5. Hasil pengujian dinilai berdasarkan kesesuaian fungsi sistem terhadap keluaran yang diharapkan.

Black Box Testing memiliki beberapa teknik perancangan kasus uji yang umum digunakan untuk mengidentifikasi dan mengevaluasi fungsi sistem berdasarkan masukan dan keluaran yang diharapkan. Teknik-teknik tersebut antara lain sebagai berikut [24]:

1. ***Equivalence Partitioning***: Teknik ini membagi seluruh nilai masukan ke dalam sejumlah partisi atau kelompok, baik partisi valid maupun tidak valid. Kasus uji kemudian dirancang dari masing-masing partisi untuk mengungkap kesalahan sistem. Dengan pendekatan ini, pengujian dapat mencakup berbagai kondisi masukan secara efisien tanpa harus menguji setiap nilai satu per satu.

2. **Boundary Value Analysis:** Teknik ini berfokus pada nilai-nilai di batas domain masukan, karena kesalahan sistem sering kali ditemukan pada nilai batas tersebut. Kasus uji dirancang untuk nilai tepat pada batas, di bawah batas, dan di atas batas domain masukan yang telah ditentukan.
3. **Cause Effect Graphing:** Teknik ini memfokuskan pengujian pada hubungan logis antara kondisi masukan (*cause*) dan kondisi keluaran (*effect*) menggunakan operator Boolean. Hubungan antara sebab dan akibat tersebut digunakan sebagai dasar perancangan kasus uji.
4. **Decision Table Based Testing:** Teknik ini digunakan untuk menangani kombinasi kondisi masukan yang besar. Tabel keputusan memuat seluruh kemungkinan nilai kondisi beserta tindakan yang dihasilkan, sehingga sangat berguna untuk menguji alur bisnis yang kompleks.
5. **Error Guessing:** Teknik ini mengandalkan pengalaman dan intuisi penguji dalam mengidentifikasi kemungkinan kesalahan pada sistem. Penguji yang berpengalaman mengetahui di mana dan jenis kesalahan apa yang paling sering ditemukan pada suatu sistem.

Adapun tahapan pelaksanaan *Black Box Testing* yang digunakan dalam penelitian ini mengacu pada proses berikut [7, 14, 24]:

1. **Perencanaan Pengujian (Test Plan):** Tahap pertama adalah menyusun rencana pengujian yang memuat ruang lingkup pengujian, tujuan, metode yang digunakan, serta daftar fungsi atau modul yang akan diuji. Seluruh fungsi yang tercantum dalam kebutuhan fungsional sistem menjadi cakupan pengujian untuk memastikan tidak ada fungsi yang terlewat.
2. **Perancangan Kasus Uji (Test Case):** Kasus uji disusun untuk setiap fungsi sistem yang akan diuji menggunakan teknik *Equivalence Partitioning*. Teknik ini dilakukan dengan mengelompokkan data masukan ke dalam kelas-kelas yang dianggap mewakili kondisi valid maupun tidak valid, sehingga jumlah kasus uji dapat diminimalkan tanpa mengurangi efektivitas pengujian. Dalam penyusunannya, setiap kasus uji memuat beberapa komponen penting, yaitu ID kasus uji, deskripsi pengujian, data masukan (*input*), langkah pengujian, hasil yang diharapkan, hasil yang diperoleh, serta status akhir pengujian berupa *pass* atau *fail* [14].
3. **Persiapan Data Uji:** Data masukan disiapkan sesuai dengan skenario yang telah dirancang, mencakup data yang sesuai kondisi normal (*valid*) maupun data yang tidak sesuai (*invalid*) untuk memastikan sistem dapat menangani berbagai kondisi masukan dengan benar.

4. **Pelaksanaan Pengujian:** Setiap kasus uji dijalankan satu per satu sesuai skenario yang telah dirancang berdasarkan penggunaan nyata pengguna sistem. Penguji memberikan masukan ke dalam sistem dan mengamati respons atau keluaran yang dihasilkan secara langsung.
5. **Pencatatan dan Perbandingan Hasil:** *Output* aktual yang dihasilkan sistem dicatat dan dibandingkan dengan *output* yang diharapkan. Apabila *output* aktual sesuai dengan yang diharapkan, fungsi dinyatakan **Berhasil** (*pass*). Apabila tidak sesuai, fungsi dinyatakan **Tidak Berhasil** (*fail*) dan dicatat sebagai temuan yang memerlukan perbaikan
6. **Analisis dan Pelaporan :** Seluruh hasil pengujian didokumentasikan secara sistematis dan dianalisis untuk mengetahui tingkat keberhasilan fungsional dari setiap fungsi yang diuji.

Persentase keberhasilan pengujian digunakan untuk mengetahui tingkat keberhasilan sistem dalam memenuhi skenario pengujian yang telah ditentukan. Persentase tersebut diperoleh dengan membandingkan jumlah *test case* yang berhasil dijalankan (*pass*) terhadap jumlah keseluruhan *test case* yang diuji, kemudian dikalikan 100% [14].

$$\text{Persentase Keberhasilan} = \frac{\text{Jumlah Test Case Pass}}{\text{Jumlah Test Case Total}} \times 100\% \dots\dots\dots (1)$$

Temuan yang dinyatakan tidak berhasil (*fail*) perlu ditindaklanjuti melalui proses perbaikan sebelum sistem diimplementasikan secara operasional agar fungsi sistem dapat berjalan sesuai dengan kebutuhan yang telah ditetapkan [14].

Dalam penelitian ini, *Black Box Testing* dilakukan untuk menguji seluruh fungsi utama pada sistem manajemen stok dan penjualan Codel Coffee, meliputi proses *login*, pengelolaan data menu, pengelolaan stok, transaksi penjualan, proses pembayaran, penyesuaian stok, serta penyajian laporan stok dan laporan penjualan. Teknik *Equivalence Partitioning* dipilih karena sistem banyak melibatkan masukan data pada formulir yang perlu diuji untuk kondisi valid maupun tidak valid, seperti validasi data *login*, validasi formulir penambahan menu, dan validasi transaksi penjualan [14, 24].

2.10.2 Usability

Usability merupakan salah satu atribut kualitas perangkat lunak yang mengukur seberapa mudah antarmuka sistem dapat digunakan oleh pengguna yang dituju. Sebuah sistem informasi yang berhasil dibangun secara teknis dan fungsional belum tentu dapat diterima dengan baik oleh

pengguna apabila tidak memperhatikan aspek kemudahan penggunaan. Oleh karena itu, evaluasi *usability* menjadi bagian yang tidak terpisahkan dari proses pengembangan sistem informasi, khususnya sistem yang akan digunakan secara langsung oleh pengguna dalam kegiatan operasional sehari-hari [7, 25].

Definisi *usability* yang paling banyak diacu adalah yang dikemukakan oleh Nielsen, yang menyatakan bahwa *usability* merupakan atribut kualitas yang menilai seberapa mudah antarmuka pengguna digunakan. Menurut Nielsen, *usability* bukan sekadar satu dimensi, melainkan terdiri dari lima komponen yang saling melengkapi, yaitu [25]:

1. *Learnability* (Kemudahan Dipelajari): Seberapa mudah pengguna baru dapat menyelesaikan tugas-tugas dasar ketika pertama kali berinteraksi dengan sistem. *Learnability* yang baik memungkinkan pengguna menjadi produktif dalam waktu singkat tanpa memerlukan pelatihan yang panjang.
2. *Efficiency* (Efisiensi): Seberapa cepat pengguna yang sudah terbiasa dengan sistem dapat menyelesaikan tugas-tugasnya. Sistem yang efisien memungkinkan pengguna berpengalaman bekerja dengan cepat dan tepat.
3. *Memorability* (Kemudahan Diingat): Seberapa mudah pengguna dapat kembali menggunakan sistem setelah tidak menggunakannya dalam jangka waktu tertentu. Sistem dengan *memorability* yang baik tidak mengharuskan pengguna mempelajari ulang cara penggunaannya setiap kali kembali menggunakannya.
4. *Errors* (Penanganan Kesalahan): Seberapa banyak kesalahan yang dilakukan pengguna saat menggunakan sistem, seberapa serius kesalahan tersebut, dan seberapa mudah pengguna dapat memulihkan diri dari kesalahan yang terjadi. Sistem yang baik meminimalkan jumlah kesalahan dan menyediakan mekanisme pemulihan yang jelas.
5. *Satisfaction* (Kepuasan): Seberapa menyenangkan penggunaan sistem bagi pengguna. Komponen ini mengukur aspek subjektif dari pengalaman pengguna, yaitu apakah sistem terasa nyaman dan memuaskan untuk digunakan.

Dalam konteks evaluasi sistem informasi, *usability* memiliki peran yang sangat penting karena berhubungan langsung dengan tingkat penerimaan pengguna terhadap sistem. Sistem dengan *usability* yang tinggi akan lebih mudah diadopsi oleh pengguna, mengurangi kebutuhan pelatihan, dan meningkatkan efisiensi kerja secara keseluruhan [7, 25]. Pada penelitian ini, aspek *usability* dievaluasi untuk mengukur sejauh mana sistem manajemen stok dan penjualan yang

dikembangkan untuk Codel Coffee dapat digunakan dengan mudah, efisien, dan memuaskan oleh *owner* maupun *staff* dalam menjalankan kegiatan operasional harian.

2.10.3 System Usability Scale (SUS)

System Usability Scale (SUS) merupakan instrumen pengukuran *usability* yang dikembangkan oleh John Brooke pada tahun 1986 dan dipublikasikan secara formal pada tahun 1996. SUS dirancang sebagai alat yang cepat, mudah digunakan, dan dapat diterapkan pada berbagai jenis sistem untuk mengukur persepsi subjektif pengguna terhadap kemudahan penggunaan sistem secara keseluruhan. Sejak diperkenalkan, SUS telah menjadi salah satu instrumen *usability* yang paling banyak digunakan di dunia karena kesederhanaannya tanpa mengorbankan keandalan hasil pengukuran [25].

System Usability Scale (SUS) merupakan instrumen yang andal dan serbaguna untuk mengukur persepsi subjektif pengguna terhadap *usability* suatu sistem. Instrumen ini mampu memberikan evaluasi yang cepat dan mudah tanpa mengurangi kualitas hasil pengukuran, sehingga banyak digunakan dalam berbagai penelitian maupun pengembangan sistem. Selain itu, SUS menghasilkan satu skor tunggal yang merepresentasikan penilaian menyeluruh terhadap *usability* sistem, sehingga memudahkan perbandingan hasil antar sistem maupun antar kelompok pengguna [26].

SUS terdiri dari sepuluh pernyataan yang harus dinilai oleh responden menggunakan skala *Likert* dengan rentang 1 (*Strongly Disagree*) hingga 5 (*Strongly Agree*). Pernyataan bernomor ganjil bersifat positif, sedangkan pernyataan bernomor genap bersifat negatif. Kesepuluh pernyataan SUS adalah sebagai berikut [25, 26]:

1. Saya pikir saya akan sering menggunakan sistem ini.
2. Saya merasa sistem ini terlalu kompleks dan tidak perlu.
3. Saya merasa sistem ini mudah digunakan.
4. Saya pikir saya membutuhkan bantuan dari orang yang ahli teknis untuk menggunakan sistem ini.
5. Saya merasa berbagai fungsi dalam sistem ini terintegrasi dengan baik.
6. Saya pikir terdapat banyak ketidakkonsistenan dalam sistem ini.
7. Saya bayangkan kebanyakan orang akan mudah mempelajari sistem ini dengan cepat.
8. Saya merasa sistem ini sangat rumit untuk digunakan.

9. Saya merasa sangat percaya diri menggunakan sistem ini.
10. Saya perlu mempelajari banyak hal sebelum dapat menggunakan sistem ini.

Perhitungan skor SUS dilakukan melalui dua langkah berikut [25, 26]:

1. Untuk pernyataan bernomor **ganjil** (1, 3, 5, 7, 9): kontribusi skor = nilai skala yang dipilih dikurangi 1.

$$\text{Kontribusi Skor} = \text{Nilai Skala yang dipilih} - 1 \dots \dots \dots (2)$$

2. Untuk pernyataan bernomor **genap** (2, 4, 6, 8, 10): kontribusi skor = 5 dikurangi nilai skala yang dipilih.

$$\text{Kontribusi Skor} = 5 - \text{Nilai Skala yang dipilih} \dots \dots \dots (3)$$

Jumlahkan seluruh kontribusi skor dari sepuluh pernyataan, kemudian kalikan dengan 2,5 untuk memperoleh skor akhir SUS dalam rentang 0 hingga 100.

$$\text{Skor SUS} = (\text{Total Nilai Poin}) \times 2,5 \dots \dots \dots (4)$$

Berdasarkan hasil berbagai pengujian usability yang menggunakan SUS, skor rata-rata yang diperoleh adalah 68 sehingga nilai tersebut digunakan sebagai ambang batas penerimaan sistem. Sistem dengan skor di atas 68 dinilai dapat diterima oleh pengguna, sedangkan sistem dengan skor di bawah 68 menunjukkan perlunya perbaikan lebih lanjut terhadap usability sistem [26].

Tabel 2. 3 Rentang Skor SUS

Rentang Skor SUS	Interpretasi
<51	<i>Awful</i> (Buruk)
51 – 67	<i>Poor</i> (Di Bawah Rata – rata)
68 – 80	<i>OK / Good</i> (Dapat diterima)
81 – 90	<i>Excellent</i> (Sangat Baik)
>91	<i>Best Imaginable</i> (Terbaik)

Dalam penelitian ini, SUS digunakan untuk mengukur tingkat *usability* sistem manajemen stok dan penjualan yang dikembangkan untuk Codel Coffee. Pengujian SUS dilakukan setelah responden menggunakan sistem secara langsung, dengan cara mengisi kuesioner sepuluh

pernyataan SUS berdasarkan pengalaman mereka. Responden terdiri dari *owner* dan *staff* Codel Coffee sebagai pengguna langsung sistem. Hasil skor SUS yang diperoleh kemudian diinterpretasikan menggunakan tabel penilaian kata sifat di atas untuk mengetahui tingkat penerimaan pengguna terhadap sistem yang dikembangkan, sehingga dapat memberikan gambaran apakah sistem yang dibangun layak digunakan secara operasional dari sisi kemudahan penggunaan [25, 26].

