

BAB II

KAJIAN LITERATUR

2.1 Sistem Informasi

Sistem merupakan sekumpulan elemen atau komponen yang saling berhubungan dan bekerja sama untuk mencapai tujuan tertentu. Setiap sistem memiliki bagian-bagian yang saling berinteraksi serta menjalankan fungsi masing-masing dalam suatu proses yang terorganisasi. Pada umumnya, suatu sistem terdiri dari proses masukan (*input*), proses pengolahan (*process*), serta keluaran (*output*) yang saling berkaitan sehingga menghasilkan suatu hasil yang diharapkan [9,10].



Gambar 2.1 Alur Pemrosesan Data [10]

Informasi merupakan data yang telah diolah melalui suatu proses tertentu sehingga memiliki makna dan nilai bagi penggunanya. Informasi yang dihasilkan harus memiliki kualitas yang baik, seperti akurat, relevan, dan tepat waktu agar dapat digunakan secara efektif dalam mendukung berbagai aktivitas organisasi. Informasi yang berkualitas akan membantu pengguna dalam memahami suatu kondisi serta menjadi dasar dalam proses pengambilan keputusan [11].

Sistem informasi dapat diartikan sebagai suatu sistem yang mengintegrasikan berbagai komponen, seperti perangkat keras, perangkat lunak, data, prosedur, serta sumber daya manusia, dalam suatu proses yang terstruktur untuk menghasilkan informasi yang bermanfaat. Sistem informasi berfungsi untuk mengumpulkan, mengolah, menyimpan, dan mendistribusikan data sehingga mendukung kegiatan operasional maupun manajerial dalam organisasi, seperti pengolahan transaksi, penyusunan laporan, dan sebagai dasar dalam proses pengambilan keputusan oleh pihak manajemen [12,13]. Dalam penelitian ini, konsep sistem informasi digunakan sebagai dasar untuk mendukung pengelolaan data operasional perusahaan.

2.1.1 Jenis-Jenis Sistem Informasi

Sistem informasi dalam suatu organisasi dapat diklasifikasikan ke dalam beberapa jenis berdasarkan fungsi serta kebutuhan penggunaannya. Setiap jenis sistem informasi memiliki peran berbeda dalam mendukung kegiatan operasional maupun manajerial organisasi. Dengan adanya berbagai jenis sistem informasi, organisasi dapat memanfaatkan sistem yang sesuai dengan kebutuhan dalam pengelolaan data dan informasi [14].

Beberapa jenis sistem informasi antara lain sebagai berikut [14]:

1. Sistem Informasi Keuangan

Sistem informasi keuangan digunakan untuk mendukung pengelolaan data keuangan dalam suatu organisasi. Sistem ini membantu bagian keuangan dalam mengolah data transaksi serta menghasilkan informasi yang berkaitan dengan kondisi keuangan organisasi, seperti laporan arus kas maupun informasi mengenai pengalokasian serta pengendalian sumber daya keuangan.

2. Sistem Informasi Akuntansi

Sistem informasi akuntansi menyediakan informasi mengenai kondisi keuangan organisasi, baik bagi pihak internal maupun eksternal. Informasi yang dihasilkan biasanya disajikan dalam bentuk laporan keuangan, laporan periodik, maupun analisis tertentu yang dapat digunakan untuk mendukung kegiatan pengelolaan keuangan organisasi.

3. Sistem Informasi Manufaktur

Sistem informasi manufaktur digunakan untuk mendukung kegiatan produksi dalam suatu organisasi. Sistem ini menyediakan informasi yang berkaitan dengan proses produksi, seperti pengelolaan persediaan, penjadwalan produksi, pengendalian kualitas produk, serta pelaporan biaya produksi sehingga kegiatan produksi dapat berjalan secara lebih efektif.

4. Sistem Informasi Sumber Daya Manusia

Sistem informasi sumber daya manusia digunakan untuk mengelola informasi yang berkaitan dengan tenaga kerja dalam organisasi. Sistem ini mencakup data karyawan, penggajian, penilaian kinerja, serta informasi lain yang berkaitan dengan pengelolaan sumber daya manusia.

5. Sistem Informasi Manajemen

Sistem informasi manajemen digunakan untuk menyediakan informasi yang dibutuhkan pihak manajemen dalam menjalankan fungsi perencanaan, pengendalian, dan pengambilan keputusan. Sistem ini memanfaatkan berbagai sumber daya, seperti

manusia, teknologi, dan prosedur, untuk menghasilkan informasi yang berguna bagi organisasi.

Pada penerapannya, suatu sistem informasi dapat memiliki lebih dari satu fungsi sesuai dengan kebutuhan pengguna. Sistem yang dikembangkan pada penelitian ini tidak hanya digunakan untuk mencatat transaksi, tetapi juga membantu pengelolaan stok, piutang, dan laporan keuangan. Hal ini menunjukkan bahwa sistem informasi yang dirancang memadukan beberapa fungsi untuk mendukung kegiatan perusahaan.

2.1.2 Pengembangan Sistem Informasi

Pengembangan sistem informasi merupakan suatu rangkaian proses terstruktur yang dikenal dengan istilah *System Development Life Cycle* (SDLC). Proses ini bertujuan untuk menciptakan, memodifikasi, atau meningkatkan sistem guna memenuhi kebutuhan pengguna serta mendukung efisiensi operasional organisasi. Tahapan dalam pengembangan sistem umumnya mencakup analisis kebutuhan (*requirement analysis*), perancangan arsitektur (*design*), implementasi kode program (*implementation*), pengujian fungsionalitas (*testing*), serta pemeliharaan berkelanjutan (*maintenance*). Pendekatan yang sistematis dalam SDLC sangat diperlukan untuk menjamin bahwa perangkat lunak yang dihasilkan memiliki kualitas yang baik, andal, dan selaras dengan ekspektasi para pemangku kepentingan [15].

Dalam praktiknya, penerapan metodologi pengembangan sistem yang terstruktur menjadi kunci keberhasilan dalam membangun solusi teknologi informasi, terutama pada sektor UMKM yang masih menghadapi kendala pencatatan manual. Salah satu pendekatan yang adaptif terhadap kebutuhan pengguna adalah metode *prototype*. Sebagaimana ditunjukkan dalam penelitian yang dilakukan pada UMKM MertaBeauty, penggunaan metode *prototype* dalam pengembangan sistem informasi penjualan berbasis *web* mampu menghasilkan solusi yang tepat guna. Melalui tahapan iteratif yang melibatkan pengguna secara langsung, mulai dari identifikasi kebutuhan awal, perancangan desain, pembuatan purwarupa, evaluasi, hingga revisi berulang. Sistem yang dikembangkan berhasil mengintegrasikan proses pencatatan transaksi, pengelolaan stok produk, serta penyajian laporan penjualan dalam satu platform terpadu. Pendekatan ini tidak hanya meminimalkan kesenjangan antara ekspektasi pengguna dan fungsionalitas sistem, tetapi juga memastikan bahwa antarmuka yang dihasilkan mudah dioperasikan oleh pemilik usaha yang sebelumnya terbiasa dengan metode manual. Hasil akhir penelitian tersebut berupa sistem informasi yang

mampu mempercepat proses transaksi, menyediakan data analisis penjualan secara *real-time*, serta mendukung perluasan jangkauan pemasaran melalui platform digital [16].

2.2 Teori dalam Pengembangan Sistem

Dalam pengembangan sistem informasi untuk optimalisasi pengelolaan stok, penjualan, dan keuangan pada CV Jayanti Muliatama, diperlukan beberapa teori yang menjadi dasar dalam memahami kebutuhan sistem. Landasan teori yang digunakan meliputi sistem informasi penjualan, distribusi produk, manajemen persediaan, piutang usaha, serta laporan keuangan. Kelima teori tersebut dipilih karena berkaitan langsung dengan proses operasional perusahaan, mulai dari pencatatan transaksi, pengelolaan produk, pemantauan piutang, hingga penyajian informasi keuangan.

Landasan teori tersebut berfungsi sebagai dasar konseptual dalam menentukan kebutuhan, alur kerja, dan fitur utama pada sistem yang dikembangkan. Setiap teori memberikan batasan dan arah terhadap proses yang akan diterapkan agar sistem tidak hanya berfungsi sebagai media pencatatan, tetapi juga sebagai alat bantu pengolahan informasi. Dengan demikian, teori-teori tersebut digunakan untuk memperkuat keterkaitan antara permasalahan operasional, rancangan sistem, dan informasi yang dihasilkan.

2.2.1 Sistem Informasi Penjualan

Sistem informasi penjualan merupakan bagian dari sistem informasi bisnis yang berperan dalam mengelola aktivitas penjualan secara terstruktur dan sistematis. Sistem ini mencakup serangkaian prosedur seperti perencanaan, pencatatan, perhitungan, pembuatan dokumen, serta penyediaan informasi yang berkaitan dengan kegiatan penjualan untuk mendukung kebutuhan manajemen dalam suatu organisasi. Sistem informasi penjualan tidak hanya berfungsi sebagai alat pencatatan transaksi, tetapi juga mampu mengintegrasikan seluruh proses penjualan sehingga data yang dihasilkan menjadi lebih terorganisasi dan mudah diakses. Dengan adanya sistem ini, perusahaan dapat mengelola informasi penjualan secara lebih efektif dibandingkan dengan metode manual [17]. Sistem informasi penjualan juga berfungsi untuk mengumpulkan dan memproses data transaksi serta menghasilkan informasi yang berkaitan dengan penjualan dan keuangan sebagai dasar dalam pengambilan keputusan bisnis [18].

Beberapa fungsi utama sistem informasi penjualan meliputi [17,18]:

1. Mengelola proses penjualan

Sistem digunakan untuk mengelola seluruh aktivitas penjualan mulai dari pemesanan hingga transaksi selesai, sehingga mengurangi kesalahan operasional dan meningkatkan efisiensi.

2. Melakukan pencatatan dan pengolahan data transaksi

Sistem berfungsi untuk mencatat dan mengolah data transaksi penjualan secara otomatis, mencakup perhitungan serta penyimpanan data sehingga menghasilkan informasi yang lebih akurat.

3. Menyediakan informasi bagi manajemen

Sistem menghasilkan berbagai informasi yang dapat digunakan oleh manajemen untuk melakukan pengawasan, evaluasi, serta perencanaan strategi bisnis ke depan.

4. Meningkatkan efisiensi operasional

Penggunaan sistem memungkinkan proses bisnis dilakukan secara lebih cepat dan efisien tanpa bergantung pada metode manual, sekaligus menghemat waktu, tenaga, dan biaya.

5. Mendukung transaksi penjualan secara *online* dan *offline*

Sistem berbasis *web* mampu menangani transaksi baik secara langsung (*offline*) maupun secara *online* dalam satu sistem terintegrasi, memberikan fleksibilitas dalam melayani pelanggan dari berbagai wilayah.

6. Mempermudah promosi dan pemasaran produk

Sistem berbasis *web* memungkinkan informasi produk dapat diakses secara luas dan tidak terbatas waktu, meningkatkan efektivitas promosi dibanding media yang bersifat sementara.

Transaksi penjualan menjadi bagian penting karena menghasilkan data yang berpengaruh terhadap proses operasional lainnya. Data yang berasal dari transaksi penjualan tidak hanya menunjukkan jumlah dan nilai penjualan, tetapi juga dapat berkaitan dengan perubahan stok produk, pembentukan piutang pada transaksi kredit, serta penyusunan laporan. Oleh karena itu, teori sistem informasi penjualan membantu memperjelas bahwa fitur penjualan perlu dirancang sebagai proses yang saling terhubung dengan pengelolaan persediaan, piutang, dan informasi keuangan.

2.2.2 Distribusi Produk

Distribusi produk merupakan bagian dari aktivitas *supply chain management* yang berperan dalam mengatur aliran produk dan informasi dari produsen hingga ke konsumen akhir. Dalam sistem rantai pasok, distribusi tidak hanya berkaitan dengan pemindahan fisik

produk, tetapi juga mencakup integrasi data dan informasi yang mendukung kelancaran proses penyaluran produk secara efektif dan efisien. Dengan adanya sistem yang terintegrasi, proses distribusi dapat berjalan lebih optimal serta mampu meningkatkan akurasi dalam pengambilan keputusan [19].

Secara konseptual, distribusi produk merupakan proses perencanaan, pelaksanaan, dan pengendalian arus produk dari titik asal ke titik konsumsi guna memenuhi kebutuhan pelanggan. Distribusi mencakup berbagai aktivitas yang saling terhubung, seperti penyimpanan, transportasi, pengemasan, serta pemrosesan pesanan. Selain itu, distribusi juga melibatkan aliran informasi yang menjadi pendukung utama dalam memastikan kelancaran proses distribusi secara menyeluruh [20]. Distribusi produk memiliki peran penting dalam mendukung operasional perusahaan, terutama dalam memastikan ketersediaan produk, ketepatan waktu pengiriman, serta efisiensi biaya distribusi. Dalam praktiknya, proses distribusi yang masih dilakukan secara manual sering kali menimbulkan kendala seperti keterlambatan pencatatan, kesalahan data, serta kurangnya transparansi, sehingga diperlukan sistem yang mampu mengatasi permasalahan tersebut secara lebih terstruktur [21].

Beberapa aktivitas utama dalam distribusi produk meliputi [20]:

1. Penyimpanan produk (pergudangan),
2. Transportasi atau pengiriman produk,
3. Pengemasan dan penanganan produk,
4. Serta pemrosesan pesanan (*order processing*).

Proses distribusi produk perlu dipahami karena berkaitan langsung dengan alur perpindahan produk dari persediaan hingga sampai kepada pelanggan. Setiap produk yang keluar melalui proses penjualan perlu tercatat dengan baik agar perubahan stok dapat diketahui dan proses pemenuhan kebutuhan pelanggan tidak terhambat. Pemahaman terhadap distribusi produk membantu sistem diarahkan untuk mencatat alur produk secara lebih terstruktur serta menjaga keterkaitan antara transaksi penjualan dan ketersediaan stok.

2.2.3 Manajemen Persediaan

Persediaan (*inventory*) merupakan produk atau aset yang dimiliki oleh perusahaan yang memiliki nilai ekonomi dan digunakan untuk mendukung kegiatan operasional maupun untuk memenuhi kebutuhan tertentu. Persediaan dapat berupa bahan baku, produk dalam proses, maupun produk jadi yang berperan dalam menjaga kelancaran aktivitas perusahaan. Manajemen persediaan (*inventory management*) adalah proses pengelolaan, pengawasan,

dan pengendalian terhadap persediaan produk atau aset dalam suatu organisasi. Tujuan utamanya adalah memastikan ketersediaan produk yang cukup untuk memenuhi kebutuhan tanpa menyebabkan kelebihan stok yang dapat meningkatkan biaya penyimpanan. Manajemen persediaan mencakup berbagai kegiatan seperti perencanaan persediaan, pencatatan dan pelacakan produk, pengendalian stok, serta pengadaan dan distribusi produk agar persediaan dapat dikelola secara teratur [22].

Jenis persediaan dalam perusahaan umumnya diklasifikasikan berdasarkan perannya dalam proses operasional [22]. Persediaan bahan baku (*raw materials*) merupakan produk yang digunakan sebagai bahan dasar dalam proses produksi dan belum mengalami pengolahan. Persediaan produk dalam proses (*work in process*) adalah produk yang sedang dalam tahap pengerjaan dan masih memerlukan proses lanjutan sebelum menjadi produk akhir. Adapun persediaan produk jadi (*finished goods*) adalah produk yang telah selesai diproduksi dan siap untuk digunakan atau dipasarkan kepada konsumen.

Pengendalian persediaan (*inventory control*) merupakan proses pengelolaan jumlah produk agar tidak terjadi kelebihan maupun kekurangan stok. Pengendalian ini bertujuan untuk menjaga keseimbangan antara permintaan dan ketersediaan produk dalam perusahaan. Dalam pelaksanaannya, pengendalian dilakukan melalui pencatatan produk masuk dan keluar, pemantauan kondisi stok, serta penentuan batas minimum dan maksimum persediaan. Sistem teknologi informasi memiliki peranan penting dalam mendukung pengendalian persediaan karena dapat membantu proses pencatatan, pemantauan, dan pelaporan persediaan secara *real-time* sehingga mempermudah pengelolaan stok dan meningkatkan efektivitas operasional [22]. Penerapan konsep manajemen persediaan membantu sistem mengurangi risiko selisih stok, mempermudah pemantauan ketersediaan produk, dan mendukung kelancaran proses penjualan.

2.2.4 Piutang Usaha dan Pengelolaannya

Piutang usaha merupakan salah satu aset lancar yang timbul akibat transaksi penjualan produk atau jasa secara kredit kepada pelanggan. Piutang ini mencerminkan hak perusahaan untuk menerima kas di masa mendatang dalam jangka waktu tertentu, sehingga memiliki peran penting dalam mendukung operasional dan arus kas perusahaan. Piutang usaha muncul sebagai konsekuensi dari penjualan kredit yang dilakukan untuk meningkatkan volume penjualan [23]. Piutang usaha juga berpengaruh terhadap kelancaran arus kas perusahaan karena keterlambatan pembayaran dapat menghambat aktivitas operasional [24]. Dalam praktiknya, piutang usaha memiliki risiko yang perlu diperhatikan,

antara lain keterlambatan pembayaran dan piutang tidak tertagih yang dapat memengaruhi kondisi keuangan apabila tidak dikelola dengan baik [25].

Pengelolaan piutang usaha merupakan proses pengendalian terhadap piutang agar dapat ditagih secara tepat waktu, yang mencakup aktivitas pencatatan, pemantauan, dan penagihan piutang [24]. Proses ini diperlukan agar perusahaan dapat mengetahui jumlah piutang, status pembayaran, serta pelanggan yang masih memiliki kewajiban pembayaran. Dengan pengelolaan yang baik, risiko keterlambatan pembayaran dan kesalahan pencatatan piutang dapat diminimalkan.

Dalam penerapannya pada sistem, pengelolaan piutang dilakukan melalui [23]:

1. Pencatatan piutang dari transaksi penjualan
Setiap transaksi penjualan kredit dicatat sebagai piutang agar data dapat terpantau dengan baik.
2. Pemantauan status piutang pelanggan
Pemantauan dilakukan untuk mengetahui kondisi piutang, baik yang belum jatuh tempo maupun yang sudah melewati batas waktu pembayaran.
3. Penagihan piutang
Penagihan dilakukan untuk memastikan pelanggan melakukan pembayaran sesuai dengan ketentuan.
4. Pengendalian data piutang
Pengendalian diperlukan untuk menjaga keakuratan data dan mengurangi kesalahan pencatatan.

Selain itu, evaluasi piutang dilakukan melalui analisis umur piutang dan penilaian kemungkinan piutang tidak tertagih agar perusahaan dapat meminimalkan risiko piutang bermasalah serta menjaga kelancaran arus kas [25]. Pengelolaan piutang diperlukan agar transaksi penjualan kredit dapat dipantau secara jelas dan tidak hanya berhenti pada pencatatan tagihan. Informasi mengenai jumlah tagihan, pembayaran yang telah diterima, sisa piutang, dan status pembayaran menjadi dasar penting dalam mengetahui kewajiban pelanggan. Konsep piutang usaha membantu sistem diarahkan untuk menelusuri data piutang, memantau pembayaran pelanggan, serta mendukung proses penagihan secara lebih teratur.

2.2.5 Laporan Keuangan

Laporan keuangan merupakan sarana penyampaian informasi keuangan suatu entitas usaha yang menggambarkan keberlangsungan dan keberhasilan operasional bisnis dalam

periode tertentu. Lebih dari sekadar dokumen pencatatan, laporan keuangan berfungsi sebagai indikator pengambilan keputusan manajerial yang meliputi penetapan harga, pengembangan bidang usaha, pemasaran, hingga pengendalian aset. Dalam konteks UMKM, laporan keuangan yang berkualitas sangat ditentukan oleh kualitas sistem pencatatan dan pengolahan data keuangan yang digunakan. Namun demikian, sebagian besar pelaku UMKM di Indonesia masih menghadapi keterbatasan dalam penyusunan laporan keuangan yang transparan dan akuntabel akibat rendahnya literasi akuntansi serta penggunaan sistem pembukuan manual yang belum terstandarisasi [26]. Kondisi ini berdampak pada rendahnya kualitas laporan keuangan yang menghambat pelaku usaha dalam melakukan evaluasi kinerja dan pengambilan keputusan secara tepat [27].

Perusahaan dagang termasuk UMKM memerlukan beberapa jenis laporan keuangan utama yang dapat dihasilkan secara otomatis melalui implementasi sistem pembukuan digital. Laporan laba rugi menyajikan hasil kinerja keuangan dalam suatu periode tertentu, mencerminkan selisih antara pendapatan dari kegiatan penjualan dengan beban-beban yang dikeluarkan, sehingga pemilik usaha dapat mengetahui kondisi laba atau rugi tanpa mengandalkan perkiraan semata [27]. Laporan posisi keuangan atau neraca menggambarkan kondisi keuangan pada suatu titik waktu tertentu, mencakup posisi aset, kewajiban, dan ekuitas pemilik usaha sebagai dasar penilaian kemampuan perusahaan memenuhi kewajibannya. Laporan arus kas melaporkan pergerakan kas masuk dan keluar dari seluruh aktivitas operasional, investasi, dan pendanaan selama satu periode, yang sangat krusial bagi usaha dagang dan distribusi untuk memantau likuiditas dan merencanakan alokasi keuangan. Selain ketiga laporan utama tersebut, perusahaan dagang juga membutuhkan laporan pendukung berupa data persediaan produk dan status piutang pelanggan sebagai bagian integral dari sistem informasi akuntansi yang terintegrasi [26].

Dalam era transformasi digital, digitalisasi laporan keuangan menjadi kebutuhan strategis bagi UMKM agar mampu bersaing dan berkembang secara berkelanjutan. Digitalisasi memberikan manfaat signifikan berupa peningkatan efisiensi operasional dan kemudahan akses terhadap informasi keuangan secara *real-time*, sehingga pemilik usaha tidak lagi harus mencatat transaksi secara manual [27]. Penerapan sistem informasi akuntansi berbasis digital termasuk yang berbasis *website* terbukti memberikan pengaruh positif terhadap perkembangan UMKM karena mampu memperbaiki efisiensi proses bisnis, menghasilkan informasi yang akurat dan andal, serta mendukung pengambilan keputusan yang lebih rasional [28]. Digitalisasi pembukuan tidak semata-mata dipahami sebagai penggunaan perangkat lunak, melainkan sebagai proses perubahan pola pikir pelaku UMKM

dalam mengelola keuangan secara lebih profesional. Transparansi dan akuntabilitas laporan keuangan meningkat secara signifikan karena setiap transaksi terdokumentasi secara digital, sekaligus meningkatkan kepercayaan pihak perbankan dan mitra usaha sehingga membuka peluang lebih besar bagi UMKM untuk mengakses pembiayaan formal [26].

Penyajian informasi keuangan diperlukan agar kondisi usaha dapat dipantau secara lebih jelas dan terarah. Data penjualan, pembayaran, piutang, pendapatan, dan pengeluaran perlu diolah secara terstruktur agar menghasilkan informasi keuangan yang mudah dipahami. Konsep laporan keuangan membantu sistem diarahkan untuk menyajikan informasi mengenai pendapatan, pengeluaran, keuntungan, dan kerugian sebagai bahan pemantauan dan pengambilan keputusan usaha.

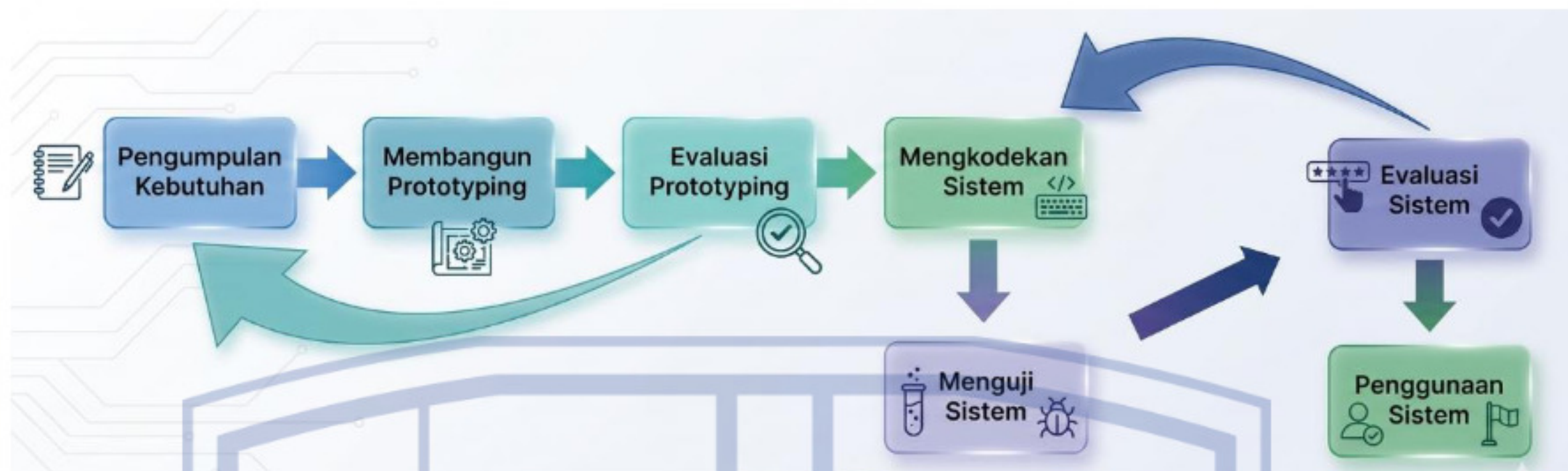
2.3 Metode *prototype*

Metode *prototype* merupakan salah satu metode dalam siklus hidup pengembangan sistem yang didasarkan pada konsep model kerja (*working model*). Menurut [29], model *prototype* adalah suatu teknik yang dapat digunakan untuk mengumpulkan, merancang, dan membangun sistem secara cepat berdasarkan kebutuhan informasi yang ada, di mana hasil desain disajikan kepada klien dalam bentuk prototipe sistem untuk kemudian dievaluasi dan dijadikan referensi pengembangan lebih lanjut. Senada dengan hal tersebut, [30] mendefinisikan metode *prototype* sebagai teknik pengembangan sistem yang menggunakan model sederhana dari sistem yang akan dibangun, sehingga memungkinkan pengembang untuk mengevaluasi dan mengoptimalkan sistem sebelum diimplementasikan secara utuh.

Penelitian [31] juga menjelaskan bahwa metode *prototype* mengutamakan model bekerja (*working model*) sebagai acuan dasar pengembangan. Penerapan metode ini dinilai tepat karena memiliki beberapa kelebihan, di antaranya: (1) terjalinnya komunikasi yang baik antara pengembang dan pengguna, (2) pengembang dapat bekerja lebih optimal dalam menentukan kebutuhan sistem, (3) pengguna berperan aktif selama proses pengembangan, serta (4) lebih efisien dalam hal waktu pengembangan karena pengguna telah mendapatkan gambaran awal dari *prototype* yang dihasilkan. Kelebihan tersebut menjadikan metode *prototype* sesuai digunakan dalam pengembangan sistem yang membutuhkan evaluasi dan penyesuaian berdasarkan kebutuhan pengguna.

Metode *prototype* dilaksanakan secara bertahap, mulai dari pengembangan prototipe sederhana hingga sistem perangkat lunak yang lengkap dan siap digunakan [29]. Setiap tahapan dalam metode ini memberikan ruang bagi pengguna untuk memberikan masukan

terhadap rancangan sistem. Masukan tersebut kemudian digunakan sebagai dasar perbaikan sebelum sistem diimplementasikan.



Gambar 2.2 Tahapan Metode *prototype* [30]

Berdasarkan [30], tahapan dalam proses *prototyping* adalah sebagai berikut:

1. **Pengumpulan Kebutuhan**
Pengembang dan pengguna berkolaborasi untuk merumuskan keseluruhan format perangkat lunak, menentukan semua kebutuhan yang diperlukan, serta menetapkan sistem yang akan dikembangkan.
2. **Membangun *prototype***
Tahap ini melibatkan pembuatan desain sementara yang berfokus pada penyajian kepada pengguna, mencakup pembuatan *input* dan format *output* yang dapat diujicobakan, sehingga pengguna dapat memperoleh gambaran umum mengenai cara kerja perangkat lunak.
3. **Evaluasi *prototype***
Pengguna mengevaluasi prototipe yang telah dibangun untuk memastikan kesesuaiannya dengan kebutuhan dan harapan. Apabila belum sesuai, *prototype* akan direvisi dengan mengulangi tahap-tahap sebelumnya.
4. **Mengkodekan Sistem**
Prototipe yang telah disetujui ditransformasikan ke dalam kode pemrograman yang sesuai.
5. **Menguji Sistem**
Setelah perangkat lunak selesai dikembangkan, dilakukan pengujian sebelum sistem digunakan, mencakup berbagai metode seperti *white box*, *black box*, dan pengujian arsitektur [31].

6. Evaluasi Sistem

Sistem kembali dievaluasi oleh pengguna untuk memastikan bahwa seluruh kebutuhan telah terpenuhi. Jika masih terdapat kekurangan, sistem diperbaiki dan diuji ulang.

7. Penggunaan Sistem

Setelah perangkat lunak diterima oleh pengguna, sistem siap diimplementasikan dan digunakan secara penuh.

2.4 Teknologi yang Digunakan dalam Pengembangan Sistem

Pengembangan sistem informasi berbasis *web* pada CV Jayanti Muliatama membutuhkan dukungan teknologi yang tepat agar sistem yang dibangun dapat berjalan secara optimal dan sesuai dengan kebutuhan operasional perusahaan [32]. Pemilihan teknologi dalam penelitian ini didasarkan pada hasil analisis kebutuhan sistem yang meliputi pengelolaan data produk, data pelanggan, transaksi penjualan, piutang pelanggan, pencatatan keuangan, serta penyajian laporan secara lebih cepat, efektif, dan efisien [33]. Oleh karena itu, teknologi yang digunakan harus mampu mendukung proses pengolahan data dan penyajian informasi secara terstruktur.

Teknologi yang digunakan dalam pengembangan sistem ini memanfaatkan platform berbasis *web*, *framework backend* Laravel untuk mengelola proses dan logika sistem, MySQL sebagai sistem manajemen basis data relasional, React.js untuk membangun antarmuka pengguna, Inertia.js untuk menghubungkan sisi *backend* dan *frontend*, Tailwind CSS untuk mendukung penyusunan tampilan antarmuka, serta Figma untuk perancangan desain awal sistem [34-35]. Melalui penerapan teknologi tersebut, sistem yang dikembangkan diharapkan dapat membantu perusahaan dalam mengelola data dan memperoleh informasi yang dibutuhkan dalam kegiatan operasional [33]. Pemilihan teknologi tersebut disesuaikan dengan kebutuhan sistem agar proses pengembangan dapat dilakukan secara lebih terstruktur.

2.4.1 Website

Website merupakan sistem berbasis *web* yang dapat diakses oleh pengguna melalui *browser*, sehingga pengguna tidak perlu melakukan instalasi khusus pada setiap perangkat yang digunakan. Karakteristik ini menjadikan *website* sebagai media yang praktis untuk mendukung penggunaan sistem dan akses informasi pada berbagai perangkat [36]. Kemudahan akses tersebut menjadikan *website* sesuai digunakan sebagai media penerapan sistem informasi.

Dalam penerapannya, *website* bekerja dengan konsep *client-server*, yaitu pengguna mengakses sistem dari sisi *client*, sedangkan pemrosesan data dan layanan sistem dijalankan pada sisi *server*. Pola kerja ini memungkinkan pertukaran data dan penyajian informasi berlangsung dalam satu sistem yang saling terhubung, sehingga mendukung pengelolaan data secara lebih terstruktur [37]. Konsep ini mendukung proses komunikasi antara pengguna dan sistem dalam pengolahan data.

Pemanfaatan *website* dalam sistem informasi memberikan kemudahan dalam menyatukan fungsi sistem dan penyajian informasi dalam satu media. Dengan demikian, *website* sesuai digunakan sebagai dasar penerapan sistem informasi pada perusahaan [38]. Hal ini mendukung penggunaan *website* sebagai media utama dalam sistem yang dikembangkan.

2.4.2 Framework Laravel

Laravel merupakan *framework* berbasis PHP yang digunakan dalam pengembangan aplikasi *web*. *Framework* ini didukung oleh arsitektur *Model View Controller* (MVC), yaitu pola pengembangan yang membagi aplikasi ke dalam tiga bagian utama, yakni *Model* untuk pengelolaan data, *View* untuk tampilan antarmuka, dan *Controller* untuk mengatur alur proses di antara keduanya. Dengan arsitektur tersebut, proses pengembangan sistem dapat dilakukan dengan struktur yang lebih jelas dan terorganisasi [39].

Dalam pengembangan sistem berbasis *web*, Laravel digunakan pada sisi *backend* untuk mendukung pengelolaan proses aplikasi dan pengolahan data. *Framework* ini juga berperan dalam mengatur *routing*, menerima dan memproses *request* dari pengguna, serta menjalankan logika sistem agar alur kerja aplikasi dapat berlangsung secara lebih terstruktur [40]. Penggunaan Laravel pada aplikasi berbasis *web* menunjukkan bahwa sistem yang dibangun dapat digunakan dengan mudah, berjalan sesuai fungsi yang dirancang, dan mampu memenuhi kebutuhan pengguna [41].

2.4.3 MySQL (Database Management System)

MySQL merupakan sistem manajemen basis data yang termasuk ke dalam *Relational Database Management System* (RDBMS). MySQL memanfaatkan bahasa *Structured Query Language* (SQL) untuk melakukan pengambilan dan perubahan data secara sistematis, seperti *select*, *insert*, *update*, dan *delete*. Dengan karakteristik tersebut, MySQL dapat mendukung pengelolaan data secara terstruktur pada sistem informasi [42].

MySQL juga mendukung penyusunan basis data relasional melalui tahapan perancangan konseptual, logikal, dan fisik. Dalam implementasinya, MySQL memungkinkan pembentukan *database* dan tabel dengan penentuan *field*, tipe data, lebar *field*, serta *Primary Key* yang disusun sesuai prinsip basis data relasional. Susunan tersebut membantu hubungan antardata menjadi lebih jelas dan terorganisasi [43].

Pada sistem berbasis *web*, MySQL digunakan sebagai basis data untuk mendukung pengolahan data secara efisien dan aman. Peran tersebut menjadikan MySQL sesuai digunakan sebagai media pengelolaan data pada sistem yang dikembangkan, terutama karena kebutuhan data pada aplikasi perlu ditangani secara terpusat dan mendukung proses kerja sistem secara berkelanjutan [44]. Oleh karena itu, MySQL menjadi komponen penting dalam pengelolaan basis data pada sistem yang dikembangkan.

2.4.4 React.js

React.js merupakan *library* JavaScript yang digunakan untuk membangun antarmuka pengguna pada sisi *frontend*. React.js menerapkan pendekatan berbasis komponen, sehingga elemen tampilan pada aplikasi dapat disusun ke dalam bagian-bagian yang lebih kecil dan digunakan kembali sesuai kebutuhan pengembangan sistem [45]. Dengan pendekatan tersebut, tampilan aplikasi dapat dikembangkan secara lebih fleksibel.

Dalam pengembangan *website*, React.js mendukung penyusunan antarmuka yang lebih interaktif dan terstruktur. Penerapannya pada *frontend* memudahkan pengembangan tampilan aplikasi sesuai fungsi yang dirancang, sedangkan pada pendekatan *Single Page Application* (SPA), React.js membantu proses interaksi pengguna berlangsung dengan lebih efisien karena perubahan tampilan tidak selalu memerlukan pemuatan ulang halaman secara penuh [40,46]. Kemampuan tersebut membantu sistem menampilkan data dan fitur secara lebih cepat kepada pengguna.

2.4.5 Inertia.js

Inertia.js merupakan *library* yang digunakan untuk menghubungkan sisi *server-side* dan *client-side* dalam pengembangan aplikasi *web*. Inertia.js tidak digunakan sebagai *framework* utama, melainkan sebagai penghubung antara Laravel pada sisi *server* dan *framework* pada sisi klien, sehingga pengembangan aplikasi dapat dilakukan tanpa membangun *Application Programming Interface* (API) secara terpisah [47]. Penggunaan Inertia.js membantu pengembangan aplikasi menjadi lebih sederhana karena proses komunikasi data dapat dilakukan secara langsung.

Dalam penerapannya, Laravel tetap digunakan untuk menangani proses pada sisi *server*, sedangkan React digunakan untuk membangun tampilan pada sisi klien. Inertia.js berperan dalam meneruskan data dan hasil proses dari Laravel ke komponen React, sehingga data yang diproses pada *backend* dapat langsung digunakan pada tampilan aplikasi [40]. Dukungan tersebut membantu sistem menampilkan data dari *server* ke halaman pengguna secara lebih praktis.

Penerapan Inertia.js bersama Laravel dan React juga telah digunakan pada sistem berbasis *web* yang responsif dan terintegrasi. Integrasi tersebut mendukung pembentukan aplikasi yang lebih terstruktur, memudahkan pengelolaan data, serta membantu penyajian informasi berjalan dengan lebih baik [48]. Hal ini menunjukkan bahwa Inertia.js dapat mendukung pengembangan sistem yang terhubung antara sisi proses dan tampilan.

2.4.6 Tailwind CSS

Tailwind CSS merupakan *framework* CSS yang digunakan pada sisi *front-end* untuk membantu penyusunan tampilan *website*. Tailwind CSS menyediakan pendekatan berbasis *utility classes* yang memberi keleluasaan dalam mengatur *style* dan *layout* sesuai kebutuhan desain, sehingga mendukung pembentukan antarmuka yang lebih fleksibel [49]. Pendekatan ini memudahkan pengembang dalam menyesuaikan tampilan sistem sesuai rancangan antarmuka.

Penggunaan Tailwind CSS juga mendukung penyusunan tampilan yang responsif dan selaras dengan rancangan visual *website*. Penerapannya pada pengembangan *website* dan sistem berbasis *web* menunjukkan bahwa Tailwind CSS dapat digunakan untuk membentuk antarmuka yang rapi, menarik, dan sesuai dengan kebutuhan tampilan aplikasi [50,51]. Kemampuan tersebut menjadikan Tailwind CSS sesuai digunakan dalam pengembangan tampilan sistem informasi.

2.4.7 Figma

Figma merupakan alat bantu desain antarmuka yang digunakan dalam proses perancangan *user interface* (UI) dan *user experience* (UX). Penggunaan Figma membantu pengembang dalam merancang antarmuka sistem agar sesuai dengan kebutuhan pengguna dan tujuan pengembangan sistem [52]. Hal ini menjadikan Figma sesuai digunakan sebagai alat bantu dalam tahap perancangan antarmuka sistem.

Figma digunakan untuk menyusun *wireframe* sebagai gambaran awal struktur halaman sebelum antarmuka direalisasikan ke dalam kode. Melalui *wireframe*, susunan

menu, konten, dan navigasi dapat direncanakan lebih awal sehingga rancangan halaman menjadi lebih terarah [53]. Perancangan *wireframe* juga membantu mengurangi kesalahan dalam penyusunan antarmuka sistem.

Selain digunakan untuk merancang antarmuka, Figma juga mendukung pembuatan *prototype* sistem. Melalui *prototype*, rancangan dapat divisualisasikan dalam bentuk alur dan tampilan yang lebih jelas, sehingga memudahkan pengembang dalam memahami gambaran sistem yang akan dibangun [54]. Dengan demikian, Figma berperan dalam mendukung tahap perancangan dan evaluasi awal sistem.

2.5 Tinjauan Objek Penelitian: CV Jayanti Muliatama

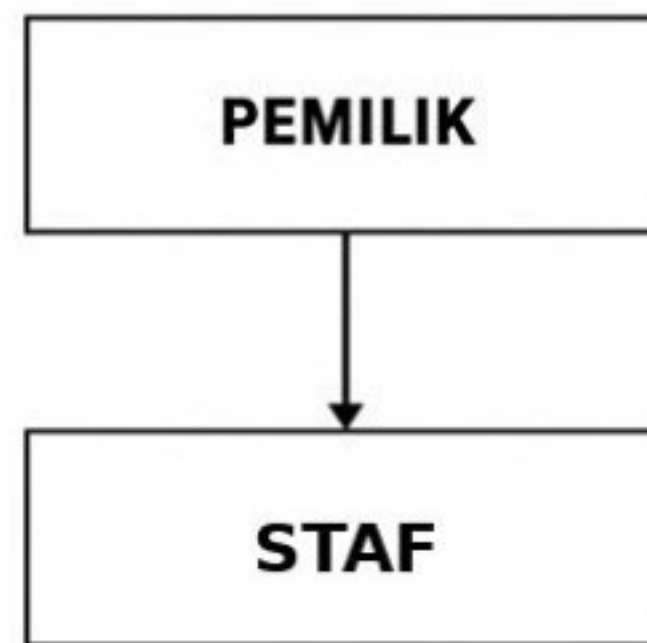
Objek penelitian ini adalah CV Jayanti Muliatama. Tinjauan objek penelitian diperlukan untuk memberikan gambaran awal mengenai kondisi perusahaan sebagai tempat penerapan sistem. Gambaran tersebut menjadi dasar dalam memahami kebutuhan pengembangan sistem informasi.

Pembahasan mengenai objek penelitian ini meliputi profil perusahaan, proses bisnis saat ini, identifikasi aktor, studi kasus pengelolaan operasional, serta arsitektur sistem informasi yang diusulkan. Uraian tersebut digunakan untuk memperjelas keterkaitan antara kondisi perusahaan dan sistem yang dirancang. Dengan demikian, pengembangan sistem dapat diarahkan sesuai dengan kebutuhan CV Jayanti Muliatama.

2.5.1 Profil Singkat CV Jayanti Muliatama

CV Jayanti Muliatama merupakan perusahaan distributor peralatan dan perlengkapan hidrolik yang berlokasi di Jl. Karya Sehati No. 20, Kota Medan. Perusahaan ini menyediakan berbagai kebutuhan komponen hidrolik yang digunakan dalam bidang industri, manufaktur, dan alat berat, dengan salah satu produk utama berupa *hydraulic hose*. Dalam kegiatan operasionalnya, perusahaan melakukan pembelian produk, pengelolaan stok, penjualan kepada pelanggan, serta pengelolaan piutang pelanggan.

Untuk mendukung kegiatan operasional tersebut, CV Jayanti Muliatama memiliki struktur organisasi yang menggambarkan pembagian peran di dalam perusahaan. Struktur organisasi diperlukan untuk mengetahui pihak yang bertanggung jawab dalam menjalankan aktivitas perusahaan secara umum. Struktur organisasi CV Jayanti Muliatama ditunjukkan pada Gambar 2.3.



Gambar 2.3 Struktur Organisasi CV Jayanti Muliatama

2.5.2 Proses Bisnis Saat Ini dan Identifikasi Aktor

Dalam struktur organisasi, kegiatan operasional pada CV Jayanti Muliatama berada di bawah tanggung jawab pemilik dan dibantu oleh karyawan dalam pelaksanaannya. Sistem yang akan dikembangkan melibatkan beberapa aktor kunci dari pihak internal perusahaan maupun eksternal yang berperan dalam sistem tersebut. Identifikasi aktor diperlukan untuk memperjelas peran dan hak akses masing-masing pengguna dalam sistem.

Aktor-aktor yang terlibat meliputi:

1. Admin
 - a. Peran: Aktor internal yang memiliki hak akses penuh terhadap sistem.
 - b. Tugas Utama: Mengelola data produk, data pelanggan, transaksi pembelian, transaksi penjualan, piutang pelanggan, serta laporan yang mencakup stok, penjualan, piutang, dan keuangan dengan hak akses penuh terhadap seluruh fitur sistem.
2. Staf Operasional
 - a. Peran: Aktor internal yang membantu kegiatan operasional perusahaan.
 - b. Tugas Utama: Melakukan penambahan dan perubahan data produk, data pelanggan, serta mencatat transaksi penjualan.
3. Masyarakat Umum
 - a. Peran: Aktor eksternal yang berperan sebagai pengunjung *website* publik.
 - b. Tugas Utama: Melihat informasi produk dan profil perusahaan

2.5.3 Tinjauan Studi Kasus: Pengelolaan Operasional Stok, Penjualan, dan Keuangan

Salah satu fokus utama dari penelitian ini adalah untuk mendukung kegiatan operasional internal pada CV Jayanti Muliatama, khususnya dalam pengelolaan stok produk, transaksi penjualan, piutang pelanggan, dan pencatatan keuangan. Keempat aspek tersebut

saling berkaitan dalam menunjang kelancaran operasional serta pengambilan keputusan perusahaan. Keterkaitan tersebut menjadi dasar dalam menentukan fitur utama pada sistem yang dirancang.

Sistem yang dirancang merupakan sistem informasi internal yang digunakan untuk membantu aktivitas operasional perusahaan agar lebih terstruktur. Penggunaan sistem internal bertujuan untuk mempermudah admin dan staf operasional dalam menjalankan tugas masing-masing. Hal ini menjadi dasar dalam penyusunan mekanisme sistem yang diusulkan.

Adapun mekanisme sistem yang diusulkan adalah sebagai berikut:

1. Pengelolaan Data Produk: Admin dapat menambahkan, mengubah, menghapus, serta mengatur status aktif atau nonaktif data produk, sedangkan staf operasional hanya dapat menambahkan dan mengubah data produk.
2. Transaksi Penjualan: Admin dan staf operasional dapat mencatat transaksi penjualan yang dilakukan kepada pelanggan.
3. Pengelolaan Piutang Pelanggan: Admin dapat mencatat, memantau jumlah piutang, serta mengelola status pembayaran pelanggan.
4. Penyajian Laporan: Admin dapat melihat laporan stok, penjualan, piutang, dan keuangan.

2.5.4 Arsitektur Sistem Informasi

Arsitektur sistem dibuat untuk mendukung kebutuhan akses yang berbeda antara pengguna umum dan pengguna internal. Perbedaan akses tersebut menjadi dasar dalam pembagian sistem menjadi *website* publik dan *website* internal. Pembagian ini diperlukan agar setiap pengguna dapat mengakses fungsi sistem sesuai dengan perannya.

Untuk memenuhi kebutuhan seluruh aktor dan memisahkan fungsi publik dengan fungsi internal, arsitektur sistem yang dikembangkan dibagi menjadi dua (2) bagian utama, yaitu:

1. *Website* Publik (*Landing Page*): Berfungsi sebagai halaman informasi yang menampilkan profil perusahaan, informasi produk, detail produk, serta kontak CV Jayanti Muliatama. *Website* ini dapat diakses oleh masyarakat umum sebagai media informasi dan pengenalan perusahaan tanpa melalui proses *login*.

2. *Website Internal* (Sistem Informasi): Merupakan sistem yang digunakan oleh admin dan staf operasional melalui proses autentikasi (*login*) untuk mengelola data produk, transaksi penjualan, piutang pelanggan, serta melihat laporan operasional perusahaan.

2.6 Unified Modeling Language

Unified Modeling Language (UML) merupakan suatu standar bahasa pemodelan visual yang diterapkan dalam proses analisis dan perancangan sistem perangkat lunak berorientasi objek. UML berfungsi sebagai sarana untuk menyusun cetak biru (*blueprint*) sistem yang terstandarisasi, mencakup visualisasi alur proses bisnis, pendefinisian kelas-kelas dalam bahasa spesifik, serta penggambaran interaksi antarkomponen di dalam sistem. Keberadaan UML sebagai notasi standar bertujuan untuk menjembatani komunikasi antara pengembang sistem dengan pemangku kepentingan, sehingga meminimalkan ambiguitas dan memastikan kesamaan persepsi terhadap arsitektur sistem yang akan dibangun [55].

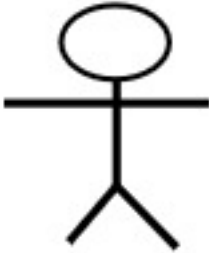
Dalam pengembangan sistem informasi berbasis *web*, terdapat beberapa alat yang digunakan untuk memodelkan sistem sebelum tahap implementasi dilakukan. Pada penelitian ini, pemodelan sistem dilakukan menggunakan dua alat UML, yaitu *use case diagram* dan *activity diagram*. *use case diagram* digunakan untuk menggambarkan interaksi antara aktor dengan fungsionalitas sistem, sedangkan *activity diagram* digunakan untuk menggambarkan aliran kerja sistem serta aliran kejadian yang terjadi antara aktor dan sistem [55].

2.6.1 Use Case Diagram

Use Case Diagram merupakan bagian dari UML (*Unified Modeling Language*) yang berfungsi untuk menggambarkan interaksi antara aktor dengan fungsionalitas yang terdapat dalam suatu sistem [56]. Diagram ini mendeskripsikan "apa" yang dilakukan oleh sistem dari sudut pandang pengguna tanpa merinci aspek teknis implementasinya. Dalam konteks pengembangan sistem berbasis *web*, *use case diagram* dimanfaatkan untuk mengidentifikasi hak akses setiap aktor terhadap fitur-fitur yang tersedia pada sistem, seperti *login*, pengelolaan data, hingga pembuatan laporan [31].

Dalam penyusunan *use case diagram*, penggunaan simbol yang tepat diperlukan agar hubungan antara aktor dan fungsi sistem dapat digambarkan secara jelas. Setiap simbol memiliki makna tertentu yang membantu menjelaskan bentuk interaksi di dalam sistem. Simbol-simbol yang digunakan dalam *use case diagram* ditunjukkan pada Tabel 2.1.

Tabel 2.1 Simbol *Use Case Diagram* [55]

Simbol	Nama	Keterangan
	Aktor	Aktor adalah entitas di luar sistem yang berinteraksi dengan sistem informasi, baik berupa manusia, proses, maupun sistem lain. Aktor merepresentasikan peran yang dijalankan saat berinteraksi dengan sistem dan umumnya diawali dengan kata benda.
	<i>Use Case</i>	<i>Use Case</i> merupakan fungsi atau layanan yang disediakan oleh sistem sebagai bagian yang saling berinteraksi dengan aktor melalui pertukaran pesan. Penamaan <i>use case</i> biasanya diawali dengan kata kerja yang menunjukkan aktivitas yang dilakukan.
	Association atau Asosiasi	Association adalah hubungan komunikasi antara aktor dengan <i>use case</i> yang menunjukkan adanya interaksi atau keterlibatan aktor dalam suatu fungsi sistem.
	Generalization atau Generalisasi	Generalization adalah hubungan antara <i>use case</i> yang bersifat umum dengan <i>use case</i> yang lebih spesifik, di mana <i>use case</i> tertentu merupakan turunan dari <i>use case</i> yang lebih umum.

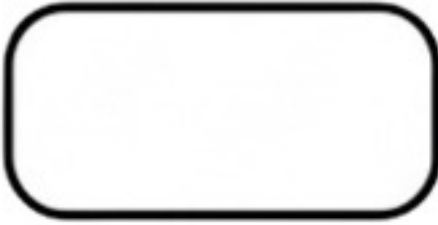
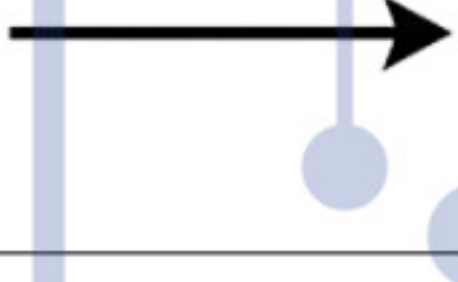
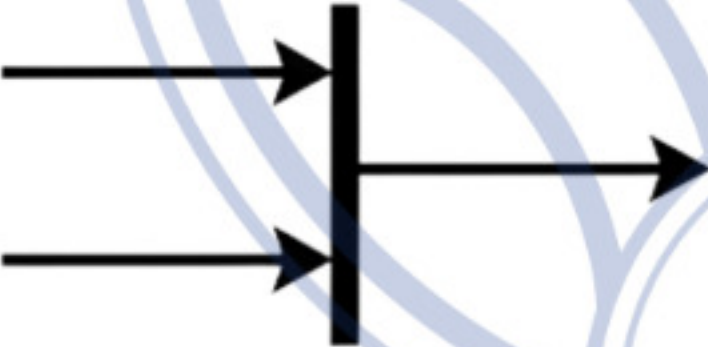
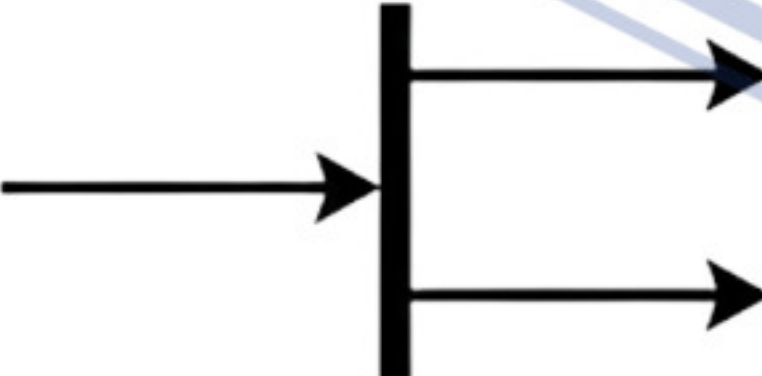
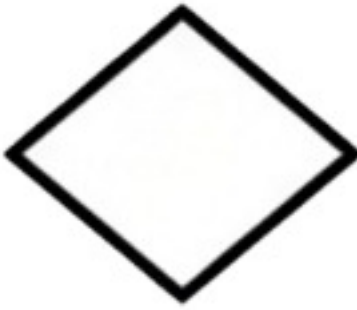
Simbol	Nama	Keterangan
 	Include	Include adalah relasi antar <i>use case</i> di mana suatu <i>use case</i> membutuhkan <i>use case</i> lain sebagai bagian dari prosesnya. <i>Use Case</i> yang disertakan dapat selalu dijalankan atau menjadi prasyarat sebelum <i>use case</i> utama dijalankan.
	Extend	Extend merupakan relasi yang menunjukkan adanya penambahan perilaku pada suatu <i>use case</i> utama, di mana <i>use case</i> tambahan dapat berdiri sendiri tanpa bergantung sepenuhnya pada <i>use case</i> utama.

2.6.2 Activity Diagram

Activity Diagram adalah diagram yang menggambarkan aliran fungsionalitas atau alur kerja (*workflow*) dari suatu sistem. Diagram ini digunakan untuk menunjukkan urutan aktivitas yang terjadi dalam suatu fitur, mulai dari proses dimulai hingga berakhir. Setiap aktivitas dalam diagram ini direpresentasikan secara berurutan sehingga memudahkan pemahaman mengenai alur proses yang berjalan, dan pada tahap pemodelan sistem, *activity diagram* juga dapat digunakan untuk menggambarkan aliran kejadian yang terjadi antara aktor dan sistem [31].

Dalam penyusunan *activity diagram*, simbol-simbol digunakan untuk menunjukkan aktivitas, alur proses, keputusan, serta awal dan akhir suatu proses. Penggunaan simbol yang sesuai membantu alur kerja sistem digambarkan secara lebih jelas dan mudah dipahami. Simbol-simbol yang digunakan dalam *activity diagram* ditunjukkan pada Tabel 2.2.

Tabel 2.2 Simbol *Activity Diagram* [55]

No	Bentuk Simbol	Nama Simbol	Fungsi Simbol
1		Activity	Menyatakan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2		Control Flow	Menunjukkan Urutan Eksekusi.
3		Object Flow	Menunjukkan aliran objek dari sebuah action atau activity ke action.
4		Start Point	Menyatakan bahwa suatu objek dibentuk atau diawali.
5		End Point	Menyatakan bahwa suatu objek dihentikan.
6		Join/Penggabungan	Menyatakan untuk menggabungkan kembali activity atau action yang paralel.
7		Fork	Menyatakan untuk memecah suatu activity menjadi activity atau action yang paralel.
8		Decision	Menunjukkan suatu keputusan/tindakan yang harus diambil pada kondisi tertentu.

2.7 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah diagram yang digunakan dalam tahap perancangan basis data untuk menggambarkan entitas, atribut, dan relasi antardata secara visual. ERD membantu pengembang sistem untuk benar-benar memahami kebutuhan data dan cara kerjanya secara menyeluruh sebelum implementasi [57]. Pemahaman yang tepat terhadap ketiga komponen ERD terbukti berpengaruh terhadap kualitas rancangan basis data yang dihasilkan, serta dapat mengurangi duplikasi data sekaligus memudahkan implementasi basis data sesuai kebutuhan sistem. ERD terdiri dari tiga komponen utama [58]:

1. Entitas (*Entity*)

Entitas adalah objek, ide, atau peristiwa baik yang bersifat nyata maupun abstrak yang datanya perlu disimpan dalam sistem basis data. Entitas umumnya dilambangkan dengan simbol berbentuk persegi panjang dalam diagram ERD. Dalam kehidupan sehari-hari, entitas dapat berupa pelanggan, produk, transaksi, atau pegawai. Entitas dibedakan menjadi dua jenis:

a. Entitas Kuat

Entitas yang dapat berdiri sendiri dan memiliki *primary key* sebagai pengenalan uniknya tanpa bergantung pada entitas lain. Contohnya adalah entitas Pelanggan, Produk, dan Pengguna dalam sistem informasi toko.

b. Entitas Lemah

Entitas yang tidak memiliki identitas sendiri dan keberadaannya bergantung pada entitas kuat. Contohnya adalah Detail Pesanan yang bergantung pada entitas Pesanan.

Pada sistem informasi berbasis *web*, identifikasi entitas dilakukan dengan menganalisis kebutuhan data dari proses bisnis yang berjalan. Sebagai contoh, pada perancangan ERD sistem POS untuk Toko Sembako Istiqomah, terdapat lima entitas utama yaitu *user*, pelanggan, produk, jenis produk, dan penjualan [59].

2. Atribut (*Attribute*)

Atribut merupakan informasi deskriptif yang dimiliki oleh sebuah entitas untuk membedakan satu data dengan data lainnya di dalam sistem. Secara visual, atribut digambarkan dengan simbol berbentuk elips yang terhubung ke entitas yang bersangkutan. Sebagai contoh, entitas Admin dalam sistem e-commerce Ratu Komputer

memiliki atribut seperti id admin, kode admin, nama, *email*, jenis kelamin, *password*, jabatan, nomor telepon, alamat, foto, dan kode cabang [60].

3. Relasi (*relationship*)

Relasi menggambarkan hubungan antara dua entitas atau lebih di dalam basis data. Relasi antarentitas ditentukan berdasarkan hubungan fungsional yang terjadi dalam proses bisnis, dan hubungan ini menjadi dasar pembentukan *foreign key* guna menjaga integritas referensial antartabel [60]. ERD memvisualisasikan relasi antar entitas secara jelas sebelum diterapkan ke dalam struktur tabel basis data yang sesungguhnya [57]. Terdapat tiga jenis kardinalitas relasi dalam ERD, yaitu:

a. Satu ke Satu (One-to-One / 1:1)

Relasi ini terjadi ketika satu kejadian pada entitas pertama hanya dapat berhubungan dengan tepat satu kejadian pada entitas kedua, dan sebaliknya. Jenis relasi ini umumnya digunakan ketika dua entitas memiliki keterkaitan yang sangat erat.

b. Satu Ke Banyak (One-to-Many / 1:N)

Relasi ini merupakan jenis yang paling umum digunakan dalam perancangan basis data, di mana satu entitas dapat berhubungan dengan banyak entitas lain, namun entitas di sisi "banyak" hanya dapat terhubung dengan satu entitas di sisi "satu". Contoh konkretnya dapat ditemukan pada sistem informasi inventaris, di mana satu entitas Cabang dapat memiliki banyak entitas Admin, namun setiap Admin hanya terkait dengan satu Cabang [60].

c. Banyak Ke Banyak (Many-to-Many / M:N)

Relasi ini muncul ketika satu entitas dapat berhubungan dengan banyak entitas lain, dan begitu pula sebaliknya. Dalam implementasi basis data relasional, relasi jenis ini tidak dapat diwujudkan secara langsung sehingga memerlukan entitas perantara (*junction table*) sebagai penghubung. Pada sistem POS, relasi antara entitas Produk dan Penjualan merupakan contoh relasi banyak ke banyak yang di atasi melalui tabel detail transaksi [59].

2.8 Pengujian Sistem (*Black Box Testing*)

Black Box Testing atau disebut juga *Behavioral Testing* merupakan teknik pengujian yang berfokus pada persyaratan fungsional perangkat lunak tanpa memperhatikan struktur internal kode. Pengujian ini dilakukan dengan mengevaluasi apakah keluaran (*output*) yang dihasilkan oleh sistem telah sesuai dengan masukan (*input*) yang diberikan berdasarkan spesifikasi kebutuhan yang telah ditetapkan. Metode ini memungkinkan penguji untuk

menentukan serangkaian kondisi masukan yang sepenuhnya melatih semua persyaratan fungsional suatu program. Sebaliknya, *White Box Testing* adalah teknik pengujian yang memeriksa logika internal, struktur kode, dan alur program secara langsung. Meskipun demikian, *Black Box Testing* bukan merupakan alternatif dari *White Box Testing*, melainkan pendekatan pelengkap yang dapat mengungkap kelas kesalahan yang berbeda. Pengujian dengan metode *Black Box Testing* berupaya menemukan kesalahan dalam beberapa kategori, antara lain fungsi yang salah atau tidak ada, kesalahan antarmuka, kesalahan dalam struktur data atau akses basis data, kesalahan perilaku atau kinerja, serta kesalahan dalam inisialisasi dan terminasi sistem [61,62].

Dalam implementasinya, metode *Black Box Testing* memiliki berbagai teknik dalam pelaksanaannya. Salah satu teknik yang paling banyak digunakan adalah *Equivalence Partitioning*, yaitu teknik pengujian yang membagi domain *input* ke dalam kelas-kelas ekuivalen yang berbeda untuk mengurangi jumlah kasus pengujian sekaligus memastikan cakupan yang komprehensif. Teknik ini dilakukan dengan membagi *input* menjadi dua bagian, yaitu *input* yang *valid* dan *invalid*, untuk memastikan bahwa *output* yang dihasilkan sesuai dengan kasus pengujian yang dilakukan. Beberapa kelebihan teknik ini antara lain memberikan cakupan domain *input* yang luas dengan memilih subset yang lebih kecil dari setiap kelas ekuivalen, serta mencegah terjadinya duplikasi pengujian [63].

Tahapan pengujian *Black Box Testing* dilakukan dengan menentukan fitur yang akan diuji, menyusun skenario pengujian, menentukan data masukan, menjalankan pengujian, serta mencatat hasil pengujian. Setiap skenario pengujian disusun berdasarkan fungsi sistem yang telah ditentukan, kemudian keluaran yang dihasilkan sistem dibandingkan dengan keluaran yang diharapkan. Tujuan utama dari *Black Box Testing* adalah untuk mengidentifikasi kesalahan seperti fungsi yang tidak berjalan sebagaimana mestinya, ketidaksesuaian antara *input* dan *output*, serta berbagai bentuk kesalahan lainnya yang dapat mengganggu operasional sistem [62]. Pengujian dilakukan terhadap seluruh proses yang ada pada sistem, mulai dari proses autentikasi (*login*), pengelolaan data, hingga pembuatan laporan, guna memastikan keseluruhan fitur utama berfungsi secara optimal [36]. Sebagai bagian dari dokumentasi penjaminan mutu, hasil pengujian disajikan dalam bentuk tabel *test case* yang memuat kolom fitur yang diuji, skenario pengujian, data masukan, keluaran yang diharapkan (*expected result*), keluaran yang dihasilkan (*actual result*), serta status pengujian (*pass/fail*) [64]. Sistem dinyatakan layak digunakan apabila seluruh skenario pengujian menghasilkan keluaran yang valid dan sesuai dengan spesifikasi kebutuhan [18,65].

2.9 User Acceptance Testing (UAT)

User Acceptance Testing (UAT) atau pengujian penerimaan pengguna merupakan tahap pengujian akhir yang bertujuan untuk memastikan bahwa sistem yang dibangun telah memenuhi kebutuhan dan harapan pengguna akhir. UAT dilakukan setelah sistem lolos dari pengujian fungsional (*alpha testing*) dan melibatkan pengguna secara langsung untuk mengevaluasi kesesuaian sistem dengan proses bisnis nyata. Menurut Pressman (2010), UAT merupakan pengujian yang dilakukan oleh pengguna akhir untuk memvalidasi bahwa perangkat lunak siap digunakan di lingkungan produksi [66].

Tujuan utama UAT adalah untuk memperoleh konfirmasi dari pengguna bahwa sistem telah berfungsi sesuai dengan kebutuhan operasional dan memberikan manfaat yang diharapkan. Pengujian ini juga berfungsi untuk mengidentifikasi potensi kekurangan atau ketidaksesuaian yang mungkin terlewat pada tahap pengujian sebelumnya. Dalam konteks pengembangan sistem informasi untuk UMKM, UAT menjadi tahap krusial karena pengguna akhir adalah pihak yang paling memahami alur kerja dan kebutuhan bisnis sehari-hari [66].

Pengukuran tingkat penerimaan pengguna dalam *User Acceptance Testing* (UAT) dilakukan dengan pendekatan deskriptif kuantitatif menggunakan instrumen kuesioner berbasis skala Likert lima tingkat, mulai dari bobot 1 (Tidak Setuju) hingga 5 (Sangat Setuju). Evaluasi UAT dapat mencakup empat aspek utama, yaitu kemudahan penggunaan (*usability*), kepuasan pengguna (*user satisfaction*), fungsionalitas sistem (*functionality*), dan kinerja sistem (*system performance*) [67].

Analisis data dilakukan dengan menghitung skor aktual, yaitu total bobot jawaban responden, kemudian dibandingkan dengan skor ideal yang diperoleh dari jumlah responden, jumlah pernyataan, dan bobot tertinggi. Tingkat penerimaan sistem dihitung menggunakan Persamaan (2.1) [67].

$$\text{Persentase UAT} = \left(\frac{\text{Skor Aktual}}{\text{Skor Ideal}} \right) \times 100\% \quad (2.1)$$

Keterangan:

Persentase UAT = persentase tingkat penerimaan pengguna terhadap sistem.

Skor Aktual = total skor yang diperoleh dari hasil penilaian responden.

Skor Ideal = skor maksimum yang dapat diperoleh berdasarkan jumlah responden, jumlah pernyataan, dan bobot tertinggi.

Selain persentase penerimaan, nilai rata-rata hasil penilaian UAT dihitung menggunakan Persamaan (2.2).

$$\text{Nilai Rata-rata} = \frac{\text{Total Skor Aktual}}{\text{Jumlah Responden} \times \text{Jumlah Pernyataan}} \quad (2.2)$$

Keterangan:

Nilai Rata-rata = rata-rata skor penilaian UAT.

Total Skor Aktual = jumlah seluruh skor yang diperoleh dari seluruh responden.

Jumlah Responden = jumlah responden yang mengikuti pengujian UAT.

Jumlah Pernyataan = jumlah pernyataan yang dinilai oleh setiap responden.

Hasil persentase tersebut kemudian dikelompokkan ke dalam kriteria kelayakan sistem, di mana nilai 81%–100% termasuk kategori Sangat Layak / Sangat Diterima, 61%–80% Layak / Diterima, 41%–60% Cukup Layak, 21%–40% Kurang Layak, dan 0%–20% Tidak Layak [67]. Penggunaan UAT berbasis skala Likert dapat memberikan gambaran kuantitatif mengenai tingkat penerimaan pengguna terhadap sistem.