

BAB II

KAJIAN PUSTAKA

A. Kajian Teoritis

1. Sistem

Sistem merupakan sekumpulan komponen atau elemen yang saling berinteraksi dan bekerja sama untuk mencapai tujuan tertentu (Santi et al., 2023). Setiap komponen dalam sistem memiliki fungsi masing-masing dan saling terhubung melalui proses *Input*, proses, dan output. Dalam konteks organisasi, sistem digunakan untuk mendukung berbagai aktivitas operasional agar dapat berjalan secara terstruktur, efektif, dan efisien. Konsep tersebut sesuai dengan sistem informasi pada UD Ayam Berkah, dimana data pembelian ayam, setoran *Outlet*, dan biaya operasional menjadi *Input* yang diproses oleh sistem sehingga menghasilkan output berupa laporan keuangan dan perhitungan laba rugi. Perancangan sistem perlu memperhatikan hubungan antar bagian yang membentuk sistem agar proses pengembangan dapat digambarkan secara lebih terstruktur dan menyeluruh melalui pemodelan UML. Menurut Sunarsa et al., (2024) sistem juga memerlukan sdm yang mumpuni untuk mengoperasikannya agar tercipta konsep sistem yang sinergis . Berdasarkan uraian tersebut, dapat disimpulkan bahwa sistem adalah sekumpulan elemen yang saling berhubungan dan bekerja secara terpadu untuk mengolah masukan menjadi keluaran yang mendukung tercapainya tujuan tertentu.

2. Sistem Informasi

Sistem informasi merupakan suatu sistem yang terdiri dari komponen-komponen yang saling berinteraksi dalam mengolah data menjadi informasi yang memiliki nilai guna bagi organisasi maupun pengguna. Menurut Dita Nurmadewi, (2025:8) di era bisnis modern, sistem informasi memegang peranan krusial untuk mengelola data mulai dari pengumpulan hingga pendistribusiannya sehingga manajemen dapat mengambil keputusan strategis dengan lebih efektif.. Menurut Bratha,W,G,E., (2022) Sistem Informasi Manajemen (SIM) mengintegrasikan perangkat keras, perangkat lunak, basis data, prosedur, dan pengguna untuk menghasilkan informasi akurat yang membantu manajemen mengambil keputusan strategis. Selain itu, penerapan sistem informasi berbasis digital mampu meningkatkan efektivitas pengelolaan data dan efisiensi proses bisnis pada organisasi modern (Wisnu, 2024). Berdasarkan uraian tersebut, dapat disimpulkan bahwa sistem informasi adalah sistem terintegrasi yang berfungsi mengolah data menjadi informasi guna mendukung aktivitas operasional organisasi secara efektif.

3. Sistem Informasi Manajemen

Sistem Informasi Manajemen (SIM) merupakan sebuah sistem prosedural yang terintegrasi antara manusia dan mesin berbasis komputer, yang berfungsi untuk mengumpulkan, mengolah, menyimpan, dan menyajikan data operasional menjadi informasi yang komprehensif. Dalam tatanan akademis, SIM tidak hanya berfungsi sebagai alat pencatatan elektronik, melainkan sebagai instrumen strategis yang menyediakan data

historis dan aktual guna mendukung fungsi perencanaan, pengawasan, koordinasi, serta pengambilan keputusan strategis (*decision making*) bagi pihak manajemen atau pemilik usaha (Bratha,W,G,E., 2022). Implementasi Sistem Informasi Manajemen Keuangan umumnya dirancang untuk memusatkan semua pencatatan keuangan secara *real-time* untuk meningkatkan efisiensi, akurasi, dan struktur operasional yang ada. Sehubungan dengan implementasi tersebut diharapkan sistem dapat berkembang dengan terintegrasi sehingga dapat meminimalisir kesalahan dan memudahkan pelaku usaha dalam melakukan monitoring (Jibran et al., 2025).

Penyusunan landasan teori serta arah Pembangunan SIM dalam penelitian ini diperkuat oleh beberapa kajian ilmiah teoretis dari penelitian terdahulu yang relevan. Kompleksitas operasional pada bisnis yang memiliki banyak cabang memerlukan sistem pengolahan data yang tersentralisasi agar tidak terjadi fragmentasi informasi. Sebagaimana dikaji oleh dalam penelitiannya mengenai sistem manajemen berbasis *web*, implementasi SIM mampu mengotomatisasi aliran data operasional, stok, dan pengeluaran harian yang awalnya terpisah secara manual menjadi terpusat dalam satu sistem tunggal. Pada sistem yang dirancang untuk UD. Ayam Berkah, teori integrasi operasional ini diterapkan secara langsung pada modul logistik *multi-outlet*, di mana *Admin* dapat mengelola data manifest pengiriman, informasi keuangan secara terpusat guna mengeliminasi terjadinya kerancuan dan kesalahan dalam pelaporan

keuangan dari setiap *Outlet* .

Selain masalah logistik, aspek krusial lain dalam keberlangsungan usaha distribusi adalah pengawasan keuangan yang ketat terhadap biaya-biaya variabel di lapangan. Penggunaan sistem informasi manajemen berbasis *website* terbukti mampu mendigitalisasi pencatatan konvensional menjadi otomatis, sehingga meningkatkan efisiensi operasional dan mempercepat pembuatan laporan penjualan terstruktur bagi pemilik usaha untuk memonitor bisnisnya, serta mengurangi potensi human error (Isma et al., 2023). Dalam penelitian ini, teori efisiensi operasional tersebut diimplementasikan untuk memonitor aliran kas masuk (*cash-inflow*) berupa setoran harian dari setiap *Outlet*, serta aliran kas keluar (*cash-outflow*) yang meliputi biaya operasional gudang, biaya bahan bakar minyak (BBM), dan uang makan supir. Melalui integrasi ini, SIM berfungsi sebagai alat kontrol manajerial yang valid bagi *Owner* untuk memantau performa finansial seluruh cabang dan mencegah terjadinya kebocoran dana (*fraud*).

Sebuah Sistem Informasi Manajemen dinyatakan berhasil diimplementasikan apabila memiliki akurasi pengolahan data yang tinggi serta dikembangkan dengan metode yang adaptif terhadap kebutuhan pengguna. Penerapan pendekatan *Agile* dalam rancang bangun sistem informasi manajemen keuangan berbasis *web* sangat efektif dan dinamis dalam proses pembangunan perangkat lunak serta meminimalkan risiko *human error* pada penyusunan laporan keuangan. Atas dasar teoretis tersebut, sistem pada UD. Berkah Ayam ini dikembangkan menggunakan

metode *Agile Development* untuk memastikan Pembangunan sistem berjalan dinamis dan sesuai kebutuhan nyata pengguna. Selain itu, sistem ini dipadukan dengan konsep *Responsive Web Design* (RWD) guna memfasilitasi aksesibilitas dua aktor pada perangkat yang berbeda, yaitu perangkat desktop untuk *Admin* operasional dan perangkat *smartphone* untuk monitoring *Owner*, yang kemudian divalidasi kualitas fungsionalnya menggunakan *Black-Box Testing* serta dievaluasi kemudahan penggunaannya melalui *System Usability Scale* (SUS).

4. Website

Aplikasi berbasis *website* atau *web -based application* merupakan sebuah perangkat lunak yang dieksekusi di lingkungan *server (server-side)*, di mana pengguna dapat berinteraksi dengan aplikasi tersebut melalui peramban *web (web browser)* yang terhubung dengan jaringan internet. Berdasarkan kajian dari Maulana et al., (2023), aplikasi berbasis *website* memiliki keunggulan komparatif yang signifikan dalam aspek aksesibilitas dan pemeliharaan, di mana pengguna tidak perlu melakukan proses instalasi perangkat lunak secara lokal pada perangkat keras, melainkan cukup mengakses *Uniform Resource Locator* (URL) sistem. Hal ini memberikan efisiensi tinggi bagi organisasi dalam melakukan pembaruan data maupun pemeliharaan sistem secara terpusat tanpa mengganggu aktivitas operasional di sisi klien.

Dalam konteks operasional UD. Ayam Berkah, pemilihan arsitektur berbasis *website* merupakan langkah strategis untuk mengatasi

kendala geografis dan fragmentasi data akibat pengelolaan multi-*Outlet*. Selaras dengan hal tersebut, implementasi sistem berbasis *web* terbukti efektif dalam mengintegrasikan berbagai cabang bisnis yang terpisah secara geografis ke dalam satu basis data pusat (*centralized database*) secara *real-time* (Salbuanda & Esabella, 2026). Dengan menerapkan teknologi ini, seluruh data transaksi harian, setoran keuangan dari berbagai *Outlet*, hingga manifes logistik pengiriman armada dapat langsung tersinkronisasi pada saat itu juga, sehingga meminimalkan risiko kehilangan data akibat pencatatan konvensional.

Fleksibilitas aplikasi berbasis *web* juga sangat mendukung pembagian hak akses (*user privilege*) yang dinamis untuk memenuhi kebutuhan multi-pengguna. Sebagaimana dijelaskan oleh Yuricha & Phan, (2023), arsitektur *web* modern memungkinkan pengembang untuk membangun sistem yang aman dengan memisahkan fungsi kontrol dan visualisasi data sesuai dengan peran masing-masing pengguna. Pada sistem UD. Ayam Berkah, hal ini memfasilitasi *Admin* dalam mengoperasikan modul peng*Inputan* data logistik yang kompleks secara intensif melalui perangkat desktop, sementara di saat yang bersamaan, *Owner* dapat melakukan monitoring laporan keuangan secara fleksibel melalui perangkat *smartphone*.

Untuk mendukung mobilitas tersebut, arsitektur *website* ini dipadukan dengan konsep *Responsive Web Design* (RWD). Konsep ini menjamin bahwa struktur visual, ukuran elemen, dan tata letak menu sistem

akan secara otomatis beradaptasi secara dinamis mengikuti resolusi layar perangkat yang digunakan, baik saat diakses menggunakan monitor desktop oleh *Admin* maupun melalui layar *smartphone* oleh *Owner*. Integrasi antara teknologi aplikasi berbasis *web* dan fleksibilitas responsivitas ini menjadi pilar utama dalam mewujudkan Sistem Informasi Manajemen yang adaptif, efisien, dan akurat.

5. *Framework*

Framework merupakan kerangka kerja perangkat lunak yang menyediakan struktur dasar, library, serta aturan Pembangunan untuk mempermudah proses Pembangunan aplikasi secara terorganisir. Pada proses Pembangunan sistem framework berfungsi agar sistem berkembang menjadi lebih cepat dan terstruktur. Framework juga membantu pengembang tanpa perlu membangun seluruh komponen aplikasi/sistem dari awal.

Pada dasarnya Framework menyediakan fitur seperti routing, autentikasi, pengelolaan database, serta keamanan aplikasi. Dengan hadirnya fitur – fitur bawaan framework tersebut, diharapkan mempermudah pengembang dalam membangun dan mengelola sistem yang dibuat. Dalam praktik Pembangunan sistem aplikasi modern sendiri, framework dapat digunakan baik pada backend maupun frontend.

Menurut Sinlae et al, (2024) Framework juga dapat membantu menjaga konsistensi struktur kode, sehingga mempermudah pengembang melakukan proses pemeliharaan sistem. Selain itu Gunawan et al., (2024)

pemanfaatan framework dapat mendukung pembangunan aplikasi berbasis web dengan menyediakan berbagai fasilitas yang membantu proses pengembangan aplikasi. Berdasarkan uraian tersebut, dapat disimpulkan bahwa framework merupakan kerangka kerja perangkat lunak yang digunakan untuk mempercepat, menstrukturkan, dan mempermudah proses Pembangunan aplikasi.

6. *Agile*

Agile Development merupakan salah satu metode Pembangunan perangkat lunak yang menggunakan pendekatan iteratif dan inkremental. Dalam penerapannya, proses Pembangunan dilakukan secara bertahap melalui perencanaan, Pemabngunan, pengujian, serta evaluasi berulang. Metode ini sangat menekankan pentingnya komunikasi aktif antara tim pengembang dan pengguna. Melalui komunikasi tersebut, sistem yang dibangun dapat menyesuaikan dengan kebutuhan pengguna yang dinamis selama proses berjalan.

Menurut Schüll et al., (2023), *Agile* memungkinkan tim menghasilkan perangkat lunak secara cepat melalui proses eksekusi yang berulang. Setiap eksekusi yang selesai akan menghasilkan bagian sistem tertentu yang siap pakai. Bagian sistem inilah yang kemudian dievaluasi secara langsung oleh pengguna di setiap tahapannya. Berdasarkan masukan tersebut, pengembang dapat segera melakukan perbaikan demi menyempurnakan sistem.

Sementara itu, Russo, (2021) menyatakan Penerapan metode *Agile*

dalam pembuatan sistem berbasis web, makin banyak dipilih karena terbukti efektif menekan anggaran, mempercepat waktu rilis aplikasi, serta menyatukan visi antara pengembang dan pemangku kepentingan.. Efektivitas ini lahir dari pendekatan Agile yang fleksibel dan berbeda dibanding metode konvensional yang kaku. Ketika metode konvensional mewajibkan tahapan secara berurutan, Agile justru memberi ruang adaptasi sehingga perubahan fungsionalitas dapat dilakukan kapan saja selama proses pembangunan sistem.

Penelitian oleh Samosir & Nengsih, (2024) menjelaskan bahwa penerapan *Agile* terbukti meningkatkan kesesuaian sistem informasi. Peningkatan kualitas ini terjadi karena adanya tahapan evaluasi berkala yang melibatkan pengguna secara intensif. Interaksi yang kuat antara pengguna dan pengembang meminimalkan risiko kesenjangan ekspektasi. Sejalan dengan konsep metode *Agile* yang melakukan pendektan secara iteratif dan umpan balik sehingga menghasilkan sistem yang diproduksi menjadi jauh lebih berkualitas dan berkembang lebih relevan dengan kondisi nyata di lapangan.

Di sisi lain, Susilo & Azimah, (2023) menyatakan metode agile memudahkan proses transaksi sistem informasi keuangan berbasis *website*. Karakteristik *Agile* dinilai mampu mengakomodasi perubahan kebutuhan yang sering terjadi selama proses implementasi. Hal ini menjadikan *Agile* sebagai pilihan terbaik untuk proyek yang dinamis. Oleh karena itu, metode ini sangat cocok untuk sistem yang membutuhkan penyesuaian berkala.

Berdasarkan beberapa pendapat ahli di atas, dapat disimpulkan

bahwa *Agile* mengutamakan fleksibilitas dan perbaikan berkelanjutan. Karakteristik utama dari metode ini terletak pada komunikasi intensif serta proses iterasi yang konsisten. Alasan tersebut mendasari pemilihan metode *Agile* dalam Pembangunan Sistem Informasi Manajemen Operasional UD Ayam Berkah. Evaluasi secara bertahap sangat dibutuhkan agar sistem yang dibangun benar-benar sesuai dengan kebutuhan mitra.

7. *MySQL*

MySQL merupakan sistem manajemen basis data relasional (*Relational Database Management System/RDBMS*) yang digunakan untuk menyimpan, mengelola, dan memproses data secara terstruktur menggunakan bahasa *Structured Query Language (SQL)* (Wahyudi et al., 2022). *MySQL* banyak digunakan dalam Pembangunan aplikasi berbasis *website* maupun *mobile* karena sifatnya yang ringan, stabil, serta mudah diintegrasikan dengan berbagai framework Pembangunan seperti *Agile*.

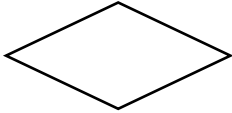
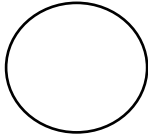
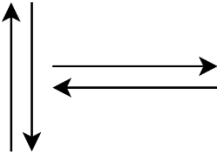
Dalam Pembangunan sistem informasi menurut Mo, (2025) *MySQL* berperan dalam pembangunan sistem basis data karena memiliki stabilitas dan skalabilitas yang baik serta mampu memproses data dalam jumlah besar secara efisien, sehingga data dapat diakses, dikelola, dan diperbarui secara efisien. Pada penelitian ini, *MySQL* digunakan untuk menyimpan data pengguna, data produk ayam potong, transaksi pemesanan, serta data lokasi yang digunakan dalam sistem berbasis *Location Based Service (LBS)*. Selain itu, Putri Wahyuni et al., (2024) menyatakan fleksibilitas *MySQL* dalam sistem informasi berbasis *website* karena kemampuan fitur –

fiturnya dalam mengelola sistem transaksi sekala besar dengan aman dan kon konsisten. Berdasarkan uraian tersebut, dapat disimpulkan bahwa *MySQL* merupakan sistem manajemen basis data relasional yang digunakan untuk mengelola data aplikasi secara terstruktur, efisien, dan terintegrasi dengan sistem backend.

8. *Flowchart*

Flowchart merupakan *Diagram* alur yang digunakan untuk menggambarkan langkah-langkah proses dalam suatu sistem secara sistematis menggunakan simbol-simbol tertentu seperti terminator, proses, *Input/output*, decision, dsb (Nazila & Yahfizham, 2023). *Flowchart* berfungsi sebagai alat bantu dalam menganalisis serta merancang sistem sehingga alur kerja sistem dapat dipahami dengan lebih jelas sebelum tahap implementasi dilakukan. Menurut Luxchita et al., (2025) *flowchart* digunakan sebagai media visual untuk menggambarkan logika proses sistem sehingga memudahkan pengembang dalam memahami alur program serta mengidentifikasi potensi kesalahan sejak tahap perancangan. *Flowchart* membantu memperjelas interaksi jalannya proses transaksi antara pengguna, sistem, dan *Admin* sehingga proses pembangunan sistem menjadi lebih terstruktur. Hal ini didukung oleh penelitian yang menyatakan bahwa penggunaan *flowchart* dalam perancangan sistem informasi dapat membantu visualisasi alur proses serta meningkatkan pemahaman terhadap sistem sebelum tahap implementasi dilakukan untuk menghindari proses yang ambigu (Khesya, 2021).

Tabel 2. 1 Simbol *Flowchart*

Simbol	Nama	Deskripsi
	Terminal	Untuk menampilkan awal dan akhir dari serangkaian tahapan.
	Pemrosesan	Untuk menampilkan proses apapun yang dilakukan sistem.
	Keputusan	Untuk menunjukan titik manapun dalam tahapan, untuk menentukan tindakan selanjutnya.
	Konektor dalam halaman	Untuk menjalin koneksi antara elemen – elemen pada flowchart yang berlanjut di halaman yang sama.
	Garis Alir	Untuk menghubungkan tanda – tanda.

Sumber : (Nazila & Yahfizham, 2023)

9. UML

Unified Modeling Language (UML) merupakan bahasa pemodelan standar yang digunakan untuk merancang, memvisualisasikan, dan mendokumentasikan sistem perangkat lunak (Dewi & Voutama, 2024). UML berfungsi menggambarkan struktur dan perilaku sistem melalui berbagai jenis *Diagram* , seperti *Use Case Diagram* , *class Diagram* , *Sequence Diagram* , dan *Activity Diagram* . Penggunaan bahasa pemodelan ini memberikan kemudahan dalam memahami hubungan antara pengguna dengan sistem serta

meminimalkan kesalahan pada tahap Pembangunan. *UML* merupakan metode yang mampu merepresentasikan komponen sistem secara terstruktur dan logis sehingga proses perancangan menjadi lebih efektif.

Sejalan dengan hal tersebut, penggunaan *UML* bertujuan menggambarkan alur sistem dan interaksi antara pengguna dengan sistem melalui *Diagram* terstruktur . Melalui pemodelan yang disediakan, pengembang dapat memahami kebutuhan fungsional dan non-fungsional sistem secara mendalam sebelum proses implementasi dilakukan. Selain itu, visualisasi yang jelas pada *UML* juga berkontribusi besar dalam meningkatkan efisiensi komunikasi antara analis sistem, pengembang, dan pengguna. Dengan demikian, penerapan ragam *Diagram* ini dapat meminimalisir kesenjangan pemahaman antara pihak-pihak yang terlibat dalam proyek (Hidayati et al., 2023).

Salah satu komponen krusial dalam perancangan sistem menggunakan *UML* adalah *Use Case Diagram*. *Use Case Diagram* digunakan untuk menggambarkan hubungan antara aktor dengan sistem serta fungsi utama yang tersedia di dalamnya. *Diagram* ini sangat membantu dalam mengidentifikasi kebutuhan fungsional sistem dan memberikan gambaran mengenai seluruh layanan yang dapat diakses oleh pengguna. Melalui pendekatan visual tersebut, proses analisis kebutuhan dapat dilakukan secara lebih sistematis agar sistem yang dikembangkan benar-benar sesuai dengan ekspektasi.

Pada penelitian ini, *UML* diterapkan untuk merancang *Sistem Informasi*

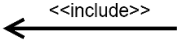
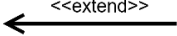
Manajemen Operasional dan Perhitungan Laba Rugi Multi-Outlet UD Berkah Ayam berbasis *website*. Pemodelan tersebut digunakan untuk memvisualisasikan interaksi antara aktor yaitu *owner*, *Admin*, dan bagian operasional dalam mengelola seluruh data penting perusahaan. Data-data yang dikelola meliputi informasi *Outlet* , harga ayam harian, data produksi, biaya operasional, hingga laporan laba rugi akhir. Melalui implementasi UML, proses Pembangunan diharapkan dapat berjalan terstruktur dan menghasilkan sistem yang sesuai dengan kebutuhan operasional UD Ayam Berkah.

9.1 Use Case Diagram

Use Case Diagram merupakan bagian *Diagram UML (Unified Modeling Language)* yang berfungsi untuk menggambarkan hubungan interaksi antar aktor dengan sistem serta menunjukkan fungsi – fungsi utama yang tersedia dalam sistem (Sunarsa et al., 2024). Selain itu *Use Case Diagram* juga mempermudah pemahaman alur kerja sistem karena, pemodelan pemahaman antara actor dan sistem disajikan secara jelas dan terstruktur. Dalam proses perancangan, *Use Case Diagram* digunakan untuk mengidentifikasi kebutuhan fungsional yang harus tersedia pada sistem serta menggambarkan layanan atau fitur yang dapat diakses oleh pengguna. *Diagram* ini gambaran umum mengenai sistem yang dikembangkan, sehingga mempermudah komunikasi antara pengembang dan pengguna dalam memahami proses yang berjalan. Selain itu, penggunaan *Use Case Diagram* menjadi dasar dalam Pembangunan *Diagram UML* lainnya seperti

Activity Diagram dan *Sequence Diagram*, sehingga proses perancangan sistem dapat dilakukan secara sistematis dan terarah. Dengan demikian, menurut Nasiri et al., (2021) *Use Case Diagram* memiliki peran penting dalam tahap analisis dan perancangan sistem agar sistem yang dibangun sesuai dengan kebutuhan pengguna.

Tabel 2. 2 Simbol *Use Case Diagram*

Simbol	Nama	Keterangan
	Aktor	Entitas yang berinteraksi dengan sistem, baik berupa manusia maupun sistem lain yang berada di luar sistem.
	<i>Use Case</i>	Fungsi atau layanan yang disediakan oleh sistem kepada aktor untuk mencapai tujuan tertentu.
	<i>Association</i>	Hubungan antara aktor dengan <i>Use Case</i> yang menunjukkan adanya interaksi antara keduanya.
	<i>Include</i>	Relasi yang menunjukkan bahwa suatu <i>Use Case</i> selalu melibatkan <i>Use Case</i> lain sebagai bagian dari proses yang harus dijalankan.
	<i>Extend</i>	Relasi yang menunjukkan adanya fungsi tambahan yang bersifat opsional dan hanya terjadi pada kondisi tertentu.

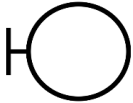
Sumber : (Ramdany, 2024)

9.2 *Sequence Diagram*

Sequence Diagram merupakan salah satu *Diagram* dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan interaksi antar komponen dalam sistem berdasarkan urutan waktu (Sunarsa et al.,

2024). *Diagram* ini menunjukkan bagaimana proses komunikasi terjadi antara aktor dengan sistem melalui pertukaran pesan (*message*) dalam menjalankan suatu fungsi tertentu. Berdasarkan penelitian oleh Aryani et al., (2024), UML digunakan sebagai bahasa pemodelan untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak, dimana salah satu tahapannya adalah penggunaan *Diagram* seperti *Sequence Diagram* untuk menggambarkan alur proses sistem. *Sequence Diagram* membantu dalam memodelkan perilaku sistem secara dinamis dengan menampilkan urutan interaksi antar objek sehingga memudahkan dalam memahami alur kerja sistem yang dibangun. Dengan demikian, *Sequence Diagram* berperan penting dalam tahap perancangan sistem karena dapat memberikan gambaran yang jelas mengenai proses yang terjadi dalam sistem secara terstruktur dan sistematis.

Tabel 2. 3 Simbol *Sequence Diagram*

Simbol	Nama	Keterangan
	Aktor	Entitas yang berinteraksi dengan sistem, baik berupa manusia maupun sistem lain yang berada di luar sistem.
	<i>Entity Class</i>	Komponen yang digunakan untuk menyimpan dan mengelola data dalam sistem.
	<i>Boundary Class</i>	Komponen yang berfungsi sebagai antarmuka antara aktor.

Simbol	Nama	Keterangan
	<i>Control Class</i>	Mengatur jalannya proses serta mengelola logika sistem dalam menghubungkan <i>boundary</i> dengan <i>entity</i> .

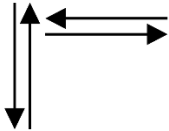
Sumber : (Ramdany, 2024)

9.3 Activity Diagram

Activity Diagram merupakan salah satu *Diagram* dalam *Unified Modeling Language (UML)* yang digunakan untuk menggambarkan alur aktivitas atau proses kerja dalam suatu sistem secara berurutan (Sunarsa et al., 2024). *Diagram* ini menampilkan urutan kegiatan mulai dari kondisi awal hingga kondisi akhir yang terjadi dalam sistem. *UML* digunakan sebagai bahasa pemodelan untuk mendokumentasi, menspesifikasikan, dan membangun sebuah sistem, dimana *Activity Diagram* berperan dalam memodelkan proses bisnis serta alur kerja sistem secara logis dan terstruktur. *Activity Diagram* membantu dalam memahami proses yang terjadi dalam sistem dengan menampilkan langkah-langkah aktivitas yang dilakukan oleh pengguna maupun sistem. Dengan adanya *Activity Diagram*, pengembang dapat mengetahui alur proses secara menyeluruh sehingga memudahkan dalam analisis dan perancangan sistem yang lebih sistematis.

Tabel 2. 4 Simbol *Activity Diagram*

Simbol	Nama	Keterangan
	<i>Initial node</i>	Titik awal dimulainya suatu proses atau aktivitas dalam sistem.
	<i>Action</i>	Langkah atau proses yang dilakukan dalam sistem

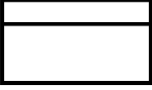
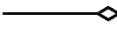
Simbol	Nama	Keterangan
	<i>Decision</i>	yang menggambarkan suatu kegiatan tertentu
	<i>Line Connector</i>	Percabangan yang digunakan untuk menentukan pilihan berdasarkan kondisi tertentu
	<i>Final Node</i>	Alur yang menghubungkan antar aktivitas yang menunjukkan urutan proses dalam sistem
		Titik akhir dari suatu proses atau aktivitas dalam sistem

Sumber : (Ramdany, 2024)

9.4 Class Diagram

Class Diagram merupakan salah satu *Diagram* dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan struktur sistem secara statis. *Diagram* ini menunjukkan hubungan antar kelas dalam sistem, termasuk atribut (data) dan metode (fungsi) yang dimiliki oleh masing-masing kelas. UML digunakan sebagai bahasa pemodelan untuk merancang sistem perangkat lunak, dimana *Class Diagram* berperan dalam menggambarkan struktur data serta hubungan antar komponen dalam sistem. *Class Diagram* membantu pengembang dalam memahami bagaimana data disusun dan saling berhubungan sehingga memudahkan dalam proses implementasi sistem. Dengan adanya *Class Diagram*, perancangan sistem menjadi lebih terstruktur karena menggambarkan hubungan antar entitas secara jelas, sehingga dapat dijadikan sebagai dasar dalam pembuatan database maupun Pembangunan sistem lebih lanjut.

Tabel 2. 5 Simbol *Class Diagram*

Simbol	Nama	Keterangan
	<i>Class</i>	Representasi dari objek yang memiliki atribut dan metode yang menggambarkan suatu entitas dalam sistem.
	<i>Attribute</i>	Data atau informasi yang dimiliki oleh suatu <i>class</i> .
	<i>Aggregation</i>	Hubungan antara <i>class</i> yang menunjukkan hubungan bagian dari suatu keseluruhan.
	Generalisasi	Hubungan pewarisan (<i>inheritance</i>) antara <i>class</i> induk dan <i>class</i> turunan.

Sumber : (Ramdany, 2024)

10. Blacbox Testing

Black Box Testing merupakan metode pengujian perangkat lunak yang berfokus pada pengujian fungsi sistem berdasarkan kesesuaian antara masukan (*Input*) dan keluaran (output) yang dihasilkan. Metode ini dilakukan tanpa melihat struktur kode program, tetapi dengan mengamati bagaimana sistem memberikan respon terhadap aktivitas yang dilakukan oleh pengguna. Pengujian ini bertujuan untuk memastikan setiap fitur yang tersedia dapat berjalan sesuai dengan kebutuhan fungsional sistem. Menurut Huda et al., (2022), *Black Box Testing* digunakan untuk melakukan validasi terhadap fungsi sistem dengan memberikan skenario pengujian berdasarkan proses *Input* dan output yang dihasilkan oleh sistem.

Wulandari et al., (2022) menjelaskan bahwa pengujian sistem menggunakan *Black Box Testing* diperlukan untuk memastikan setiap bagian sistem telah berjalan sesuai prosedur dan fungsi yang telah dirancang sebelum

diterapkan kepada pengguna akhir. Pengujian dilakukan dengan melakukan percobaan terhadap fitur sistem berdasarkan kebutuhan yang telah ditentukan. Metode ini membantu pengembang mengetahui apakah terdapat kesalahan pada proses maupun hasil keluaran sistem. Dengan demikian, *Black Box Testing* dapat digunakan untuk mengetahui tingkat kesesuaian sistem terhadap kebutuhan pengguna.

Berdasarkan analisis Wintana et al., (2022) menjelaskan bahwa *Black Box Testing* lebih mudah dan cepat digunakan, karena hanya menguji fungsionalitas sistem berdasarkan interaksi pengguna dengan sistem. Pengujian dilakukan dengan memeriksa apakah setiap fitur mampu menghasilkan keluaran yang sesuai dengan harapan pengguna. Metode ini tidak membutuhkan pemeriksaan terhadap kode program sehingga pengujian dapat dilakukan berdasarkan sisi fungsional sistem. Oleh karena itu, metode ini sesuai digunakan untuk menguji sistem informasi yang memiliki berbagai fitur operasional.

Pada penelitian ini, *Black Box Testing* diterapkan untuk menguji Sistem Informasi Manajemen Operasional dan Perhitungan Laba Rugi *Multi-Outlet* UD Berkah Ayam berbasis *website*. Pengujian dilakukan terhadap fitur utama sistem seperti proses *login*, pengelolaan data *Outlet*, *Input* harga ayam harian, pencatatan produksi ayam, pengelolaan biaya operasional, proses perhitungan laba rugi, serta pembuatan laporan. Pengujian dilakukan untuk memastikan seluruh fungsi sistem berjalan sesuai dengan kebutuhan *admin* dan *owner*. Hasil pengujian diharapkan dapat menunjukkan bahwa sistem

telah mampu membantu proses pengelolaan operasional UD Berkah Ayam secara efektif.

B. Kajian Empiris

Kajian empiris digunakan untuk menunjukkan penelitian terdahulu yang memiliki keterkaitan dengan penelitian ini sebagai dasar dalam Pembangunan “Rancang Bangun Sistem Informasi Dan Perhitungan *Multi-Outlet* Berbasis *Website* Menggunakan Metode *Agile Development*” pada UD Ayam Berkah. Penelitian sebelumnya memberikan gambaran mengenai penerapan sistem informasi berbasis *website* dalam membantu pengelolaan operasional usaha, pencatatan data, pengolahan laporan keuangan, serta penggunaan metode pembangunan sistem sesuai kebutuhan pengguna. Penelitian Kapoh et al., (2021) membahas pengujian sistem informasi berbasis *website* menggunakan metode *Black Box Testing*. Hasil penelitian menunjukkan bahwa sistem dapat membantu pengelolaan data secara lebih terintegrasi. Selain itu, *Black Box Testing* efektif untuk memastikan kesesuaian *Input* dan output sistem. Penelitian ini relevan dengan sistem UD Berkah Ayam pada fitur operasional seperti data *Outlet*, produksi, biaya, dan laporan laba rugi.

Penelitian Pebriadi et al., (2024) membahas sistem informasi keuangan berbasis *website* dengan pendekatan *Agile*. Hasil penelitian diharapkan bahwa sistem dapat menyederhanakan alur operasional, meningkatkan transparansi laporan keuangan, dan mengotomatisasi transaksi keuangan. Selaras dengan hal itu, sistem juga mampu menghasilkan laporan keuangan yang lebih akurat dan efisien. Hal ini relevan dengan penelitian UD Berkah Ayam yang berfokus pada

perhitungan laba rugi dari data produksi dan biaya operasional. Penerapan metode *Agile* pada Pembangunan sistem informasi berbasis *website* melalui pendekatan Systematic Literature Review. Hasil penelitian Asad et al, (2023) menunjukkan bahwa metode *Agile* semakin banyak digunakan dalam pengembangan perangkat lunak karena mampu menyesuaikan diri terhadap perubahan kebutuhan serta memungkinkan perangkat lunak dikembangkan secara bertahap. Pendekatan tersebut relevan dengan pembangunan sistem pada UD Berkah Ayam karena pengembangan sistem dilakukan secara bertahap untuk menyesuaikan kebutuhan operasional, seperti pengelolaan *outlet*, pencatatan pembelian ayam, setoran *outlet*, biaya operasional, serta penyusunan laporan laba rugi.

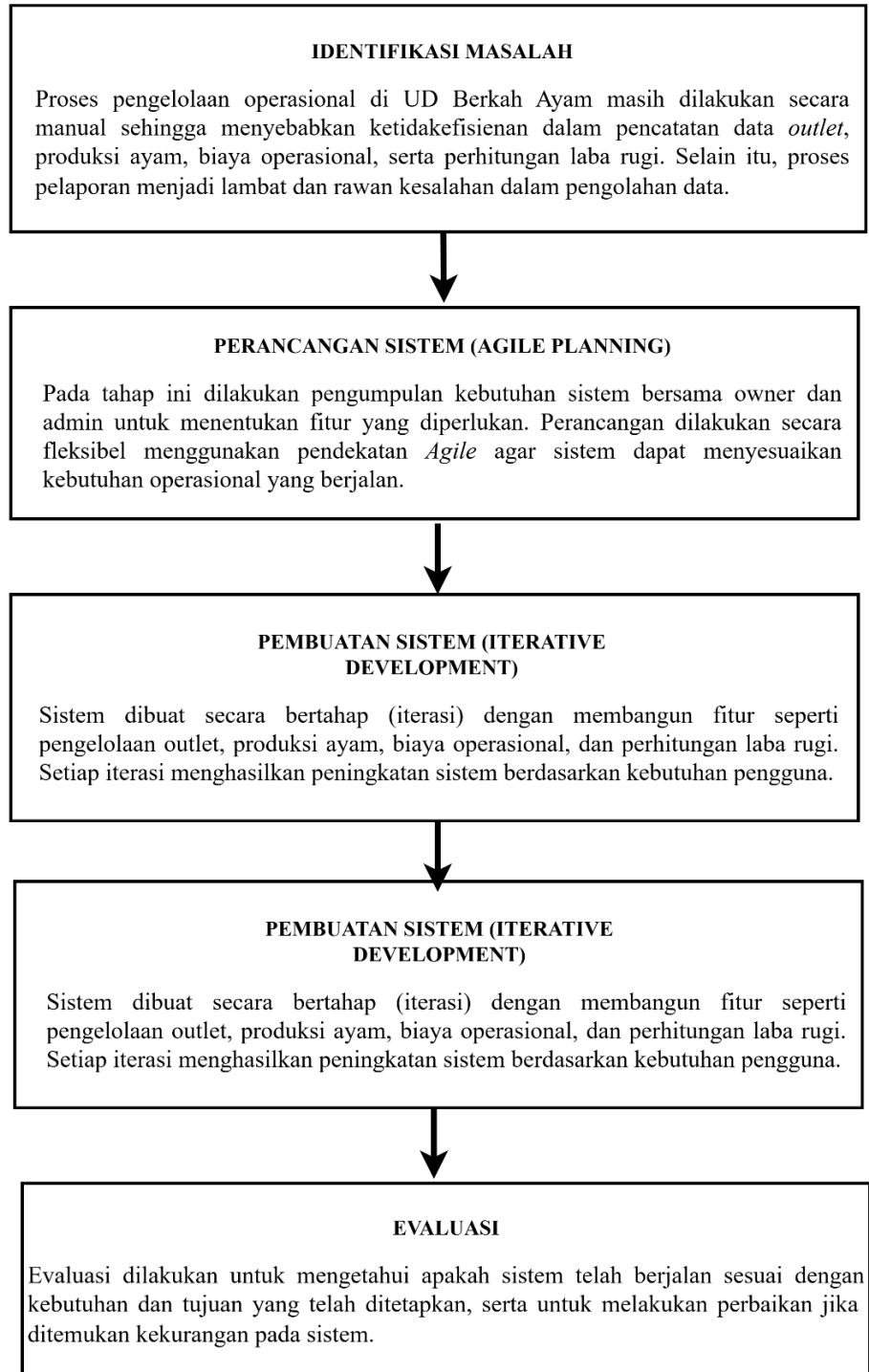
Penelitian internasional Edison et al., (2022) membahas perbandingan metode *Agile* pada Pembangunan perangkat lunak skala besar. Hasil penelitian menunjukkan bahwa organisasi dapat memilih dan mengadopsi metode *Agile* yang sesuai dengan konteks pengembangannya, sedangkan proses penerapan *Agile* pada skala besar dapat melibatkan eksperimen, kegagalan, dan pembelajaran. Penelitian tersebut memiliki keterkaitan dengan sistem UD Berkah Ayam karena pengembangan sistem *multi-outlet* memerlukan metode yang dapat disesuaikan dengan konteks dan kebutuhan pengembangan sistem.

Penelitian internasional Yong, (2021) membahas penerapan *Agile Software Development* melalui kajian sistematis terhadap berbagai pendekatan Pembangunan perangkat lunak. Penelitian tersebut menjelaskan bahwa pengembangan *Agile* dilakukan secara bertahap dan berorientasi pada kebutuhan

pengguna sehingga mampu merespons perubahan kebutuhan perangkat lunak dengan cepat. Penelitian tersebut relevan dengan pembangunan sistem UD Berkah Ayam karena sistem dikembangkan secara bertahap dan dapat disesuaikan dengan kebutuhan operasional serta proses bisnis yang berjalan.

Berdasarkan kajian empiris tersebut, dapat disimpulkan bahwa penerapan sistem informasi berbasis *website* dapat membantu meningkatkan efektivitas pengelolaan data operasional dan keuangan dibandingkan proses manual. Selain itu, metode *Agile* mendukung Pembangunan sistem yang fleksibel karena memungkinkan adanya evaluasi dan perubahan sesuai kebutuhan pengguna. Penelitian ini berfokus pada Pembangunan Sistem Informasi Logistik dan Keuangan Multi-*Outlet* UD Berkah Ayam yang mencakup pengelolaan pembelian ayam, setoran *Outlet*, biaya operasional, serta perhitungan laba rugi otomatis untuk membantu monitoring dan pengambilan keputusan.

C. Kerangka Berpikir



Gambar 2. 1 Kerangka Berpikir