

BAB II

TINJAUN LITERATUR

2.1. Penelitian Terdahulu

2.2. Tabel 1.2 *literature review* (SLR)

No	Peneliti & Tahun	Objek Penelitian	Metode / Algoritma	Dataset	Tujuan Penelitian	Hasil Penelitian
1	(Hamidy, 2024)	Sistem manajemen persediaan (objek umum organisasi)	+FIFO (First In First Out)	Data stok barang dan transaksi persediaan	Menguji efektivitas penerapan FIFO dalam sistem manajemen persediaan menggunakan model DeLone & McLean	istem berbasis FIFO menunjukkan efektivitas sangat tinggi (100%) pada dimensi kualitas sistem, penggunaan, dan dampak bersih, serta meningkatkan efisiensi pengendalian stok

2	(Nafisah & Yamasari, 2024)	Gudang Lembaga Manajemen Infaq (LMI) Surabaya	FIFO, Waterfall	Data barang masuk dan keluar gudang	Merancang sistem pergudangan digital untuk mengatasi pencatatan manual dan penumpukan barang	Sistem memenuhi standar ISO 9126 dengan nilai kesesuaian 1,83 serta meningkatkan akurasi dan kecepatan pengelolaan stok
3	Gustayo & hermawan (2020)	Perusahaan konstruksi (inventori & bahan baku)	FIFO	Data bahan baku dan inventori proyek	Mengoptimalkan pendataan dan pengendalian bahan baku melalui sistem inventori	Penerapan FIFO mampu mengurangi kesalahan pencatatan, meningkatkan transparansi, dan mendukung produktivitas perusahaan
4	(Fajriani et al., 2022)	Klinik (persediaan obat)	FIFO	Data stok dan distribusi obat	Mengurangi kesalahan pencatatan dan risiko kedaluwarsa obat melalui sistem terkomputerisasi	Sistem FIFO meningkatkan akurasi data, efisiensi pelaporan, dan memastikan

						obat lama digunakan terlebih dahulu
5	(Nafisah & Yamasari, 2024)	Sistem informasi stok barang (organisasi multi-unit)	FIFO	Data inventaris dan distribusi barang	Mengoptimalkan manajemen gudang melalui sistem stok berbasis web	Sistem FIFO meningkatkan efisiensi, menjaga kualitas barang, serta mencegah penimbunan stok

2.2 First In First Out (FIFO)

Metode First In First Out (FIFO) merupakan metode pengelolaan persediaan yang menetapkan bahwa barang yang pertama kali masuk akan menjadi barang yang pertama kali diproses atau dikeluarkan. Prinsip ini mengasumsikan bahwa barang yang diperoleh lebih awal dianggap digunakan atau dijual terlebih dahulu, sehingga biaya perolehan barang yang masuk pertama kali akan dibebankan lebih dahulu sebagai harga pokok penjualan. Dalam sistem pencatatan fisik, FIFO mengasumsikan bahwa aliran barang mengikuti urutan kedatangan secara nyata. Sementara itu, pada sistem pencatatan perpetual, perhitungan harga pokok penjualan dilakukan setiap kali terjadi transaksi penjualan berdasarkan urutan barang yang pertama masuk (Candra Pamungkas et al., 2023)

Secara konseptual, metode FIFO juga menyatakan bahwa barang dagangan yang pertama kali dibeli merupakan barang yang pertama kali dijual. Dengan demikian, perhitungan harga pokok penjualan didasarkan pada harga perolehan barang yang masuk lebih awal, sedangkan persediaan akhir akan mencerminkan harga perolehan barang yang terakhir dibeli (Tria Tirtaliyany Agustin, 2022)

2.3. Metode Last In First Out (LIFO)

Metode LIFO adalah pencatatan persediaan yang mengasumsikan persediaan yang terakhir masuk (dibeli) akan dikeluarkan (dijual atau dipakai) pertama kali. MTKP (masuk terakhir keluar pertama). (Aprilia et al., 2020)

2.4. Persediaan (inventori)

Persediaan menurut Kamus Besar Bahasa Indonesia (KBBI) diartikan sebagai suatu kegiatan menyiapkan atau merencanakan sesuatu agar tersedia saat dibutuhkan. Dalam konteks perusahaan, khususnya yang bergerak di bidang produksi, ketersediaan persediaan bahan baku merupakan hal yang sangat penting untuk mendukung kelancaran proses produksi serta memenuhi kebutuhan dan preferensi pelanggan. Penumpukan atau penyediaan bahan baku dalam jumlah tertentu menjadi syarat utama dalam penyusunan jadwal produksi. Apabila bahan baku tersedia secara memadai, maka proses produksi dapat berjalan lebih optimal serta meminimalkan risiko terjadinya kekurangan bahan baku dan keterlambatan pengiriman produk kepada pelanggan. Kondisi tersebut penting untuk menjaga stabilitas operasional perusahaan dan menghindari kerugian maupun dampak negatif terhadap reputasi perusahaan (Sumaryanto et al., 2022)

2.5 Rancangan Database


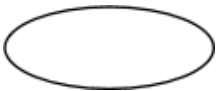
2.5.1. Unified Modeling Language (UML)




Pemodelan merupakan proses perancangan perangkat lunak yang dilakukan sebelum tahap pengodean (coding). Pada sistem yang bersifat kompleks, pemahaman secara menyeluruh sering kali menjadi sulit, sehingga diperlukan representasi dalam bentuk model untuk mempermudah analisis dan perancangan. Semakin tinggi tingkat kompleksitas sistem yang dikembangkan, semakin penting pula penerapan teknik pemodelan yang tepat dan terstruktur. Unified Modeling Language (UML) merupakan bahasa pemodelan berbasis grafis yang memiliki fitur lengkap untuk mendukung proses perancangan sistem). (Booch & Jacobson, 1996). UML telah menjadi standar industri dalam kegiatan perancangan, dokumentasi, dan visualisasi perangkat lunak, serta menyediakan kerangka pemodelan yang sistematis dan terstandarisasi.

2.5.2. Use Case Diagram

Use case diagram adalah salah satu bentuk pemodelan yang digunakan untuk menggambarkan perilaku (behavior) sistem informasi yang akan dibangun. Diagram ini menunjukkan interaksi antara aktor dan sistem, di mana aktor dapat terdiri dari satu atau lebih pihak yang berinteraksi dengan sistem. Secara sederhana, use case diagram menjelaskan fungsi-fungsi yang tersedia dalam sistem serta pihak-pihak yang memiliki hak untuk menjalankannya. (Gemino & Parker, 2009). Dalam penamaannya, setiap use case sebaiknya menggunakan kata kerja yang sederhana, jelas, dan mudah dipahami agar merepresentasikan aktivitas yang dilakukan secara tepat. Berikut merupakan simbol-simbol yang digunakan dalam *use case* antara lain:

Tabel 2.1 Simbol-simbol *Use case Diagram*

No	Simbol	Nama	Deskripsi
1		<i>Actor</i>	Aktor merupakan entitas yang dapat berupa individu, proses, maupun sistem lain yang berinteraksi dengan sistem informasi yang dirancang, tetapi berada di luar batasan sistem tersebut.
2		<i>Use case</i>	Use case adalah representasi fungsi atau layanan yang disediakan oleh sistem informasi yang akan dibangun. Setiap use case menggambarkan unit aktivitas yang memungkinkan terjadinya pertukaran pesan atau interaksi antara aktor dan


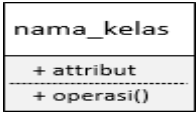

			sistem.
3		<i>Association</i>	Association merupakan hubungan komunikasi antara aktor dan use case yang menunjukkan adanya interaksi atau keterlibatan aktor dalam menjalankan fungsi yang terdapat pada use case tersebut.
4		<i>Extend</i>	Extend adalah relasi yang menunjukkan adanya penambahan fungsi pada suatu use case. Use case tambahan ini bersifat opsional dan tetap dapat berdiri sendiri meskipun tanpa dijalankannya relasi perluasan tersebut.
5		<i>Include</i>	Include merupakan relasi antar use case yang menunjukkan bahwa suatu use case memerlukan use case lain sebagai bagian dari proses pelaksanaannya. Dengan kata lain, use case yang disertakan menjadi syarat atau bagian yang harus dijalankan agar fungsi utama dapat berjalan dengan baik.




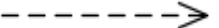

2.5.3. Class Diagram

Class diagram merupakan representasi struktur sistem yang menggambarkan pendefinisian kelas-kelas yang dibentuk dalam proses pembangunan sistem. Setiap class memiliki atribut dan operasi (metode). Atribut adalah variabel-variabel yang dimiliki oleh suatu class, yang berfungsi untuk menjelaskan karakteristik atau properti dan biasanya dituliskan dalam bentuk teks di dalam kotak class. Sementara itu, operasi atau metode merupakan fungsi-fungsi yang dimiliki oleh class untuk menjalankan suatu proses tertentu. (Bhatt & Nandu, 2021) Secara umum, class diagram berfungsi untuk mendeskripsikan jenis-jenis objek yang terdapat dalam sistem yang akan dikembangkan serta hubungan statis di antara objek-objek tersebut. Diagram ini juga menjelaskan properti dan operasi yang dimiliki oleh setiap class serta batasan-batasan yang terdapat dalam relasi antarobjek. Selain itu, class diagram menggambarkan struktur class (kelas), package (paket), dan objek beserta hubungan yang menyertainya, seperti asosiasi, pewarisan (inheritance), dan hubungan kepemilikan (containment), maupun relasi lainnya.

Adapun simbol-simbol yang digunakan dalam class diagram adalah sebagai berikut:

Tabel 2.2 Simbol-simbol *Class Diagram*

No.	Nama	Deskripsi
1	 <i>Package</i>	Sebuah bungkus dari satu atau lebih <i>class</i> .
2	 <i>Class</i>	Kelas pada struktur sistem.
3	 <i>Interface</i>	Sama dengan konsep antarmuka dalam

			Pemrograman berorientasi objek.
4		<i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity.
5		<i>Directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain.
6		<i>Generalization</i>	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).
7		<i>Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas.
8		<i>Aggregation</i>	Relasi antar kelas dengan makna semua bagian (whole-part).







2.5.4. Sequence Diagram

Sequence diagram atau diagram sekuens merupakan sebuah diagram yang menggambarkan kelakuan (*behaviour*) objek yang ada pada *use case* dengan mendefinisikan waktu hidup (*lifetime*) dari objek tersebut dan pesan yang diterima dan dikirimkan antar objek. Maka dari itu, dalam membuat diagram sekuens sebelumnya pembuat harus mengetahui objek-objek apa saja yang terlibat dalam sebuah *usecase* beserta metode atau operasi apa yang dimiliki *class* yang diinstansiasi menjadi objek itu.

Penggambaran diagram sekuens sebanding dengan banyaknya

pendefinisian *use case* yang mempunyai proses sendiri atau semua *use case* yang sudah didefinisikan interaksinya atau jalannya pesan. (Swain et al., n.d.). Dapat disimpulkan jika semakin banyak *use case* yang didefinisikan, maka semakin banyak pula diagram sekuens yang harus dibuat. Simbol-simbol yang digunakan dalam *sequence diagram* adalah:

Tabel 2.3 Simbol-simbol *Sequence Diagram*

No.	Simbol	Nama	Deskripsi
1		An Actor	Menggambarkan orang yang sedang berinteraksi dengan sistem.
2		Entity Class	Menggambarkan hubungan kegiatan yang akan dilakukan.
3		Boundary Class	Menggambarkan sebuah penggambaran dari form.
4		Control Class	Menggambarkan penghubung antara boundary dengan tabel.
5		A focus of control	Menggambarkan tempat mulai dan berakhirnya sebuah message (pesan).
6		A line of life	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.



2.5.5. Activity Diagram




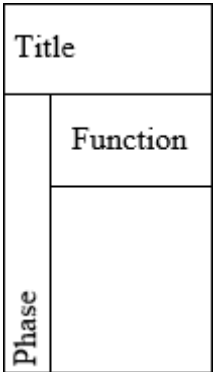
Activity diagram atau diagram aktivitas merupakan representasi visual yang digunakan untuk menggambarkan alur kegiatan atau proses kerja dalam

suatu sistem maupun proses bisnis (Sugiarti, 2018). Perlu dipahami bahwa diagram ini berfokus pada aktivitas yang terjadi di dalam sistem, bukan pada tindakan yang dilakukan oleh aktor. Selain itu, diagram aktivitas juga mampu merepresentasikan proses yang berjalan secara paralel.

Secara umum, activity diagram menunjukkan aliran aktivitas dalam sistem yang sedang dirancang, mulai dari bagaimana suatu proses dimulai, kemungkinan keputusan yang terjadi di tengah proses, hingga bagaimana proses tersebut berakhir. Dalam UML, activity diagram dapat dianggap sebagai bentuk khusus dari state diagram, di mana sebagian besar state merepresentasikan aksi, dan perpindahan antarstate biasanya terjadi setelah suatu aktivitas selesai dijalankan (Touseef, n.d.). Dengan demikian, activity diagram tidak menjelaskan perilaku internal sistem maupun interaksi detail antar subsistem secara spesifik, melainkan menggambarkan jalur dan rangkaian aktivitas pada tingkat konseptual atau level tinggi secara umum. Adapun simbol-simbol yang digunakan dalam penggambaran *activity diagram* adalah sebagai berikut:

Tabel 2.4 Simbol-simbol *Activity Diagram*

No.	Simbol	Nama	Deskripsi
1		Status awal	Status awal merupakan titik dimulainya suatu proses dalam diagram aktivitas, yang menandai awal jalannya sistem.
2		Aktivitas	Aktivitas menggambarkan tindakan atau proses yang dijalankan oleh sistem dan umumnya dituliskan menggunakan kata kerja.

3		<i>Decision</i>	Percabangan (decision) menunjukkan adanya lebih dari satu pilihan aktivitas yang dapat dilakukan berdasarkan kondisi tertentu.
4		Join	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5		Status akhir	Status akhir menandakan berakhirnya rangkaian aktivitas dalam sistem, yang menunjukkan bahwa proses telah selesai dijalankan.
6		Swimlane	Swimlane digunakan untuk memisahkan pihak atau bagian organisasi yang bertanggung jawab terhadap setiap aktivitas yang terjadi dalam proses tersebut.