

BAB III

LANDASAN TEORI

3.1. Definisi Perancangan

Perancangan adalah suatu tahapan untuk menetapkan apa yang akan dilakukan dengan memanfaatkan beragam metode, yang mencakup penjelasan mengenai struktur, detil elemen, serta kendala yang akan dihadapi selama proses pelaksanaan (Shafirina Arsydini Shidqi et al., 2025). Perancangan merupakan sebuah langkah atau tahapan dalam menciptakan atau merencanakan suatu hal dengan memanfaatkan metode untuk mendefinisikan target yang ingin diraih (Fauzi et al., 2022). Perancangan merupakan langkah-langkah terstruktur untuk menetapkan cara mana sistem atau perangkat lunak akan direalisasikan, dengan sasaran untuk memproduksi dokumen spesifikasi terperinci mengenai elemen, struktur, antarmuka, prosedur, dan alur kerja sistem yang didasarkan pada kebutuhan yang telah dikenali (Rizki Simbolon et al., 2025).

3.2. Sistem Informasi

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi yang menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (Edo Prandahlan & Veri Ilhadi, 2025). Salah satunya adalah sistem informasi untuk memenuhi kebutuhan perusahaan untuk menangani masalah penjualan, pembelian barang dan pembuatan laporan, sistem penjualan serta pembelianya belum berjalan secara maksimal seperti yang diharapkan, oleh karena itu, perusahaan perlu sistem yang lebih baik dengan memakai sistem yang berbasis komputer. (Effendi E et al., 2023)

3.3. Definisi *Inventory* Gudang

Inventory (persediaan) adalah kumpulan stok barang yang dikelola dan dipantau untuk memastikan pemenuhan pesanan secara akurat, yang melibatkan pencatatan

secara sistematis pada sistem manajemen stok untuk mencegah *stock out* atau selisih antara data *online* dan gudang fisik (Nur Izzatusshafa An-Nisaa et al., 2025).

Inventaris merujuk pada pencatatan jumlah fisik barang yang ada, yang dibandingkan dengan data yang tercatat dalam sistem penyimpanan untuk menilai akurasi stok (Nursyanti & Partisia, 2024).

Inventory barang gudang adalah kumpulan barang yang dicatat dan diawasi melalui sistem *web* untuk mempermudah pengelolaan stok gudang, sehingga tujuannya adalah meningkatkan efisiensi dan keakuratan dalam pencatatan barang (Adinda Mutiara Adzani, 2024).

3.4. Website

Website adalah sebuah kumpulan halaman *web* yang dapat diakses publik dan saling terkait, biasanya dibangun untuk menyajikan informasi yang dapat diakses melalui jaringan internet. *Website* terdiri dari berbagai konten seperti teks, gambar, suara dan animasi, dan bisa dibuat serta dikelola oleh individu, kelompok, atau organisasi untuk berbagai keperluan (FItrah et al., 2025). *website* adalah kumpulan informasi yang terdiri dari halaman *web* yang saling terhubung, disediakan oleh perorangan atau organisasi, dan dapat diakses kapan saja melalui perangkat seperti komputer atau ponsel (Afdhaluddin et al., 2023). *website* adalah kumpulan halaman yang menyimpan informasi dalam bentuk data digital seperti teks, gambar, video, audio, dan animasi yang dapat diakses melalui internet (Rifai et al., 2025).

3.5. Agile

Metode Agile adalah sebagian dari *Software Development Life Cycle (SDLC)* yang banyak digunakan dalam pengembangan sistem informasi berbasis *web*. Agile adalah pendekatan pengembangan perangkat lunak yang menekankan kolaborasi tim, adaptasi terhadap perubahan, dan iterasi/penyampaian produk secara bertahap sepanjang siklus hidup proyek. Pendekatan ini memungkinkan tim merespons kebutuhan pengguna yang berubah lebih cepat dibandingkan model tradisional (Nova et al., 2022). Metode Agile dijelaskan sebagai pendekatan manajemen proyek yang mendukung kolaborasi tim, fleksibilitas, adaptasi terhadap perubahan

kebutuhan pengguna, serta pengembangan bertahap melalui iterasi pendek untuk evaluasi dan penyempurnaan berkelanjutan (Fajri et al., 2024).

1. Metode *Scrum*

Metode *Scrum* digunakan sebagai pendekatan iteratif dan inkremental dalam mengembangkan aplikasi sistem informasi. *Scrum* dilakukan melalui tahapan seperti *Sprint Planning*, *Daily Scrum*, *Sprint Review*, dan *Sprint Retrospective* untuk mengelola pekerjaan pengembangan secara bertahap dan adaptif terhadap perubahan kebutuhan pengguna (Arsa et al., 2024). *Scrum* dijelaskan sebagai metode Agile yang mengadopsi pendekatan iteratif dan inkremental, memungkinkan pengembangan sistem secara bertahap dan adaptif terhadap perubahan kebutuhan pengguna. Dengan *Scrum*, siklus kerja dibagi menjadi *Sprint-Sprint* pendek yang meningkatkan fleksibilitas dan respons tim terhadap perubahan kebutuhan (Zuhri et al., 2025). *Scrum* sebagai metodologi pengembangan sistem informasi yang memungkinkan tim bekerja secara kolaboratif, iteratif, dan adaptif melalui tahapan seperti *Daily Scrum*, *Sprint Review*, dan *Sprint Retrospective* untuk merespons perubahan kebutuhan dan meningkatkan kualitas hasil kerja (Hasypa Aulia, 2021).

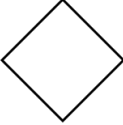
3.6. *Flowchart*

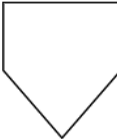
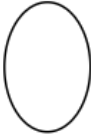
Flowchart atau bagan alur adalah diagram yang menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses dari suatu program. Setiap langkah digambarkan dalam bentuk diagram dan dihubungkan dengan garis atau arah panah.

Flowchart berperan penting dalam memutuskan sebuah langkah atau fungsionalitas dari sebuah proyek pembuatan program yang melibatkan banyak orang sekaligus. Selain itu dengan menggunakan bagan alur proses dari sebuah program akan lebih jelas, ringkas, dan mengurangi kemungkinan untuk salah penafsiran. Penggunaan *flowchart* dalam dunia pemrograman juga merupakan cara yang bagus untuk menghubungkan antara kebutuhan teknis dan *non-teknis*.

Pada dasarnya simbol-simbol dalam *flowchart* memiliki arti yang berbeda-beda. Berikut adalah simbol-simbol yang sering digunakan dalam proses pembuatan *flowchart* (Rony Setiawan, 2021).

Tabel 3. 1 Simbol *flowchart*

SIMBOL	NAMA	FUNGSI
	<i>Terminator</i>	Permulaan/akhir program
	<i>Garis Alir (Flow Line)</i>	Arah aliran program
	<i>Preparation</i>	Proses analisis/pemberian harga awal
	<i>Proses</i>	Proses yang dilakukan oleh komputer
	<i>Input/Output Data</i>	Proses input atau <i>output</i> tanpa tergantung jenis peralatannya
	<i>Predefined Process (Sub Program)</i>	Simbol untuk melaksanakan suatu bagian (sub bagian)/prosedur
	<i>Decision</i>	Perbandingan pernyataan, menyeleksi data yang memberikan pilihan untuk langkah selanjutnya

SIMBOL	NAMA	FUNGSI
	<i>Display</i>	Menunjukkan <i>output</i> yang tampil di komputer
	<i>Off Page Connector</i>	Penghubung bagian-bagian <i>flowchart</i> yang berada pada halaman berbeda
	<i>On Page Connector</i>	Penghubung bagian-bagian <i>flowchart</i> yang berada pada satu halaman

Sumber 1 Jurnal Rony Setiawan, 2021

3.7. Unified Modelling Language (UML)

Unified Modelling Language adalah bahasa pemodelan *visual* yang digunakan untuk merancang dan memodelkan sistem perangkat lunak.

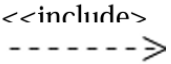
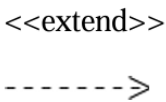
Dilansir dari *Visual Paradigm*, UML mencakup notasi grafis yang digunakan untuk merepresentasikan berbagai aspek dari sistem, termasuk struktur sistem, perilaku sistem, interaksi antara objek, dan lingkungan di mana sistem beroperasi (Aulia Reta Faulina, 2025)

3.7.1. Use Case Diagram

Use case diagram merupakan *visualisasi* dari beberapa komponen, seperti aktor, *use case*, dan relasi antar komponen. Beberapa simbol atau notasi digunakan dalam penggambaran fungsionalitas sebuah sistem dalam *use case diagram*. Melalui *use case diagram*, dapat membantu analis dalam penyusunan kebutuhan (*requirement*) pengembangan sistem. *Use case diagram* dipakai untuk menjelaskan perancangan

sistem kepada *user* dan melakukan perancangan semua fitur yang ada pada sistem yang akan dibangun (Siska Narulita, 2024).

Tabel 3. 2 Simbol *Use Case Diagram*

SIMBOL	NAMA	KETERANGAN
	Aktor	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi itu sendiri.
	<i>Association</i>	Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
	<i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum- khusus) antar dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>)
	<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya
	<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri

SIMBOL	NAMA	KETERANGAN
	<i>System</i>	Memverifikasikan paket yang menampilkan sistem secara terbatas
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi)
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

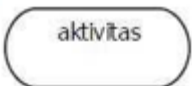
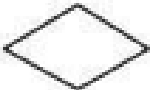
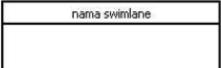
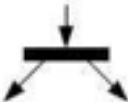
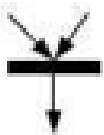
Sumber 2. Siska Narulita, 2024

3.7.2. Activity Diagram

Activity diagram adalah diagram aktivitas yang dapat digunakan memodelkan proses-proses tindakan yang dieksekusi atau menjelaskan komunikasi data. Biasanya diagram ini digunakan untuk desain tingkat tinggi seperti memodelkan proses bisnis dan desain tingkat rendah seperti untuk mendeskripsikan algoritma yang akan dipakai. (M. Rhifky Wayahdi, 2023)

Activity Diagram (diagram aktivitas) adalah diagram yang memodelkan proses-proses yang terjadi dalam sebuah sistem. (Agustio, 2024)

Tabel 3. 3 Simbol *Activity Diagram*

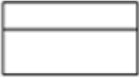
SIMBOL	NAMA	DESKRIPSI
	Status awal	Satus awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
	Aktivitas	Aktivitas yang dilakukan sistem, sktivitas biasanya diawali dengan kata kerja
	Percabangan/ <i>decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
	Penggabungan/ <i>join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
	Status akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi
	<i>Fork</i>	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel
	<i>Join</i>	Digunakan untuk menunjukkan kegiatan yang digabungkan

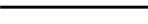
Sumber 3. Jurnal Shafira Arsydini Shidqi

3.7.3. Class Diagram

Class Diagram adalah diagram yang menggambarkan hubungan pada semua class atau tabel yang ada pada database. Diagram kelas atau *Class Diagram* menggambarkan kerangka sistem dari segi pengertian kelas-kelas yang akan dibuat untuk membangun sistem. Database yang akan digunakan untuk *web* inventaris barang terdiri dari beberapa tabel utama yang saling terkoneksi diantaranya tabel pengguna, tabel barang, table laporan, tabel barang masuk dan tabel barang keluar.(Zhafira, 2025)

Tabel 3. 4 Simbol *Class Diagram*

SIMBOL	NAMA	DESKRIPSI
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagai perilaku dan struktur data dari <i>object</i> yang ada di atasnya <i>object</i> induk (<i>ancestor</i>)
	<i>N-Ary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 <i>object</i>
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan <i>suet</i> hasil yang terukur bagi <i>suet</i> aktor
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh <i>suet object</i>
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada <i>suet</i> elemen mandiri (<i>independent</i>) akan mempengaruhi

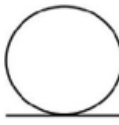
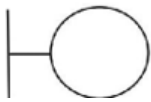
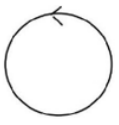
		elemen yang bergantung padanya elemen yang tidak mandiri
	<i>Association</i>	Apa yang menghubungkan antara <i>object</i> satu dengan <i>object</i> lainnya

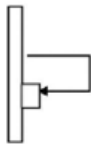
Sumber 4. Jurnal Suharni, Eel Susilowati, Fahrial Pakusadewa

3.7.4. Sequence Diagram

Sequence diagram atau diagram urutan adalah sebuah diagram yang digunakan untuk menjelaskan dan menampilkan interaksi antar objek-objek dalam sebuah sistem secara terperinci. Selain itu sequence diagram juga akan menampilkan pesan atau perintah yang dikirim, beserta waktu pelaksanaannya. Objek-objek yang berhubungan dengan berjalannya proses operasi biasanya diurutkan dari kiri ke kanan. (Rony Setiawan, 2021)

Tabel 3. 5 Simbol Sequence Diagram

SIMBOL	NAMA	DESKRIPSI
	Aktor	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi
	<i>Entity Class</i>	Gambaran sistem sebagai landasan dalam menyusun basis data
	<i>Boundary Class</i>	Menangani komunikasi antar lingkungan sistem
	<i>Control Class</i>	Bertanggung jawab terhadap kelas-kelas terhadap objek yang berisi logika

SIMBOL	NAMA	DESKRIPSI
	<i>Recursive</i>	Menggambarkan pesan/hubungan objek itu sendiri yang menunjukkan urutan kejadian yang terjadi
	<i>Activation</i>	Mewakili proses durasi aktivasi sebuah operasi
	<i>Life Line</i>	Komponen yang digambarkan garis putus terhubung dengan objek

Sumber 5. Rony Setiawan, 2021

3.8. Draw.io

Draw.io adalah perangkat lunak pembuat diagram sumber terbuka (*open source*) berbasis *web* yang memungkinkan pengguna membuat berbagai jenis diagram seperti *flowchart*, UML, dan skema jaringan. Penekanan di berikan pada kemudahannya karena tidak memerlukan instalasi (berbasis browser) dan integrasinya dengan Google Drive (Jumriati, 2024). Draw.io adalah platform desain grafis vektor yang fleksibel. Fokusnya adalah pada kemampuan draw.io dalam memfasilitasi pembuatan dokumentasi teknis sistem perangkat lunak secara kolaboratif dan gratis, yang menjadikannya alternatif utama bagi aplikasi berbayar seperti Microsoft Visio (Permana, 2023). Draw.io didefinisikan sebagai media pembelajaran *visual* yang interaktif. Menekankan bahwa draw.io adalah alat bantu berpikir komputasional yang mengubah logika abstrak menjadi representasi *visual* (*flowchart*) yang mudah dipahami oleh siswa (Firdaus, 2025).

3.9. User interface (UI)

User Interface didefinisikan sebagai bagian *visual* dari perangkat lunak yang menjembatani interaksi antara pengguna dan sistem, mencakup elemen estetika seperti warna, tipografi dan tata letak untuk meningkatkan efisiensi penggunaan (Rahayu, 2024). *User Interface* dijelaskan sebagai mekanisme komunikasi antara manusia dan komputer melalui tampilan grafis (GUI) yang bertujuan untuk membuat interaksi menjadi sederhana dan seefisien mungkin (Pratama, 2025). User interface adalah titik kontak dimana interaksi manusia-komputer terjadi, yang kini beralih dari sekedar *visual* statis menjadi antarmuka yang adaptif dan responsive terhadap perilaku pengguna (Zhang, 2023).

3.10. Database

Database didefinisikan sebagai sekumpulan data yang terorganisir, memungkinkan akses dan pengelolaan yang efisien melalui sistem manajemen basis data untuk operasi pengambilan dan pembaruan secara terstruktur (Dery rahmawan, 2025). Basis data adalah bermacam-macam informasi yang secara metodis disimpan di komputer dan dapat diakses melalui program Komputer (DBMS) (Diarta, 2025). Menjelaskan evolusi pengertian basis data dari model relasional (SQL) tradisional menuju model yang lebih fleksibel (NoSQL) sesuai dengan perkembangan data modern (Handoko, 2024).

3.11. Visual Studio Code

Visual studio code adalah alat bantu utama dalam pembelajaran pemrograman yang meningkatkan pemahaman konseptual dan keterampilan *coding*. Penulis menyoroti antarmuka yang intuitif serta fitur *intellisense* dan debugging terintegrasi sebagai keunggulan utamanya (Nia Ani Murani, 2025). *Visual Studio Code* merupakan sebuah text editor ringan dan handal buatan *Microsoft* untuk sistem operasi *multiplatform* (*Windows, Linux, Mac*) editor ini mendukung secara langsung Bahasa *JavaScript, TypeScript*, dan *Node js*, serta Bahasa lainnya melalui *plugin* (Salamah, 2021).

3.12. Framework

Framework adalah sekumpulan instruksi atau program yang telah terstandarisasi untuk memudahkan pengembangan dalam membangun sebuah aplikasi tanpa harus menulis kode dari awal (boilerplate) penggunaan *framework* meningkatkan konsistensi dan keamanan kode (Hanun, 2025). *Framework* didefinisikan struktur konseptual yang menyediakan fungsionalitas umum yang dapat diubah secara khusus oleh pengembang (Wijaya, 2025).