

## BAB II

### LANDASAN TEORI

#### 2.1 Penelitian Terdahulu

Berikut adalah beberapa penelitian terdahulu yang relevan dengan perancangan sistem manajemen stok berbasis website, serta penerapan metode *User-Centered Design* (UCD) dan *Agile Development*.

Tabel 2. 1 Jurnal Penelitian terdahulu

No	Nama Peneliti, Tahun, Judul penelitian	Tujuan Penelitian	Perbedaan
1.	(Nazila Ramadina, 2024) Pengembangan Sistem Informasi Manajemen Persediaan Barang Berbasis Website Menggunakan Metode Safety Stock dan Reorder Point	Mengembangkan sistem informasi manajemen persediaan barang berbasis website untuk membantu proses pengendalian stok dengan metode Safety Stock dan Reorder Point.	Fokus pada metode penghitungan stok. Penelitian ini lebih menekankan perhitungan matematis dibanding desain UI/UX atau metode pengembangan perangkat lunak.
2.	(Nugroho & Sari, 2023) Analisis UI/UX Menggunakan Metode User Centered-Design pada Aplikasi TSP Mobile	Menganalisis kualitas UI/UX aplikasi TSP Mobile menggunakan metode UCD dengan pengukuran aspek usability.	Fokus pada aplikasi mobile dan analisis UI/UX. Tidak mengembangkan sistem penuh dan tidak terkait dengan pengelolaan stok.
3.	(Minasa et al., 2024) Sistem Informasi Pengelolaan Inventaris	Mengembangkan sistem pengelolaan inventaris berbasis	Fokus pada proses iteratif Agile, namun tidak menggabungkan

No	Nama Peneliti, Tahun, Judul penelitian	Tujuan Penelitian	Perbedaan
	UMKM Berbasis Web dengan Pendekatan Agile	web dengan metode Agile Development untuk UMKM.	UCD atau memperhatikan aspek antarmuka pengguna secara eksplisit.
4.	(Rura & Ardiansyah, 2023) Rancang Bangun Sistem Informasi Inventory Barang Berbasis Web (Studi Kasus: Toko Dina Beauty Care)	Merancang dan mengembangkan sistem informasi inventory berbasis web untuk toko retail guna meningkatkan efisiensi operasional.	Menggunakan metode Waterfall, tidak fokus pada pendekatan UCD atau Agile.
5.	(Handayani et al., 2023) Perancangan Sistem Informasi Inventory Barang Berbasis Web Menggunakan Metode Agile Software Development	Merancang sistem manajemen inventaris berbasis web menggunakan metode Agile.	Hanya menggunakan Agile Development, tanpa pendekatan User-Centered Design dalam desain UI/UX sistemnya.

Berdasarkan Tabel 2.1 lima penelitian terdahulu yang telah dianalisis, dapat disimpulkan bahwa masing-masing penelitian memiliki fokus dan pendekatan yang berbeda dalam membangun sistem informasi inventaris. Beberapa penelitian seperti oleh Handayani et al. (2023) dan Minasa et al. (2024) menekankan penggunaan *Agile Development* sebagai metode fleksibel dalam pengembangan sistem. Di sisi lain, Ramadina & Asmunin (2024) fokus pada penghitungan stok matematis melalui metode *Safety Stock* dan *Reorder Point*.

Sementara itu, Nugroho & Sari (2023) memberikan kontribusi penting dalam aspek analisis kualitas UI/UX melalui pendekatan *User-Centered Design* (UCD),

meskipun tidak membangun sistem inventaris secara langsung. Penelitian oleh Rura & Ardiansyah (2023) masih menggunakan metode tradisional *Waterfall*, tetapi tetap relevan karena menggarap topik yang serupa, yakni pengelolaan stok berbasis web.

Dengan menggabungkan pendekatan UCD dan Agile dalam penelitian ini, sistem yang dibangun diharapkan dapat lebih optimal dari sisi kebutuhan pengguna serta fleksibel dalam pengembangannya, yang belum secara lengkap diterapkan oleh satu pun dari lima penelitian tersebut secara bersamaan.

## 2.2 Website

Website adalah kumpulan halaman yang saling terhubung dan berisi informasi dalam bentuk teks, gambar, suara, atau animasi, yang dapat diakses melalui jaringan internet. Menurut (Gregorius, 2000), website merupakan kumpulan halaman web yang saling terhubung dan file-filenya saling terkait. Web terdiri dari page atau halaman, dan kumpulan halaman yang dinamakan homepage. *Homepage* berada pada posisi teratas, dengan halaman-halaman terkait berada di bawahnya. Biasanya setiap halaman di bawah *homepage* disebut *child page*, yang berisi *hyperlink* ke halaman lain dalam web.

Penelitian (Bekti, 2015) menyebutkan bahwa website adalah halaman informasi yang disediakan melalui jalur internet sehingga bisa diakses di seluruh dunia selama terkoneksi dengan jaringan internet. Website merupakan komponent atau kumpulan komponent yang terdiri dari teks, gambar, suara animasi sehingga lebih merupakan media informasi yang menarik dikunjungi.

Menurut (Lukamnul, 2004), website merupakan fasilitas internet yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen pada website disebut dengan web page dan link dalam website memungkinkan pengguna bisa berpindah dari satu page ke page lain (*hyper text*), baik diantara page yang disimpan dalam server yang sama maupun server diseluruh dunia.

Dari beberapa definisi di atas, dapat disimpulkan bahwa website merupakan media penyampai informasi yang bersifat global, mudah diakses kapan pun dan di mana pun selama terhubung ke jaringan internet. Website dapat digunakan dalam berbagai aspek seperti komunikasi, pendidikan, bisnis, hingga layanan publik.

### 2.2.1 Jenis-Jenis Website

1. Website Pribadi (*Personal Website*): Digunakan untuk kepentingan individu, seperti blog atau portofolio online.
2. Website Perusahaan (*Corporate Website*): Digunakan oleh perusahaan untuk memberikan informasi tentang produk atau layanan mereka.
3. Website *E-commerce*: Digunakan untuk kegiatan jual beli secara online.
4. Website Pendidikan : Digunakan oleh institusi pendidikan untuk memberikan informasi akademik dan layanan pendidikan.
5. Website Pemerintah: Digunakan oleh instansi pemerintah untuk memberikan informasi dan layanan publik.

Dengan memahami pengertian dan jenis-jenis website dari (FIKOM UDBFIKOM UDB, 2023), kita dapat lebih efektif dalam memanfaatkan teknologi ini sesuai dengan kebutuhan dan tujuan yang ingin dicapai.

### 2.3 Manajemen stok

Manajemen stok adalah proses pengaturan dan pengawasan terhadap barang-barang yang dimiliki oleh suatu perusahaan atau usaha, baik dalam bentuk bahan mentah, barang setengah jadi, maupun produk jadi. Tujuan utama dari manajemen stok adalah memastikan bahwa barang tersedia dalam jumlah yang cukup untuk memenuhi kebutuhan operasional dan permintaan pelanggan tanpa menyebabkan kelebihan persediaan. Hal ini penting karena stok yang tidak terkelola dengan baik bisa menyebabkan kerugian, baik dari segi biaya penyimpanan maupun risiko barang rusak atau kedaluwarsa (Jean, 2024).

Dalam praktiknya, manajemen stok melibatkan berbagai aktivitas seperti pencatatan keluar masuk barang, pengendalian jumlah persediaan, pengecekan kondisi barang, serta perencanaan pembelian barang baru. Proses ini memerlukan sistem yang tertata agar informasi mengenai jumlah dan lokasi barang selalu akurat dan mudah diakses. Pengelolaan yang rapi juga membantu menghindari kesalahan dalam pemesanan, seperti pembelian barang yang sebenarnya masih tersedia dalam jumlah cukup.

Manajemen stok yang baik dapat memberikan banyak manfaat bagi suatu usaha, antara lain efisiensi biaya, kelancaran operasional, serta kepuasan pelanggan karena ketersediaan produk yang stabil. Selain itu, pengelolaan stok yang tepat juga

mempermudah proses pengambilan keputusan, seperti kapan harus melakukan pemesanan ulang atau berapa banyak barang yang perlu disediakan dalam periode tertentu. Oleh karena itu, meskipun terlihat sebagai proses sederhana, manajemen stok memiliki peran strategis dalam keberlangsungan dan pertumbuhan sebuah usaha.

### **2.3.1 Pentingnya Manajemen Stok**

Manajemen stok yang efektif memiliki peran penting dalam operasional perusahaan, antara lain:

1. Mengurangi Biaya Operasional: Dengan mengelola stok secara efisien, perusahaan dapat mengurangi biaya penyimpanan dan risiko barang rusak atau kadaluarsa.
2. Meningkatkan Kepuasan Pelanggan: Ketersediaan barang yang tepat waktu dan jumlah yang sesuai dapat meningkatkan kepuasan pelanggan dan loyalitas mereka terhadap Perusahaan.
3. Mengoptimalkan Arus Kas: Manajemen stok yang baik membantu dalam perencanaan pembelian dan penjualan, sehingga arus kas perusahaan dapat terkelola dengan lebih baik.

Penelitian juga menunjukkan bahwa praktik manajemen stok yang baik memberikan perhatian langsung terhadap biaya, aliran kas, dan kepuasan pelanggan dalam kinerja rantai pasokan penelitian juga menunjukkan bahwa praktik manajemen stok yang baik memberikan perhatian langsung terhadap biaya, aliran kas, dan kepuasan pelanggan dalam kinerja rantai pasokan (Jean, 2024).

### **2.3.2 Metode dalam Manajemen Stok**

Terdapat beberapa metode yang umum digunakan dalam manajemen stok, antara lain:

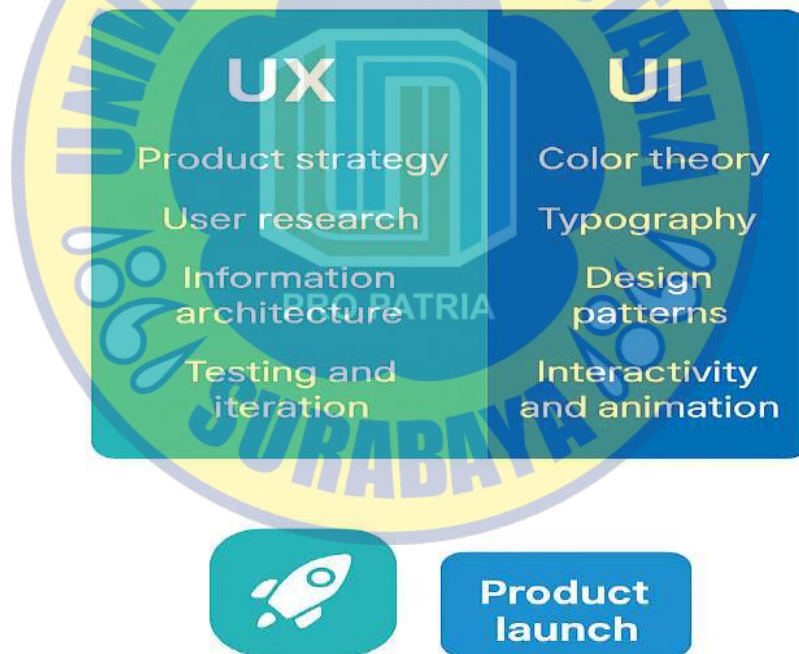
1. *Just-In-Time* (JIT): Metode ini bertujuan untuk meminimalkan jumlah stok dengan menerima barang tepat pada saat dibutuhkan dalam proses produksi atau penjualan.
2. *Economic Order Quantity* (EOQ): Metode ini menghitung jumlah pesanan optimal untuk meminimalkan total biaya persediaan, termasuk biaya pemesanan dan penyimpanan.

3. ABC Analysis: Metode ini mengklasifikasikan barang berdasarkan tingkat kepentingannya, sehingga perusahaan dapat memfokuskan pengelolaan pada barang yang paling bernilai.

Dengan menerapkan metode-metode tersebut, perusahaan dapat meningkatkan efisiensi manajemen stok dan mendukung pencapaian tujuan bisnis secara keseluruhan (SAID AL RUQEISHI & ULLAH, 2024).

## 2.4 UI/UX Design

UI (*User Interface*) dan UX (*User Experience*) adalah dua aspek penting dalam pengembangan sebuah sistem atau aplikasi digital yang berfokus pada interaksi pengguna. UI merujuk pada tampilan visual yang dilihat dan digunakan oleh pengguna, seperti tombol, ikon, warna, dan tata letak halaman. Desain UI yang baik harus menarik secara estetika dan mudah dipahami agar pengguna bisa menjalankan fungsinya tanpa kebingungan (CMD DESIGN, 2023).



Gambar 2. 1 UI/UX Design

Sementara itu, pada Gambar 2.1 UX berkaitan dengan keseluruhan pengalaman pengguna saat berinteraksi dengan sistem, mulai dari seberapa mudah digunakan hingga seberapa puas pengguna terhadap proses dan hasil akhirnya. UX mencakup alur penggunaan, kemudahan navigasi, kecepatan akses, dan kenyamanan dalam mencapai tujuan tertentu melalui sistem tersebut. Pengalaman pengguna yang baik

akan mendorong pengguna untuk terus menggunakan produk digital tersebut dan merekomendasikannya ke orang lain (*CMD DESIGN, 2023*).

UI dan UX saling berkaitan dan tidak bisa dipisahkan. UI yang menarik tanpa didukung oleh UX yang baik bisa membuat pengguna merasa frustrasi. Sebaliknya, sistem dengan UX yang bagus tapi tampilan antarmuka yang membingungkan juga akan menurunkan kepuasan pengguna. Oleh karena itu, dalam merancang aplikasi atau website, penting untuk memperhatikan kedua aspek ini secara seimbang agar menghasilkan produk yang tidak hanya fungsional, tetapi juga nyaman dan menyenangkan digunakan (*CMD DESIGN, 2023*).

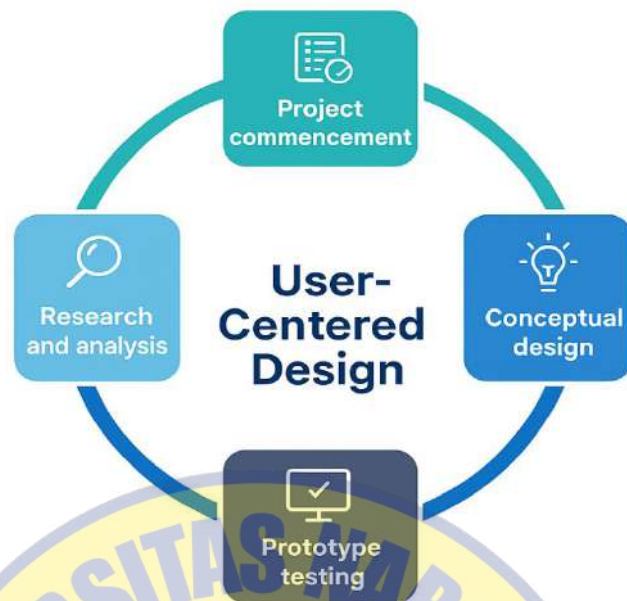
UX mencakup lima elemen berlapis, yaitu:

1. *Strategy* – Tujuan bisnis dan kebutuhan pengguna.
2. *Scope* – Fitur dan konten.
3. *Structure* – Interaksi dan arsitektur informasi.
4. *Skeleton* – Desain antarmuka, navigasi, dan sistem informasi.
5. *Surface* – Tampilan visual.

Desain UI/UX yang baik bertujuan untuk meningkatkan kepuasan pengguna, mempermudah interaksi, serta mempercepat proses pencapaian tujuan pengguna saat menggunakan system (*Dian Kartika, 2025*).

## **2.5 User-Centered Design**

*User-Centered Design* (UCD) adalah pendekatan dalam merancang sistem atau produk yang berfokus sepenuhnya pada kebutuhan, keinginan, dan keterbatasan pengguna. Dalam metode ini, pengguna dilibatkan sejak awal proses perancangan hingga tahap akhir pengujian. Tujuannya adalah menciptakan solusi yang benar-benar sesuai dengan cara pengguna berpikir dan berinteraksi, sehingga produk yang dihasilkan lebih mudah digunakan dan lebih efektif dalam menyelesaikan tugas yang diinginkan (*Emma, 2024*).



Gambar 2. 2 *User-Centered Design*

Proses pada Gambar 2.2 UCD biasanya dilakukan secara bertahap dan iteratif. Artinya, desain awal akan terus disempurnakan berdasarkan masukan dari pengguna melalui pengujian langsung atau wawancara. Dengan pendekatan ini, kesalahan desain dapat diminimalkan sejak awal, dan kepuasan pengguna bisa lebih terjamin. UCD tidak hanya membuat sistem menjadi lebih ramah pengguna, tetapi juga meningkatkan efisiensi, efektivitas, dan penerimaan produk di dunia nyata.

Dengan menerapkan *User-Centered Design*, pengembang dapat menciptakan produk yang tidak hanya berfungsi dengan baik, tetapi juga memberikan pengalaman yang menyenangkan dan relevan bagi pengguna. Pendekatan ini membantu menjembatani kesenjangan antara apa yang dibutuhkan pengguna dengan apa yang dirancang oleh tim pengembang. Hasil akhirnya adalah sistem yang lebih responsif terhadap kebutuhan nyata pengguna, yang pada gilirannya dapat meningkatkan loyalitas pengguna dan keberhasilan produk di pasar.

### **2.5.1 Manfaat *User-Centered Design***

1. Meningkatkan Kepuasan Pengguna: Dengan memahami kebutuhan dan preferensi pengguna, produk yang dirancang akan lebih sesuai dengan harapan mereka, meningkatkan kepuasan dan loyalitas.

2. Mengurangi Biaya dan Waktu Pengembangan: Identifikasi masalah usability sejak awal dapat mengurangi biaya perbaikan di tahap selanjutnya dan mempercepat proses pengembangan.
3. Meningkatkan Produktivitas: Produk yang mudah digunakan memungkinkan pengguna menyelesaikan tugas dengan lebih efisien, meningkatkan produktivitas.

*User-Centered Design (UCD)* membantu menciptakan produk digital yang intuitif dan mudah digunakan dengan melibatkan pengguna sejak awal—mengurangi potensi kesalahan dan memperkuat kesesuaian produk dengan kebutuhan pengguna melalui proses iteratif yang mendalam (Emma, 2024).

### **2.5.2 Tahapan *User-Centered Design***

1. Penelitian Pengguna: Mengumpulkan informasi tentang kebutuhan, tujuan, dan perilaku pengguna melalui wawancara, survei, atau observasi.
2. Desain: Mengembangkan prototipe atau wireframe berdasarkan data pengguna yang telah dikumpulkan.
3. Pengujian: Melibatkan pengguna dalam pengujian prototipe untuk mendapatkan umpan balik dan mengidentifikasi area yang perlu diperbaiki.
4. Implementasi: Mengembangkan produk akhir berdasarkan desain yang telah divalidasi oleh pengguna.
5. Evaluasi: Setelah implementasi, melakukan evaluasi berkelanjutan untuk memastikan produk terus memenuhi kebutuhan pengguna.

### **2.6 *Agile Development***

*Agile Development* adalah pendekatan dalam pengembangan perangkat lunak yang menekankan fleksibilitas, kolaborasi tim, iterasi pendek, dan kemampuan beradaptasi terhadap perubahan kebutuhan pengguna. Pendekatan ini muncul sebagai respons terhadap keterbatasan metode tradisional seperti *Waterfall* yang bersifat linear dan kurang tanggap terhadap perubahan selama proses pengembangan berlangsung. Agile mendorong komunikasi yang intens dan pengembangan yang bertahap, sehingga memungkinkan perbaikan terus-menerus berdasarkan umpan balik dari pengguna (Chibuike Daraojimba et al., 2024).



Gambar 2. 3 Agile Development

Dalam praktiknya, pada Gambar 2.3 Agile membagi proses pengembangan menjadi beberapa siklus pendek yang disebut sprint. Setiap sprint biasanya berlangsung selama satu hingga empat minggu dan menghasilkan bagian kecil dari sistem yang sudah bisa diuji atau digunakan. Setelah setiap sprint selesai, tim akan melakukan evaluasi dan perencanaan ulang untuk sprint berikutnya. Dengan cara ini, produk bisa berkembang secara bertahap sambil tetap menjaga kualitas dan relevansi terhadap kebutuhan pengguna.

### 2.6.1 Prinsip dan Pendekatan Agile

Agile berlandaskan pada *Agile Manifesto* yang terdiri dari empat nilai utama dan dua belas prinsip. Di antaranya:

1. Individu dan interaksi lebih penting daripada proses dan alat.
2. Perangkat lunak yang berfungsi lebih penting daripada dokumentasi yang lengkap.
3. Kolaborasi dengan pelanggan lebih penting daripada negosiasi kontrak.
4. Merespons perubahan lebih penting daripada mengikuti rencana yang kaku.

Salah satu pendekatan populer dalam Agile adalah *Scrum*, yang membagi proyek menjadi sprint (periode pengembangan singkat, biasanya 2–4 minggu). Dalam tiap sprint, tim fokus pada penyelesaian fitur tertentu yang dapat langsung diuji (*Chibuike Daraojimba et al., 2024*).

## 2.6.2 Keunggulan *Agile Development*

### 1. Fleksibel terhadap perubahan

Agile memungkinkan perubahan kebutuhan pengguna bahkan di tengah proses pengembangan, tanpa mengganggu alur kerja secara signifikan. Hal ini penting untuk proyek yang berkembang seiring waktu.

### 2. Iterasi singkat dan terukur

Agile menggunakan siklus iteratif (*sprint*) yang singkat, biasanya 1–4 minggu, yang memungkinkan pengujian dan evaluasi produk lebih cepat serta peningkatan berkelanjutan dari satu iterasi ke iterasi berikutnya.

### 3. Keterlibatan aktif pengguna

Metode ini mendorong kolaborasi erat antara pengembang dan pengguna. *Feedback* pengguna secara langsung membantu memastikan bahwa sistem sesuai dengan kebutuhan nyata mereka.

### 4. Peningkatan kualitas produk

Dengan pengujian yang dilakukan secara berkala di setiap iterasi, bug atau kesalahan dapat ditemukan dan diperbaiki lebih awal, sehingga kualitas perangkat lunak menjadi lebih terjaga.

### 5. Tim lebih adaptif dan produktif

Agile mendorong komunikasi terbuka dan kolaborasi dalam tim. Ini membantu menciptakan lingkungan kerja yang responsif dan efisien.

### 6. Waktu ke pasar lebih cepat

Karena pengembangan dilakukan secara inkremental, bagian dari produk bisa langsung digunakan meskipun keseluruhan belum selesai. Hal ini mempercepat proses rilis ke pengguna akhir.

Metodologi Agile terbukti meningkatkan fleksibilitas terhadap perubahan, iterasi singkat, keterlibatan pengguna, kualitas produk, produktivitas tim, dan kecepatan time-to-market (*Chibuike Daraojimba et al., 2024*).

## 2.7 Scrum

Scrum adalah kerangka kerja (framework) dalam Agile Development yang berfokus pada pengembangan perangkat lunak secara iteratif dan inkremental. Framework ini banyak digunakan karena mampu meningkatkan fleksibilitas tim, memperbaiki komunikasi, serta menghasilkan produk yang sesuai kebutuhan

pengguna melalui proses adaptasi yang berkelanjutan. Scrum bekerja dengan siklus singkat yang disebut sprint, di mana setiap sprint menghasilkan bagian produk yang siap diuji atau digunakan. (Schwaber & Sutherland, 2020)

### **2.7.1 Sprint Planning**

Sprint Planning adalah fase awal dalam setiap sprint di mana tim, bersama Product Owner, menetapkan tujuan sprint dan memilih item dari Product Backlog ke dalam Sprint Backlog. Dalam kerangka SPM-Scrum, Sprint Planning membantu tim menentukan fitur prioritas yang bisa diselesaikan dalam waktu sprint yang terstruktur. (Santoso & Amalia, 2021)

### **2.7.2 Daily Scrum**

Daily Scrum adalah pertemuan singkat harian (sekitar 15 menit) yang dimaksudkan untuk mempertemukan tim pengembang agar melaporkan progres, mengidentifikasi hambatan, dan merencanakan pekerjaan berikutnya. Studi implementasi Agile-Scrum pada sistem informasi MBKM menunjukkan bahwa daily stand-up bermanfaat menjaga sinkronisasi tim dan mempercepat adaptasi saat hambatan muncul. (Aruan et al., 2024)

### **2.7.3 Sprint Review**

Sprint Review adalah pertemuan akhir sprint untuk mendemonstrasikan hasil kerja kepada stakeholder. Hasil implementasi metode Scrum pada sistem magang menunjukkan Sprint Review memungkinkan tim menerima masukan langsung dan meningkatkan kualitas system. (Nurmasani et al., 2024)

### **2.7.4 Sprint Retrospective**

Sprint Retrospective dilakukan setelah Review untuk melakukan evaluasi terhadap proses kerja — apa yang berjalan baik dan perlu diperbaiki. Penelitian yang sama menunjukkan bahwa Retrospective membantu tim menyempurnakan alur kerja dan fitur sesuai masukan pengguna. (Nurmasani et al., 2024)

## **2.8 Frontend dan Backend**

Dalam pengembangan aplikasi web, sistem umumnya dibagi menjadi dua bagian utama, yaitu *frontend* dan *backend*.

*Frontend* adalah bagian dari aplikasi yang langsung terlihat dan digunakan oleh pengguna (*client-side*). *Frontend* mencakup elemen visual, desain antarmuka, serta interaksi pengguna. Teknologi yang biasa digunakan dalam pengembangan

*frontend* meliputi HTML, CSS, JavaScript, dan framework seperti React, Vue, atau Angular (Vijay Chauhan, 2022).

*Backend* adalah bagian yang berjalan di sisi server (*server-side*) dan bertanggung jawab untuk mengelola logika bisnis, penyimpanan data, autentikasi, dan komunikasi dengan database. Teknologi yang sering digunakan antara lain Node.js, PHP, Python (Django/Flask), serta sistem manajemen basis data seperti MySQL atau MongoDB (Vijay Chauhan, 2022).

Frontend dan backend bekerja bersama melalui API (*Application Programming Interface*), sehingga menciptakan sistem yang utuh, efisien, dan interaktif.

## 2.9 Perangkat Lunak

Dalam proses perancangan dan pengembangan sistem manajemen stok berbasis website untuk Usaha Dagang Asri Raya, digunakan beberapa perangkat lunak pendukung yang membantu dalam tahap desain antarmuka, pemodelan sistem, dan pengembangan kode program. Berikut adalah penjelasan dari masing-masing perangkat lunak:

### 2.9.1 Figma

Figma adalah perangkat lunak desain antarmuka berbasis cloud yang digunakan untuk membuat wireframe, prototipe, dan desain visual antarmuka pengguna (UI). Figma mendukung kolaborasi secara real-time dan sangat cocok diterapkan dalam pendekatan *User-Centered Design* karena memudahkan pengujian desain bersama pengguna (Sonali Kapoor & Aarti, 2024).

Contoh penggunaan:

1. Mendesain tampilan antarmuka halaman dashboard, login, dan manajemen stok.
2. Membuat prototipe interaktif untuk uji coba pengguna sebelum implementasi *frontend*.

### 2.9.2 Visual Studio Code

VS Code adalah *source-code editor* ringan dan multiplatform yang mendukung banyak bahasa pemrograman seperti JavaScript, HTML, CSS, dan PHP. VS Code dilengkapi fitur seperti ekstensi, IntelliSense, debugging, dan integrasi Git, yang sangat mendukung proses pengembangan sistem berbasis web (Visual Studio Code, 2025).

Contoh penggunaan:

1. Mengembangkan frontend sistem menggunakan React Vite.
2. Mengatur struktur file backend dan frontend.
3. Menyambungkan sistem dengan database serta menulis API

### 2.9.3 Draw.io

Draw.io adalah aplikasi berbasis web yang digunakan untuk membuat diagram alur, use case, flowchart, dan diagram sistem lainnya. Aplikasi ini mendukung ekspor dalam berbagai format dan mudah diintegrasikan dengan Google Drive atau perangkat lokal (*Kymerly Fergusson, 2018*).

Contoh penggunaan:

1. Membuat diagram *Use Case* dan *Entity Relationship* Diagram (ERD).
2. Membuat flowchart alur proses manajemen stok.
3. Visualisasi struktur sistem sebelum proses pengembangan dilakukan.

