

## **BAB II**

### **TINJAUAN PUSTAKA DAN LANDASAN TEORI**

#### **2.1 Tinjauan Pustaka**

Penelitian sebelumnya dilakukan pada tahun 2023 oleh Calvin Pang dan Muhammad Ardiansyah dengan judul “Analisis Desain Dan Pengembangan *Point Of Sales* Astakona Anugerah Abadi Menggunakan Metode *Extreme Programming*” [12]. Dengan menerapkan metode *Extreme Programming*. Dalam penelitian ini, penulis bertujuan untuk mengintegrasikan elemen digital ke dalam sistem pencatatan yang awalnya menggunakan kertas, dengan tujuan mengurangi penggunaan kertas melalui implementasi sistem elektronik berbasis *web*. Pendekatan yang digunakan dalam penelitian ini adalah metode *Extreme Programming* (XP).

Penerapan metode XP juga dilakukan pada penelitian dengan judul “Perancangan Aplikasi *Point Of Sale* Berbasis *Web* pada Belig Coffe Menggunakan *Extreme Programming*” [10] yang dilaksanakan oleh Tomi Rizmawan, Daffa Satria Maulana, Bara Pentium Alarsy, dan Roeslan Djulatov pada tahun 2023. Penelitian ini dilakukan Di Kedai Batu Belig Coffee, sistem kasir masih menggunakan pendekatan manual, termasuk dalam pembuatan nomor setruk atau kuitansi yang dilakukan secara tulis tangan. Hal ini dapat mengakibatkan masalah seperti duplikasi nomor transaksi dan risiko kehilangan, rusak, atau setruk yang tercecer. Hasilnya adalah aplikasi kasir yang memudahkan transaksi dan menyimpan laporan penjualan secara terstruktur dalam *database* yang dapat dicetak.

Penelitian yang dilakukan oleh Muhammad Rafi Raihan, dan Deny Hidayatullah pada 2022 juga membahas pengembangan aplikasi POS dengan judul penelitian “Pengembangan Sistem *Point of Sale* Berbasis *User Centered Design*” [13]. Pada penelitian ini, Sistem *Point of Sale* (POS) diperlukan untuk mengotomatisasi proses pencatatan persediaan barang dan laporan keuangan di UD Tebar Gembira, sebuah usaha mikro, kecil, dan menengah (UMKM) dalam penjualan minuman herbal. Kurangnya pemahaman pemilik usaha tentang pencatatan persediaan dan laporan keuangan menyebabkan kesulitan dalam melihat jumlah persediaan dan pengeluaran-pendapatan. Sistem POS memungkinkan untuk

mengintegrasikan pengelolaan transaksi penjualan, manajemen inventaris, penghitungan keuangan, dan fitur lainnya. Penelitian sebelumnya juga mendukung pengembangan sistem POS dalam berbagai konteks, yang meningkatkan efisiensi dan kualitas layanan. Dalam konteks UD Tebar Gembira, pengembangan sistem POS berbasis *User Centered Design* akan membantu dalam meningkatkan pengelolaan usaha dan memudahkan pemilik dalam mengelola data dan laporan keuangan. Proses pengembangan dimulai dari studi literatur dan identifikasi masalah, lalu pengumpulan data, perancangan dan implementasi sistem, pengujian, dan penulisan laporan.

Pada tahun 2021, penelitian yang dilakukan oleh Suprianto, Muhammad Fadlan, dan Denis Prayogi dengan judul “Perancangan Aplikasi *Point of Sale* Berbasis *Web* pada Toko *Project Salfa Tarakan*” [11]. Pada penelitian tersebut, pengembangan aplikasi yang dilakukan dengan latar belakang permasalahan sulitnya pemilik toko dalam melakukan rekapitulasi laporan penjualan dan pemasukan akibat menggunakan sistem pencatatan yang manual. Pengembangan *webiste* POS pada penelitian ini menggunakan metode *waterfall*. Proses pengembangan *website* pada penelitian ini dimulai dengan merancang kebutuhan sistem melalui wawancara dengan pemilik toko, lalu merancang kebutuhan *hardwere* dan sistem, setelah itu implementasi, integrasi, pengujian, dan operasional serta perawatan. Selama pengembangan, bahasa pemrograman utama yang digunakan adalah PHP.

Dengan menerapkan metode serupa, yaitu metode *waterfall*. pada tahun 2021 Khalida Nistrina dan Anisa Rahmania melakukan penelitian dengan judul “Sistem Informasi *Point Of Sale* Berbasis *Website* studi Kasus: PT Barokah Kreasi Solusindo (Artpedia)” [14]. Pada penelitian ini, ditemukan beberapa masalah dalam pengelolaan transaksi dan data di PT Barokah Kreasi Solusindo (Artpedia), termasuk selisih stok barang, keterlambatan dalam proses *order*, serta kesulitan dalam manajemen data transaksi penjualan dan pembelian. Akibatnya, pelayanan kepada pelanggan tidak maksimal, dan pelaporan memerlukan waktu yang lama. Penggunaan sistem *Point of Sale* (POS) diperlukan untuk mengatasi masalah-masalah ini, meningkatkan efisiensi, menghindari kesalahan, dan memberikan

notifikasi terkait penawaran perusahaan. Implementasi POS akan membantu perusahaan dalam mengoptimalkan operasionalnya, meningkatkan kinerja, dan memberikan layanan yang lebih baik kepada pelanggan. Pada pengembangannya, sistem POS dibuat menggunakan kerangka kerja atau *framework Laravel*. Proses pengembangan yang dilakukan pada penelitian ini adalah pengumpulan data, analisis sistem, lalu *design* dan pengkodean, dan implementasi.

Penelitian mengenai pengembangan aplikasi *Point of Sale* juga dilakukan oleh Eko Junirianto dan Nia Kurniadin pada tahun 2020 dengan judul “Pengembangan Aplikasi *Point Of Sale* Berbasis Android Menggunakan Metode *Rapid Application Development*” [15]. Penelitian ini dilakukan dengan latar belakang diperlukannya aplikasi POS dalam bidang usaha industri perdagangan untuk mengatasi masalah kesalahan manusia dalam pendataan produk dan transaksi. Sistem POS diharapkan mampu mengurangi kesalahan dan meningkatkan efisiensi dalam pendataan barang dan pemesanan produk. Pada penelitian ini digunakan metode *Rapid Application Development* (RAD) digunakan untuk mengembangkan sistem POS dengan waktu yang singkat, efisiensi, dan produktivitas yang tinggi. Tujuan penelitian ini adalah membangun sistem aplikasi POS dengan metode RAD dan pendekatan berorientasi objek untuk meningkatkan efisiensi transaksi dan pembuatan laporan. Penelitian ini mengembangkan aplikasi POS dalam *platform mobile* dan *website* dengan tahapan perancangan untuk sistem *website*, perancangan API, perancangan untuk sistem *mobile*, lalu integrasi dan pengujian.

Pada tabel 2.1 dijelaskan ringkasan mengenai penelitian terdahulu yang relevan dengan dengan penelitian yang akan dilakukan oleh peneliti

**Tabel 2. 1** Kajian Pustaka

No	Judul dan Penulis	Hasil	Perbedaan
1	Analisis Desain Dan Pengembangan <i>Point Of Sales</i> Astakona Anugerah Abadi Menggunakan Metode <i>Extreme Programming</i> oleh Calvin Pang dan Muhammad Ardiansyah(2023) [12]	Aplikasi POS yang dikembangkan dengan metode XP bertujuan untuk meminimalisir penggunaan kertas dan menggantinya menjadi penggunaan <i>web</i> .	Pada penelitian ini belum menerapkan PWA yang akan diterapkan pada penelitian ini
2	Perancangan Aplikasi <i>Point Of Sale</i> Berbasis <i>Web</i> pada Belig Coffe Menggunakan <i>Extreme Programming</i> oleh Tomi Rizmawan, Daffa Satria Maulana, Bara Pentium Alarsy, dan Roeslan Djulatov (2023) [10]	Penelitian ini menjadikan aplikasi POS sebagai solusi untuk permasalahan dalam sistem kasir manual, termasuk risiko duplikasi nomor transaksi dan kehilangan, rusak, atau tercecernya setruk.	Pada penelitian ini belum menerapkan PWA yang akan diterapkan pada penelitian ini
3	Pengembangan Sistem <i>Point of Sale</i> Berbasis <i>User Centered Design</i> oleh Muhammad Rafi Raihan, dan Deny Hidayatullah (2022) [13]	Aplikasi <i>Point of Sale</i> belum digunakan di UD Tebar Gembira, yang masih mengandalkan sistem manual dalam pengelolaan data penjualan dan laporan.. Penggunaan program <i>point of sale</i> diharapkan mempermudah pencarian data,	Penelitian ini menggunakan bahasa <i>Javascript</i> dengan <i>framework</i> Next Js dengan menerapkan <i>Responsive Progressive Web app</i> dan menerapkan metode <i>Extreme Programming</i>

		mengelola transaksi, dan menghemat waktu dengan akurasi.	
4	Perancangan <i>Aplikasi Point of Sale</i> Berbasis <i>Web</i> pada Toko <i>Project Salfa Tarakan</i> oleh Suprianto, Muhammad Fadlan , dan Denis Prayogi (2021) [11]	Membuat sistem POS berbasis <i>website</i> untuk mempermudah rekapitulasi laporan penjualan dan pemasukan dengan metode <i>waterfall</i>	<i>Website</i> yang akan dibuat oleh penulis dibuat menggunakan bahasa pemrograman <i>Javascript</i> dan menerapkan <i>Progressive Web App</i> dan metode yang digunakan adalah <i>Extreme Programming</i>
5	Sistem Informasi <i>Point Of Sale</i> Berbasis <i>Website</i> studi Kasus: PT Barokah Kreasi Solusindo (Artpedia) oleh Khalida Nistrina dan Anisa Rahmania (2021) [14]	Sistem POS yang telah dibangun menampilkan laporan dan grafik untuk dengan tujuan utama sistem ini adalah untuk menyederhanakan manajemen data yang telah diotomatisasi dan memungkinkan potensi pengembangan ke arah sistem berbasis <i>mobile</i> di masa depan.	<i>Framework</i> yang digunakan di penelitian ini adalah <i>Laravel</i> , sedangkan pada penelitian yang akan dilakukan oleh penulis menggunakan <i>Next.js</i> dan menerapkan <i>Progressive Web App</i> dan metode yang digunakan adalah <i>Extreme Programming</i>
6	Pengembangan Aplikasi <i>Point Of Sale</i> Berbasis Android Menggunakan Metode <i>Rapid Application Development</i> oleh Eko Junirianto dan Nia Kurniadin (2020) [15]	Membuat sistem POS dengan metode RAD dan dikembangkan di dua platform yaitu Android dan <i>Website</i>	Pengembangkan aplikasi POS yang akan dibuat penulis merupakan aplikasi berbasis <i>website</i> yang <i>responsive</i> dengan berbagai perangkat dan menerapkan menerapkan <i>Progressive Web App</i> untuk mendapatkan hasil layaknya aplikasi <i>mobile</i> dan metode yang digunakan adalah <i>Extreme Programming</i>

## 2.2 Landasan Teori

Pada penelitian ini, diperlukan pemahaman mengenai beberapa teori yang relevan dan akan digunakan selama proses pengembangan aplikasi *Point of Sale* dalam penelitian ini.

### 2.2.1 *Point of Sale*

Salah satu sistem yang berguna untuk mendukung proses transaksi pada yaitu sistem *Point Of Sale*[16]. Aplikasi *Point Of Sale* atau yang sering disingkat POS adalah suatu sistem informasi yang berfungsi sebagai alat pencatatan transaksi penjualan dan terintegrasi dengan perangkat kasir [17]. POS merujuk pada aktivitas yang difokuskan pada penjualan, bertujuan untuk mempermudah proses transaksi dengan mencakup aspek-aspek seperti pengelolaan persediaan dan pelaporan [18]. Oleh karena itu, sistem ini tidak hanya digunakan untuk merinci dan mencatat transaksi penjualan dengan lebih akurat, tetapi juga untuk efisien mengelola stok barang sesuai dengan kebutuhan bisnis yang spesifik [19].

### 2.2.2 *Website*

*Website* pertama kali dikembangkan pada tahun 1989 untuk Particle Physic oleh Tim Barners Lee tepatnya di European Laboratory. Pada awalnya sebuah *website* berfungsi sebagai media untuk berbagi dokumen antar ilmunan [20]. *Website* dapat diartikan sebagai sistem dokumen *hypertext* yang terkait satu sama lain melalui internet yang berarti selama terhubung ke internat, sebuah *website* akan mampus diakses asalkan terdapat *web server* dan *web client* untuk menjalankannya [20].

#### 2.2.2.1 *Respopnsive Website*

*Website* yang responsif adalah jenis desain *website* yang bisa dengan mudah disesuaikan untuk digunakan di berbagai perangkat, termasuk komputer, tablet, dan ponsel. Desain ini secara otomatis menyesuaikan ukuran dan tampilan *website*, yang

membuat pengalaman pengguna lebih nyaman. Selain itu, teknologi responsif dalam desain *web* memberikan peluang baru untuk berintegrasi dengan beragam perangkat yang digunakan [21].

#### **2.2.2.2 Progressive Web App**

*Progressive Web App* merupakan teknologi yang pada tahun 2015 dikembangkan oleh developer dari Google Chrome. *Progressive Web App* yang dikenal dengan PWA merupakan teknik pengembangan perangkat lunak yang menggabungkan teknologi *web*, *desktop*, dan *native* [22].

*Progressive Web Application* atau PWA adalah sebuah teknologi berbasis situs *web* yang dibangun dengan menggunakan teknologi *web* modern, namun memiliki kemampuan untuk berfungsi seperti aplikasi *mobile*. Berbeda dengan aplikasi *native* Android yang terbatas pada sistem operasi Android, PWA dapat dipasang dan digunakan baik pada perangkat dengan sistem operasi Android maupun iOS [23].

PWA adalah aplikasi berbasis *web* yang terlihat seperti halaman web biasa, namun memiliki kemampuan untuk memberikan pengalaman pengguna yang lebih kaya, termasuk kemampuan untuk beroperasi tanpa koneksi internet, memberikan notifikasi, dan mengakses perangkat keras [24].

#### **2.2.3 Service Worker**

PWA bekerja pada sebuah *website* dengan bantuan *service worker*. *Service worker* adalah sebuah skrip yang berjalan di latar belakang browser tanpa ketergantungan pada halaman *web* utama, sehingga tidak memerlukan interaksi dengan halaman web atau pengguna [24]. *Service worker* saat ini mendukung fitur-fitur seperti notifikasi dan sinkronisasi latar belakang. *Service worker* adalah suatu jenis pekerjaan *JavaScript*

yang tidak memiliki kemampuan akses langsung ke model objek dokumen [24].

*Service Worker* merupakan elemen kunci dalam pengembangan aplikasi web progresif (PWA) yang memungkinkan pengendalian terhadap permintaan jaringan, penggunaan *cache* untuk meningkatkan kinerja, serta memungkinkan akses konten dalam mode *offline* [25].

Dengan memanfaatkan *Fetch API*, pengguna dapat mengubah permintaan jaringan dan merespons dengan konten yang telah tersimpan dalam *cache* ketika dalam kondisi *offline* [25]. Selain itu, melalui *Background Sync API*, aktivitas dapat dijadwalkan hingga perangkat pengguna memperoleh konektivitas yang stabil, memastikan pengiriman data yang penting dan pembaruan aplikasi secara teratur saat pengguna terhubung ke internet [25]. Dapat disimpulkan bahwa dengan fitur-fitur ini, *service worker* memainkan peran yang penting dalam menciptakan pengalaman *web* responsif.

#### **2.2.4 User Interface**

*User Interface* (UI) adalah suatu cara atau antarmuka yang digunakan untuk memfasilitasi interaksi antara manusia dan sistem komputer. UI sering kali dianggap sebagai pengganti *Human Computer Interaction* (HCI), mencakup semua bentuk interaksi yang dilakukan oleh manusia terhadap komputer. Dalam konteks ini, UI mencakup elemen-elemen seperti tata letak, elemen grafis, dan kontrol yang memungkinkan pengguna berkomunikasi dengan sistem komputer secara efektif. Inti dari UI adalah memberikan pengalaman pengguna yang intuitif dan efisien, memastikan bahwa interaksi antara manusia dan komputer berjalan dengan lancar dan memenuhi kebutuhan pengguna [26].





## 2.2.5 UML


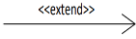

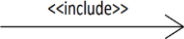
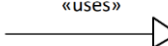
UML (*Unified Modeling Language*) adalah sebuah bahasa berbasis gambar yang digunakan untuk visualisasi, penjelasan, perancangan, dan dokumentasi sistem perangkat lunak yang menggunakan pendekatan berorientasi objek [27]. UML juga memberikan standar untuk membuat panduan sistem, termasuk konsep proses bisnis, perancangan kelas dalam bahasa pemrograman tertentu, struktur basis data, serta elemen-elemen yang diperlukan dalam pengembangan perangkat lunak [28].

### 2.2.5.1 Diagram *Use Case*

Diagram *use case* merupakan alat pemodelan yang digunakan untuk menggambarkan bagaimana satu atau lebih aktor berinteraksi dengan sistem informasi yang sedang dikembangkan. Tujuan dari *use case* adalah untuk mengidentifikasi dan mendokumentasikan fungsionalitas yang ada dalam sistem tersebut, sekaligus menentukan siapa yang berperan dalam menggunakan fungsionalitas tersebut [28]. Beberapa simbol yang sering digunakan dalam *use case* dapat dilihat pada Tabel 2.2 dibawa ini:

**Tabel 2. 2** Simbol-Simbol *Use Case Diagram*[28]






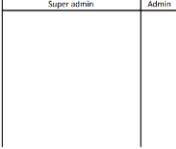
Simbol	Nama	Deskripsi
	<i>Use Case</i>	Fungsionalitas sistem direpresentasikan sebagai unit-unit yang berkomunikasi melalui pertukaran pesan, dengan ciri khas kata kerja di awal nama <i>use case</i> .
	Aktor	Aktor adalah entitas, baik berupa individu, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang sedang dibangun. Meskipun simbol aktor mungkin menggambarkan manusia, aktor dapat mewakili berbagai

		jenis entitas. Biasanya, aktor diidentifikasi dengan kata benda di awal nama aktor.
	Asosiasi	Hubungan antara aktor dan use case yang terlibat dalam suatu skenario <i>use case</i> atau <i>use case</i> yang memiliki interaksi dengan aktor.
	Ekstensi	Relasi use case tambahan adalah ketika suatu <i>use case</i> tambahan ditambahkan ke dalam <i>use case</i> yang ada, dan <i>use case</i> tambahan tersebut memiliki kemampuan untuk berdiri sendiri, tanpa tergantung pada <i>use case</i> utama yang menjadi tujuannya. Konsep ini serupa dengan prinsip warisan dalam pemrograman berorientasi objek.
	Generalisasi	Hubungan generalisasi dan spesialisasi dalam use case mengindikasikan bahwa ada hubungan "umum-khusus" antara dua use case, di mana satu use case adalah versi yang lebih generik atau umum dari yang lain. Use case yang lebih generik disebut use case induk, sementara yang lebih spesifik adalah use case anak. Use case anak mewarisi fungsi dari use case induk, tetapi mungkin juga memiliki tambahan atau variasi dalam fungsionalitasnya.
 	<i>Includes / Uses</i>	Relasi <i>use case</i> tambahan adalah hubungan di mana suatu <i>use case</i> membutuhkan <i>use case</i> utama sebagai syarat atau prasyarat agar dapat berfungsi.

### 2.2.5.2 Diagram Activity

Diagram aktivitas, atau yang juga dikenal sebagai diagram kegiatan, digunakan untuk menggambarkan aliran kerja atau aktivitas dalam suatu sistem, proses bisnis, atau menu perangkat lunak. Poin utama dari diagram aktivitas adalah untuk menunjukkan aktivitas-aktivitas yang dapat dilakukan oleh sistem, daripada fokus pada apa yang dilakukan oleh aktor-aktor yang terlibat [28]. Simbol-simbol yang sering digunakan dalam diagram *activity* dapat dilihat melalui Tabel 2.3 bawah ini:





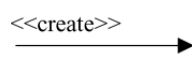
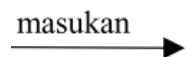
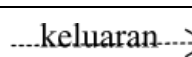
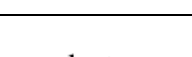
**Tabel 2. 3** Simbol-Simbol *Activity Diagram* [28]

Simbol	Nama	Deskripsi
	<i>Initial State</i>	Status awal atau Initial State adalah kondisi awal yang terjadi ketika sistem pertama kali beroperasi.
	<i>Final State</i>	Status Akhir atau Final State adalah kondisi akhir dari siklus atau tahap tertentu dalam suatu proses atau daur hidup.
	Aktivitas	Aktivitas adalah tindakan yang terjadi di dalam sistem dan biasanya dimulai dengan kata kerja.
	Percabangan	Percabangan adalah aktivitas di mana terdapat berbagai pilihan kegiatan yang dapat diambil.
	Penggabungan	Asosiasi penggabungan adalah proses di mana lebih dari satu aktivitas digabungkan menjadi satu aktivitas tunggal.
	<i>Swimlane</i>	Ini digunakan untuk memisahkan unit atau entitas organisasi yang bertanggung jawab terhadap berbagai aktivitas yang terjadi.

### 2.2.5.3 Diagram Sequence

*Sequence diagram* berfungsi untuk menggambarkan urutan interaksi di antara komponen-komponen sistem. Diagram ini memungkinkan kita untuk menjelaskan interaksi yang terjadi saat sebuah *use case* dilaksanakan dan urutan di mana interaksi tersebut berlangsung. *Sequence diagram* menyajikan banyak informasi mengenai interaksi, namun keunggulan utama dari diagram ini adalah kemampuannya dalam menyampaikan urutan peristiwa dalam sebuah interaksi dengan cara yang sederhana dan jelas [29]. Simbol-simbol yang sering digunakan dalam diagram *sequence* dapat dilihat melalui Tabel 2.4 bawah ini:

**Tabel 2. 4** Simbol-Simbol *Sequence Diagram*

Simbol	Nama	Deskripsi
 nama aktor	<i>Aktor</i>	Orang atau proses yang akan berinteraksi dengan sistem yang akan dikembangkan atau dibuat.
	<i>Life Line</i>	Garis yang berguna untuk menyatakan kehidupan suatu objek
 objek:kelas	<i>Object / Class</i>	Objek atau kelas yang berisi berguna untuk mengirim ataupun menerima pesan
	<i>Active Object</i>	Berfungsi sebagai penanda bahwa objek dalam keadaan aktif.
	Pesan tipe <i>Create</i>	Menyatakan <i>object</i> yang mengirim pesan
	Pesan tipe <i>Input</i>	Menyatakan <i>object</i> yang mengirim masukan
	Pesan tipe <i>Output</i>	Menyatakan <i>object</i> yang mengirim hasil
	Pesan tipe <i>destroy</i>	Menyatakan <i>object</i> yang mengakhiri siklus <i>object</i> lain

## 2.2.6 Framework

*Framework* adalah kerangka kerja yang digunakan dalam pengembangan *website*. Menggunakan *framework* dapat memperpendek waktu pembuatan *website*, mempermudah perbaikan, dan memberikan berbagai keuntungan, seperti menyediakan struktur yang baik dalam program karena *framework* memiliki *library* dan fungsi yang siap digunakan [30].

### 2.2.6.1 Next Js

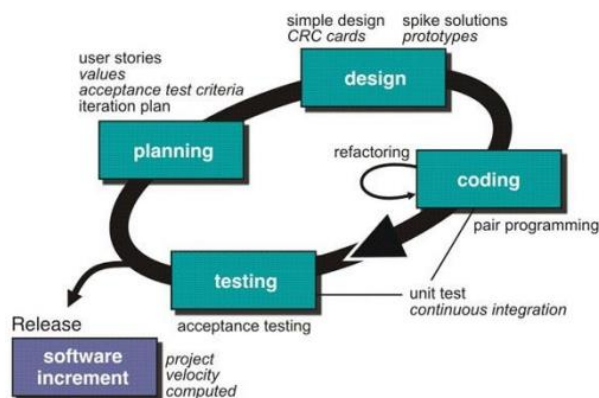
Next.js adalah suatu kerangka kerja React yang digunakan dalam pembangunan aplikasi *web*. Dalam pemanfaatannya, digunakan komponen React untuk pembangunan antarmuka pengguna, sedangkan Next.js menyediakan berbagai tambahan dan optimalisasi. Selain itu, Next.js juga secara otomatis mengelola beragam alat yang diperlukan dalam pengembangan React, termasuk proses *bundling* dan kompilasi, sehingga memungkinkan fokus utama diberikan pada tahap pengembangan aplikasi daripada menghabiskan waktu dalam konfigurasi [31].

### 2.2.6.2 Tailwind CSS

Tailwind CSS adalah sebuah kerangka kerja CSS yang menawarkan pendekatan alternatif dibandingkan dengan gaya penulisan CSS tradisional. Saat ini, Tailwind CSS semakin mendapatkan perhatian dari para pengembang, dan memiliki tingkat kepuasan yang tinggi di antara para pengembang yang telah mencobanya. Kerangka kerja ini memberikan berbagai kelas utilitas CSS tingkat rendah yang dapat digunakan untuk membangun antarmuka dengan lebih efisien [32].

### 2.2.7 Extreme Programming

*Extreme Programming* atau dapat disingkat XP adalah sebuah metode pengembangan perangkat lunak yang sederhana yang pertama kali dikembangkan oleh Kent Beck, Ron Jeffries, dan Ward Cunningham. XP adalah salah satu metode yang paling populer dan telah dikenal luas. Salah satu tujuan XP adalah untuk mengatasi kebutuhan yang tidak jelas dan perubahan kebutuhan yang cepat [33]. *Extreme Programming* (XP) adalah sebuah metodologi pengembangan perangkat lunak yang termasuk dalam kelompok metodologi *agile*. Metodologi ini mengutamakan kegiatan pengkodean sebagai hal utama yang dilakukan sepanjang seluruh tahap dalam siklus pengembangan perangkat lunak [34]



**Gambar 2. 1** Fase Metode XP [34]

Tahapan dari metode *Exteme Programming* sesuai dengan gambar 2.1 adalah sebagai berikut:

#### 2.2.7.1 Planning

Tahapan perencanaan atau *planning* merupakan langkah awal dalam pembangunan sistem dengan metode XP, di mana beberapa kegiatan perencanaan dilakukan, seperti identifikasi permasalahan, analisis kebutuhan, hingga penetapan jadwal pelaksanaan pembangunan sistem [35].

Pada tahapan *planning*, perencanaan dilakukan melalui *planning meeting*. Pertemuan dimulai dengan meninjau kebutuhan pengguna, dan setiap anggota tim dapat mengajukan pertanyaan untuk memahami sepenuhnya kebutuhan pengguna. Selanjutnya, tim menyampaikan dan menetapkan estimasi waktu penyelesaian. Di akhir pertemuan, pengguna membuat keputusan mengenai fitur mana yang akan diimplementasikan [36]. Luaran yang diharapkan dari tahap *planning meeting* dapat berupa list masalah ataupun daftar kebutuhan fitur pengguna [35]. Tabel di bawah ini merupakan contoh *output* dari tahapan *planning meeting*:

**Tabel 2. 5** Contoh Kebutuhan Pengguna [37]

<b>Jenis Kebutuhan</b>	
<b>Kebutuhan Pengguna</b>	<b>Kebutuhan Sistem</b>
Pengguna dapat mengakses halaman <i>website</i> dan mengisi <i>form login</i>	Sistem mengalihkan ke halaman <i>dashboard</i> setelah <i>login</i>
Pengguna dapat memasukkan data barang	Sistem menampilkan stok barang yang tersedia
Pengguna dapat memasukkan data transaksi	Sistem akan Kembali ke halaman <i>login</i> jika pengguna menekan tombol <i>logout</i>

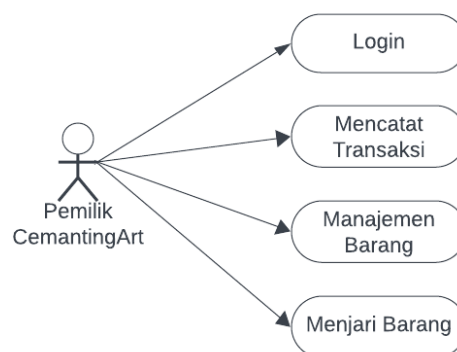
Dalam setiap iterasi, tim pengembangan dan pelanggan bertemu untuk mendefinisikan pekerjaan yang akan diselesaikan melalui pengembangan "*user stories*". *User story* merupakan deskripsi pengalaman atau aksi pengguna pada situs *web*. Setiap deskripsi *user story* umumnya terdiri dari satu hingga tiga kalimat. Kegiatan ini merupakan kelanjutan dari perencanaan yang dilakukan pada tahap awal, dimana *user stories* memberikan landasan konkret untuk pekerjaan yang akan dilakukan dalam setiap iterasi [36]. Contoh kalimat *user story* misalnya adalah :

**Tabel 2. 6** Contoh User Story [36]

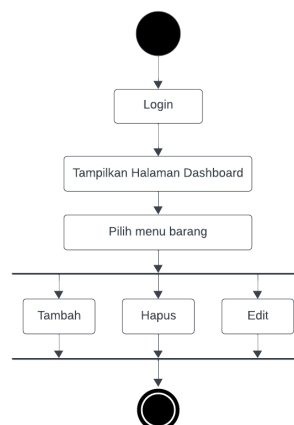
No	User Story
1	sistem yang dibuat mengizinkan pemilik usaha menginputkan data transaksi yang terjadi
2	Sistem yang dibuat harus mampu <i>update</i> otomatis ketika ada barang yang berkurang ketika pemilik usaha melakukan transaksi

### 2.2.7.2 Design

Dalam tahap perancangan atau tahapan *design*, langkah awal akan melibatkan pembuatan pemodelan sistem berdasarkan analisis kebutuhan yang telah dikumpulkan pada tahapan sebelumnya. Penggunaan *Unified Modeling Language* (UML) dengan berbagai jenis diagram seperti diagram *use case* dan diagram *activity* yang akan digunakan dalam pemodelan sistem [10]. Gambar di bawah ini adalah contoh UML menggunakan jenis diagram *use case* dan *activity* diagram:

**Gambar 2. 2** Contoh Diagram *Use Case* [27]





**Gambar 2.3** Contoh Diagram *Activity* [27]

Dalam merancang sistem, tumpuan utama harus diberikan pada prinsip-prinsip kesederhanaan, fleksibilitas, dan kemudahan pemeliharaan. Hal ini diperlukan karena desain sistem bertujuan untuk menciptakan solusi yang tidak hanya efisien dalam pengembangan awal tetapi juga dapat bersifat responsif terhadap evolusi kebutuhan dan perubahan proyek secara keseluruhan [12].

### 2.2.7.3 Coding

Merupakan suatu proses pembuatan perangkat lunak yang melibatkan implementasi sesuai dengan desain yang telah disiapkan sebelumnya. Dalam metodologi *Extreme Programming* (XP), dianjurkan untuk mengembangkan perangkat lunak dengan menerapkan teknik *pair programming* atau melakukan pengerjaan perangkat lunak secara kolaboratif [38]. Dalam tahapan ini perlu dilakukan *coding* yang sesuai standar baik untuk penamaan sampai struktur folder yang baik guna mengurangi kebingungan [36].

Tahapan *coding* dalam pengembangan suatu aplikasi biasanya sistem akan dibagi menjadi dua, yaitu untuk *frontend*

dan *backend* [34]. Pada tahapan pengkodean dari sisi *frontend* atau *coding* akan dilakukan proses penerjemahan dari perancangan UML menjadi sebuah *user interface* dan dari sisi *backend* akan merancang basis data menggunakan bahasa pemrograman. Sehingga *output* dari tahapan ini akan berupa tampilan dan fitur dari sebuah sistem yang akan dibangun [39].

#### 2.2.7.4 Testing

Pengujian atau *testing* merupakan langkah yang dilakukan setelah selesai proses pengkodean. Tujuan dari *testing* yang dilakukan adalah untuk mengevaluasi kinerja sistem yang telah dikembangkan dan memastikan bahwa sistem tersebut berfungsi dengan baik sesuai dengan kebutuhan pengguna. Pengujian dilakukan untuk memastikan kualitas dan kelayakan sistem [40]. Pada tahapan ini dibuat daftar *test case* dari daftar halaman dan masing – masing fitur yang ada di halaman tersebut untuk dilakukan pengujian [39].

Ada beberapa cara untuk melakukan pengujian sistem, salah satunya dengan menggunakan metode *blackbox*. Keuntungan dari metode ini adalah pada penentuan penguji atau *tester* tidak harus memiliki pengetahuan akan bahasa pemrograman[41]. Tabel di bawah ini merupakan contoh halaman dan fitur yang akan dilakukan pengujian menggunakan *blackbox testing* :

**Tabel 2. 7** Contoh Daftar Pengujian Fitur pada Halaman login [39]

No	Proses	<i>Expected Output</i>
1	Mengakses halaman <i>website</i> di <i>browser</i>	Menampilkan halaman <i>login</i>
2	Mengisi <i>form login</i> dan klik tombol login	Mengalihkan ke halaman <i>dashboard</i>

### 2.2.8 Blackbox Testing

*Blackbox testing* adalah metode pengujian perangkat lunak yang dilakukan dengan memberikan masukan ke sistem dan memeriksa apakah *output* yang dihasilkan sesuai dengan yang diharapkan atau tidak [42]. *Black Box Testing* melakukan pengujian perangkat lunak tanpa perlu memeriksa detail program perangkat lunak. Dalam pengujian ini, fokusnya adalah pada fungsi-fungsi yang terlihat dari luar program [42]. Salah satu metode *blackbox* yang cukup populer adalah *Equivalence partitioning*, Metode ini didasarkan pada pemisahan masukan dan keluaran suatu komponen sesuai spesifikasi. Asumsinya adalah masukan yang sama akan menghasilkan respons yang sama [43]. Tabel di bawah ini merupakan contoh *test case* pada pengujian dengan *Equivalence partitioning* pada pengujian *blackbox*:

**Tabel 2. 8** Contoh *Test Case* [44]

Kode	Skenario	Ekspetasi <i>Output</i>	Hasil Pengujian
1	Mengakses halaman <i>website</i> di <i>browser</i>	Menampilkan halaman <i>login</i>	Berhasil
2	Mengisi <i>form login</i> dan klik tombol <i>login</i>	Mengalihkan ke halaman <i>dashboard</i>	Berhasil

Hasil dari metode ini akan melihat apakah fungsi pada suatu halaman sudah bekerja dengan baik dan sesuai harapan. Dengan demikian akan diketahui fitur apa saja yang belum sesuai dengan ekspektasi yang diharapkan dan akan diperbaiki [45]. Perhitungan nilai kesuksesan dalam pengujian diperoleh dengan perhitungan seperti pada rumus 2.1 [43] :

$$\text{Nilai Kesuksesan} = \frac{\text{pengujian sukses}}{\text{total pengujian}} \times 100 \quad (2.1)$$

Berdasarkan ketentuan perhitungan di atas akan diketahui persentase kesuksesan pengujian halaman dan fitur di dalamnya dengan cara

membandingkan jumlah pengujian sukses dengan banyaknya pengujian yang dilakukan [43].