

BAB II LANDASAN TEORI

2.1 *State of The Art*

Penelitian-penelitian sebelumnya yang relevan berperan sebagai acuan dalam memperjelas arah dan ruang lingkup penelitian ini, sehingga dapat dilakukan secara lebih terarah dan mendalam. Adapun beberapa penelitian terkait disajikan dalam tabel berikut.

Tabel 2. 1 *State of The Art*

No	Nama Peneliti	Judul Penelitian	Metode Penelitian	Hasil Pembahasan
1.	Calvin Pang, Muhammad Ardiansyah Universitas Internasional Batam JURNAL RESPONSIF, Vol. 5 No.1 Februari 2023, pp. 52~60 E-ISSN : 2685- 6964	Penerapan Metode <i>Extreme Programming</i> pada Perancangan Aplikasi <i>Point of Sales</i> Berbasis Android (Studi Kasus Kedai Kopi Mokara) https://www.journal.mediapublikasi.id/index.php/oktal/arti	<i>Extreme Programming</i>	Hasil dari penelitian ini meliputi aplikasi untuk penjualan, serta mengedepankan kesimpulan dan respon dari pelanggan.

No	Nama Peneliti	Judul Penelitian	Metode Penelitian	Hasil Pembahasan
		cle/view/1943/2331		
2.	Heri Setiawan, Susana Dwi Yulianti Kusuma Fakultas Ilmu Komputer, Teknik Informatika, Universitas Pamulang OKTAL : Jurnal Ilmu Komputer dan Science Volume 2, No.12, Desember 2023 ISSN 2828-2442 (media online) Hal 3268-3282	Rancang Bangun Aplikasi <i>Point of Sale</i> Berbasis Android Menggunakan <i>Extreme Programming</i> (Studi Kasus: Beringin Laundry Purwakarta) https://ejournal.itn.ac.id/index.php/jati/article/view/10875	<i>Extreme Programming</i>	Hasil penelitian ini yaitu pengembangan sistem <i>mobile</i> berbasis Android untuk mendukung operasional kedai kopi Mokara. Sistem ini mencakup absensi, transaksi penjualan, dan rekapitulasi laporan keuangan, sehingga mempermudah pencatatan kehadiran, pengelolaan penjualan, serta pemantauan keuangan secara lebih efisien.
3.	Rifqi Fadhilah, Yudhi Raymond	Rancang Bangun Aplikasi <i>Point</i>	<i>Extreme Programming</i>	Penelitian ini menghasilkan aplikasi <i>Point Of</i>

No	Nama Peneliti	Judul Penelitian	Metode Penelitian	Hasil Pembahasan
	Ramadhan,Lise Sri AndarMuni Teknik Informatika, Sekolah Tinggi Teknologi Wastukencana Purwakarta JATI (Jurnal Mahasiswa Teknik Informatika) Vol. 8No. 5, Oktober 2024 ISSN : 2598- 828X	<i>of Sales</i> Menggunakan Metode <i>Extreme</i> <i>Programming</i> (Studi Kasus Hadi Sport Shop) https://jurnal.uantan.ac.id/index.php/justin/article/view/78761/75676602885		<i>Sale</i> berbasis Android yang dapat melakukan pencatatan transaksi, nota digital yang mengurangi penggunaan kertas (<i>paperless</i>), dan pembuatan laporan otomatis
4.	Maulana Dimyatia dan Alon Jala Tirta Program Studi Rekayasa Perangkat Lunak, Institut	Analisa dan Perancangan Sistem <i>Point</i> <i>Of Sale</i> Menggunakan Metode <i>Extreme</i> <i>Programming</i>	<i>Extreme</i> <i>Programming</i>	Hasil penelitian menunjukkan bahwa aplikasi Hadi Sport <i>Point</i> <i>Of Sale</i> (POS) mampu mengatasi masalah pencatatan dan pencarian

No	Nama Peneliti	Judul Penelitian	Metode Penelitian	Hasil Pembahasan
	Teknologi Telkom Purwokerto Jurnal Sistem dan Teknologi Informasi p-ISSN : 2460- 3562 / e-ISSN : 2620-8989 DOI:10.26418/j ustin.v12i3.7876 1 Vol. 12, No. 3, Juli 2024 p	dengan PHP Native Berbasis <i>Website</i> https://journal. mediapublikasi .id/index.php/o ktal/article/vie w/3255/2962		barang. Aplikasi ini memungkinkan pembaruan data barang secara otomatis dan pencarian produk yang lebih cepat, sehingga meningkatkan efisiensi kerja kasir.
5.	Rahmat Ali Saif Fakultas Ilmu Komputer, Teknik Informatika, Universitas Pamulang, OKTAL : Jurnal Ilmu Komputer dan Science Volu me 3, No.8 Agustus 2024	Rancang Bangun Sistem Informasi <i>Point Of Sales</i> (POS) https://reposito ry.nusamandiri .ac.id/repo/file s/248165/down load/Rancang- Bangun- Sistem- Informasi-	<i>Extreme Programming</i>	Penelitian ini menghasilkan sebuah sistem <i>Point of Sale (POS)</i> yang kegunaanya sudah sesuai dengan kebutuhan transaksi dan pengelolaan stock barang pada Toko Amar, yaitu proses transaksi yang lebih efisien dan stock barang yang

No	Nama Peneliti	Judul Penelitian	Metode Penelitian	Hasil Pembahasan
	ISSN 2828-2442(media online) Hal 1925-1932	Point-of-Sales-(POS).pdf		lebih terdata dengan baik.
6.	Dony Oscar, Hendri , Jefi , Muchammad Ichwanul Muslim , Muhammad Fahmi Universitas Bina Sarana Informatika, Indonesia Jurnal Vocational Teknik Elektronika dan Informatika http://ejournal.unp.ac.id/index.php/voteknika/index Vol. 11, No. 3, September 2023	Rancang Bangun Sistem Informasi <i>Point Of Sales</i> (POS) https://repository.nusamandiri.ac.id/repository/file/248165/download/Rancang-Bangun-Sistem-Informasi-Point-of-Sales-(POS).pdf	<i>Extreme Programming</i>	Penelitian ini menghasilkan sistem <i>Point Of Sales</i> (POS) yang dirancang berhasil membantu pengelola toko dalam menjalankan transaksi, mempermudah pengelolaan laporan transaksi, serta menyimpan data dengan lebih aman dan terkelola.

No	Nama Peneliti	Judul Penelitian	Metode Penelitian	Hasil Pembahasan
	P- ISSN: 2302-3295, E-ISSN : 2716-3989			
7.	Handhira Bayu Pradhana Putra, Sulistyo Dwi Sancoko Program Studi Informatika Universitas Teknologi Yogyakarta Infotek : Jurnal Informatika dan Teknologi Vol. 7No. 1, Januari 2024 Hal.195-204e- ISSN 2614-8773	Penerapan Sistem <i>Point Of Sales</i> (POS) Berbasis Android Untuk Peningkatan Kinerja Usaha https://core.ac. uk/reader/5993 69930	<i>Extreme Programming</i>	Penelitian ini menemukan bahwa aplikasi <i>Point Of Sales</i> (POS), yang berbasis Android, membantu mengelola toko Mapan dengan lebih mudah. Hasil pengujian sistem dengan metode <i>blackbox</i> menunjukkan bahwa aplikasi ini bekerja dengan baik dan semua fungsi berfungsi dengan baik.
8.	Agus Suharto, Lia Winarti Universitas Pamulang Program Studi Teknik	Rancang Bangun Sistem <i>Point Of Sales</i> (POS) Dengan Metode Personal	<i>Extreme Programming</i>	Temuan penelitian ini, aplikasi <i>Point Of Sales</i> (POS) yang dikembangkan dengan metode

No	Nama Peneliti	Judul Penelitian	Metode Penelitian	Hasil Pembahasan
	Informatika STMIK Eresha, Jurnal Teknologi Informasi ESIT Vol. XV No. 01April 2020	<i>Extreme Programming</i> (Studikamus:Ke dairatu) https://openjournal.unpam.ac.id/index.php/ESIT/article/view/14040		Personal <i>Extreme Programming</i> berhasil mengatasi masalah pencatatan manual, mempercepat proses transaksi, serta memungkinkan pengembangan sistem yang fleksibel sesuai kebutuhan pengguna.
9.	Cavell Angga Widjaja,Christia n Timotius Wahyudi, Violitt a Yesmaya, DenniseAdriant o ComputerScienc e Department, School of Computer Science, Bina	<i>Analysis and Reporting System Point of Sales for Food and Beverage (F&B)</i> https://www.semanticscholar.org/paper/Analysis-and-Reporting-System-Point-	<i>Extreme Programming</i>	Temuan penelitian ini mengindikasikan bahwa sistem <i>Point Of Sales</i> (POS) yang berplatform iOS mampu meningkatkan efisiensi dan keamanan dalam operasional restoran. Hal ini dilakukan melalui

No	Nama Peneliti	Judul Penelitian	Metode Penelitian	Hasil Pembahasan
	Nusantara University, (E-ISSN 2250-2459, Scopus Indexed, ISO 9001:2008 Certified Journal, Volume 11, Issue 12, December 2021)	of-Sales-for-Widjaja-Wahyudi/5652b5bd5b9fe0cd08e4e88dcad0d76e780bdabb		fitur analisis penjualan, pemantauan staf, pengelolaan meja, serta catatan transaksi yang ringkas.
10.	Yusril Aziz, Muhammad Alda Universitas Islam Negeri Sumatera utara, JURNAL INOVTEK POLBENG - SERI INFORMATIK A, VOL.9,NO.2, 2024 ISSN:2527-9866	Aplikasi <i>Mobile System Point Of Sales</i> (POS) Penjualan Kopi pada Langit Coffee Space https://jurnal.polbeng.ac.id/index.php/ISI/article/view/214/66	<i>Extreme Programming</i>	Hasil penelitian ini menunjukkan bahwa aplikasi berhasil meningkatkan efisiensi dengan menurunkan ketidaktepatan dalam pencatatan stok dapat mencapai 30%, serta mempercepat proses penyusunan laporan keuangan hingga 50%, dan memberikan fleksibilitas untuk

No	Nama Peneliti	Judul Penelitian	Metode Penelitian	Hasil Pembahasan
				pengembangan fitur lanjutan dibandingkan aplikasi <i>Point Of Sales</i> (POS) lainnya.
Perbedaan penelitian terdahulu dengan penelitian yang tertulis lakukan adalah :				
11.	Anisa Fitria	Perancangan Aplikasi <i>Point of Sale</i> (POS) Berbasis <i>Website</i> di Restoran Sajiwa Steak Majalaya Menggunakan Metode <i>Extreme Programming</i>	<i>Extreme Programming</i>	Penelitian ini menciptakan sistem <i>Point of Sale</i> (POS) yang mana berbasis <i>website</i> memakai Next.js dengan efisiensi teknologi modern dan dilengkapi dengan fitur utama dashboard laporan penjualan real-time. Selain itu, sistem ini dilengkapi fitur pemantauan log aktivitas untuk manajer, yang meningkatkan

No	Nama Peneliti	Judul Penelitian	Metode Penelitian	Hasil Pembahasan
				pengawasan operasional.

2.2 Perancangan Sistem

Perancangan sistem adalah proses utama dalam pengembangan sistem informasi, yang mencakup proses pembuatan gambar, perencanaan, serta penyusunan atau pengaturan berbagai komponen yang terisolasi menjadi sebuah unit yang kompak serta berperan. Perancangan sistem pun mencakup konfigurasi bagian perangkat lunak dan perangkat keras sistem. Menurut Fauzi et al., (2022) perancangan sistem merupakan serangkaian tindakan yang menjelaskan dengan rinci bagaimana sistem yang baru akan dioperasikan. Tujuan dari rancangan sistem merupakan untuk menetapkan alur serta data yang dibutuhkan untuk sistem baru demi memenuhi kebutuhan pengguna.

Menurut Mulyani dalam (Arman & Maberur, 2022) Perancangan sistem merupakan langkah menentukan proses dan data yang dibutuhkan oleh sistem yang baru. Tujuan dari perancangan sistem adalah untuk memenuhi kebutuhan pengguna sistem serta memberikan gambaran yang jelas dan desain yang menyeluruh.

Sejalan dengan penjelasan tersebut, O'Brien dan Marakas dalam (Vivi, 2022) mengemukakan bahwa perancangan sistem adalah proses yang melibatkan pembuatan dan penentuan cara untuk memproses informasi sistem berdasarkan analisis yang dilakukan, hal ini memungkinkan untuk memenuhi kebutuhan pengguna, mencakup desain tampilan pengguna, data yang diperlukan, serta berbagai aktivitas dalam proses.

2.3 Aplikasi

Perangkat lunak (*software*) atau program komputer yang beroperasi pada sistem dan dirancang untuk melakukan tugas tertentu disebut aplikasi. Istilah "aplikasi" berasal dari kata "penerapan" atau "penggunaan" dalam bahasa Inggris.

Secara harfiah, "aplikasi" adalah implementasi perangkat lunak atau *software* yang dirancang untuk melakukan tugas tertentu.

Menurut Hengky W. Pramana, dalam (Adlan Al Hawari Nasution & Suryana, 2023) Aplikasi merupakan perangkat lunak yang dirancang untuk melakukan banyak hal, seperti layanan masyarakat, bisnis, iklan, *game*, dan banyak lagi.

Secara istilah aplikasi adalah program yang siap untuk digunakan yang memungkinkan penggunaan aplikasi lain untuk mencapai tujuan tertentu dan dirancang untuk melakukan tugas tertentu untuk pengguna jasa aplikasi. Sebagaimana dinyatakan dalam kamus komputer eksekutif, "aplikasi" mengacu pada pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi. Dalam kebanyakan kasus, ini mengacu pada komputasi dan pemrosesan data yang diharapkan atau diinginkan.

2.4 Point Of Sale (POS)

Menurut Nugraha, Wardani, dan Sukarmayasa dalam (Saif, 2024) Sistem *Point of Sale* merupakan sebuah perangkat lunak yang dimanfaatkan oleh usaha minimarket dan toko untuk mengelola informasi terkait transaksi pembelian, penjualan, pengembalian, dan pembelian ulang. Informasi ini umumnya sangat krusial bagi pemilik bisnis swalayan, organisasi, serta usaha kecil.

Point Of Sale (POS) adalah suatu sistem aplikasi yang digunakan di bisnis minimarket atau toko untuk mengelola pengolahan data transaksi pembelian, penjualan eceran, pengembalian barang, dan pelaporan transaksi yang secara umum dibutuhkan dalam pengambilan keputusan strategi oleh pengusaha swalayan, badan organisasi, atau perusahaan kecil dan menengah (Arman & Maberur, 2022).

Desain sistem aplikasi *Point Of Sale* (POS) dapat meningkatkan kualitas layanan kepada konsumen, seperti mempercepat proses perhitungan harga dan jumlah barang yang dibeli, serta mengurangi ketergantungan pada pencatatan manual untuk kuantitas barang. Oleh sebab itu, perusahaan perlu menerapkan sistem otomatis untuk meningkatkan efisiensi dan meminimalkan risiko kesalahan informasi.

2.5 *Website*

Website merupakan sekumpulan halaman yang menyimpan informasi digital seperti teks, gambar, animasi, audio, dan video, yang dapat diakses melalui internet, sehingga orang di seluruh dunia bisa melihatnya. Menurut Maharani (2025) *website* dalam istilahnya adalah sekumpulan halaman web—halaman dalam sebuah *website*, yang biasanya berkumpul dalam satu *domain* atau *subdomain* di dalam *World Wide Web* (WWW), yang merupakan keseluruhan situs web yang bisa diakses publik di internet.

Website tidak hanya berguna untuk menyebarkan informasi, tetapi juga dapat berfungsi sebagai platform untuk berbagai kebutuhan, seperti toko *online*, profil organisasi, media promosi, dan layanan publik. Dengan kemajuan teknologi, mereka sekarang menjadi bagian penting dari pembangunan citra, komunikasi, dan layanan digital di berbagai bidang (Miftahuljannah & Suharso, 2023).

2.6 *Unified Modelling Language (UML)*

Unified Modeling Language merupakan suatu pendekatan atau bahasa yang bersifat pemodelan visual yang digunakan untuk menggambarkan, menentukan, mengembangkan, dan mendokumentasikan sistem perangkat lunak, terutama yang berbasis objek. (Hamas & Imaduddin, 2019)

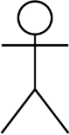
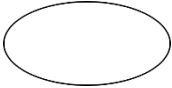
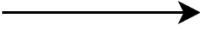
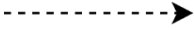
UML juga mempermudah proses analisis dan perancangan aplikasi atau software menjadi lebih mudah karena menyediakan seperangkat aturan dan notasi grafis yang membantu dalam spesifikasi sistem.

2.6.1 *Use Case Diagram*

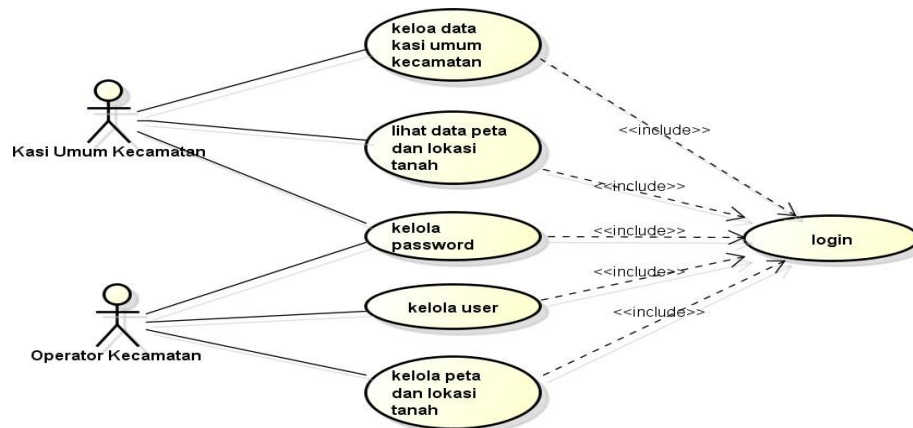
Use Case Diagram adalah salah satu jenis diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk mensimulasikan interaksi antara sistem dan aktor luar yang berfungsi sebagai penggunanya. *Use case* juga berfungsi untuk menerangkan hubungan antara sistem dan lingkungan sekitarnya, serta menggambarkan dan mewakili aktor, *use case*, dan ketergantungan. (Hamas & Imaduddin, 2019)

Berikut ini adalah beberapa simbol yang digunakan saat membuat diagram *use case*.

Tabel 2. 2 Contoh Simbol- Simbol Use Case

Simbol	Nama	Keterangan
	<i>Actor</i>	Digunakan untuk menjelaskan seseorang yang sedang berinteraksi dengan sistem.
	<i>Use Case</i>	Menggambarkan bagaimana sistem bertindak tetapi tidak menunjukkan strukturnya yang dalam.
	<i>Association</i>	Jalur komunikasi di antara aktor dengan <i>use case</i> yang saling terlibat.
	<i>Generalisasi</i>	Menunjukkan keahlian Aktor agar bisa berkontribusi pada <i>use case</i> .
..... <<extend>>	<i>Extend</i>	Menambahkan tindakan ke dalam <i>use case</i> utama yang tidak mengetahui mengenai hal itu.
<include>>	<i>Include</i>	Menambahkan tindakan ke dalam <i>use case</i> utama yang secara jelas menjelaskan penambahan tersebut.

Dengan merujuk pada simbol – simbol yang ada dalam diagram *use case*, maka dapat digambarkan seperti pada contoh di bawah ini.



Gambar 2. 1 Contoh *Use Case Diagram*

Sumber : <https://images.app.goo.gl/5cRXLfcN3cS3yzDr9>

2.6.2 *Class Diagram*







Class diagram merupakan diagram struktur dalam UML yang menampilkan kelas-kelas beserta atribut, metode, dan hubungan antar kelas untuk memodelkan struktur statis sistem secara terstruktur dan terstandar.

Class diagram sangat bermanfaat untuk memvisualisasikan susunan kelas dalam sebuah sistem. Diagram ini sering digunakan dalam pemodelan sistem yang berbasis objek. *Class diagram* memiliki tujuan untuk menggambarkan jenis objek dalam sistem serta keterkaitannya dengan objek lainnya. (Suharni et al., 2023)

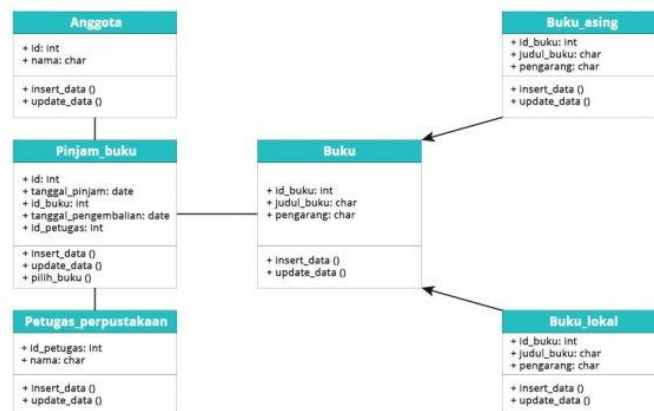
Simbol-simbol yang ada dalam *class diagram* dapat dijelaskan dalam sebuah tabel dibawah ini.

Tabel 2. 3 Simbol – Simbol *Class Diagram*

No	Simbol	Nama	Kegiatan
1.	—	<i>Generalization</i>	Objek anak (<i>descendent</i>) mewarisi sifat dan fungsi dari objek induk (<i>ancestor</i>).
2.		<i>Nary Asociation</i>	Upaya untuk menghindari hubungan dengan lebih dari dua

No	Simbol	Nama	Kegiatan
			hal.
3.		<i>Class</i>	Grup dari objek yang memiliki karakteristik dan operasi yang sama.
4.		<i>Collaboration</i>	Deskripsi dari urutan tindakan yang ditampilkan sistem yang menghasilkan hasil yang dapat diukur untuk seorang aktor
5.		<i>Realization</i>	Suatu objek melakukan operasi yang benar.
6.		<i>Dependency</i>	Hubungan di mana elemen yang tidak mandiri bergantung pada elemen yang mandiri ketika elemen yang mandiri mengalami perubahan.
7.		<i>Association</i>	Apa yang menghubungkan satu objek ke objek lainnya.

Dengan merujuk pada simbol – simbol yang ada dalam diagram *class diagram*, maka dapat digambarkan seperti pada contoh di bawah ini.



Gambar 2. 2 Contoh Class Diagram

Sumber : <https://www.dicoding.com/blog/>

2.6.3 Activity Diagram





Activity diagram adalah salah satu jenis diagram dalam UML yang digunakan untuk memodelkan alur kerja atau tahapan sistem dari awal hingga akhir. Diagram ini menampilkan urutan aktivitas beserta percabangan logika, transisi antar langkah, dan titik awal hingga akhir proses.

Berbeda dengan diagram UML lainnya, activity diagram memiliki kemiripan dengan flowchart tradisional namun dengan notasi dan aturan yang lebih terstandarisasi. (Hamam & Imaduddin, 2019)

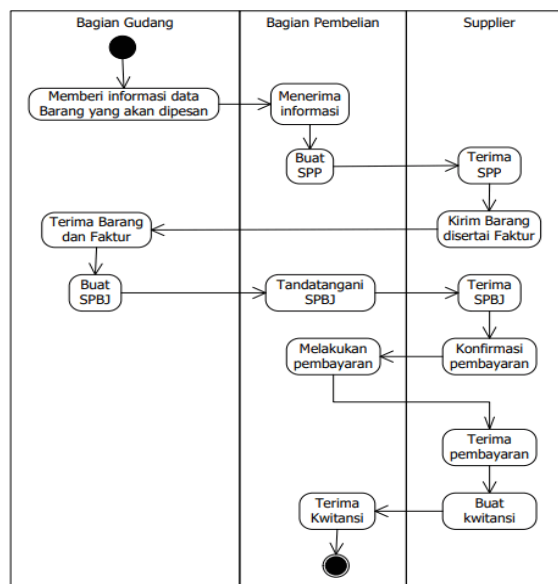
Simbol-simbol yang terdapat pada *Activity Diagram* ditunjukkan sebagai berikut.

Tabel 2. 4 Simbol-simbol Activity Diagram

No	Simbol	Nama	Kegiatan
1.	●	<i>Initial Node</i>	Menunjukkan dimana alur kerja dimulai.

2.		<i>Activity</i>	Menunjukkan cara setiap jenis antarmuka berkomunikasi satu sama lain.
3.		<i>Decision</i>	Menguraikan beberapa kegiatan untuk mencapai suatu keputusan.
4.		<i>Activity</i> <i>Final node</i>	Menggambarkan aktivitas terakhir dari sistem.
5.		<i>Fork Node</i>	Dalam fase tertentu, satu aliran akan bertransformasi menjadi berbagai aliran.

Dengan merujuk pada simbol – simbol yang ada dalam diagram *use case*, maka dapat digambarkan seperti pada contoh di bawah ini.



Gambar 2. 3 Contoh Activity Diagram

Sumber : <https://www.dicoding.com/blog/>

2.7 Framework NextJS

Berikut merupakan bagian yang terdapat pada NextJS :

2.7.1 Pengertian *Framework*

Ian Sommerville (2011) dalam (Maharani, 2025) menyatakan bahwa kerangka kerja memfasilitasi para pengembang untuk memanfaatkan komponen perangkat lunak yang sudah tersedia. Kerangka kerja dibuat untuk memperlancar proses pembuatan perangkat lunak dan memungkinkan pengembang untuk memulai dari awal. Dengan adanya *framework*, memungkinkan pengembang untuk berkonsentrasi pada logika bisnis aplikasi sementara memenuhi kebutuhan dasar – dasar seperti pengelolaan basis data, antarmuka pengguna, dan koneksi jaringan. Ini sesuai dengan gagasan modern tentang rekayasa perangkat lunak yang menekankan penggunaan ulang komponen untuk meningkatkan efisiensi, mengurangi biaya, dan mempercepat proses pengembangan perangkat lunak.

2.7.2 Pengertian NextJS

Framework *React open-source* yang digunakan untuk membuat aplikasi web modern yang cepat, *scalable*, dan ramah SEO. Ini memiliki fitur seperti *rendering server-side*, pembuatan situs statis, dan kemudahan pengelolaan *routing*. Menurut Guillermo Rauch dalam (Maharani, 2025), *next.js* merupakan suatu kerangka kerja React yang dirancang untuk mendukung para pembuat aplikasi dalam menciptakan aplikasi web yang *contemporary*, lebih efektif, multifungsi, dan optimal. Tujuan utama dari *framework* ini adalah untuk memberi pengembang solusi untuk membangun aplikasi *full-stack* dengan performa yang luar biasa tanpa mengorbankan pengalaman pengembang.

2.7.3 Fitur-Fitur NextJS

Next.js menawarkan sejumlah fitur unggulan yang mendukung pengembangan aplikasi web modern yang lebih efisien, antara lain:

1. *Server-Side Rendering (SSR)*: Pendekatan ini memungkinkan halaman web dirender di sisi *server* sebelum dikirim ke *browser*, yang mempercepat waktu muat dan meningkatkan kualitas SEO.

2. Halaman statis dihasilkan saat proses *build*, memungkinkan akses yang sangat cepat dan efisien.
3. *Incremental Static Regeneration (ISR)*: Halaman statis dapat diperbarui secara otomatis tanpa perlu membangun ulang seluruh situs.
4. *Routing* berbasis file: Ini memudahkan pengelolaan navigasi dengan mengubah setiap file dalam folder pages menjadi rute aplikasi.
5. SEO Friendly: SSR dan SSG meningkatkan visibilitas *website* dengan membuat konten lebih mudah diindeks oleh mesin pencari.

2.7.4 Alur Kerja NextJS

Alur kerja Next.js dimulai dengan pembuatan proyek melalui CLI. Selanjutnya, pengembang mengatur struktur folder, khususnya folder halaman, untuk routing otomatis. Saat pengguna mengakses sebuah halaman, Next.js akan menentukan apakah halaman harus dirender di server (SSR), diambil dari hasil build statis (SSG), atau diupdate secara inkremental (ISR). Untuk keperluan dinamis, Next.js juga mendukung rendering client-side (CSR) menggunakan React. Selain itu, Next.js menawarkan *API Routes* untuk *prefetching* otomatis dan logika *backend*, yang meningkatkan kecepatan navigasi antar halaman. Aplikasi tetap responsif dan mudah dikembangkan karena optimalisasi otomatis, yang mencakup pemisahan kode dan optimasi gambar.

2.7.5 Keunggulan NextJS

Keunggulan fitur Next.js meliputi:

1. Performa Tinggi: SSR, SSG, dan optimasi gambar membuat aplikasi lebih cepat dan responsif.
2. Ramah SEO: Konten yang dirender di server mudah diindeks oleh mesin pencari, meningkatkan peringkat *website*.
3. Pengalaman Pengembang Lebih Baik: Konfigurasi minimal, *fast refresh*, dan dokumentasi lengkap mempercepat proses pengembangan.
4. Kemudahan Integrasi API dan *Backend*: API Routes memungkinkan *backend* terintegrasi langsung dalam satu proyek.

2.8 Database

Database, juga dikenal sebagai basis data, merupakan sekumpulan informasi yang diatur dan dikelola dengan cara tertentu sehingga saling berhubungan dan mudah diakses, diatur, dan diubah menggunakan perangkat lunak atau aplikasi khusus. Menurut Joni Kurniawan, (2019) Database memungkinkan orang untuk mencari, menyimpan, memperbarui, dan menghapus data. Database adalah gabungan dari file dan tabel, dengan setiap tabel terdiri dari entri yang diorganisasikan sesuai dengan kolom-kolom yang terdapat di dalamnya.

Menurut (Riyan Dirgantara et al., 2023), Database adalah sekumpulan data yang saling berhubungan dan disimpan dengan pengendalian redundansi untuk mendukung satu atau lebih aplikasi secara efisien. Data tersebut disusun sedemikian rupa agar dapat diakses tanpa memandang program yang digunakan, serta memberikan kemudahan untuk menambah data baru, mengubah data yang sudah ada, dan menarik data dari basis data dengan cara yang teratur dan terkendali.

2.9 MySQL

MySQL merupakan perangkat lunak untuk mengelola basis data yang berbasis relasi dan bersifat *open source*, menggunakan bahasa pemrograman terstruktur SQL untuk menangani, menyimpan, dan memodifikasi informasi yang tersaji dalam tabel yang saling berhubungan. Menurut Edi Winarno & Ali Zaki (2014) dalam (Joni Kurniawan, 2019) bahwa MySQL adalah program database yang bertipe data relasional, yang berarti menyimpan data dalam bentuk tabel-tabel yang saling berhubungan.

2.10 Laragon

Laragon adalah perangkat lunak lokal gratis yang menawarkan lingkungan pengembangan web lengkap dengan berbagai komponen siap pakai seperti Apache/Nginx, MySQL/MariaDB, PHP, dan Node.js. (Sulistiya Putra et al., 2019)

Dengan menggunakan program ini, pengembang dapat membuat, menguji, dan mengelola aplikasi web secara offline tanpa melakukan konfigurasi manual

yang kompleks. Dengan antarmuka yang mudah digunakan dan dukungan alat bantu seperti Composer, Xdebug, dan PHPMyAdmin, Laragon menjadi solusi praktis dan efisien untuk pengembangan web lokal. Fitur unggulan lainnya termasuk dukungan multi-versi PHP, portabilitas, dan konsumsi memori rendah.

2.11 Visual Studio Code

Visual Studio Code adalah aplikasi pengeditan kode yang dibuat oleh Microsoft untuk platform Windows, Linux, dan macOS. Aplikasi ini dilengkapi dengan fitur seperti penyorotan sintaks, penyelesaian kode pintar, potongan kode, pengorganisasian kode, dan integrasi dengan git serta GitHub. Sangat fleksibel, pengguna bisa menyesuaikan tema, pintasan keyboard, preferensi, dan menambahkan ekstensi untuk fitur tambahan (Joni Kurniawan, 2019). *Visual Studio Code* dirancang untuk memudahkan pengembang dalam menulis, mengedit, dan mengelola kode program dengan mendukung beberapa bahasa pemrograman, seperti JavaScript, Python, PHP, Java, C++, dan lain-lain.

2.12 Tailwind CSS

Tailwind CSS adalah *framework* CSS *utility-first* yang memungkinkan kamu membangun antarmuka langsung dari *class-class* siap pakai di HTML. Menurut Nauval Azhar dalam jurnal (Puan Maharani, 2025) tailwind adalah *framework utility-first* yang pertama kali dikembangkan oleh Adam Wathan dan dirilis pada akhir tahun 2017. Seiring waktu, *framework* ini menjadi favorit banyak developer, baik yang fokus di front-end maupun back-end. Karena kemudahannya, Tailwind sangat membantu developer back-end yang kurang mahir dalam CSS. Biasanya, Tailwind dipakai bersama *framework* seperti Laravel, atau JavaScript *framework* seperti React dan Vue.

2.13 Google Chrome

Google Chrome merupakan aplikasi peramban website yang diperlukan untuk menampilkan tata letak halaman web dan berperang sebagai sarana untuk menjelajahi internet (Sonny & Rizki, 2021) . Chrome terkenal dengan antarmuka pengguna yang sederhana dan mudah digunakan, serta kinerja yang cepat dan stabil.

Selain itu, memungkinkan sinkronisasi data seperti *bookmark*, riwayat, dan pengaturan melalui akun *Google*, yang memungkinkan pengguna mengakses data mereka secara mudah di berbagai perangkat.

2.14 Balsamiq Wireframe

Wireframe merupakan suatu struktur desain dari sebuah produk, yang mana proses pembuatannya dilakukan pada fase awal pengembangan produk. Menurut Junilia dalam, (Alfiarto et al., 2024) menjelaskan bahwa dalam tahap wireframe, dibahas berbagai aspek penting seperti fitur, konten, antarmuka, dan elemen-elemen lainnya secara mendetail.

Balsamiq Wireframe juga sebagai alat prototyping yang memudahkan pembuatannya melalui fitur *drag-and-drop* komponen UI, tanpa memerlukan keahlian pemrograman. Alat ini mempercepat iterasi desain, memfasilitasi kolaborasi tim, dan membantu memvisualisasikan konsep produk sejak dini, sehingga dapat menghemat waktu dan mengurangi risiko kesalahan pada tahap pengembangan selanjutnya.

2.15 Blackbox Testing

Blackbox testing merupakan pendekatan dalam menguji perangkat lunak yang difokuskan pada fungsi dan perilaku aplikasi dari perspektif pengguna tanpa harus memahami atau memperhatikan struktur internal, kode program, atau desain sistem. *Black box testing* adalah cara untuk menilai fungsionalitas suatu sistem tanpa mempertimbangkan struktur di dalamnya. Metode ini juga memiliki kelebihan yang menjadikannya lebih unggul dibandingkan beberapa teknik pengujian lainnya. (Ruki et al., 2024).

2.16 Metode Extreme Programming

Extreme programming adalah pendekatan untuk mengembangkan metodologi pengembangan perangkat lunak cepat yang berfokus pada pengkodean, yang merupakan tugas utama dalam semua tahapan siklus pengembangan perangkat lunak Borman dalam jurnal (Muhammad Farhan et al., 2024). *Extreme*

programming menekankan kerja sama tim, perubahan kebutuhan pengguna, dan penerapan prinsip dan teknik praktis dalam pengembangan perangkat lunak sehari-hari.

Extreme programming, menurut (Supriyatna, 2018) adalah salah satu metode pengembangan perangkat lunak yang paling populer dan dikenal luas karena tujuannya untuk menciptakan tim dengan ukuran kecil sampai menengah, tanpa perlu membangun tim besar. Metode ini sangat cocok untuk tim kecil hingga menengah, terutama pada proyek yang mengalami perubahan frekuensi pada kebutuhan dan harapan klien. *Extreme programming* terdiri dari beberapa tahap penting: perencanaan (*planning*), perancangan (*design*), pengkodean (*coding*), dan pengujian (*testing*).

2.17 Tahapan Metode Extreme Programming

Dalam metode *Extreme Programming*, ada beberapa tahapan diantaranya :

1. *Planning*

Pada tahap ini, tim pengembang bekerja sama dengan pihak terkait untuk mengenali kebutuhan dan menentukan elemen-elemen krusial yang perlu dikembangkan. Proses ini mencakup membuat cerita pengguna, menentukan prioritas fitur, dan membuat rencana pengembangan yang terorganisir dan terfokus. Sangat penting untuk melibatkan stakeholder secara aktif untuk memastikan bahwa kebutuhan pengguna dipahami dan dipenuhi saat mengembangkan aplikasi.

2. *Design*

Pada tahap desain, sistem model dibuat berdasarkan analisis kebutuhan yang telah diperoleh. Selain itu, juga dilakukan pembuatan model basis data untuk menunjukkan hubungan antar data. Sebagai bahasa pemodelan sistem, *Unified Modelling Language (UML)*, terdiri dari berbagai diagram, seperti *Use-Case Diagram*, *Activity Diagram*, *Component Diagram*, dan *Deployment Diagram*.

3. *Coding*

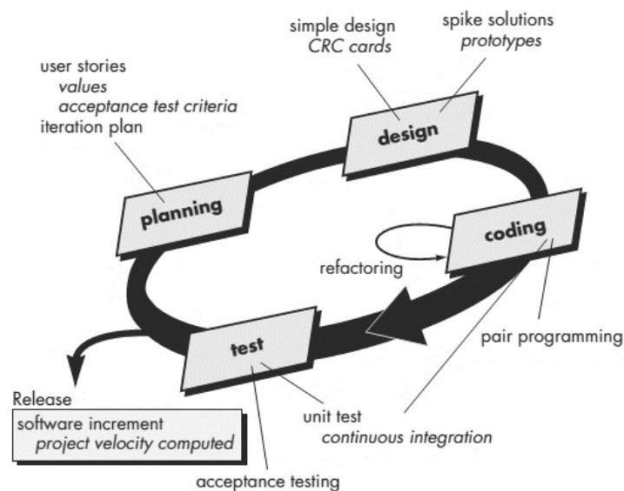
Desain yang telah dibuat diterjemahkan ke dalam bahasa pemrograman pada tahap pengkodean. Praktik *Extreme Programming* seperti *pair*

programming (dua programmer bekerja sama pada satu komputer), *refactoring* (penyempurnaan kode secara berkala), dan integrasi berkelanjutan digunakan untuk mengkodekan. Bahasa pemrograman yang digunakan yaitu JavaScript dan MySQL. Laragon, framework JavaScript NextJS, mendukung bahasa pemrograman ini.

4. *Testing*

Setelah langkah pengkodean selesai kemudian dilakukan tahap pengujian terhadap aplikasi yang sudah dibangun, pengguna sistem menentukan fitur dan fungsionalitas sistem secara keseluruhan dan kemudian meninjau aplikasi tersebut.

Berikut merupakan gambar dari tahapan metode *Extreme programming* :



Gambar 2. 4 Tahapan Metode *Extreme Programming*

Sumber : Buku Merancang Aplikasi dengan Metodologi *Extreme Programming*

2.18 Kelebihan *Extreme Programming*

Secara umum, kelebihan *Extreme Programming* mencakup hal-hal berikut:

1. **Fleksibilitas Waktu dan Biaya:** *Extreme Programming* memfokuskan pengkodean inti dan menghilangkan aktivitas yang tidak produktif, yang mempercepat dan mengurangi biaya pengembangan perangkat lunak.

2. Minim Risiko: Siklus pengembangan yang pendek dan pengujian rutin mengurangi kemungkinan kegagalan proyek. Perubahan dapat segera ditindaklanjuti dan diterapkan.
3. Sederhana: Kode yang dibuat cenderung mudah dipahami dan mudah diperbaiki, yang memudahkan perbaikan di masa depan.
4. Proses Transparan dan Akuntabel: Semua tahapan proses pengembangan dapat dipertanggungjawabkan, dan tim membuat komitmen nyata.
5. Pengembangan Lebih Cepat: Bug dapat dideteksi lebih awal dengan pengujian rutin, sehingga perangkat lunak yang dibuat dapat segera digunakan.

2.19 Kekurangan Extreme Programming

Adapun kekurangan *Extreme Programming* secara umum adalah:

1. Kurangnya Dokumentasi: Karena fokusnya pada komunikasi langsung dan pengkodean, *Extreme Programming* cenderung tidak memiliki banyak dokumentasi formal. Saat pemeliharaan jangka panjang atau pergantian anggota tim dapat membuat hal ini sulit.
2. Ketergantungan pada Kolaborasi Intensif: *Extreme Programming* membutuhkan klien dan seluruh tim untuk berpartisipasi secara aktif selama proyek; tanpa komunikasi yang baik, proses dapat terganggu.
3. Tidak Dianjurkan untuk Proyek Besar dan Tim Besar: *Extreme Programming* bekerja lebih baik dengan tim kecil hingga menengah. Proyek besar dengan banyak anggota bisa sulit untuk berkomunikasi dan berkolaborasi.
4. Fokus Berlebihan pada Kode: Karena *Extreme Programming* berfokus pada pengkodean dan pengujian, desain dan dokumentasi kadang-kadang menjadi kurang perhatian.
5. Biaya Tinggi untuk *Pair Programming*: *Pair programming* membutuhkan dua orang untuk satu tugas, sehingga membutuhkan lebih banyak tenaga kerja.

2.20 Perbedaan Extreme Programming, Agile dan Waterfall

1. *Extreme Programming*

Extreme Programming merupakan salah satu metode dalam pengembangan perangkat lunak yang tergolong dalam kelompok *Agile*. *Extreme Programming* mengutamakan kecepatan dan kemudahan dengan siklus pengembangan yang singkat, iterasi yang cepat, dan sedikit dokumentasi. Praktik seperti *pair programming* (di mana dua programmer bekerja sama pada satu komputer), integrasi berkelanjutan, pengujian unit menyeluruh, dan komunikasi intensif dengan pengguna adalah ciri khas *Extreme Programming*.

2. *Agile*

Agile adalah metode pengembangan perangkat lunak yang menekankan keterlibatan pengguna, keterlibatan pengguna, iterasi, dan kemampuan untuk menyesuaikan diri dengan perubahan kebutuhan. *Agile* Sangat cocok untuk proyek dengan kebutuhan yang dinamis dan sering berubah karena *agile* memungkinkan perubahan cepat tanpa mengganggu proses keseluruhan.

3. *Waterfall*

Waterfall adalah metode pengembangan perangkat lunak yang bersifat linier dan bertahap. Proses analisis kebutuhan, desain, pengimplementasian, pengujian, dan pemeliharaan harus dilakukan secara berurutan. *Waterfall* adalah pilihan yang baik untuk proyek dengan kebutuhan yang jelas dan stabil, karena setiap perubahan setelah suatu fase selesai dapat menjadi sangat sulit dan mahal.

Berdasarkan perbedaan antara metode *Extreme Programming*, *Agile*, dan *Waterfall*, penulis memutuskan untuk menggunakan metode *Extreme Programming* dalam pengembangan sistem *Point of Sale* (POS) di Restoran Sajiwa Steak Majalaya. Metode *Extreme Programming* dipilih karena dinilai paling sesuai dengan kebutuhan pengembangan sistem yang dinamis dan menuntut kecepatan serta kemampuan beradaptasi terhadap perubahan. *Extreme Programming* memungkinkan proses pengembangan dilakukan secara iteratif dan adaptif,

sehingga setiap perubahan kebutuhan dari pihak restoran dapat segera diakomodasi tanpa harus menunggu siklus pengembangan selesai secara keseluruhan. Selain itu, *Extreme Programming* menekankan pentingnya komunikasi intensif dan kolaborasi antara tim pengembang dengan pengguna, sehingga umpan balik yang didapatkan dapat langsung diterapkan untuk penyempurnaan sistem. Praktik-praktik seperti *pair programming* dan pengujian berkelanjutan yang diterapkan dalam *Extreme Programming* juga sangat bermanfaat untuk menjaga kualitas aplikasi, memastikan sistem *Point of Sale* (POS) dapat berjalan optimal, minim gangguan, serta mudah dalam proses pemeliharaan dan pengembangan lebih lanjut. Dengan berbagai keunggulan tersebut, *Extreme Programming* dinilai sebagai pendekatan yang paling efektif dan efisien untuk mendukung kebutuhan pengembangan sistem *Point of Sale* (POS) di Restoran Sajiwa Steak Majalaya.