

BAB II

TINJAUAN PUSTAKA

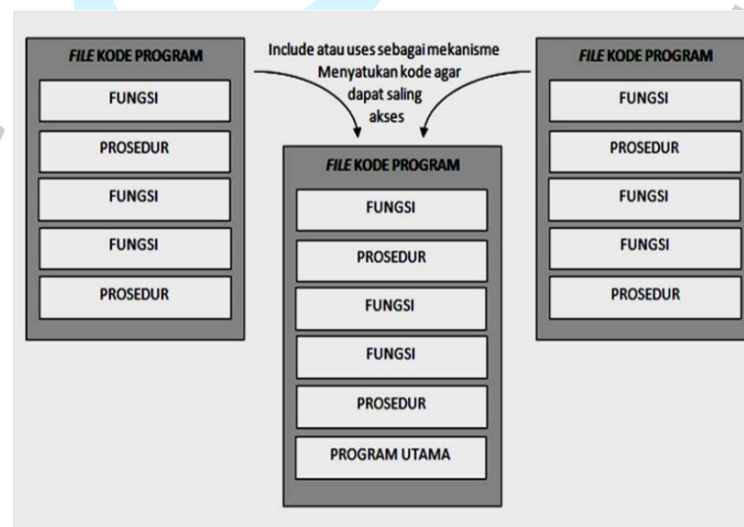
2.1 Teori Dasar

2.1.1 Rancang Bangun

Menurut Nurhayati et al. (2017), rancang bangun adalah proses mengubah hasil analisis ke dalam perangkat lunak. Ini dapat digunakan untuk membangun atau memperbaiki sistem yang sudah ada. Penggambaran, perencanaan, dan penyusunan komponen yang terpisah menjadi satu kesatuan yang utuh dan berfungsi juga termasuk dalam rancang bangun. Dengan demikian, dapat dikatakan bahwa rancang bangun adalah tahap awal dalam mengembangkan ide atau sketsa yang belum pernah terjadi sebelumnya, yang kemudian dikembangkan menjadi desain yang dapat digunakan. Oleh karena itu, tujuan rancang bangun tugas akhir ini adalah untuk menghasilkan sebuah konsep yang dirancang yang melibatkan penggambaran, perancangan, dan penyusunan yang terstruktur. Ada dua metode yang digunakan dalam rancang bangun sistem :

1. Pendekatan Terstruktur

Konsep atau perspektif pemrograman yang dikenal sebagai pendekatan terstruktur membagi program berdasarkan cara kerjanya. Menurut kebergantungan antarfungsi, fungsi dan prosedur dicatat dengan cara yang berurut dari atas ke bawah. Berikut contoh ilustrasi untuk pendekatan terstruktur :



Gambar 2.1 Ilustrasi Pemrograman Terstruktur

Sumber : (Sutanti et al., 2020)

2. Pendekatan *Object Oriented*

Pendekatan Object Oriented merupakan pendekatan yang menggunakan objek dan class. Pendekatan ini memberikan kemudahan dalam membuat sebuah program. Dalam hal ini, object oriented menjadi pendekatan dalam penelitian ini karena menggunakan UML (Unified Modeling Language) sebagai alat bantu memodelkan sistem.

2.1.2 SDLC (System Development Life-Cycle)

Membangun sistem informasi bisa diibaratkan seperti membangun sebuah rumah. Awalnya, semua dimulai dari ide dasar tentang apa yang ingin dibuat. Setelah itu, ide ini dituangkan ke dalam sketsa sederhana yang diperlihatkan kepada pelanggan. Sketsa tersebut kemudian diperbaiki dan disesuaikan beberapa kali hingga pelanggan merasa cocok dengan hasilnya. Selanjutnya, dibuat cetak biru yang lebih detail, mencakup elemen-elemen seperti jenis keran air atau letak soket telepon. Akhirnya, rumah dibangun berdasarkan cetak biru tersebut, meskipun terkadang ada beberapa perubahan kecil yang dilakukan sesuai permintaan pelanggan selama proses pembangunan.

Proses ini sejalan dengan empat fase utama dari System Development Life Cycle (SDLC), yang terdiri dari perencanaan, analisis, desain, dan implementasi. Setiap proyek mungkin menempatkan penekanan yang berbeda pada tiap fase atau menggunakan pendekatan yang berbeda untuk menyelesaikan fase-fase tersebut, tetapi semuanya tetap mencakup keempat elemen ini. Setiap fase terdiri dari beberapa langkah yang menghasilkan output, yaitu dokumen atau file yang memberikan pemahaman lebih lanjut tentang proyek.

SDLC merupakan proses bertahap yang digunakan dalam pengembangan sistem informasi, di mana setiap fase menyempurnakan hasil dari fase sebelumnya. Proyek dimulai dengan fase analisis untuk memberikan gambaran umum tentang sistem, dilanjutkan dengan desain untuk memperinci detail teknis, dan diakhiri dengan implementasi untuk menghasilkan sistem yang sesuai kebutuhan. Pendekatan ini memungkinkan tim proyek untuk menggunakan berbagai teknik dalam menyelesaikan langkah-langkah dan menghasilkan deliverables seperti dokumen atau evaluasi. Meskipun sering kali dilakukan secara berurutan, SDLC

juga dapat diterapkan secara iteratif atau inkremental, tergantung kebutuhan proyek. (Dennis et al., 2015)

1. Planning

Sistem informasi perlu dibangun dan caranya tim proyek akan melaksanakan pengembangannya, tahap planning adalah langkah pertama. Ini terdiri dari dua langkah utama, langkah inisiasi proyek dan manajemen proyek. Langkah inisiasi proyek bertujuan untuk mengidentifikasi nilai bisnis sistem terhadap perusahaan. Pertanyaan utama yang harus dijawab adalah bagaimana sistem ini dapat mengurangi biaya atau meningkatkan pendapatan. Ide untuk sistem baru sering kali berasal dari departemen di luar TI, seperti pemasaran atau akuntansi, melalui dokumen permintaan sistem (system request).

Dokumen permintaan sistem adalah ringkasan singkat tentang kebutuhan bisnis dan penjelasan mengenai bagaimana sistem tersebut akan memberikan nilai bisnis. Departemen TI bekerja sama dengan pihak atau departemen yang mengajukan permintaan (project sponsor) untuk melakukan analisis kelayakan. Hasil dari dokumen permintaan sistem dan analisis kelayakan ini kemudian disampaikan kepada komite persetujuan sistem informasi yang akan memutuskan apakah proyek tersebut layak dilanjutkan. Tahap berikutnya setelah proyek disepakati adalah manajemen proyek. Pada tahap ini, manajer proyek membuat rencana kerja, membentuk tim, dan menetapkan teknik-teknik untuk membantu tim dalam mengelola dan mengarahkan rencana sepanjang SDLC (Software Development Life Cycle). Hasil dari langkah ini adalah rencana proyek yang menjelaskan bagaimana tim proyek akan mengembangkan sistem tersebut. Proses planning memastikan bahwa proyek memiliki dasar yang jelas dan strategi pelaksanaan yang terorganisir untuk mencapai tujuan yang diinginkan.

2. Analysis

Tahap analisis melibatkan identifikasi terhadap siapa pengguna sistem, fungsi yang akan dijalankan oleh sistem, serta konteks penggunaan sistem, termasuk lokasi dan waktu penggunaannya. Fase ini mencakup evaluasi terhadap sistem yang telah ada, pencarian peluang peningkatan, serta

perancangan konsep untuk sistem yang baru. Terdapat tiga langkah utama dalam tahap ini: Langkah pertama adalah menyusun strategi analisis yang akan menjadi panduan bagi tim proyek. Strategi ini biasanya melibatkan evaluasi terhadap sistem yang sedang berjalan saat ini (as-is system) serta identifikasi permasalahan yang ada, kemudian merancang pendekatan untuk mengembangkan sistem baru (to-be system).

Langkah kedua mencakup proses pengumpulan kebutuhan, yang dapat dilakukan melalui wawancara atau penyebaran kuesioner. Informasi yang diperoleh dianalisis bersama dengan masukan dari sponsor proyek serta pihak terkait lainnya. Hasil analisis ini digunakan untuk merancang konsep sistem baru, yang kemudian menjadi dasar dalam pembuatan model analisis bisnis guna menggambarkan bagaimana operasional bisnis akan berjalan jika sistem baru diterapkan. Langkah terakhir adalah menyusun seluruh hasil analisis, konsep sistem, serta model bisnis ke dalam sebuah dokumen yang disebut proposal sistem. Dokumen ini disampaikan kepada sponsor proyek dan pemangku kepentingan utama, seperti anggota komite persetujuan, untuk menentukan apakah proyek dapat dilanjutkan ke tahap berikutnya

3. Design

Fase desain bertujuan untuk menentukan cara kerja sistem secara rinci, mencakup infrastruktur perangkat keras, perangkat lunak, dan jaringan, serta elemen seperti antarmuka pengguna, formulir, laporan, program, basis data, dan file yang dibutuhkan. Meskipun keputusan strategis mengenai sistem telah dibuat pada fase analisis, tahap desain lebih fokus pada perincian operasional sistem. Fase ini terdiri dari empat langkah utama:

- **Strategi Desain**

Langkah awal adalah merancang strategi pengembangan sistem. Keputusan ini mencakup apakah sistem akan dikembangkan oleh tim internal, diserahkan kepada pihak ketiga seperti perusahaan konsultan, atau menggunakan perangkat lunak siap pakai

- **Desain Arsitektur Dasar**

Langkah berikutnya adalah menentukan desain arsitektur sistem, yang mencakup perangkat keras, perangkat lunak, dan jaringan yang akan

digunakan. Biasanya, sistem baru akan berintegrasi atau memperbarui infrastruktur yang sudah ada dalam organisasi. Selain itu, desain antarmuka pengguna juga dirancang, termasuk metode navigasi seperti menu dan tombol, serta formulir dan laporan yang akan digunakan.

- **Spesifikasi Basis Data dan File**

Langkah ketiga adalah merancang spesifikasi basis data dan file, yang mencakup jenis data yang akan disimpan serta lokasi penyimpanannya.

- **Desain Program**

Langkah terakhir adalah menyusun desain program, yang mencakup daftar program yang perlu dikembangkan beserta fungsionalitasnya masing-masing.

4. Implementation

Fase final dalam Siklus Hidup Pengembangan Sistem (SDLC) adalah fase penerapan, bagaimana sistem benar-benar dikembangkan atau diadopsi jika menggunakan perangkat lunak siap pakai. Fase ini sering menjadi fokus utama karena, dalam banyak kasus, merupakan tahap yang paling memakan waktu dan biaya dalam seluruh proses pengembangan. Fase implementasi terdiri dari tiga langkah utama :

- **Desain Sistem**

Tahap awal adalah desain sistem, di mana sistem dibuat serta diperiksa guna menjamin bahwa sistem berfungsi sesuai dengan desain. Karena biaya yang ditimbulkan oleh kesalahan tergolong banyak, pengujian menjadi salah satu tahap vital dalam implementasi. Banyak organisasi memberikan perhatian lebih pada pengujian daripada menulis program itu sendiri.

- **Instalasi Sistem**

Tahap kedua adalah proses instalasi sistem, di mana sistem lama dihentikan dan sistem baru mulai dioperasikan. Salah satu aspek krusial dalam transisi ini adalah penyusunan rencana pelatihan yang bertujuan untuk membimbing pengguna dalam mengoperasikan sistem baru serta

membantu mereka beradaptasi dengan perubahan yang ditimbulkan oleh implementasi sistem tersebut.

- **Rencana Dukungan**

Tahap akhir adalah penyusunan rencana dukungan. Rencana ini umumnya meliputi evaluasi setelah implementasi, baik secara formal maupun informal, serta metode terstruktur untuk mengidentifikasi perubahan, baik skala besar maupun kecil, yang perlu dilakukan pada s



2.1.3 UML (Unified Modeling Language)



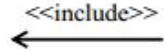
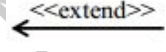
Unified Modeling Language (UML) umumnya digunakan untuk memodelkan sistem yang berbasis konsep berorientasi objek. Menurut Satzinger, Jackson, & Burd (2015), UML merupakan serangkaian standar untuk membangun model dan notasi yang dirancang khusus dalam pengembangan berbasis objek. Shelly dan Rosenblatt (2012) menjelaskan bahwa UML adalah metode yang sering dipakai untuk visualisasi dan dokumentasi perangkat lunak dalam tahap perancangan sistem. Dengan demikian, UML dapat dianggap sebagai metode pemodelan visual yang berfungsi sebagai alat untuk merancang atau mengembangkan perangkat lunak berorientasi objek (Arief Yanto Rukmana et al., 2023). Untuk membuat suatu model, UML memerlukan diagram-diagram yang diartikan sebagai berikut :

1. Usecase Diagram

Usecase Diagram adalah salah satu diagram dalam UML (Unified Modelling Language) yang memperlihatkan bagaimana sistem berinteraksi dengan aktor-aktor yang terlibat.

Tabel 2.1 Tabel Penjelasan Usecase


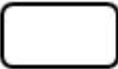


SIMBOL	NAMA SIMBOL	KETERANGAN
	Aktor	Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Usecase</i>	Abstraksi dan interaksi antara sistem dan actor



	<i>Association</i>	Abstraksi dari penghubung antara actor dengan <i>use case</i>
	Generalisasi	Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	<i>Include</i>	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	<i>Extend</i>	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

2. Activity Diagram

Diagram Aktivitas merupakan diagram yang menunjukkan proses yang terjadi di dalam sistem.

Tabel 2.2 Tabel Penjelasan Diagram Activity

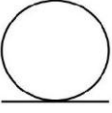
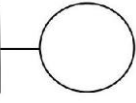
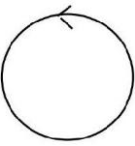
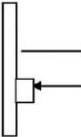
SIMBOL	NAMA	KETERANGAN
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas yang biasanya diawali dengan kata kerja
	Percabangan/Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu
	Penggabungan/Join	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu



	Status akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggungjawab terhadap aktivitas yang terjadi

3. Sequence Diagram

Sequence diagram merupakan diagram yang menjelaskan interaksi objek berdasarkan urutan waktu dan dapat menggambarkan urutan yang harus dilakukan untuk dapat menghasilkan sesuatu.

Tabel 2.3 Tabel Penjelasan Diagram Sequence

GAMBAR	NAMA	KETERANGAN
	Entity Class	Gambaran sistem sebagai landasan dalam menyusun basis data
	Boundary Class	Menangani komunikasi antar lingkungan sistem
	Control Class	Bertanggung jawab terhadap kelas-kelas, terhadap objek yang berisi logika
	Recursive	Pesan untuk dirinya

	Activation	Mewakili proses durasi aktivasi sebuah operasi
	Life Line	Komponen yang digambarkan garis putus terhubung dengan objek

4. Class Diagram

Class Diagram adalah diagram dalam UML yang digunakan untuk memodelkan struktur statis dari sebuah sistem, seperti kelas, atribut, operasi, dan hubungan antar kelas. Kelas merepresentasikan entitas atau konsep dalam sistem, di mana setiap kelas memiliki atribut sebagai karakteristiknya dan operasi sebagai fungsi yang bisa dilakukan kelas tersebut. Atribut juga memiliki visibility (public, protected, atau private) yang menentukan aksesibilitas atribut dari luar kelas.

Hubungan antar kelas digambarkan melalui garis penghubung, seperti asosiasi, yang menunjukkan bagaimana kelas saling berinteraksi. Dalam hubungan asosiasi, multiplicity digunakan untuk menggambarkan jumlah minimum dan maksimum objek yang bisa saling terhubung, seperti seorang pasien bisa memiliki banyak janji temu (*0..*), sedangkan setiap janji temu hanya memiliki satu pasien (*1*). Jika terdapat informasi tambahan dalam hubungan, digunakan association class untuk mewakilinya.

Generalization menunjukkan hubungan pewarisan, di mana subkelas mewarisi properti dan operasi dari superkelas. Misalnya, "Dokter" dan "Perawat" adalah subkelas dari "Pegawai", sehingga mewarisi atribut dan operasi dari kelas induk. Generalization ditampilkan dengan garis solid dan panah berlubang menuju superkelas. Selain itu, terdapat hubungan aggregation dan composition untuk menggambarkan relasi "bagian dari". Aggregation (diamond putih) menunjukkan hubungan logis, seperti "Roda" adalah bagian dari "Kendaraan" tetapi dapat dipindahkan. Sementara composition (diamond hitam) menunjukkan hubungan fisik yang lebih kuat, seperti "Pintu" hanya menjadi bagian dari satu "Mobil".

Kesimpulannya, class diagram berfungsi untuk memodelkan struktur sistem dengan mendefinisikan kelas, atribut, operasi, dan hubungan antar kelas.

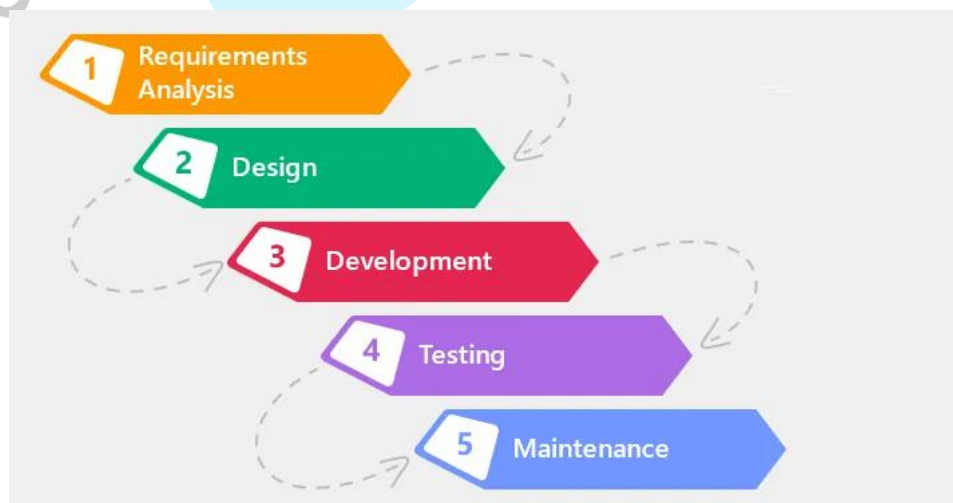
Generalization digunakan untuk pewarisan, sedangkan aggregation dan composition menggambarkan relasi "bagian dari" dalam berbagai tingkat keterkaitan. Dengan memahami elemen-elemen ini, struktur sistem dapat diorganisir dan direpresentasikan dengan jelas.

2.1.4 Metodologi

Metodologi dalam pengembangan perangkat lunak adalah suatu kerangka kerja atau pendekatan yang digunakan untuk merencanakan, mengelola, dan melaksanakan proses pengembangan sistem secara sistematis. Untuk metodologi pengembangan perangkat lunak bisa dibilang cukup banyak variasinya yang dibagi menjadi model pengembangan tradisional dan model pengembangan modern. Model pengembangan tradisional cukup banyak dipergunakan karena memiliki kelebihan tertentu, seperti mudah dipahami, struktur yang jelas, dan cocok untuk pengembangan aplikasi yang memiliki kebutuhan stabil dan terprediksi. Model pengembangan modern juga cukup banyak digunakan, dan model pengembangan modern bisa dilakukan pada sistem tradisional jika diperlukan. Dalam metode pengembangan modern, biasanya seringkali dianggap sebagai model yang ringan karena hanya membutuhkan dokumentasi yang lebih sedikit dibandingkan model sebelumnya.

1. Model Pengembangan Tradisional

A. Model Waterfall

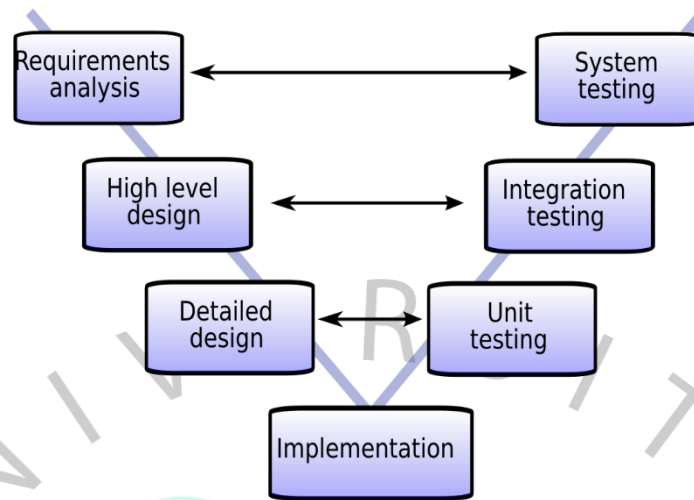


Gambar 2.2 Metode Pengembangan Waterfall
Sumber : <https://inovasi.ac.id/wiki/metode-sdlc-waterfall/>

Model Waterfall adalah metode pengembangan perangkat lunak yang terdiri dari serangkaian fase linier yang meliputi analisis kebutuhan, desain, implementasi, pengujian, dan pemeliharaan. Tahap awal, yaitu analisis kebutuhan, berfokus pada pemahaman tujuan dan fitur sistem melalui interaksi dengan pemangku kepentingan. Setelah kebutuhan sistem dirumuskan, tahap desain dimulai dengan perancangan arsitektur, database, antarmuka pengguna, dan spesifikasi teknis lainnya. Selanjutnya, implementasi dilakukan dengan menulis dan mengintegrasikan kode untuk membentuk sistem perangkat lunak yang lengkap. Pada tahap pengujian, sistem diuji untuk memastikan kesesuaiannya dengan kebutuhan yang telah ditetapkan, mencakup pengujian fungsional, non-fungsional, integrasi, dan sistem. Fase terakhir adalah pemeliharaan, yang melibatkan perbaikan bug, peningkatan fitur, dan pembaruan keamanan secara berkala untuk menjaga kinerja sistem.

Model Waterfall memiliki beberapa kelebihan, seperti struktur yang jelas, dokumentasi yang lengkap, dan cocok untuk proyek dengan kebutuhan stabil. Namun, model ini memiliki kekurangan, termasuk kurang fleksibel terhadap perubahan, tidak memungkinkan umpan balik hingga tahap akhir, dan berisiko tinggi jika kesalahan baru ditemukan di akhir proses. Model ini sering digunakan pada proyek-proyek pemerintah, organisasi mapan, dan proyek kecil dengan persyaratan yang tidak berubah. Tahap perencanaan menjadi langkah awal yang sangat penting untuk mengidentifikasi potensi kekurangan sebelum proses pengembangan dimulai (Ghanghro & Sawand, 2021).

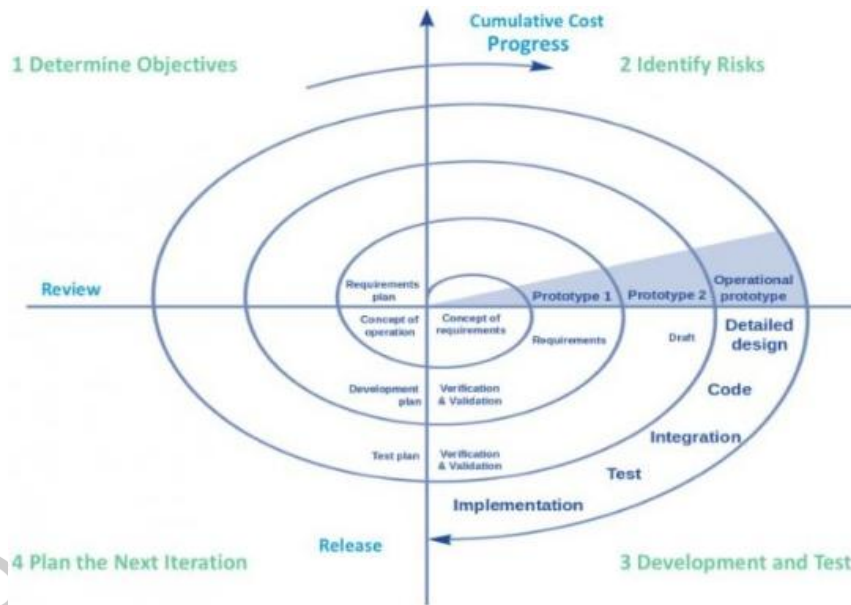
B. Model V-Model



Gambar 2.3 Metode Pengembangan V-Model
Sumber : https://id.wikipedia.org/wiki/Model_V

Model V-Model adalah pengembangan dari Model Waterfall yang memberikan penekanan lebih besar pada proses pengujian. Setiap tahap pengembangan perangkat lunak memiliki tahap pengujian yang terhubung secara langsung, membentuk pola "V". Model ini sering digunakan pada proyek yang membutuhkan pengujian yang ketat dan terstruktur. Meskipun memerlukan waktu dan sumber daya yang signifikan, koordinasi yang baik antara tim pengembangan dan pengujian sangat penting untuk memastikan kelancaran proses. Namun, model ini memiliki keterbatasan dalam menanggapi perubahan kebutuhan atau dinamika proyek yang berkembang.

C. Model Spiral

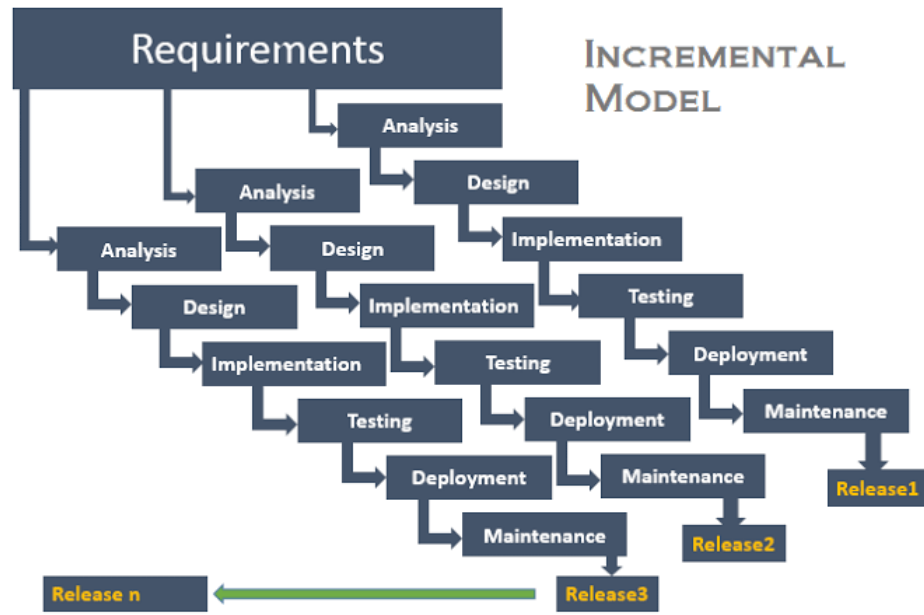


Gambar 2.4 Metode Pengembangan Spiral

Sumber : <https://blockmoney.co.id/blogs/6-metode-pengembangan-perangkat-lunak-waterfall-rad-agile-prototype-dll/>

Model Spiral adalah metode pengembangan perangkat lunak yang menggabungkan pendekatan iteratif dari model prototipe dengan pengendalian sistematis dari model linear. Diajukan oleh Barry Boehm pada tahun 1986, model ini dirancang untuk proyek-proyek besar, kompleks, dan berisiko tinggi. Pengembangan dilakukan secara bertahap melalui iterasi yang mencakup empat tahap utama: perencanaan, analisis risiko, rekayasa, dan evaluasi. Keunggulan model ini terletak pada fleksibilitasnya terhadap perubahan dan kemampuannya untuk mengelola risiko secara efektif sepanjang siklus hidup proyek. Namun, model ini memiliki kelemahan, seperti kompleksitas yang tinggi, kebutuhan akan manajemen risiko yang cermat, serta waktu dan biaya yang lebih besar dibandingkan dengan model linear. Model Spiral sering digunakan dalam proyek yang membutuhkan pengembangan bertahap dan fokus kuat pada pengelolaan risiko (Ghanghro & Sawand, 2021).

D. Model Incremental

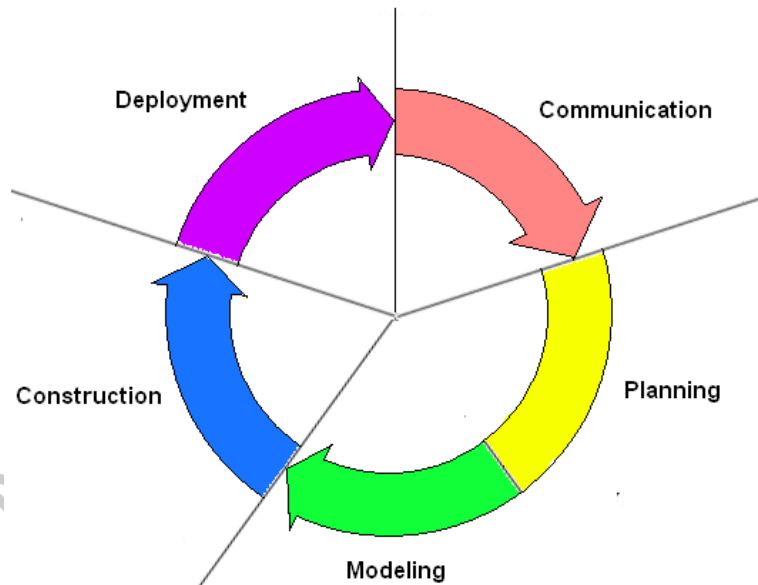


Gambar 2.5 Metode Pengembangan Incremental

Sumber : https://sanjayawisnu33.github.io/Proses_Pengembangan/

Model Incremental adalah metode pengembangan perangkat lunak di mana sistem dikembangkan dan diselesaikan secara bertahap, dengan setiap tahap menambahkan fungsionalitas baru. Pendekatan ini sering digunakan pada proyek besar yang memiliki fungsionalitas kompleks, karena memungkinkan pembagian proyek menjadi bagian-bagian yang lebih kecil dan mudah dikelola. Model ini mengadaptasi pendekatan Waterfall dengan tahapan yang tumpang tindih, sehingga menghasilkan fungsionalitas yang dapat digunakan lebih awal, meskipun proyek belum sepenuhnya selesai. Dalam pelaksanaannya, model ini dapat dimulai dengan serangkaian persyaratan awal yang lengkap atau dengan tujuan umum yang dipecah menjadi bagian-bagian kecil yang diimplementasikan secara bertahap. Namun, model ini memiliki kelemahan, seperti potensi ketidaknyamanan akibat persyaratan yang belum lengkap, perlunya antarmuka yang terdefinisi dengan baik, dan tantangan dalam menerapkan ulasan formal atau audit pada sistem yang belum sepenuhnya selesai.

E. Model Prototyping



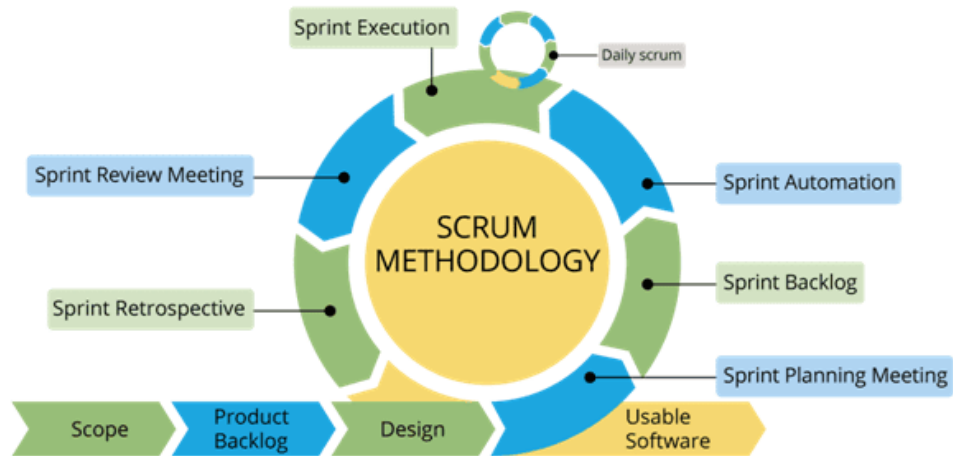
Gambar 2.6 Metode Prototyping

Sumber : <https://blogger-sharearea.blogspot.com/2013/11/perbedaan-metode-prototyping-dan-metode.html>

Model Prototyping adalah pendekatan pengembangan perangkat lunak yang berfokus pada pembuatan prototipe sebagai model awal sistem. Prototipe ini digunakan untuk mengenali kebutuhan, mengevaluasi desain, dan mendapatkan umpan balik dari pengguna sebelum sistem final dikembangkan. Berbeda dengan model lain, model ini tidak menetapkan persyaratan secara kaku sebelum memasuki tahap desain, pengembangan, dan pengujian. Proses dimulai dengan memahami kondisi awal, sehingga klien dapat memiliki gambaran awal tentang sistem yang akan dihasilkan. Umpan balik dari klien membantu tim pengembang menyesuaikan sistem sesuai kebutuhan. Prototipe sering kali tidak mencerminkan logika akhir yang akan diimplementasikan, tetapi berfungsi untuk mengidentifikasi kebutuhan khusus pengguna yang mungkin terlewatkan selama perancangan. Model ini sangat bermanfaat dalam memastikan sistem yang dikembangkan sesuai dengan ekspektasi pengguna (Ghanghro & Sawand, 2021).

2. Model Pengembangan Modern

A. Model Scrum

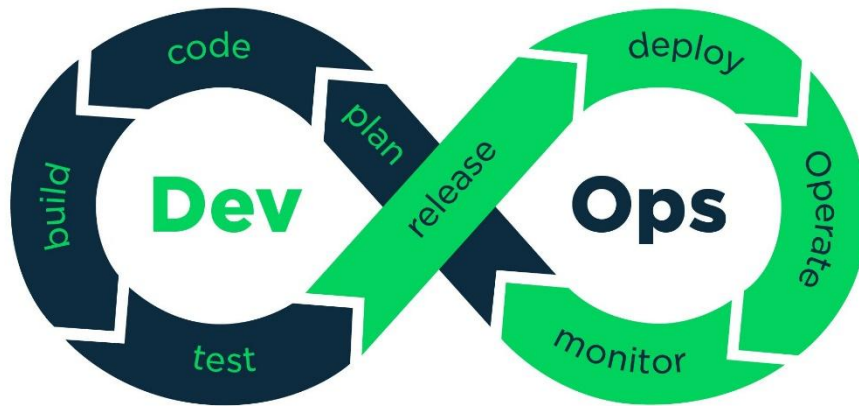


Gambar 2.7 Metode Scrum

Sumber : <https://bif.telkomuniversity.ac.id/scrum-prinsip-agile-dan-tahapan-dalam-metode-scrum/>

Scrum adalah kerangka kerja populer dalam model Agile yang mengutamakan pengembangan perangkat lunak melalui iterasi singkat bernama *sprint*, dengan durasi satu hingga empat minggu. Setiap *sprint* dimulai dengan perencanaan, dilanjutkan dengan pembangunan, pengujian, dan demonstrasi hasil kepada pemangku kepentingan. Pendekatan ini menekankan transparansi, kolaborasi tim, serta pembaruan berkelanjutan berdasarkan umpan balik. Proses Scrum terbagi dalam tiga fase utama: Pra-pertandingan, Pertandingan, dan Pasca-pertandingan. Scrum dimulai dengan Rapat Perencanaan Sprint untuk menentukan kebutuhan proyek, dan diakhiri dengan Sprint Retrospective yang mengevaluasi apakah tujuan proyek telah tercapai. Setiap iterasi memiliki durasi tetap yang mendorong tim bekerja secara efisien dan menyelesaikan tugas lebih cepat melalui kolaborasi yang intensif. Keberhasilan Scrum sangat bergantung pada komunikasi antar anggota tim dan kehadiran anggota yang berpengalaman dalam menyelesaikan proyek tepat waktu, sehingga memastikan hasil pengembangan yang berkualitas.

B. Model DevOps

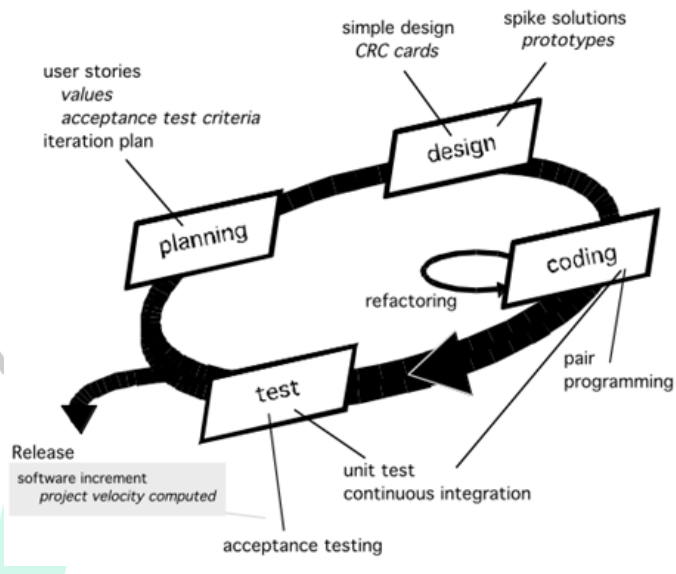


Gambar 2.8 Metode DevOps

Sumber : <https://www.logique.co.id/blog/2021/05/28/apa-itu-devops/>

DevOps adalah pendekatan pengembangan perangkat lunak yang mendorong kolaborasi erat antara tim pengembangan (Development) dan tim operasional (Operations). Tujuan utamanya adalah mengintegrasikan proses pengembangan dan pengiriman secara efisien untuk mempercepat rilis perangkat lunak sekaligus meningkatkan kualitas dan keandalan produk. Pendekatan ini menggunakan otomatisasi, monitoring, serta praktik CI/CD untuk memastikan pengiriman yang cepat dan berkualitas. Berbagai alat seperti Confluence, GitHub, Bitbucket, Jenkins, Chef, Puppet, Docker, dan Splunk mendukung tim DevOps dengan menyederhanakan tugas mereka. Alat-alat ini juga berfungsi sebagai solusi SaaS (Software as a Service) dan PaaS (Platform as a Service) untuk memfasilitasi kolaborasi tim pengembangan dan operasional (Narang & Mittal, 2022).

C. Extreme Programming

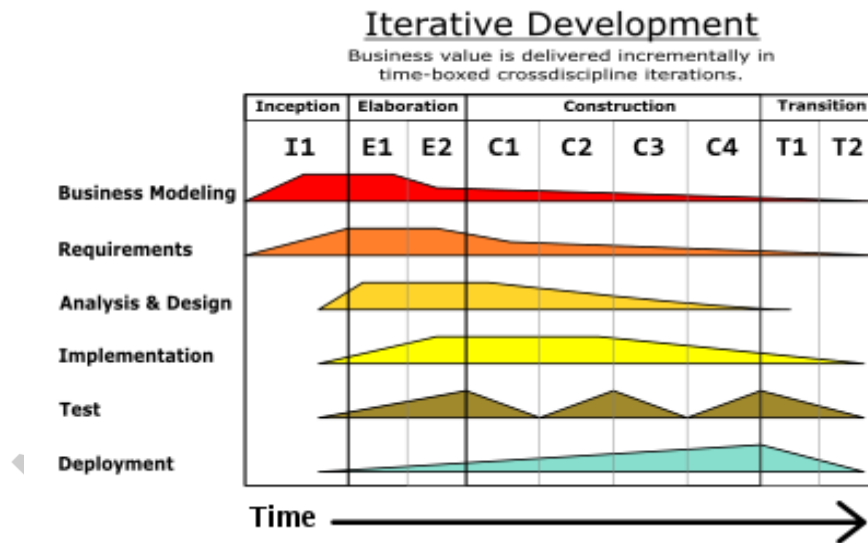


Gambar 2.9 Metode Extreme Programming

Sumber : <https://ilmurplkitabersama.blogspot.com/2020/03/metode-extreme-programming-contoh.html>

Extreme Programming (XP), yang diperkenalkan oleh Kent Beck pada tahun 1996, adalah metodologi Agile yang fleksibel, ringan, dan dirancang untuk mengatasi kebutuhan yang tidak jelas atau sering berubah. XP lebih cocok untuk tim kecil hingga menengah dan berfokus pada penerapan nilai, prinsip, serta praktik yang dilakukan secara konsisten untuk menghasilkan perangkat lunak berkualitas tinggi. Metodologi ini menekankan kepuasan pelanggan melalui umpan balik cepat dan pelepasan produk secara berkala, sehingga masalah dapat diidentifikasi lebih awal, biaya pengembangan ditekan, dan produk akhir dihasilkan dengan kualitas lebih baik. Proses XP terdiri dari enam fase: Eksplorasi, Perencanaan, Iterasi hingga rilis, Produksi, Pemeliharaan, dan Penyelesaian. Dibandingkan dengan Scrum, XP memiliki fitur yang lebih terdefinisi, seperti kepemilikan kode, pengujian, teknik validasi, standar pengkodean, dan permintaan persyaratan, yang mendukung pengembangan perangkat lunak secara sistematis dan efisien.

D. Unified Process

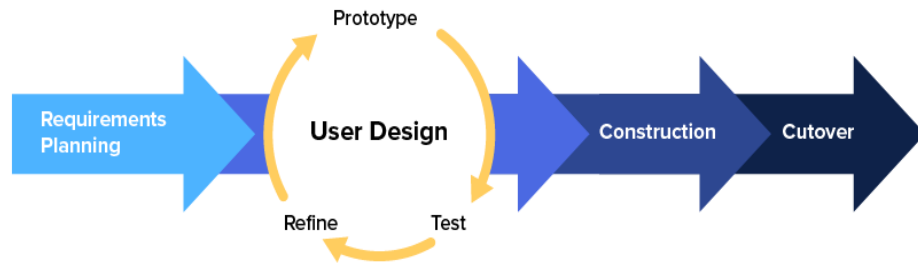


Gambar 2.10 Metode Unified Process

Sumber : <https://medium.com/@andrerahardjo/pengertian-rup-rational-unified-process-1bec9c664458>

Pada awal 1990-an, pengembangan perangkat lunak mengalami pergeseran paradigma menuju pendekatan berbasis objek, yang mendorong munculnya metode seperti Object-Oriented Analysis and Design (OOAD) dan Rational Unified Process (RUP). Setelah diakuisisi oleh IBM pada tahun 2003, RUP berkembang menjadi Unified Process (UP) dan terintegrasi dengan alat serta praktik pengembangan IBM. Unified Process adalah model pengembangan perangkat lunak yang bersifat iteratif dan inkremental, dengan tahapan yang meliputi pemodelan, analisis, desain, implementasi, pengujian, dan pemeliharaan. UP menekankan dokumentasi yang lengkap, manajemen risiko yang efektif, dan iterasi berulang untuk meningkatkan kualitas produk perangkat lunak. Meskipun bukan bagian dari metodologi Agile, UP memiliki konsep seperti iterasi, adaptabilitas, dan orientasi pengguna yang memengaruhi perkembangan Agile. Elemen-elemen dari UP, seperti iterasi dan pendekatan berbasis pengguna, diadopsi oleh metodologi seperti Scrum dan Extreme Programming (XP). Unified Process terdiri dari empat fase utama: Inception, Elaboration, Construction, dan Transition.

E. Rapid Application Development (RAD)



Sumber : <https://agus-hermanto.com/blog/detail/metode-pengembangan-rad-rapid-application-development>

Rapid Application Development (RAD) adalah sebuah pendekatan pengembangan perangkat lunak yang berfokus pada kecepatan dan fleksibilitas. Dalam metode ini, proses pengembangan berlangsung secara cepat melalui pembuatan prototipe yang berulang atau iteratif, memungkinkan pengembang untuk segera memberikan solusi awal yang dapat langsung diuji oleh pengguna. Umpan balik pengguna yang aktif menjadi komponen penting dalam RAD, sehingga setiap iterasi atau prototipe dapat disesuaikan dan diperbaiki sesuai dengan kebutuhan yang berubah atau persyaratan baru dari pengguna. Dibandingkan dengan pendekatan pengembangan perangkat lunak tradisional yang lebih kaku, RAD menawarkan desain yang lebih singkat dan memprioritaskan hasil yang dapat segera digunakan meskipun proses pengembangannya masih berlangsung.

Metode ini juga ideal untuk proyek dengan tenggat waktu ketat, karena melibatkan tim yang lebih kecil dan berfokus pada penyelesaian kerangka kerja dasar sejak awal pengembangan. Dengan pendekatan iteratif dan fleksibilitas tinggi, RAD memungkinkan perusahaan untuk menyesuaikan produk akhir dengan cepat, sekaligus memastikan bahwa modul yang dikembangkan tetap sesuai dengan kebutuhan dan batasan yang telah ditetapkan, sehingga pengembangan lebih efisien dan responsif terhadap umpan balik pengguna. Metode RAD memberikan fleksibilitas tinggi dalam merespon perubahan selama proses pengembangan. Mengingat sistem operasional kantin dan vending machine bisa mengalami

perubahan kebutuhan atau penyesuaian teknis, RAD memungkinkan untuk memasukkan perubahan tersebut tanpa harus memulai dari awal. Salah satu metodologi yang digunakan dalam proyek ini adalah Rapid Application Development (RAD), yang dipilih karena memungkinkan pengembangan yang cepat dan iteratif, sesuai dengan kebutuhan yang dinamis, serta melibatkan umpan balik langsung dari pengguna untuk memastikan sistem yang dihasilkan relevan dan efektif.

2.1.5 Database

Pada awal 1960-an, Charles Bachman dari General Electric merancang DBMS generasi pertama bernama *Integrated Data Store* (IDS), yang menjadi dasar model data jaringan dan distandarisasi oleh Conference on Data System Languages (CODASYL). Atas kontribusinya, Bachman menerima penghargaan Turing Award pada 1973. Akhir 1960-an, IBM mengembangkan *Information Management System* (IMS) bekerja sama dengan perusahaan penerbangan Amerika untuk menciptakan sistem SABRE yang memungkinkan akses data bersama pada jaringan komputer. Pada 1970, Edgar Codd memperkenalkan model data relasional, yang menjadi paradigma utama DBMS pada 1980-an. SQL dikembangkan untuk basis data relasional dalam proyek IBM Sistem R, distandarisasi pada akhir 1980-an oleh American National Standards Institute (ANSI) dan International Standards Organization (ISO). Akhir 1980-an hingga awal 1990-an, DBMS terus berkembang, mencakup bahasa query yang lebih canggih, model data yang lebih lengkap, dan dukungan untuk analisis data kompleks. Sistem juga diperluas untuk menyimpan tipe data baru seperti gambar dan teks, serta mendukung query yang lebih kompleks.

Basis data (*database*) terdiri dari dua kata yaitu basis dan data. Basis dapat diartikan sebagai tempat penyimpanan layaknya gudang. Sedangkan data merupakan sekumpulan fakta yang ada yang mewakili objek seperti manusia, hewan, benda, dan lainnya. Kelompok-kelompok data saling terhubung sedemikian rupa sehingga dapat digunakan secara cepat dan mudah. Sebelum adanya basis data, sistem informasi bisnis menggunakan sistem file. Namun sistem file memiliki banyak keterbatasan, seperti kesulitan dalam pengelolaan data, ketergantungan

pada struktur file, dan redundansi data yang dapat menyebabkan inkonsistensi. Basis data mengatasi masalah ini dengan menyediakan struktur yang lebih terorganisir, memudahkan pengelolaan data, serta mengurangi risiko redundansi dan kesalahan data. Dengan normalisasi yang tepat, basis data dapat menjaga konsistensi dan integritas data dalam sistem informasi. (Arief Yanto Rukmana et al., 2023).

Database merupakan komponen penting dalam sistem pengelolaan data yang terdiri dari beberapa elemen utama. Data berperan sebagai informasi yang disimpan dalam database untuk digunakan dan diolah sesuai kebutuhan. Pengelolaan data ini dilakukan oleh *DBMS (Database Management System)*, yaitu perangkat lunak yang bertugas mengatur, menyimpan, dan mengelola database agar berjalan secara optimal. Selain itu, skema menjadi bagian penting dalam database karena berfungsi sebagai struktur atau rancangan yang menentukan bagaimana data diatur dan dihubungkan satu sama lain di dalam sistem.

Database sendiri terbagi ke dalam beberapa jenis berdasarkan cara kerjanya. *Database* relasional adalah jenis *database* yang menggunakan tabel untuk menyimpan data dan mengatur relasi antar tabel tersebut, sehingga memudahkan dalam pengelolaan data yang terstruktur. Sementara itu, database *NoSQL* mengadopsi model non-relasional dengan menyimpan data dalam format seperti dokumen, grafik, key-value, atau kolom, yang cocok untuk data dengan skema fleksibel. Di sisi lain, database terdistribusi merupakan sistem database yang tersebar di berbagai lokasi fisik, namun tetap beroperasi sebagai satu kesatuan yang terintegrasi, memungkinkan akses data dari berbagai sumber secara efisien. Dengan komponen dan jenis-jenis yang beragam, database menjadi solusi efektif untuk penyimpanan, pengelolaan, dan pemrosesan data dalam berbagai kebutuhan teknologi informasi.

Untuk memastikan data dapat disimpan, diakses, dan dikelola dengan baik, basis data dirancang dengan struktur yang sistematis. Struktur ini memungkinkan data terorganisir dengan lebih efisien, sehingga memudahkan proses pengambilan informasi, pengolahan data, serta pemeliharaan integritas antar komponen di

dalamnya. Berikut adalah penjelasan mengenai struktur dasar basis data dan elemen-elemen penting yang mendukung fungsionalitasnya.

a. Database Management System (DBMS)

DBMS (Database Management System) adalah sistem perangkat lunak yang dirancang untuk mengorganisasi dan mengolah database pada komputer. Sistem ini memungkinkan pengolahan data dengan referensi yang sama dan dapat diakses oleh berbagai aplikasi. Salah satu terobosan penting DBMS adalah *Relational Database Management System* (RDBMS), yang mengorganisasikan data dalam struktur terhubung untuk memaksimalkan pengelolaan antar kumpulan data dalam database. Selanjutnya, *Distributed Relational Database Management System* (DRDBMS) memungkinkan data tersebar di lokasi berbeda, tetapi tetap dapat diakses, diperbarui, dan dikelola layaknya data dalam satu database terpusat. Berbagai perusahaan menawarkan DBMS, seperti Oracle, MySQL, Microsoft Access, SQLite, dan lainnya. DBMS bersifat *general-purpose* dengan kemampuan untuk mendefinisikan, membangun, dan memanipulasi basis data untuk berbagai aplikasi.

Pengoperasian arsitektur basis data, dibuat dengan memakai format paket bahasa, yaitu :

- **DDL (*Data Definition Language*)**

Merupakan paket bahasa DBMS yang berguna untuk melakukan spesifikasi terhadap skema basis data, seperti create table, create index, alter table, drop view, drop index.

- **DML (*Data Manipulation Language*)**

Merupakan paket DBMS yang memberikan akses kepada user untuk memanipulasi data sebagaimana yang sudah dioperasikan sebelumnya.

DML (*Data Manipulation Language*) terdiri dari dua jenis, yakni prosedural dan non-prosedural. Prosedural DML mengharuskan pengguna untuk menentukan data yang diperlukan sekaligus cara untuk mendapatkannya. Contohnya adalah dBase III, FoxBase, dan FoxPro. Sebaliknya, non-prosedural DML hanya membutuhkan pengguna untuk menyebutkan data yang diinginkan tanpa perlu mengetahui cara pengambilannya, dengan contoh seperti SQL

(*Structured Query Language*) dan QBE (*Query By Example*). Perintah umum DML meliputi *Insert*, *Select*, *Update*, dan *Delete*. Selain itu, bahasa query juga mencakup DCL (*Data Control Language*), yang berfungsi untuk mengelola otorisasi akses data serta pengaturan ruang penyimpanan. Contoh perintah DCL adalah *Grant*, *Revoke*, *Commit*, dan *Rollback*. Ketiga jenis perintah utama, yaitu DDL, DML, dan DCL, kini digabungkan dalam SQL (*Structured Query Language*). Namun, implementasi SQL bervariasi di antara vendor perangkat lunak, sehingga tidak semua fitur didukung secara universal. Beberapa contoh perangkat lunak basis data yang menggunakan SQL antara lain DB2, Ingres, Informix, Oracle, MS-Access, MySQL, PostgreSQL, Rdb, dan Sybase.

b. Struktur Basis Data

Basis data terdiri dari berbagai komponen utama yang saling bekerja sama untuk menyimpan, mengelola, dan mengakses informasi.

- Tabel merupakan komponen dasar penyimpanan data dalam basis data. Tabel terdiri dari baris (*row*) yang merepresentasikan setiap rekaman atau data secara individu dan kolom (*column*) yang menunjukkan atribut atau properti dari data tersebut. Setiap tabel biasanya berisi informasi mengenai suatu entitas atau objek tertentu, seperti data pelanggan atau produk.
- Kunci (*key*) yang berfungsi untuk mengidentifikasi data secara unik di dalam tabel. Jenis-jenis kunci antara lain: kunci primer (*primary key*), yang digunakan untuk memastikan setiap baris bersifat unik tanpa nilai duplikat atau kosong; kunci asing (*foreign key*), yang menghubungkan dua tabel dengan merujuk ke kunci primer di tabel lain; kunci alternatif (*alternate key*), yaitu atribut lain yang dapat digunakan untuk identifikasi unik selain kunci primer; serta kunci kandidat (*candidate key*), yang mencakup atribut-atribut yang berpotensi menjadi kunci primer.
- Indeks (*index*) berperan penting dalam mempercepat proses pencarian dan pengambilan data di dalam tabel. Indeks dibuat pada satu atau lebih kolom untuk meningkatkan efisiensi akses data. Selain itu, relasi (*relationship*) memungkinkan adanya hubungan antar tabel dalam basis data. Relasi ini dapat dibedakan menjadi tiga jenis: *one-to-one* (satu baris dalam satu tabel berhubungan dengan satu baris di tabel lain), *one-to-many* (satu baris

berhubungan dengan beberapa baris di tabel lain), dan many-to-many (beberapa baris di satu tabel berhubungan dengan beberapa baris di tabel lain).

- View, yaitu tabel virtual yang menyajikan data dari satu atau lebih tabel. View berguna untuk menyederhanakan query dan meningkatkan keamanan dengan membatasi akses terhadap data yang bersifat sensitif. Dengan struktur ini, basis data menjadi lebih terorganisir dan mampu mendukung kebutuhan pengelolaan data yang kompleks secara efisien.

2.1.6 Sistem Informasi

Untuk mendefinisikannya, sistem informasi digambarkan sebagai perpaduan antara manusia, proses, data, dan teknologi yang bekerjasama untuk mengolah, mengumpulkan, memproses, menyimpan, hingga menyebarkan informasi dalam suatu organisasi. Menurut Dedy Rahman Prehanto dkk. (2020), sistem informasi memproses data (input) menjadi laporan (output), yang berfungsi sebagai dasar dalam pengambilan keputusan strategis. Laudon & Laudon (2020) menambahkan bahwa sistem informasi terdiri dari komponen-komponen yang terhubung, seperti perangkat keras, perangkat lunak, data, dan manusia, yang berperan dalam meningkatkan efisiensi operasional serta memberikan keunggulan kompetitif. Selain itu, ciri-ciri sistem informasi termasuk kemampuan memperbarui informasi, mengoreksi kesalahan, serta menyediakan data yang relevan dan akurat bagi penggunaannya.

Cragg & Zinatelli (1995) dan Taherdoost (2023) mendefinisikan sistem informasi sebagai tulang punggung organisasi modern, yang terdiri dari empat komponen utama: manusia, proses, data, dan teknologi. Komponen-komponen ini bekerja bersama untuk mengelola dan menyebarkan informasi guna mendukung pengambilan keputusan serta meningkatkan efisiensi dalam menjalankan operasi bisnis. Sistem informasi tidak hanya membantu dalam pemrosesan data real-time tetapi juga meningkatkan kolaborasi antarbagian dalam organisasi, sehingga menciptakan keunggulan strategis. Dengan demikian, sistem informasi sangat penting bagi organisasi untuk merespons perubahan pasar dan kebutuhan bisnis dengan lebih cepat dan tepat. Sistem informasi sendiri terdiri akan konsep dasar

dari dua elemen utamanya, yaitu sistem dan informasi. Kedua elemen ini saling melengkapi untuk menciptakan sistem informasi yang efektif dan efisien.

1. Sistem

Menurut Fatansyah (2015:11) bahwa sistem merupakan sebuah tatanan yang terdiri dari sejumlah komponen fungsional yang saling berhubungan dan Bersama-sama memiliki tujuan untuk memenuhi proses tertentu. Menurut Jogiyanto. H. M (2005) sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama, untuk melakukan kegiatan atau untuk menyelesaikan sasaran tertentu. Dalam suatu sistem, terdapat beberapa subsistem-subsistem yang saling bekerja satu dengan yang lainnya. Kerja sama yang dilakukan sistem guna untuk mendukung semua kegiatan yang ada dalam sebuah organisasi secara rutin. Dengan menjalankan sistem yang teratur dan benar, sesuai dengan prosedur yang berlaku, maka dapat membantu kelancaran segala kegiatan yang dilakukan.

2. Informasi

Dalam kehidupan sehari-hari, kita tidak terlepas dari yang namanya informasi. Informasi bisa kita dapatkan darimana saja, dari media cetak, sosial media, dan lainnya. Namun, informasi itu sendiri berperan sangat penting bagi aspek kehidupan manusia, terutama dalam bidang teknologi. Informasi dapat diartikan sebagai pengetahuan yang diperoleh melalui proses pembelajaran, pengalaman, atau instruksi. Penggunaan informasi telah menjadi bagian penting dalam berbagai aspek kehidupan, baik pada tingkat individu maupun organisasi. Nilai dari sebuah informasi bergantung pada tujuan penggunaannya, ketelitian dalam pengolahan data, serta relevansinya terhadap ruang, waktu, dan makna. Informasi juga dimanfaatkan dalam berbagai bidang seperti pendidikan, kesehatan, kesempatan kerja, maupun produk dan layanan. Dalam penelitian ini, konsep informasi sangat penting karena menjadi dasar dalam penyusunan sistem yang akan membantu pengelolaan data secara lebih terstruktur dan akurat.

2.1.4 Computer-based Information Systems (CBIS)

Sistem informasi berbasis komputer berarti bahwa komputer memiliki peran utama dalam menjalankan sistem informasi. Meskipun secara teoritis sebuah sistem informasi dapat diterapkan tanpa komputer, dalam praktiknya sulit membayangkan sistem yang kompleks dapat berfungsi dengan baik tanpa bantuan teknologi komputer. Sistem informasi yang tepat dan efisien pada kenyataannya hampir selalu terkait dengan pengolahan informasi yang dilakukan secara komputerisasi atau berbasis komputer. (Thatok Asmony et al., 2020). Sistem informasi berbasis komputer berfungsi untuk mengumpulkan, memproses, dan menyajikan informasi melalui penggunaan teknologi komputer, yang menjadi fondasi bagi berbagai aplikasi yang digunakan dalam dunia bisnis maupun sektor lainnya. Aplikasi, sebagai salah satu komponen dalam CBIS, berperan penting dalam menjalankan fungsi-fungsi spesifik yang dirancang untuk mempermudah pengguna dalam mengakses dan mengelola data secara efisien. Dengan memanfaatkan CBIS, aplikasi dapat menjalankan tugasnya secara otomatis, meningkatkan produktivitas, dan memastikan bahwa informasi yang dihasilkan akurat dan relevan untuk pengambilan keputusan.

1. Aplikasi

Aplikasi tidak dapat berdiri sendiri, yang berarti aplikasi merupakan sistem yang bergantung terhadap sistem operasi. Aplikasi merupakan program yang dirancang dengan tujuan tertentu atau dibuat sesuai dengan adanya domain permasalahan (Yafhizham, 2019). Menurut Jogiyanto Hartono (2004) dalam buku "Pengenalan Komputer", aplikasi adalah sebuah sistem yang dirancang sedemikian rupa untuk menghasilkan informasi menggunakan sarana komputer sebagai alat penunjangnya. Secara umum, aplikasi adalah perangkat lunak atau program komputer yang dibuat untuk melakukan perintah tertentu.

Sebagai perangkat lunak yang dirancang untuk menjalankan perintah tertentu, aplikasi memiliki berbagai jenis dan bentuk yang dapat disesuaikan dengan kebutuhan pengguna. Salah satu kategori aplikasi yang semakin populer adalah aplikasi berbasis website, yang memungkinkan pengguna untuk mengakses fungsionalitasnya melalui internet menggunakan browser. Aplikasi berbasis

website ini tidak hanya menawarkan kemudahan akses dari berbagai perangkat, tetapi juga mengintegrasikan berbagai sumber data dan layanan secara real-time. Dengan kemajuan teknologi dan peningkatan konektivitas internet, aplikasi berbasis website menjadi solusi yang efektif untuk memenuhi tuntutan operasional di berbagai sektor, termasuk bisnis, pendidikan, dan layanan publik. Oleh karena itu, aplikasi yang dikembangkan dalam konteks ini memainkan peran penting dalam mendukung pengambilan keputusan dan efisiensi proses di era digital saat ini.

2. Aplikasi berbasis Web

Aplikasi berbasis web adalah program yang dikembangkan menggunakan bahasa pemrograman seperti HTML, PHP, CSS, dan JavaScript, yang memerlukan web server dan browser untuk dijalankan, seperti Chrome, Firefox, Opera, Internet Explorer, atau Microsoft Edge. Aplikasi ini dapat diakses melalui jaringan lokal (LAN) maupun internet. Salah satu keunggulan utama aplikasi web adalah data yang tersimpan secara terpusat serta kemudahan akses, yang menjadikannya semakin diminati dan mudah diterapkan di berbagai sektor kehidupan. (Janner Simamarta, dkk 2021:113).

2.1.5 Persediaan

Istilah "persediaan" dapat diartikan dengan berbagai cara, namun pada dasarnya memiliki tujuan dan makna yang serupa. Menurut C. Roll Niswonger, Philip E. Fess, dan Carl S. Warren, "Persediaan adalah barang dagangan yang disimpan untuk dijual dalam kegiatan operasional perusahaan serta barang yang sedang dalam proses produksi atau disimpan untuk keperluan tersebut". Persediaan juga merupakan aset yang mencakup berbagai komoditas yang dimiliki oleh perusahaan dengan tujuan untuk dijual dalam periode bisnis normal, atau berupa barang yang sedang dalam proses produksi maupun yang menunggu untuk digunakan dalam proses produksi. Persediaan menjadi aspek penting dalam perusahaan karena merupakan salah satu aset termahal yang dimiliki, sehingga perusahaan harus mengelola persediaan dengan manajemen yang baik untuk mengurangi biaya operasional (Arifianti, 2022).

Sistem persediaan dalam buku manajemen operasi merupakan serangkaian kebijaksanaan yang memantau dan menentukan tingkat persediaan yang dijaga, kapan persediaan diisi, dan besar pesanan yang dilakukan. Tujuan dari sistem ini adalah menjamin akan ketersediaan sumber daya yang tepat. Menurut Ikatan Akuntansi Indonesia dalam buku “Standar Akuntansi Keuangan”, persediaan adalah :

1. Tersedia untuk dijual (dalam kegiatan operasi normal)
2. Dalam proses produksi (dalam kegiatan usaha normal)
3. Dalam bentuk bahan atau perlengkapan (*supplies*) untuk digunakan proses produksi

Tujuan utama dengan adanya persediaan adalah untuk mengatasi ketidakpastian (*safety stock*), menyediakan waktu tambahan bagi pengelola produksi dan pembelian, sertaantisipasi terhadap perubahan permintaan dan penawaran. Menurut Heizer dan Render yang dikutip dari buku Manajemen Operasi (2019), untuk mengakomodasi fungsi persediaan, perusahaan harus memiliki keempat jenis persediaan, diantaranya:

1. Persediaan bahan baku (*raw material inventory*), yaitu bahan baku yang belum memasuki proses produksi, untuk memisahkan para pemasok dari proses produksi
2. Persediaan barang setengah jadi (*working in process-WIP-inventory*), yaitu bahan baku yang sudah melalui tahap produksi, tetapi belum sampai menjadi sebuah produk jadi
3. MRO (*Maintenance/Repair/Operating*) atau pemeliharaan, juga diperlukan untuk mengantisipasi akan adanya kerusakan dalam salah satu proses produksi.
4. Persediaan barang jadi (*finished goods inventory*), yaitu produk akhir yang sudah jadi dan siap untuk dijual.

Alasan memiliki persediaan yang dimana salah satu metode untuk mencapai laba maksimal adalah dengan menekan biaya yang berhubungan dengan persediaan. Biaya persiapan dapat ditekan dengan memesan atau memproduksi dalam jumlah kecil, sementara biaya pemesanan dapat diminimalkan dengan

melakukan pesanan dalam jumlah besar dan jarang. Meminimalkan biaya penyimpanan mendorong perusahaan untuk menjaga jumlah persediaan tetap rendah atau bahkan nihil, sedangkan untuk menekan biaya pemesanan, perusahaan perlu memesan dalam jumlah besar, yang justru meningkatkan jumlah persediaan. Perusahaan harus menentukan strategi mana yang paling efektif untuk menekan biaya atau apakah lebih baik mengombinasikan kedua strategi tersebut. Alasan lain bagi perusahaan untuk menyimpan persediaan dalam jumlah besar adalah ketidakpastian permintaan. Dengan persediaan yang cukup, perusahaan dapat tetap memenuhi permintaan ketika terjadi peningkatan permintaan bahan atau produk yang tidak terduga, sehingga menjaga kepuasan pelanggan. (Utama, 2019).

2.1.6 Vending Machine

Vending machine adalah alat otomatis yang menyediakan berbagai jenis produk yang dapat dibeli oleh pelanggan tanpa perlu interaksi langsung dengan kasir. Konsumen hanya perlu memilih barang yang diinginkan dan mesin akan mengeluarkan produk tersebut, sehingga tidak perlu mengantri seperti di toko fisik. Produk yang bisa ditawarkan melalui vending machine sangat beragam, mulai dari makanan, minuman, perlengkapan sepeda, produk kecantikan, hingga layanan pembayaran tagihan, dan lainnya. Vending machine bertujuan mempercepat transaksi antara penjual dan pembeli. Dari sisi penjual, vending machine mengurangi kebutuhan untuk menyewa toko atau mempekerjakan karyawan, serta memanfaatkan ruang yang lebih sedikit dibandingkan toko fisik. Bagi pembeli, mesin ini memungkinkan pembelian tanpa antrian dan bisa diakses kapan saja. Selain itu, dalam situasi seperti pembatasan sosial, vending machine membantu mengurangi kontak langsung antar individu. (Ismail, 2022)

Vending machine memiliki peran penting dalam mempercepat transaksi, baik dari sisi penjual maupun pembeli, yang selaras dengan latar belakang permasalahan terkait pengelolaan penjualan di kantin Universitas Pembangunan Jaya. Dengan vending machine, penjual tidak perlu mengeluarkan biaya untuk toko fisik dan karyawan, sementara pembeli dapat membeli produk dengan cepat tanpa mengantri. Hal ini juga relevan dalam konteks pengelolaan data penjualan di kantin, di mana kehadiran vending machine yang otomatis dapat mengurangi kesalahan pencatatan

dan meningkatkan efisiensi proses transaksi, sejalan dengan tujuan pengembangan sistem yang lebih terstruktur.

2.2 Tinjauan Studi

Dalam penelitian ini, akan digunakan lima tinjauan Pustaka yang nantinya dapat mendukung penelitian. Berikut merupakan tinjauan Pustaka yang diambil :

1. **Tinjauan studi pertama** diambil dari penelitian yang berjudul **“Integrasi Aplikasi Monitoring dan Transaksi Pembayaran pada Kantin Sekolah”** yang ditulis oleh **Ardiansyah, S., Kautsar, A.I., Indahyanti, U., & Sumarno** pada tahun 2024. Penelitian ini membahas mengenai kurangnya sistem yang terintegrasi untuk monitoring dan transaksi pembayaran pada kantin sekolah. Sistem kantin yang masih konvensional menimbulkan berbagai kendala seperti pencatatan transaksi yang tidak efisien, kesulitan dalam pengawasan penggunaan uang saku siswa, serta lambatnya proses pembayaran. Selain itu, absennya sistem monitoring yang baik juga menjadi tantangan bagi pihak sekolah atau orang tua dalam memantau kebiasaan belanja siswa. Dalam penelitian ini, dikembangkan sebuah aplikasi yang mengintegrasikan fungsi monitoring dan transaksi pembayaran secara elektronik. Aplikasi ini dirancang untuk mempermudah pembayaran bagi siswa serta memberikan akses kepada pihak sekolah dan orang tua dalam memantau kebiasaan belanja siswa. Metode yang digunakan dalam penelitian ini adalah model *Waterfall*, yang terbukti efektif dalam implementasi sistem ini. Hasil penelitian menunjukkan bahwa aplikasi yang dikembangkan berhasil mengintegrasikan fungsi monitoring dan transaksi pembayaran dengan baik. Aplikasi ini mampu mempermudah pembayaran di kantin melalui sistem elektronik serta memungkinkan pihak sekolah dan orang tua untuk memantau pengeluaran siswa. Selain itu, aplikasi ini juga meningkatkan efisiensi operasional kantin dengan mempercepat transaksi dan mengurangi kesalahan pencatatan.

2. **Tinjauan studi kedua** diambil dari penelitian yang berjudul **“Rancang Bangun Sistem Informasi Point of Sale dengan Framework CodeIgniter pada CV**

Powershop” yang ditulis oleh **Maydianto & Ridho, M. R.** pada tahun 2021. Penelitian ini berfokus pada permasalahan pengolahan data di CV Powershop yang sering mengalami keterlambatan transaksi, kesalahan dalam pembuatan laporan, serta risiko keamanan data yang cukup tinggi. Penelitian ini bertujuan untuk merancang dan membangun sistem informasi berbasis framework CodeIgniter yang dapat meningkatkan efisiensi transaksi, mengurangi kesalahan pencatatan, serta meningkatkan keamanan data. Dalam penelitian ini, metode yang digunakan adalah *UML Development System* dengan framework CodeIgniter. Hasil penelitian menunjukkan bahwa sistem informasi yang dikembangkan berhasil mengotomatisasi proses penjualan, pengelolaan data barang, pengolahan data supplier, serta pembuatan laporan. Dengan adanya sistem ini, CV Powershop dapat meningkatkan efisiensi operasionalnya, mengurangi kesalahan pencatatan, serta menjaga keamanan data transaksi.

3. Tinjauan studi ini diambil dari penelitian yang berjudul "**Rancang Bangun Aplikasi Inventaris Barang Berbasis Android Pada PT. Nuansa Indah Mane**", yang ditulis oleh **Riana, Andi Christian, dan Yuntari Purbasari**. Penelitian ini berfokus pada permasalahan pengelolaan inventaris barang di PT. Nuansa Indah Mane yang masih menggunakan sistem manual, sehingga sering terjadi kendala dalam pencatatan data barang masuk, barang keluar, dan barang yang dikembalikan. Penelitian ini bertujuan untuk merancang dan membangun aplikasi inventaris berbasis Android guna meningkatkan efisiensi dalam pengelolaan data inventaris perusahaan. Dalam penelitian ini, metode yang digunakan adalah pengembangan perangkat lunak dengan pendekatan berbasis sistem informasi. Aplikasi yang dikembangkan dirancang agar dapat membantu pencatatan barang masuk dan keluar secara real-time serta mempermudah monitoring stok barang. Hasil penelitian menunjukkan bahwa aplikasi inventaris yang dikembangkan berhasil mengotomatisasi proses pencatatan inventaris, mengurangi kesalahan input data, serta meningkatkan efisiensi dalam manajemen barang di PT. Nuansa Indah Mane. Dengan adanya sistem ini, perusahaan dapat lebih mudah mengontrol stok barang, mempercepat proses pencatatan, dan meningkatkan akurasi data inventaris.

4. Tinjauan studi keempat diambil dari penelitian yang berjudul **“Implementasi API pada Sistem Informasi Tagihan Listrik dengan Model Extreme Programming (Studi Kasus: PLTS Kantin Universitas Pamulang)”** yang ditulis oleh **Husadif, A. & Ratama, N.** pada tahun 2023. Penelitian ini membahas permasalahan dalam pengelolaan tagihan listrik di kantin Universitas Pamulang yang masih dilakukan secara manual, sehingga proses pencatatan dan perhitungan tagihan listrik menjadi tidak efisien dan rawan kesalahan. Tujuan utama dari penelitian ini adalah membangun sistem informasi tagihan listrik yang lebih efektif dan efisien. Sistem ini dirancang untuk mempercepat pemasukan data, meningkatkan akurasi pencarian informasi, serta mempercepat perhitungan tagihan listrik agar lebih akurat. Penelitian ini menggunakan metode pengembangan sistem dengan model *Extreme Programming (XP)*. Hasil penelitian menunjukkan bahwa implementasi sistem informasi berbasis framework Laravel dapat meningkatkan efisiensi dalam pengelolaan tagihan listrik, mempermudah pencarian informasi, serta mempercepat proses pemrosesan data.

5. Tinjauan studi kelima diambil dari penelitian yang berjudul **"Aplikasi Pencatatan Keuangan Berbasis Website Dengan Metode Rapid Application Development Pada PT Samsriwi Adi Megah"**, yang ditulis oleh **Dhea Apisca, Nurhadi Surojudin, dan Edora.** Penelitian ini berfokus pada permasalahan pengelolaan pencatatan keuangan di PT Samsriwi Adi Megah yang masih menggunakan Microsoft Excel, sehingga mengalami kendala dalam pencarian data keuangan dalam periode tertentu akibat penyimpanan data yang tersebar di banyak lembar kerja atau file terpisah. Penelitian ini bertujuan untuk merancang dan membangun aplikasi pencatatan keuangan berbasis website menggunakan metode Rapid Application Development (RAD) guna meningkatkan efisiensi pengolahan data keuangan. Dengan adanya sistem yang terintegrasi, perusahaan dapat mempercepat proses pencatatan, memudahkan pembuatan laporan keuangan, serta meningkatkan pengambilan keputusan berbasis data yang lebih akurat. Hasil penelitian menunjukkan bahwa aplikasi yang dikembangkan mampu mengotomatisasi pencatatan transaksi keuangan, meningkatkan akurasi data, serta mempercepat proses pencarian informasi keuangan. Selain itu, sistem

ini memungkinkan aksesibilitas data keuangan secara lebih fleksibel, sehingga memudahkan manajemen dalam mengontrol keuangan dan membuat laporan dengan lebih cepat. Dengan adanya sistem pencatatan keuangan berbasis website ini, PT Samsriwi Adi Megah dapat meningkatkan efisiensi operasional dan transparansi dalam pengelolaan keuangan perusahaan.

