

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Berikut ini beberapa hasil penelitian serupa yang dirangkum kedalam tabel

2.1 penelitian terdahulu:

Tabel 2. 1 Penelitian Terdahulu

No	Penelitian	Hasil Penelitian
1	<p>Judul: Membangun Sistem Inventaris Laboratorium Jurusan Administrasi Bisnis Berbasis PHP dan MySQL dengan Framework Larafel dan Bootstrap</p> <p>Volume: Vol. 7 Nomor: No. 2 Penulis: Said Muhammad, Riswan Yunida Agus Irwandi, Rahma Indera, Eko Sabar Prihatin Tahun: 2021</p>	<p>Keuntungan sistem terkomputerisasi termasuk peningkatan efisiensi dan efektivitas data dengan akurasi lebih tinggi. Ini juga menghemat waktu dalam pencarian, pencatatan, dan penginputan data, serta mengurangi pekerjaan berulang dan memungkinkan pengeditan data yang mudah. Aplikasi online memungkinkan pengguna untuk mengajukan permohonan peminjaman alat di luar kampus, sehingga data inventaris Laboratorium Jurusan Administrasi Bisnis dapat tercatat dan terekam dalam semua transaksinya.</p>
2	<p>Judul: Perancangan dan Implementasi Sistem Inventaris Barang Menggunakan Metode Web Engineering (Studi Kasus : SMK Trisakti Jaya Bandar Lampung)</p> <p>Penulis: Doni Mahendra, Setiawansyah, Adhie Thyo Priandika Volume: Vol. 3 Nomor: No. 2 Tahun: 2022 Jurnal: Jurnal Teknologi dan Sistem Informasi (JTISI), Juni 2022, 33-37</p>	<p>Aplikasi inventaris barang di SMK Trisakti Jaya Bandar Lampung dirancang menggunakan model web engineering dengan tahapan customer communication, planning, modelling, construction, dan deployment. Implementasi menggunakan Bahasa pemrograman PHP dan framework CodeIgniter dengan database MySQL. Pengujian menggunakan model blackbox testing dengan hasil nilai 100% sesuai dengan pengujian fungsionalitas sistem. Hasil pengujian dengan Web Quality 4.0 mendapatkan skor total 91,53%, menunjukkan kriteria yang sangat baik.</p>
3	<p>Judul: Pengembangan Program Aplikasi Inventory Menggunakan Metode Prototype (Studi Kasus PT. Indowira Putra Paint)</p> <p>Penulis: Ruchiyat Firmansyah, Rizal Rachman</p>	<p>Pembuatan perangkat lunak aplikasi inventaris menggunakan PHP dan MySQL dapat memudahkan pengolahan data komoditas. Aplikasi diubah desainnya agar lebih dinamis dan estetis. Didesain untuk dua</p>

No	Penelitian	Hasil Penelitian
	Volume: 1 Nomor: 4 Jurnal: Jurnal Cakrawala Ilmiah, 461–472 Tahun: 2021	pengguna utama guna memfasilitasi pemrosesan data yang cepat dan akurat, serta memungkinkan direktur utama untuk membaca laporan secara kronologis tanpa harus menunggu ringkasan dari bagian gudang.
4	Judul: Penerapan Metode Waterfall untuk Perancangan Sistem Informasi Inventory pada Toko Keramik Bintang Terang Penulis: Kurniawati, Mohammad Badrul Volume: Vol. 8 Nomor: No. 2 Tahun: 2021 Jurnal: Vol. 8 No. 2 (2021): Prosisko, 47-52	Sistem web Sistem Inventory memungkinkan Admin Toko Keramik Bintang Terang untuk mencatat keluar-masuk barang secara terstruktur, memastikan stok barang tercatat dengan akurat. Sistem ini juga memudahkan proses pengecekan dan pencarian data terkait transaksi penjualan, pembelian, serta pengaruhnya terhadap stok barang, serta memberikan bantuan dalam efisiensi waktu pencatatan dan penyusunan data barang.
5	Judul: Implementasi Metode Rapid Application Development pada Pengembangan Aplikasi Inventory Barang Penulis: Siti Anisah Volume: Vol. 7 Nomor: No. 1 Tahun: 2022 Jurnal: STRING (Satuan Tulisan Riset dan Inovasi Teknologi), Agustus 2022, 57-65	Sistem informasi inventory barang dengan metode Rapid Application Development (RAD) sangat efisien dalam perancangannya dan mempercepat pengolahan data. Ini memberikan kemudahan dalam pengelolaan data persediaan barang seperti data barang, permintaan barang, barang masuk dan keluar, stock opname, dan permohonan pembelian barang. Pencarian stok barang juga lebih cepat dan mudah karena dapat diakses melalui bagian data barang dalam sistem, memungkinkan pengontrolan stok barang yang lebih akurat.

Berdasarkan penelitian terdahulu, research gap dari tabel tersebut adalah perbedaan penggunaan framework dan metode yang digunakan dalam penelitian sebelumnya. Penelitian pertama menggunakan framework Laravel, sedangkan penelitian ini menggunakan framework Bootstrap, menciptakan perbedaan signifikan dikarenakan Laravel digunakan untuk pengembangan sisi server dan logika aplikasi, sementara Bootstrap digunakan untuk pengembangan sisi klien dan desain UI [5]. Selanjutnya, penelitian kedua dan ketiga menggunakan metode Web

Engineering dan metode prototipe secara berturut-turut, yang dapat menjadi pedoman dalam penelitian ini [6][7].

Di sisi lain, penelitian keempat menggunakan metode pengembangan sistem Waterfall sedangkan penelitian ini menggunakan metode pengembangan sistem yang berbeda yaitu Rapid Application Development (RAD), namun memiliki kesamaan dalam capaian yang ingin diperoleh yaitu pembuatan aplikasi inventory, dimana lewat hasil penelitian terdahulu menemukan penggunaan metode Waterfall untuk merancang aplikasi inventory memiliki tahapan yang kaku dan tidak cukup fleksibel untuk melakukan perubahan kebutuhan pengembangan aplikasi yang muncul selama proses pengembangan pada studi kasus ini, sehingga memperkuat keputusan peneliti untuk menggunakan metode pengembangan RAD pada penelitian ini [8]. Penelitian kelima menggunakan metode Rapid Application Development (RAD), yang sejalan dengan penelitian ini, dapat menjadi referensi yang berguna dalam pengembangan [9]. Analisis research gap diharapkan dapat memunculkan inovasi baru dalam penelitian ini dimana penelitian ini menyajikan pembaruan dengan menggunakan metode pengembangan RAD untuk pembuatan aplikasi persediaan yang menganut sistem manajemen stok FIFO, sehingga penelitian ini dapat merespons perubahan kebutuhan yang muncul selama proses pengembangan dengan lebih cepat dan efisien, memastikan bahwa aplikasi inventory dapat berkembang sesuai dengan kebutuhan pengguna yang dinamis serta memperbaiki permasalahan penumpukan stok barang lama digudang, yang diharapkan dapat digunakan sebagai acuan penelitian di masa depan, serta pengembangan dari penelitian sebelumnya [5][9].

2.2 Tinjauan Teori

2.2.1 Database

2.2.1.1 Pengertian database

Database adalah kumpulan data terstruktur dan terorganisir dalam sistem komputer, disimpan dalam tabel yang terhubung melalui relasi. Tujuannya adalah menyediakan cara efisien dalam menyimpan, mengelola, dan mengakses data, memungkinkan operasi seperti pencarian, penyaringan, pembaruan, dan penghapusan data dengan mudah. Database juga memfasilitasi berbagai aplikasi dengan

menggunakan data yang sama, membantu organisasi mengelola informasi secara terpusat, mengurangi redundansi data, meningkatkan keamanan, dan memungkinkan akses bersama oleh banyak pengguna. Dalam pengembangan perangkat lunak, penggunaan database penting karena memisahkan logika aplikasi dari data, memungkinkan pengembangan aplikasi yang fleksibel, skalabel, dan mudah dipelihara.

2.2.1.2 Sejarah database

Database dimulai dibentuk pada tahun 1960-an oleh Edgar F. Codd, seorang ilmuwan komputer dari IBM, mengenalkan model relasional untuk penyimpanan dan manajemen data, yang menjadi dasar bagi pengembangan sistem database modern. Pada dekade 1970-an, IBM meluncurkan System R, produk database relasional pertama. Selama tahun 1980-an, Oracle Corporation memperkenalkan Oracle Database, yang menjadi salah satu sistem database relasional paling populer, sementara perusahaan lain seperti Microsoft dan Sybase juga memasuki pasar database. Pada 1990-an, pertumbuhan internet memicu permintaan akan database yang efisien untuk aplikasi web, yang mendorong munculnya MySQL, PostgreSQL, dan Microsoft SQL Server. Seiring dengan perkembangan teknologi komputasi dan penyimpanan data pada tahun 2000-an, database berkembang menjadi lebih canggih, termasuk database terdistribusi, berbasis cloud, dan NoSQL, yang dirancang untuk menangani berbagai jenis data dan skala yang besar.

2.2.2 Inventory

Inventory atau inventaris merujuk pada suatu sistem yang digunakan oleh perusahaan untuk mengatur dan mengelola data terkait persediaan barang atau stok yang dimilikinya. Stok tersebut melibatkan berbagai jenis, mulai dari bahan baku, barang setengah jadi, hingga barang jadi yang disimpan dalam perusahaan. Tujuan utama dari penyimpanan ini adalah

untuk memastikan ketersediaan produk guna memenuhi permintaan dan kebutuhan pelanggan secepat mungkin.

Peran inventaris sangat krusial dalam jalannya proses bisnis karena berkaitan langsung dengan pelanggan dan calon pelanggan. Oleh karena itu, diperlukan sistem manajemen yang efisien untuk mendukung kelancaran operasional bisnis. Sistem inventaris berfungsi sebagai penanda ketersediaan barang dalam perusahaan, memungkinkan perusahaan untuk melakukan peramalan (forecasting) dan membuat keputusan terkait penyediaan stok untuk memastikan kenyamanan pelanggan.

Penerapan manajemen sistem inventaris yang baik tidak hanya memberikan kemudahan bagi perusahaan dalam pengelolaan stok secara efektif dan efisien, tetapi juga dapat mengurangi potensi kesalahan dan masalah yang mungkin timbul.

2.2.3 Expedisi

Expedisi pada pabrik merupakan suatu proses yang esensial dalam rantai pasok sebuah perusahaan manufaktur. Pada dasarnya, ekspedisi ini mencakup semua kegiatan terkait dengan pengiriman dan distribusi produk dari pabrik kepada konsumen atau distributor. Di awal proses, setelah barang-barang diproduksi dan dinyatakan siap untuk dikirim, ekspedisi memainkan peran penting dalam memastikan bahwa produk mencapai tujuan akhirnya dengan aman, tepat waktu, dan dalam kondisi optimal. Selain itu, ekspedisi pada pabrik juga melibatkan perencanaan logistik yang cermat, termasuk pemilihan metode pengiriman yang paling efisien, pengaturan rute pengiriman yang optimal, dan manajemen stok untuk meminimalkan risiko kekurangan atau kelebihan persediaan.

2.3 Teori tentang Framework dan Algoritma yang digunakan

2.3.1 Rapid Application Development

2.3.1.1 Pengertian RAD

Rapid Application Development, atau dikenal sebagai RAD, adalah salah satu metode pengembangan perangkat lunak yang termasuk dalam teknik bertingkat dan menitikberatkan pada siklus pengembangan yang singkat[14]. Metode RAD dapat menjadi pilihan bagi perusahaan yang ingin mengembangkan sistem atau aplikasi dengan cepat.

2.3.1.2 Tahapan RAD

RAD sendiri memiliki beberapa tahapan yang dipisah kedalam 3 tahapan utama, yaitu:

1. Requirement Planning

Tahapan ini merupakan langkah awal dalam menerapkan metode RAD, di mana identifikasi dan analisis kebutuhan, tujuan, serta desain awal sistem dilakukan. Pengguna akhir akan terlibat sejak awal tahap ini.

2. RAD Design Workshop

Langkah berikutnya adalah merencanakan dan menganalisis solusi terhadap permasalahan yang ada. Setelah tahap ini selesai, hasil dari Workshop Desain RAD akan berupa struktur sistem dan data yang akan diimplementasikan. Pada tahap ini, pengguna akhir akan melakukan penilaian untuk menilai kelayakan sistem dan apakah perlu perbaikan lebih lanjut.

3. Implementation

Langkah terakhir dari tahapan RAD merupakan implementasi, pada tahapan ini dilakukan finalisasi pada sistem dan perancangan sistem. Setelah itu, dilakukan pengujian kepada pengguna akhir yang akan menggunakan sistem untuk memastikan kelancaran implementasinya. Jika tidak ada masalah, sistem akan diimplementasikan secara penuh. Jika masih terdapat kendala, perbaikan akan dilakukan sebelum sistem dapat diimplementasikan secara keseluruhan.

2.3.1.3 Kelebihan RAD

Metode RAD memiliki beberapa keunggulan diantaranya yaitu:

1. Mudah Diimplementasikan:

Proses pengembangan RAD cenderung lebih sederhana dan mudah diimplementasikan, terutama untuk proyek-proyek kecil hingga menengah. Ini membuatnya cocok untuk proyek-proyek dengan batasan anggaran dan waktu yang ketat.

2. Pengembangan Cepat

RAD memungkinkan pengembangan aplikasi dengan waktu yang lebih singkat dibandingkan dengan metode tradisional. Pendekatan ini memungkinkan pengembang untuk fokus pada aspek-aspek kritis dan mendesain prototipe yang dapat diuji oleh pengguna lebih awal dalam siklus pengembangan.

2.3.1.4 Kekurangan RAD

Metode RAD memiliki beberapa kekurangan diantaranya yaitu:

1. Kurang cocok untuk proyek yang memiliki kompleksitas tinggi
RAD biasanya lebih cocok untuk proyek-proyek kecil hingga menengah. Pada proyek-proyek yang besar dan kompleks, penggunaan RAD dapat menjadi kurang efisien karena kompleksitas pengelolaan proyek yang tinggi dan kebutuhan untuk struktur pengembangan yang lebih formal.

2. Kesulitan dalam menentukan kebutuhan

Proses pengembangan yang cepat dapat mengakibatkan kesulitan dalam menentukan kebutuhan dengan jelas sejak awal. Hal ini dapat menyebabkan perubahan kebutuhan yang sering, yang pada gilirannya dapat mempengaruhi jadwal dan biaya proyek.

- 3 Kurangnya komperhensi yang mendalam untuk melakukan dokumentasi karena keterbatasan waktu

Fokus pada pengembangan yang cepat kadang-kadang dapat menghasilkan kurangnya dokumentasi yang memadai. Hal ini dapat menjadi masalah jika perlu dilakukan pemeliharaan atau

perubahan pada aplikasi di masa depan, karena kurangnya panduan dan pemahaman yang baik tentang struktur aplikasi.

2.3.2 Prototyping

Pengembangan perangkat lunak dengan metode prototyping melibatkan pembuatan model awal atau prototipe aplikasi sebelum versi finalnya. Prosesnya dimulai dengan menciptakan representasi visual atau fungsionalitas aplikasi yang mencakup fitur kunci. Prototipe tersebut difungsikan untuk berinteraksi dengan pengguna, mendapatkan umpan balik, dan mengenali perbaikan yang diperlukan. Keterlibatan aktif pengguna sepanjang siklus pengembangan memastikan bahwa kebutuhan dan harapan mereka terpenuhi. Melalui iterasi dan perbaikan berdasarkan umpan balik, metode ini memungkinkan pengembangan yang lebih cepat, identifikasi risiko, serta peningkatan desain secara efisien.

2.3.3 FIFO

Metode FIFO, yang merupakan singkatan dari First In, First Out, adalah cara yang sederhana namun efisien dalam menilai dan mengeluarkan barang dalam konteks akuntansi dan manajemen persediaan. Prinsipnya adalah bahwa barang yang pertama kali dimasukkan ke dalam persediaan akan diprioritaskan untuk dikeluarkan pertama kali saat terjadi penjualan atau penggunaan. Dengan menggunakan FIFO, nilai persediaan yang tersisa dihitung berdasarkan harga barang terbaru, sehingga mencerminkan nilai pasar aktual. Keunggulan utamanya adalah bahwa dalam kondisi ketika harga barang terus naik, metode ini biasanya menghasilkan laba yang lebih tinggi pada laporan laba rugi karena biaya yang lebih rendah digunakan untuk menghitung harga barang yang terjual. Meskipun prinsipnya sederhana, FIFO tetap menjadi metode yang populer karena kemudahan pemahaman dan penerapannya, menjadikannya pilihan yang sering digunakan dalam manajemen persediaan.

2.3.4 PHP

PHP (Hypertext Preprocessor) adalah bahasa pemrograman server-side yang umumnya digunakan untuk pengembangan aplikasi web. Didesain

khusus untuk membuat halaman web dinamis, PHP memungkinkan penyisipan script langsung ke dalam HTML dan menyediakan sintaksis yang mirip dengan C dan Perl. Fleksibilitas tipe data, fungsi bawaan, serta integrasi mudah dengan berbagai database membuat PHP populer di kalangan pengembang web. Berkat pustaka dan framework seperti Laravel, PHP menyediakan alat yang kuat untuk membangun aplikasi web kompleks dengan keamanan yang memadai. Sebagai perangkat lunak sumber terbuka, PHP terus berkembang dan tetap menjadi salah satu pilihan utama dalam dunia pengembangan web.

2.3.5 MySQL

MySQL adalah sistem manajemen basis data relasional (RDBMS) yang sering digunakan untuk menyimpan dan mengelola data dalam aplikasi web. Dikembangkan oleh perusahaan Swedia, MySQL AB, MySQL kemudian diakuisisi oleh Oracle Corporation. MySQL merupakan perangkat lunak sumber terbuka, artinya kode sumbernya dapat diakses dan dimodifikasi oleh pengguna sesuai kebutuhan. MySQL mendukung bahasa SQL (Structured Query Language) untuk melakukan operasi seperti pengambilan data (query), pembaruan, penghapusan, dan penambahan data dalam database. MySQL banyak digunakan sebagai basis data backend dalam kombinasi dengan bahasa pemrograman seperti PHP, Python, dan Java untuk mengembangkan aplikasi web dinamis. Keandalannya, kinerja yang baik, dan dukungan komunitas yang luas membuat MySQL menjadi pilihan populer di kalangan pengembang perangkat lunak.

2.3.6 JavaScript

JavaScript adalah sebuah bahasa pemrograman tingkat tinggi yang biasanya digunakan untuk mengembangkan skrip atau script di sisi klien (client-side) dalam konteks pengembangan web. Didesain untuk memberikan interaktivitas pada halaman web, JavaScript memungkinkan pengembang untuk membuat elemen-elemen dinamis, mengelola peristiwa, dan berinteraksi dengan pengguna melalui peramban (browser). JavaScript

bersama dengan HTML dan CSS, merupakan satu dari tiga teknologi utama dalam pengembangan web, yang memberikan kemampuan untuk membuat tampilan dan perilaku yang dinamis. Walaupun biasanya digunakan di sisi klien, JavaScript juga dapat digunakan di sisi server dengan bantuan runtime environment seperti Node.js. JavaScript adalah bahasa pemrograman yang mendukung pemrograman berbasis objek dan memiliki dukungan luas di berbagai peramban web modern.

2.3.7 Codeigniter 3

CodeIgniter 3 adalah framework PHP yang ringan dan cepat, dirancang untuk memudahkan pengembangan aplikasi web dinamis. Dengan menggunakan pendekatan Model-View-Controller (MVC), CodeIgniter 3 memisahkan logika aplikasi, tampilan, dan data, memungkinkan pengembangan yang lebih terstruktur dan mudah dikelola[15]. Framework ini dikenal karena konfigurasi minimal yang diperlukan dan jejak memori yang kecil, serta dilengkapi dengan dokumentasi yang komprehensif, menjadikannya ideal untuk pengembang yang mencari solusi cepat dan efisien untuk membangun aplikasi web.

2.3.8 Bootstrap

Bootstrap adalah sebuah framework front-end open-source yang dikembangkan oleh Twitter. Dirancang untuk mempermudah dan mempercepat pengembangan web responsif, Bootstrap menyediakan serangkaian alat dan gaya desain (CSS) yang dapat digunakan secara konsisten di berbagai proyek. Dengan komponen-komponen siap pakai seperti tombol, formulir, navigasi, dan grid system yang responsif, Bootstrap memungkinkan pengembang untuk dengan cepat membangun antarmuka pengguna yang estetik dan dapat diakses secara mobile. Kelebihan Bootstrap juga termasuk dokumentasi yang baik, kompatibilitas lintas peramban, serta kemampuan kustomisasi yang fleksibel, menjadikannya salah satu framework front-end yang populer di dunia pengembangan web.

2.3.9 Entity Relationship Diagram

Entity Relationship Diagram (ERD) adalah metode yang digunakan untuk menggambarkan kebutuhan data suatu organisasi, biasanya oleh seorang Analis Sistem pada tahap analisis persyaratan proyek pengembangan sistem[10]. Teknik ini memberikan fondasi bagi desain database relasional yang mendasari sistem informasi yang dikembangkan. ERD, bersama dengan informasi pendukung, merupakan model data yang kemudian digunakan sebagai spesifikasi untuk database. jenis Entity Relationship Diagram yang dideskripsikan dibawah:

1. Entitas

Objek dalam kehidupan nyata yang dapat dibedakan dari objek lainnya. Entitas terdiri dari sejumlah atribut yang mengidentifikasi atau membedakan satu sama lain. Setiap entitas baru harus memiliki satu atribut unik, yang disebut sebagai kunci utama (primary key).

2. Atribut

Atribut memiliki isi dari berbagai elemen yang dapat diidentifikasi masing-masing, berikut ini kedua jenis dari atribut:

- a) Identifier (key) digunakan sebagai cara untuk menentukan sebuah entity yang unik atau biasa dibesut dengan primary key.
- b) Descriptor diperlukan untuk melakukan spesifikasi dari Karakteristik suatu entri yang unik.

3. Kardinalitas

Merupakan sejumlah himpunan dari relasi antar entitas. Berikut ini beberapa jenis dari kardinalitas:

- a) One to one, merupakan hubungan entitas A dan B yang masing-masing berjumlah satu hubungan.
- b) One to many, merupakan hubungan entitas A yang terhubung dengan B berjumlah lebih dari satu.

- c) Many to many, merupakan hubungan entitas A yang lebih dari satu dengan entitas B yang memiliki lebih dari satu pula.

2.3.10 Unified Modeling Language (UML)

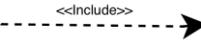
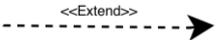
Unified Modeling Language (UML) adalah bahasa pemodelan standar yang dipakai untuk mendefinisikan, memvisualisasikan, merancang, dan mendokumentasikan komponen sistem perangkat lunak[16]. UML menyediakan berbagai notasi grafis untuk membuat model abstrak dari sistem, mencakup berbagai aspek seperti struktur, perilaku, dan interaksi sistem. Berikut ini 3 (tiga) jenis UML yang digunakan dalam penelitian:

2.3.10.1 Use Case Diagram

Use Case Diagram merupakan alat visual yang digunakan dalam pengembangan perangkat lunak dengan tujuan untuk menunjukkan interaksi antara sistem dan pengguna[16]. Ini menggambarkan fungsionalitas sistem dari sudut pandang pengguna, menampilkan aksi atau skenario yang dapat dilakukan pengguna dengan sistem tersebut. Diagram ini terdiri dari aktor, yang merupakan entitas luar yang berinteraksi dengan sistem, dan use case, yang merupakan tindakan atau fungsi sistem. Diagram use case membantu dalam memahami kebutuhan fungsional sistem dan membentuk dasar untuk merancang sistem yang sesuai dengan kebutuhan pengguna. Berikut ini tabel 2.2 yang menjelaskan mengenai jenis simbol Use Case Diagram:

Tabel 2. 2 Jenis Simbol Use Case Diagram[16]

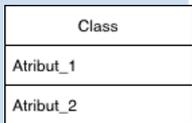
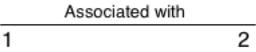
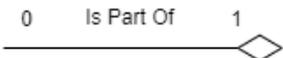
Simbol	Keterangan
	<p>Actor: Simbol aktor dalam diagram aktivitas mewakili pihak luar yang berinteraksi dengan sistem atau proses yang dipodelkan. Aktor bisa berupa pengguna manusia, sistem eksternal, atau perangkat lain yang terlibat dalam proses tersebut. Dengan memanfaatkan simbol aktor, diagram aktivitas mencerminkan cara keterlibatan entitas luar dalam proses tersebut, memfasilitasi pemahaman mengenai interaksi antara sistem atau proses dengan lingkungannya.</p>

Simbol	Keterangan
	<p>Use Case: Simbol use case dalam diagram aktivitas menggambarkan fungsi sistem yang sedang dianalisis dalam suatu proses tertentu. Representasinya berbentuk elips dengan label yang menjelaskan aksi yang dilakukan oleh sistem. Melalui simbol ini, kita dapat menggambarkan bagaimana sistem berperilaku dalam suatu skenario, membantu pemodelan dan pemahaman alur kerja secara menyeluruh.</p>
	<p>Association Relationship: Simbol association relationship dalam diagram aktivitas mengindikasikan hubungan antara dua elemen atau aktivitas, seperti aliran kontrol atau data antara keduanya. Biasanya direpresentasikan sebagai panah dengan label yang menjelaskan jenis hubungan antara aktivitas tersebut. Dengan simbol ini, diagram aktivitas membantu dalam memahami alur kerja atau proses dengan lebih baik.</p>
	<p>Include Relationship: Simbol include relationship dalam diagram aktivitas menggambarkan bagaimana satu aktivitas dapat memasukkan atau mencakup aktivitas lain dalam jalannya proses. Hal ini menunjukkan bahwa aktivitas yang dimasukkan dijalankan sebagai bagian dari aktivitas utama. Dengan menggunakan simbol ini, diagram aktivitas membantu dalam menggambarkan proses secara detail.</p>
	<p>Extend Relationship: Simbol extend relationship dalam diagram aktivitas menggambarkan bagaimana satu aktivitas dapat menambahkan fungsi tambahan ke dalam aktivitas utama tergantung pada kondisi tertentu. Dengan menggunakan simbol ini, diagram aktivitas membantu dalam memodelkan proses dengan mempertimbangkan fungsi tambahan yang mungkin dibutuhkan.</p>
	<p>Generalization Relationship: Generalization relationship dalam diagram aktivitas menunjukkan cara aktivitas yang lebih khusus menerima atribut dan perilaku dari aktivitas yang lebih umum. Ini membantu dalam menggambarkan struktur hierarki aktivitas secara singkat, memudahkan pemodelan proses yang kompleks.</p>

2.3.10.2 Class Diagram

Class Diagram bertujuan dalam memvisualisasikan struktur dari sistem, sehingga memudahkan pemahaman tentang bagaimana kelas-kelas berinteraksi dan bertanggung jawab dalam sistem yang sedang dikembangkan[16]. Berikut ini tabel 2.3 yang menjelaskan mengenai jenis simbol Class Diagram dibawah:

Tabel 2. 3 Jenis Simbol Class Diagram[16]

Simbol	Keterangan
	<p>Class:</p> <p>Class dalam diagram kelas mewakili objek atau entitas dalam sistem dengan atribut dan perilaku khusus. Kelas direpresentasikan sebagai persegi panjang yang terbagi menjadi tiga bagian, mencakup nama kelas, atribut, dan metode.</p>
	<p>Association:</p> <p>Association dalam diagram kelas menunjukkan keterkaitan antara dua atau lebih kelas dalam sistem, menggambarkan cara objek dalam satu kelas terhubung dengan objek dalam kelas lain. Ini diperlihatkan melalui garis yang menghubungkan kelas-kelas, dengan panah menunjukkan arah hubungan.</p>
	<p>Inheritance:</p> <p>Inheritance merupakan relasi yang kelas turunannya mewarisi atribut dan perilaku dari kelas induk, menggambarkan konsep pewarisan. Ini ditunjukkan oleh garis dengan panah dan segitiga di ujungnya, menunjukkan arah turunan ke induk.</p>
	<p>Aggregation:</p> <p>Aggregation adalah hubungan di mana objek dalam satu kelas berhubungan dengan objek dalam kelas lain sebagai bagian dari kesatuan yang lebih besar. Representasinya adalah garis dengan ujung berlian menunjukkan "whole-part relationship".</p>
	<p>Composition:</p> <p>Composition mirip dengan agregasi, namun lebih kuat karena objek dalam kelas satu tidak dapat ada tanpa objek dalam kelas lain. Ini direpresentasikan oleh garis dengan ujung berlian dan garis solid di ujungnya,</p>

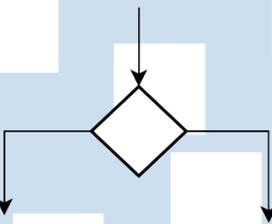
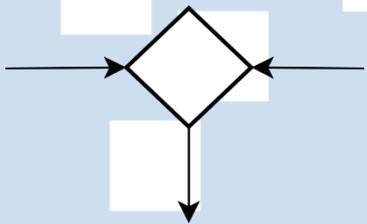
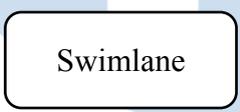
Simbol	Keterangan
	menunjukkan hubungan yang lebih erat dan terintegrasi.

2.3.10.3 Activity Diagram

Diagram aktivitas merupakan salah satu varian diagram dalam UML (Unified Modeling Language) yang dipakai untuk mengilustrasikan jalur kerja atau urutan aktivitas dalam suatu sistem atau proses bisnis[16]. Fungsi utama diagram ini adalah untuk mendukung kegiatan pemodelan, analisis, dan dokumentasi proses atau rangkaian kerja yang terjadi. Berikut ini tabel 2.4 yang menggambarkan mengenai jenis simbol Activity Diagram dibawah:

Tabel 2. 4 Jenis Simbol Activity Diagram[16]

Simbol	Keterangan
	Action: Dalam diagram aktivitas, "action" (aksi) adalah langkah dasar yang merepresentasikan tindakan atau kegiatan dalam suatu sistem atau proses.
	Activity: Activity adalah serangkaian tindakan yang membentuk suatu alur kerja atau proses dalam sistem.
	Control Flow: Control flow adalah representasi visual dari urutan eksekusi aktivitas dalam suatu proses atau sistem.
	Object Flow: Object flow adalah representasi visual dari aliran objek atau data antar aktivitas dalam suatu sistem atau proses.
	Initial Node: Initial node (simpul awal) dalam menunjukkan titik awal eksekusi alur kerja atau proses.
	Final Activity Node: Final activity node (simpul aktivitas akhir) menandakan titik akhir dari alur kerja atau proses.
	Final Flow Node:

Simbol	Keterangan
	Final Flow Node berfungsi untuk menghentikan semua jalur kontrol atau objek spesifik.
	Decision Node: Decision node (simpul keputusan) dalam adalah elemen yang digunakan untuk menunjukkan titik di mana alur kerja bercabang berdasarkan suatu kondisi atau keputusan.
	Merge Node: Merge node (simpul penggabungan) adalah simbol berbentuk titik yang menunjukkan tempat di mana jalur-jalur yang bercabang setelah decision node atau percabangan akan berkumpul kembali menjadi satu alur kerja tunggal.
	Swimlane: Swimlane (jalur renang) membagi diagram menjadi area terpisah yang mewakili entitas atau aktor yang berbeda dalam suatu proses.

2.4 Teori tentang Tools yang digunakan

2.4.1 XAMPP

XAMPP adalah sebuah paket perangkat lunak pengembangan web yang memfasilitasi pembuatan dan pengelolaan server web lokal di lingkungan pengembangan. Nama "XAMPP" berasal dari singkatan komponennya, yaitu "X" yang merujuk pada sistem operasi apapun (Windows, Linux, Mac OS X), "Apache" sebagai server web, "MySQL" sebagai sistem manajemen basis data, "PHP" sebagai bahasa pemrograman server-side, dan "Perl" sebagai bahasa pemrograman tambahan. XAMPP juga sering mencakup komponen tambahan seperti phpMyAdmin (untuk manajemen basis data MySQL melalui antarmuka web) dan beberapa alat lainnya. Dengan menggunakan XAMPP, pengembang dapat membuat dan menguji aplikasi web secara lokal sebelum mempublikasikannya ke server produksi.

XAMPP sangat berguna dalam pengembangan dan pengujian proyek-proyek web secara lokal sebelum diterapkan ke lingkungan produksi yang sesungguhnya.

2.4.2 Visual Studio Code

Visual Studio Code atau VS Code, merupakan software editor teks dan lingkungan pengembangan terintegrasi (IDE) yang dikembangkan oleh Microsoft. Dengan antarmuka pengguna yang mudah dipahami, VS Code mendukung berbagai bahasa pemrograman populer seperti Python, JavaScript, Java, dan C++. Pengguna dapat menyesuaikan editor ini dengan berbagai ekstensi untuk menambahkan fitur sesuai kebutuhan. Fitur pengeditan yang kuat, termasuk highlight sintaks, autocomplete, refactoring, dan debugging, memudahkan penulisan kode yang efisien. Integrasi bawaan dengan Git memungkinkan pengguna mengelola versi kode mereka langsung dari editor. Popularitas VS Code terus meningkat karena fleksibilitasnya dan dukungan komunitas yang luas.

