



BAB 2 LANDASAN TEORI

2.1 Penelitian Terdahulu

Dalam sebuah sistem informasi pengelolaan proposal kegiatan mahasiswa yang dibuat dengan baik dan terstruktur dapat membuat pihak-pihak yang mengoprasikannya menjadi lebih mudah dan terbantu. Karena dasar penelitian ini dibuat adalah untuk mempermudah pihak yang bersangkutan. Oleh karena itu beberapa penelitian sudah dilakukan untuk membangun sebuah sistem informasi mengenai pengelolaan proposal kegiatan mahasiswa. Berikut ini adalah penelitian terdahulu yang serupa dengan permasalahan yang ada.

Pada penelitian yang dilakukan oleh (Muslihudin & Oktafianto, 2020) dengan judul “Rancang Bangun Sistem Informasi Manajemen Proposal Kemahasiswaan.” Latar belakang penelitian ini fokus pada bidang kemahasiswaan di universitas yang memerlukan pengelolaan terstruktur untuk mendokumentasikan dan mengelola proposal kegiatan mahasiswa yang cukup banyak. Penelitian ini bertujuan mengembangkan sistem informasi untuk manajemen proposal kegiatan mahasiswa, sehingga mempermudah proses perencanaan, pengajuan, persetujuan, evaluasi, dan pelaporan kegiatan. Sistem ini dikembangkan menggunakan metode *prototyping* dan dirancang dengan *Unified Modeling Language* (UML) untuk memastikan kejelasan spesifikasi. Hasil dari penelitian ini adalah sebuah sistem informasi yang mampu mendukung proses pendataan kegiatan, pengajuan proposal, persetujuan oleh dosen pembimbing, evaluasi proposal, hingga pembuatan laporan akhir secara efektif. (Muslihudin & Oktafianto, 2020).

Selanjutnya penelitian yang dilakukan oleh (Zubaedah et al., 2022) dengan judul “Sistem Informasi



Pendataan Dan Monitoring Proposal Unit Kegiatan Mahasiswa (Ukm) Di Unmus.” Dilatarbelakangi oleh kebutuhan Universitas Musamus Merauke memiliki 18 Unit Kegiatan Mahasiswa (UKM) yang membutuhkan sistem pengelolaan proposal kegiatan yang terstandar agar proses pengajuan berjalan dengan baik. Penelitian ini bertujuan mengembangkan sistem informasi pendataan dan *monitoring* proposal kegiatan UKM untuk mendukung Minat, Bakat, dan Penalaran (MBP) dalam pengelolaan proposal. Metode penelitian survei digunakan untuk pengumpulan data lapangan dan kuisioner guna mengetahui tanggapan pengguna terhadap sistem. Hasil dari penelitian ini adalah sistem informasi yang mempermudah pendataan dan *monitoring* proposal kegiatan UKM di UNMUS, sehingga proses pengelolaan menjadi lebih efektif dan efisien. (Zubaedah et al., 2022).

Pada penelitian terdahulu yang dilakukan oleh (Famy, 2022) dengan judul “Rancang Bangun Sistem Informasi Manajemen Berbasis Web Menggunakan Nextjs Di Cv. Sanjaya Abadi Baru.” Latar belakang persaingan bisnis yang semakin ketat menuntut perusahaan untuk memanfaatkan teknologi agar tidak tertinggal dari pesaing. Penelitian ini bertujuan menerapkan sistem informasi manajemen pada CV. Sanjaya Abadi Baru di Batam untuk meningkatkan efisiensi dan efektivitas dalam menyimpan, mengelola, dan mendapatkan informasi, seperti data pemesanan dan inventori, guna mendukung pengambilan keputusan manajemen dan memberikan informasi yang cepat dan akurat kepada pelanggan. Penelitian menggunakan metode *Agile Scrum*, yang memungkinkan pengembangan aplikasi secara cepat dengan menyertakan desain diagram, desain rinci, perencanaan implementasi, pembuatan kode, dan pengujian sistem. Hasil penelitian menunjukkan bahwa sistem informasi manajemen yang diterapkan terbukti lebih efektif dan efisien dibandingkan sistem sebelumnya, membantu proses bisnis



perusahaan serta memberikan informasi yang lebih cepat, tepat, dan akurat bagi pelanggan. (Famy, 2022).

Selain itu, penelitian yang dilakukan oleh (Arizal et al., 2022) dengan judul “Metode Prototype Pada Sistem Informasi Manajemen Tugas Akhir Mahasiswa Berbasis Website” merujuk pada permasalahan tugas akhir sebagai sebuah penelitian merupakan salah satu syarat utama kelulusan mahasiswa sarjana di perguruan tinggi. Bimbingan dari dosen pembimbing menjadi salah satu cara untuk mencari solusi atas permasalahan penelitian mahasiswa. Proses bimbingan tugas akhir melibatkan beberapa pihak, yaitu administrasi kampus, dosen pembimbing, dan mahasiswa. Namun, proses ini sering kali sulit dilakukan dengan baik karena kesibukan masing-masing individu, baik dari dosen maupun mahasiswa, sehingga menghambat penyelesaian tugas akhir. Selain itu, prosedur layanan administrasi yang memakan waktu juga menjadi faktor yang memperlambat penyelesaian tugas akhir. Penelitian ini bertujuan untuk menyederhanakan dan mempercepat proses penyelesaian tugas akhir mahasiswa. Metode *prototype* digunakan untuk mengembangkan sistem informasi manajemen tugas akhir. Sistem informasi ini dirancang menggunakan diagram *use case*, *rich picture*, dan *entity relation diagram*. Pengujian dilakukan dengan metode *black box*. Hasil penelitian ini berupa sistem informasi manajemen tugas akhir yang membuat proses penyelesaian tugas akhir mahasiswa menjadi lebih efisien dan efektif. Sistem ini mempermudah layanan administrasi dan monitoring tugas akhir karena dilakukan secara sistematis dan dapat dikontrol secara real time. (Arizal et al., 2022).

Penelitian yang dilakukan oleh (Siregar et al., 2018) dengan judul “Pengembangan Aplikasi Pendaftaran Online Layanan Pencatatan Sipil Berbasis Web Menggunakan PHP dan Basis Data MySQL (Studi Kasus: Dispendukcapil Kabupaten Malang)” merujuk kepada masyarakat yang menghadapi kesulitan dalam mengakses formulir pendaftaran



karena persediaan yang sering habis di tingkat desa, sehingga mereka harus bolak-balik antara kantor Dispenduk dan kepala desa. Selain itu, tulisan tangan pada formulir sering kali sulit dibaca oleh petugas, menyebabkan ketidakakuratan data. Untuk mengatasi masalah ini, pihak Dispenduk menginginkan sistem yang dapat menggantikan proses pengajuan manual. Penelitian ini bertujuan untuk mengembangkan sistem informasi pendaftaran online berbasis web, khususnya untuk layanan penerbitan akta perkawinan, guna meningkatkan efektivitas dan efisiensi pelayanan. Sistem ini dikembangkan menggunakan metode waterfall dengan implementasi berbasis PHP serta memanfaatkan library seperti *Bootstrap*, *Javascript*, dan *Jquery*. Analisis dan perancangannya dilakukan secara terstruktur, dengan pengujian validasi dan kompatibilitas untuk memastikan fungsi sistem berjalan dengan baik. Hasil penelitian ini adalah sistem informasi pendaftaran penerbitan dokumen pencatatan sipil berbasis web yang diharapkan dapat mempermudah masyarakat dalam mengajukan penerbitan dokumen secara lebih efisien dan efektif. (Siregar et al., 2018).



Tabel 2. 1 Penelitian terdahulu

No	Peneliti	Judul Penelitian	Metode	Platform	DBMS	Hasil
1	Fajri Nugraha, Mulia Nuzulima, Arif Hidayat, Haris Hidayat. (2020)	Rancang Bangun Sistem Informasi Manajemen Proposal Kemahasiswaan	Prototype	Website	MySQL	Penelitian ini menghasilkan sistem untuk mengelola pengajuan kegiatan kemahasiswaan, meliputi pendataan, pendaftaran proposal, persetujuan dosen, evaluasi, dan pembuatan laporan.
2	Rezki Zuhairi, Zubaidah, Tatiana M. Tallia, Jombang, Jardi, Budianto, Sulaiman. (2020)	Sistem Informasi Pendataan Dan <i>Monitoring</i> Proposal Unit Kegiatan Mahasiswa (Ukm) Di Unmus	Flowchart	Website	MySQL	Sistem ini dibangun untuk memfasilitasi pendataan serta pemantauan proposal kegiatan yang diajukan oleh 18 UKM yang ada di UNMUS.
3	Steven Famy (2020)	Rancang Bangun Sistem Informasi	Agile Scrum	Website	MySQL	Sistem ini membantu perusahaan dalam



No	Peneliti	Judul Penelitian	Metode	Platform	DBMS	Hasil
		Manajemen Berbasis Web Menggunakan Nextjs Di Cv. Sanjaya Abadi Baru				mengelola data pemesanan dan inventori untuk mendukung keputusan manajemen dan memberikan informasi yang cepat serta akurat kepada pelanggan. Hasilnya, sistem baru terbukti lebih efisien dan efektif.
4	Arizka Annisa Nur Puteri, Furqan Zakaria, Dimas Febrian Priatomo. (2020)	Metode <i>Prototype</i> Pada Sistem Informasi Manajemen Tugas Akhir Mahasiswa Berbasis <i>Website</i>	Prototype	Website	Tidak menyebutkan	Sistem informasi manajemen tugas akhir yang dikembangkan meningkatkan efisiensi dan efektivitas penyelesaian tugas akhir, mempermudah administrasi, dan dapat monitoring secara sistematis dan real-time, dibandingkan prosedur lama yang lambat dan terhambat kesibukan pihak terkait.

No	Peneliti	Judul Penelitian	Metode	Platform	DBMS	Hasil
5	Janet Siregar, Isma Akmada, Djo Pradono	Pengembangan Aplikasi Pendaftaran Online Layanan Pencatatan Sipil Berbasis Web Menggunakan PHP dan Basis Data <i>MySQL</i> (Studi Kasus Dispendukcapil Kabupaten Malang)	Waterfall	Website	MySQL	Sistem pendaftaran online berbasis web yang diimplementasikan dengan PHP dan menggunakan library seperti Bootstrap, Javascript, dan JQuery, memudahkan masyarakat dalam mengajukan permohonan dokumen, serta mempercepat penerbitan dokumen kependudukan.

Penelitian ini berfokus pada pengelolaan proposal kegiatan mahasiswa di Unipdu Jombang, yang tidak hanya menyederhanakan prosedur pengajuan proposal, tetapi juga meningkatkan komunikasi antara mahasiswa dan BPKm. Dengan sistem yang dikembangkan, mahasiswa dapat dengan mudah mengajukan proposal, memantau statusnya, dan menerima *feedback* dari BPKm secara lebih cepat dan jelas. mengurangi miskomunikasi atau keterlambatan yang sering terjadi dalam proses manual.



2.2 Kajian Pustaka

2.2.1 BPKm

BPKm (Biro Pelayanan Kemahasiswaan) di Unipdu Jombang bertugas mengelola berbagai aktivitas mahasiswa, seperti membantu pelaksanaan kegiatan Ormawa (Organisasi Mahasiswa) dan UKM (Unit Kegiatan Mahasiswa), serta mengurus proses beasiswa, mulai dari seleksi hingga pengumuman penerima. BPKm juga memproses proposal kegiatan mahasiswa, memberikan persetujuan, dan mendukung pengembangan minat dan bakat mereka. Selain itu, BPKm memastikan semua kegiatan sesuai dengan kebijakan kampus. BPKm bertugas untuk memantau dan mengevaluasi kegiatan untuk memastikan semuanya berjalan lancar, serta menyediakan informasi dan konsultasi mengenai program beasiswa dan kegiatan yang membantu pengembangan mahasiswa.

2.2.2 Proposal Kegiatan Mahasiswa

Menurut (Sasongko, 2018) Proposal kegiatan adalah usulan rencana yang disusun oleh individu atau lembaga sebagai permohonan kepada pihak lain untuk mendapatkan dukungan dan dana.

2.2.3 Sistem

(Wijoyo et al., 2021) dalam bukunya yang berjudul “Sistem Informasi Manajemen” menjelaskan bahwa sistem dapat didefinisikan sebagai kumpulan elemen atau prosedur yang saling berinteraksi untuk mencapai tujuan tertentu melalui hubungan yang terstruktur. Sistem terdiri dari input, yang diproses untuk menghasilkan *output*. Menurut Ludwig Von Bertalaney, sistem adalah seperangkat unsur yang saling terikat dengan lingkungannya, sementara Anatol Rapoport dan Ackoff menekankan kesatuan elemen yang saling bergantung satu sama lain. Jadi sistem adalah kumpulan elemen atau prosedur yang saling berinteraksi dan bekerja sama, di mana

input diproses untuk menghasilkan output guna mencapai tujuan tertentu.

2.2.4 Sistem Informasi Manajemen

Menurut (Rusdiana, 2014) dalam bukunya yang berjudul “Sistem Informasi Manajemen” Sistem Informasi Manajemen (SIM) adalah sebuah sistem yang dirancang untuk mendukung proses pengambilan keputusan, koordinasi, pengendalian, analisis, dan visualisasi dalam organisasi. SIM mencakup struktur dan proses, di mana struktur mengacu pada elemen-elemen penyusun sistem seperti perangkat keras, perangkat lunak, manusia, dan prosedur kerja, sedangkan proses mencakup cara elemen-elemen tersebut berinteraksi untuk mencapai tujuan organisasi. Sistem ini terdiri dari berbagai subsistem seperti sistem pemrosesan transaksi, sistem pendukung keputusan, dan sistem informasi eksekutif yang saling berintegrasi dan merupakan bagian dari sistem organisasi yang lebih besar. SIM berfungsi untuk menangani kegiatan rutin secara efisien, menghasilkan informasi yang relevan, tepat waktu, dan akurat bagi penggunaannya. Informasi yang dihasilkan merupakan data yang telah diproses dan diolah sehingga memiliki nilai guna.

Dalam SIM, manajemen berperan untuk mengoordinasikan aktivitas organisasi melalui perencanaan, pengorganisasian, pengarahan, dan pengendalian sumber daya secara efektif dan efisien, dengan tujuan meningkatkan produktivitas dan kinerja organisasi. Dengan menggunakan SIM, organisasi dapat meningkatkan efisiensi operasional, mendukung pengambilan keputusan strategis, dan memperbaiki komunikasi antarbagian dalam organisasi.

2.2.5 Website

(Sidik, 2019) *Website* adalah sekumpulan halaman digital yang berisi informasi dalam berbagai bentuk, seperti teks, gambar, audio, video, atau animasi, yang dapat diakses melalui internet dari mana saja. *Website* terdiri dari tiga





komponen utama, yaitu kata-kata (teks), gambar, dan kode. Dalam perkembangannya, ketiga komponen tersebut diklasifikasikan menjadi tiga aspek utama: struktur (*structure*), tampilan (*presentation*), dan perilaku (*behavior*).

A. Struktur

Struktur mengatur tata letak dan organisasi informasi di dalam *website*, sehingga memudahkan pengguna memahami konten yang disajikan. Hal ini melibatkan elemen-elemen HTML (*Hypertext Markup Language*), seperti *heading*, paragraf, daftar, dan tabel, yang membantu menyusun hierarki konten dengan cara yang terstruktur dan mudah diakses oleh mesin pencari serta pengguna.

B. Tampilan

Tampilan mencakup aspek visual dari sebuah *website* yang dirancang untuk menarik perhatian pengguna. Elemen desain seperti warna, font, tata letak, dan gambar diatur menggunakan CSS (*Cascading Style Sheets*). Selain estetika, tampilan juga berkontribusi pada pengalaman pengguna (*user experience/UX*) dengan memastikan aksesibilitas dan kompatibilitas antarperangkat, seperti desktop, tablet, dan ponsel.

C. Perilaku

Perilaku mengacu pada interaksi antara pengguna dan elemen-elemen *website*. Aspek ini melibatkan penggunaan JavaScript atau teknologi pemrograman lainnya untuk menciptakan pengalaman yang dinamis dan interaktif, seperti validasi formulir, animasi, navigasi *dropdown*, atau respons terhadap klik pengguna.

Adapun 3 jenis *website* berdasarkan tujuan dan fungsinya, pertama adalah *Website Statis* adalah menyajikan informasi



yang jarang berubah, seperti profil perusahaan/organisasi, atau portofolio. Yang kedua adalah *Website* Dinamis yaitu menggunakan basis data untuk menampilkan konten yang terbaru, contohnya adalah *website* berita atau *e-commerce*. Yang terakhir adalah *Website* Interaktif yaitu *website* yang memungkinkan pengguna (*user*) untuk berkontribusi atau berinteraksi langsung, seperti forum atau media sosial. Dalam kasus ini SIM Proposal Kegiatan Mahasiswa merupakan jenis *website* dinamis, dikarenakan melibatkan *database*.

2.2.6 Database

(Andaru, 2018) menjelaskan bahwa database, juga dikenal sebagai "basis data", adalah kumpulan informasi yang disimpan secara sistematis di dalam komputer dan dapat diperiksa melalui program komputer tertentu untuk memperoleh informasi darinya. Secara umum, database merupakan sistem penyimpanan data yang terorganisir, yang memungkinkan penyimpanan informasi dalam bentuk tabel, yang terdiri dari baris dan kolom. Setiap kolom mewakili atribut dari data, sementara setiap baris mewakili satu entitas atau record. Data ini dapat berupa teks, angka, gambar, atau bentuk lainnya yang diperlukan dalam aplikasi tertentu. Database dapat digolongkan menjadi beberapa jenis berdasarkan cara penyimpanan dan pengelolaannya, yaitu:

A. Database Relasional

Data disimpan dalam tabel-tabel yang saling terhubung melalui relasi antar tabel. Contoh DBMS yang mendukung *database* relasional adalah MySQL, PostgreSQL, dan Microsoft SQL Server.

B. Database Non-Relasional (NoSQL)

Data disimpan dalam format yang lebih fleksibel, seperti dokumen, grafik, atau key-value pairs. Contoh DBMS NoSQL adalah MongoDB dan Cassandra.

C. Database Terdistribusi

Data disebar ke beberapa *server* atau lokasi geografis, tetapi dipandang sebagai satu kesatuan oleh



pengguna. Misalnya, Google Spanner dan Apache Cassandra.

2.2.6.1 Sistem Manajemen Basis Data (DBMS)

Sistem Manajemen Basis Data (DBMS) adalah perangkat lunak yang digunakan untuk mengelola *database*. DBMS menyediakan berbagai fungsi untuk menyimpan, mengelola, dan mengambil data, yang memungkinkan pengguna dan aplikasi untuk melakukan operasi seperti memasukkan, memperbarui, menghapus, dan mengambil data dari database. Beberapa fungsi utama DBMS meliputi: Penyimpanan data, keamanan data, pengelolaan transaksi, *backup* atau pemulihan data.

2.2.6.2 MySQL sebagai DBMS

Dalam penelitian ini, penulis menggunakan MySQL sebagai DBMS untuk menyimpan dan mengelola informasi. MySQL adalah salah satu DBMS relasional yang paling populer dan banyak digunakan di berbagai aplikasi, mulai dari aplikasi web hingga sistem besar. Salah satu kelebihan MySQL ini adalah *open source*, dan juga skalabilitas MySQL dapat menangani data kecil hingga besar.

2.2.6.3 Pengelolaan data dalam Database

Pengelolaan data dalam *database* dilakukan melalui *query*, yaitu perintah yang digunakan untuk berinteraksi dengan data yang tersimpan. Beberapa operasi dasar dalam SQL:

- A. SELECT digunakan untuk mengambil data dari satu atau lebih tabel yang tersedia
- B. INSERT digunakan untuk menambahkan data baru ke dalam tabel
- C. UPDATE digunakan untuk memperbarui data yang sudah ada

- D. DELETE digunakan untuk menghapus data dari tabel
- E. JOIN digunakan untuk menggabungkan data dari dua atau lebih tabel berdasarkan relasi.

2.2.7 Visual Studio Code

Visual Studio Code, atau sering disebut VS Code, adalah editor kode sumber yang dikembangkan oleh Microsoft, dirancang untuk membantu pengembang menulis, mengedit, dan mengelola kode sumber berbagai proyek pemrograman, seperti aplikasi web, desktop, *mobile*, hingga berbasis *cloud*. VS Code mendukung berbagai bahasa pemrograman dengan fitur seperti *syntax highlighting*, indentasi otomatis, dan IntelliSense yang memberikan rekomendasi kode serta dokumentasi. Editor ini memiliki debugging bawaan, terminal terintegrasi, dan mendukung pengelolaan kontrol versi seperti Git. VS Code juga menawarkan kemampuan kustomisasi tema dan pengaturan, menjadikannya alat yang fleksibel dan efisien. Sebagai editor ringan, gratis, dan *open source*, dengan dukungan komunitas global yang besar, VS Code menjadi pilihan utama bagi pengembang dari berbagai level keahlian.

2.2.8 Framework

Menurut (Nilfaidah et al., 2021) *Framework* adalah kerangka kerja yang terdiri dari kumpulan file dan fungsi yang sudah terstruktur dan terintegrasi untuk membantu *programmer* dalam menangani berbagai masalah pemrograman. *Framework* menyediakan kode program bawaan dan fungsi-fungsi yang mempermudah proses pengembangan aplikasi, seperti manajemen koneksi ke database, routing, dan autentikasi. Selain itu, *framework* membantu meningkatkan efisiensi dengan menyediakan kerangka dasar yang memungkinkan *programmer* untuk fokus pada pengembangan logika bisnis tanpa harus menulis kode dari awal. Contoh populer dari *framework* meliputi CodeIgniter, Laravel, dan





Django, yang sering digunakan dalam pengembangan aplikasi berbasis web. Dengan menggunakan *framework*, pengembang dapat memastikan standar yang konsisten, keamanan yang lebih baik, dan waktu pengembangan yang lebih cepat karena tersedianya dokumentasi lengkap.

2.2.9 CodeIgniter

Menurut (Nilfaidah et al., 2021) CodeIgniter adalah sebuah *framework* yang dikembangkan dengan bahasa pemrograman PHP, dirancang untuk memudahkan *programmer* web dalam menciptakan atau mengembangkan aplikasi berbasis web. *Framework* ini menyediakan berbagai fitur lengkap dalam satu kesatuan, seperti dukungan untuk arsitektur *Model-View-Controller* (MVC), validasi data, manajemen sesi, dan koneksi ke database, yang bertujuan meningkatkan efisiensi serta konsistensi dalam pengembangan aplikasi web. CodeIgniter dikenal karena performanya yang cepat, dokumentasi yang jelas, dan ukuran yang ringan, sehingga menjadi pilihan yang ideal bagi pengembang yang membutuhkan *framework* sederhana tetapi fleksibel. Dalam penelitian ini, penulis menggunakan CodeIgniter versi 4, yang menawarkan pembaruan signifikan dibandingkan versi sebelumnya, termasuk dukungan penuh untuk PHP 7 dan 8, penggunaan *namespace*, sistem *routing* yang lebih fleksibel, serta peningkatan keamanan. Versi terbaru ini dirancang untuk memberikan pengalaman pengembangan yang lebih modern dan efisien.

2.2.10 Prototype

Menurut (Purnomo, 2017) *prototyping* merupakan suatu metode pengembangan perangkat lunak yang berfungsi sebagai versi awal suatu sistem berupa model fisik cara kerja dan fungsionalitasnya. Metode ini bertujuan untuk membangun prototipe, yaitu representasi sederhana dari sistem yang akan dikembangkan, yang berfungsi sebagai alat komunikasi antara pengembang dan pengguna. Prototipe memungkinkan



pengembang untuk memahami kebutuhan pengguna secara lebih mendalam melalui interaksi langsung dan umpan balik selama proses pengembangan. Pendekatan *prototyping* dilakukan dalam beberapa tahapan, yaitu:

- A. Pengumpulan kebutuhan awal, di mana kebutuhan dasar pengguna dicatat
- B. Pembuatan prototipe, di mana kerangka kerja dasar sistem dibangun
- C. Evaluasi prototipe oleh pengguna, yang memberikan masukan untuk penyempurnaan
- D. Penyempurnaan prototipe berdasarkan masukan hingga mencapai persetujuan pengguna. Prototipe yang telah disetujui kemudian dijadikan acuan untuk pengembangan sistem akhir.

Metode *prototyping* memiliki beberapa keunggulan, seperti meminimalkan risiko kesalahan desain karena masukan pengguna dapat segera diterapkan, mempercepat pengembangan karena fokus pada elemen utama sistem, dan meningkatkan kepuasan pengguna karena mereka terlibat langsung dalam proses. Namun, metode ini juga memiliki kelemahan, seperti risiko pengguna memiliki ekspektasi yang terlalu tinggi terhadap prototipe awal atau potensi pengembang terlalu fokus pada tampilan daripada fungsi utama sistem. Prototyping sangat cocok digunakan untuk proyek yang memiliki ketidakpastian tinggi atau melibatkan kebutuhan yang kompleks dan dinamis, sehingga memberikan fleksibilitas dalam menangani perubahan. Dengan demikian, pendekatan ini menjadi salah satu metode efektif dalam memastikan keberhasilan pengembangan sistem informasi.

2.2.11 Unified Modeling Language (UML)

UML (*Unified Modeling Language*) adalah sebuah alat atau metode dalam perancangan sistem yang membantu pengembang aplikasi untuk merancang sistem yang akan dibangun, dengan pendekatan berbasis objek (Famy, 2022).



UML memberikan visualisasi grafis dari struktur dan perilaku sebuah sistem, yang mencakup berbagai diagram seperti diagram kelas, diagram *use case*, dan diagram aktivitas. Dengan pendekatan orientasi objek, UML memodelkan komponen sistem sebagai objek-objek yang berinteraksi satu sama lain, sehingga memudahkan pengembang dalam memahami, mendokumentasikan, dan mengkomunikasikan desain sistem dengan cara yang terstruktur dan jelas.

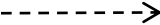
A. Class Diagram

Dalam model desain sistem, diagram kelas menunjukkan hubungan antar kelas dan memberikan penjelasan detail tentang setiap kelas, termasuk atribut, metode, dan hubungannya satu sama lain. Seperti yang ditampilkan pada Tabel 2.2 Penjelasan *Class Diagram* menjelaskan simbol-simbol dari *class diagram*.

Tabel 2. 2 Penjelasan Class Diagram

Gambar		Keterangan
nama_kelas +atribut +operasi()		Class pada struktur sistem
nama_interface		<i>Interface</i> menggunakan konsep yang serupa dengan <i>interface</i> dalam pemrograman berorientasi objek.
—		<i>Association</i> digunakan untuk menggambarkan hubungan antara satu kelas dengan kelas lainnya dalam sebuah diagram.
→		<i>Directed association</i> digunakan untuk menjelaskan



Gambar	Keterangan
	hubungan antar kelas yang dimanfaatkan oleh kelas lain.
	<i>Generalization</i> relasi antar kelas dengan arti generalisasi – spesialisasi atau umum - khusus
	<i>Dependency</i> digunakan untuk menjelaskan hubungan antara kelas yang bergantung pada kelas lain.
	<i>Aggregation</i> relasi antar kelas yang berhubungan dengan kelas lainnya

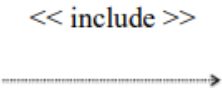
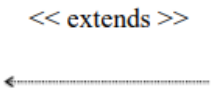
B. Use Case Diagram

Use case diagram merupakan kelakuan (*behavior*) sistem informasi yang akan dibangun. Ini digunakan untuk mengetahui fungsi apa yang ada di dalam sistem dan siapa yang berhak menggunakan fungsi tersebut. Simbol-simbolnya dapat dilihat pada Tabel 2.3 Penjelasan *Use Case Diagram*.

Tabel 2. 3 Penjelasan Use Case Diagram

Gambar	Keterangan
	<i>Use case</i> , menjelaskan fungsi dari sistem dalam bentuk unit-unit dan berhubungan dengan aktor.
	<i>Actor</i> adalah peran orang atau sistem yang lain ketika berkomunikasi dengan use case.

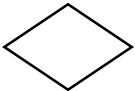


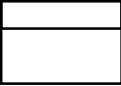
Gambar	Keterangan
	<i>Association</i> adalah penghubung antara aktor dan use case
	<i>Include</i> merupakan bagian dari use case lain yang bertujuan untuk pemanggilan sebuah fungsi dalam program.
	<i>Extend</i> adalah pengembangan dari use case lainnya yang terjadi berfungsi untuk fitur tambahan.

C. Activity Diagram

Dalam sebuah sistem atau proses bisnis, aktivitas atau aliran kerja digambarkan dalam diagram aktivitas. Diagram aktivitas menunjukkan bagaimana satu aktivitas berpindah ke aktivitas lainnya dan bagaimana keputusan atau kondisi tertentu mempengaruhi jalannya proses. Lebih jelasnya pada Tabel 2.4 Penjelasan Activity Diagram.

Tabel 2. 4 Penjelasan Activity Diagram

Gambar	Keterangan
	<i>Start point</i> , merupakan awal dari aktivitas.
	<i>Activity</i> menjelaskan sebuah proses atau aktivitas bisnis.
	<i>Decision</i> , menggambarkan percabangan yang terjadi

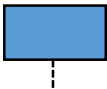
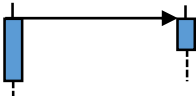
Gambar	Keterangan
	karena adanya aktivitas yang lebih dari satu.
	Penggabungan dimana beberapa aktivitas digabungkan menjadi satu.
	<i>Final</i> , merupakan tanda sebagai titik berakhirnya suatu aktivitas.
	<i>Swimlane</i> menjelaskan pembagian tanggung jawab dalam organisasi selama suatu aktivitas.

D. *Sequence* Diagram

Sequence diagram, juga dikenal sebagai diagram urutan, digunakan untuk menggambarkan gerakan objek dalam *use case*. Diagram ini menunjukkan bagaimana komponen sistem berinteraksi satu sama lain dengan mengirimkan pesan dalam urutan tertentu. Diagram ini menunjukkan siklus hidup objek, mulai dari bagaimana objek dibuat, bagaimana mereka berinteraksi, hingga kapan mereka dapat dihancurkan. *Sequence* Diagram sangat membantu dalam menentukan tugas dan tanggung jawab masing-masing objek. Simbol-simbolnya dapat dilihat pada Tabel 2.5. Penjelasan *Sequence* Diagram.



**Tabel 2. 5 Penjelasan Sequence Diagram**

Gambar	Keterangan
	<i>Actor</i> , menggambarkan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem
	<i>Life Line</i> , menjelaskan objek apa yang sedang berinteraksi.
	<i>Object Message</i> , menggambarkan interaksi antar objek yang menunjukkan urutan peristiwa yang terjadi.
	<i>Message to Self</i> , menggambarkan interaksi objek itu sendiri yang menunjukkan urutan peristiwa yang terjadi.