



BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

Landasan Teori Menjelaskan teori yang berkaitan dengan judul pada laporan akhir ini.

2.1.1 Aplikasi

Menurut Zalukhu et al., (2023) “Aplikasi adalah suatu program yang siap untuk digunakan yang di buat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta pengguna aplikasi lain yang di gunakan oleh suatu sasaran yang akan di tuju, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang di inginkan atau di harapkan maupun pemrosesan data yang di harapkan”.

Menurut Voutama & Novalia, (2021) “Aplikasi adalah program yang digunakan orang untuk melakukan sesuatu pada sistem komputer. Dan sistem aplikasi *mobile* merupakan aplikasi yang dapat digunakan walaupun pengguna berpindah dengan mudah dari satu tempat ke tempat lain- lain tanpa terjadi pemutusan atau terputusya komunikasi”.

Berdasarkan kedua pengertian tersebut dapat diartikan bahwa aplikasi merupakan suatu program yang dirancang untuk membantu pengguna dalam menyelesaikan tugas atau memecahkan masalah tertentu dengan memanfaatkan teknik pemrosesan data. Aplikasi dapat dijalankan pada berbagai perangkat, termasuk komputer maupun perangkat mobile, serta memungkinkan pengguna untuk tetap terhubung dan mengakses fungsionalitas aplikasi dengan mudah, bahkan saat berpindah lokasi. Dengan kata lain, aplikasi berfungsi sebagai alat bantu digital yang memberikan kemudahan, efisiensi, dan fleksibilitas dalam mendukung aktivitas pengguna.

2.1.2 Absensi

Menurut (Karaman et al., 2024) “Absensi adalah suatu pendataan kehadiran seseorang dalam aktivitas tertentu yang ada dalam sebuah institusi”. Menurut (Faizin, 2024) “Absensi adalah sekumpulan data kehadiran yang merupakan bagian dari proses

pelaporan dalam suatu instansi atau organisasi.”. Dari pendapat diatas penulis bisa mengartikan bahwa absensi adalah daftar para anggota sebuah institusi atau perusahaan yang mencatat kehadiran atau ketidakhadiran para anggotanya.

2.1.3 Website

Menurut (Qifli Ilhamdi & Julkarnain, 2024:4) "*website* adalah sekumpulan halaman yang berisi informasi berbentuk digital. Informasi tersebut bisa berupa text, gambar, audio, video, animasi, atau gabungan dari semuanya.”. Sedangkan (Qifli Ilhamdi & Julkarnain, 2024:4) "*Website* adalah sebuah halaman atau komponen-komponen berisi informasi-informasi yang dapat diakses melalui internet dan dapat dinikmati oleh seluruh dunia (secara global)". Berdasarkan beberapa definisi diatas, maka dapat disimpulkan bahwa *website* adalah media digital yang terdiri dari kumpulan halaman saling terkait yang berisi berbagai jenis informasi, seperti teks, gambar, suara, dan animasi.

Berdasarkan kedua pengertian tersebut, dapat diartikan bahwa *website* adalah media digital yang terdiri dari kumpulan halaman yang saling terhubung dan menyajikan berbagai jenis informasi seperti teks, gambar, audio, video, maupun animasi. Website dapat diakses secara global melalui jaringan internet dan berfungsi sebagai sarana penyampaian informasi yang efektif dan luas jangkauannya.

2.1.4 Metode pengembangan system *Waterfall*

Metode pengembangan sistem yang penulis gunakan dalam penulisan laporan disempurnakan akhir ini adalah metode *waterfall*. Menurut Supiyandi et al., (2022) “Model *waterfall* adalah model yang paling banyak digunakan untuk tahap pengembangan. Model *waterfall* ini juga dikenal dengan nama model tradisional atau model klasik. Model air terjun (*waterfall*) sering juga disebut model sekuensial linier (*Sequential linear*) atau alur hidup klasik (*Classic cycle*)”. Model air terjun ini menyediakan pendekatan alur hidup perangkat lunak secara sekuensial terurut dimulai dari analisis, desain, pengkodean, pengujian dan tahap pendukung (*support*). Adapun Pengertian lain yang menyatakan bahwa metode *waterfall* merupakan pendekatan *Software Development Life Cycle* (SDLC) paling awal yang digunakan untuk pengembangan perangkat lunak. Metode ini dilakukan dengan pendekatan yang

sistematis, mulai dari tahap kebutuhan sistem lalu menuju ke tahap analisis, desain, coding, testing/verification, dan maintenance (Nugroho et al., 2023).

Berdasarkan kedua pengertian tersebut, dapat diartikan bahwa metode Waterfall adalah model pengembangan perangkat lunak yang bersifat sistematis dan terstruktur, di mana setiap tahap dikerjakan secara berurutan mulai dari analisis kebutuhan hingga tahap pemeliharaan. Model ini memberikan alur kerja yang jelas dan terorganisir, sehingga cocok digunakan dalam pengembangan sistem yang memiliki kebutuhan dan spesifikasi yang telah ditentukan sejak awal.

2.1.5 *Unified Modeling Language (UML)*

(Widyastuti & al., 2022), UML (*Unified Modeling Language*) adalah bahasa pemodelan yang digunakan untuk mengembangkan sistem perangkat lunak yang kompleks dan besar. UML memiliki beberapa komponen, seperti *use case* diagram, *class* diagram, *Sequence* diagram, dan *state machine* diagram, yang dapat membantu meningkatkan kualitas sistem yang dihasilkan dan memudahkan pengembangan sistem yang kompleks. Menurut Limantoro dan Kristiadi (2021:8), “UML merupakan kepanjangan dari *Unified Modeling Language* yang memiliki arti bahasa pemodelan standar”.

Dari kedua pengertian diatas dapat diartikan bahwa UML adalah bahasa pemodelan yang memiliki beberapa komponen yang digunakan untuk mengembangkan sistem dalam perangkat lunak.

2.1.6 *Use Case Diagram*

(Suliyanti & Putri, 2023) “*Use case* diagram berfungsi untuk melakukan pemodelan kelakuan (behaviour) aplikasi yang akan dirancang”. Sedangkan menurut (Limantoro & Kristiadi, 2021) “Diagram *Use Case* bersifat statis. Diagram ini memperlihatkan himpunan *use case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dtuhkan serta diharapkan pengguna” *Use case* diagram adalah alat pemodelan yang digunakan untuk menggambarkan interaksi antara pengguna dan sistem melalui serangkaian fungsi yang ditawarkan.

Berdasarkan kedua pengertian tersebut, dapat diartikan bahwa *use case*

diagram adalah alat pemodelan yang bersifat statis dan digunakan untuk menggambarkan interaksi antara aktor (pengguna) dan sistem melalui serangkaian fungsi atau layanan yang disediakan. Diagram ini berfungsi untuk memodelkan perilaku sistem yang diharapkan oleh pengguna, serta membantu dalam memahami kebutuhan sistem secara menyeluruh pada tahap perancangan awal.

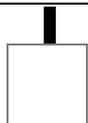
(Suliyanti & Putri, 2023) juga menjelaskan bahwa *use case* diagram memiliki beberapa komponen, seperti:

- 1) *Aktor*: pengguna sistem yang berinteraksi dengan sistem
- 2) *Use case*: sistem yang diwakili oleh aktor
- 3) *Asosiasi*: hubungan antara aktor dan use case
- 4) *Ekstensi*: ekstensi yang dapat ditambahkan ke *use case*

Tabel 2. 1 Simbol-simbol Use Case Diagram

No	Gambar	Nama	Keterangan
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang ketika pengguna mainkan berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>).
3		<i>Generalization</i>	Akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.

Lanjutan Tabel 2.1 Simbol-simbol *Use case* Diagram

No	Gambar	Nama	Keterangan
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Sumber: (Muhammad et al., 2021)

Dengan menggunakan *use case* diagram, pengembang sistem dapat memodelkan sistem perangkat lunak dengan lebih efektif dan efisien, sehingga dapat meningkatkan kualitas sistem yang dihasilkan dan memudahkan pengembangan sistem yang kompleks.

2.1.7 Class Diagram

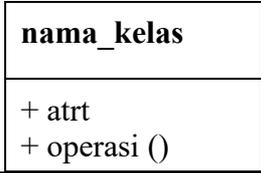
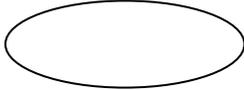
(Suliyanti & Putri, 2023) *Class* Diagram merupakan salah satu diagram UML yang berfungsi untuk menampilkan kelas kelas dan juga relasi antar kelas yang ada pada perancangan front end aplikasi kearsipan berbasis web (*e-arsip*)”.

Sedangkan (Wulandari & Nurmiati, 2022)“Diagram *Class* ialah mendeskripsikan struktur sistem asal segi pendefinisian kelas-kelas yang akan dirancang untuk menciptakan sistem”.

Dari kedua pengertian tersebut dapat diartikan bahwa *class* diagram adalah salah satu jenis diagram UML yang digunakan untuk menggambarkan struktur sistem

dengan menampilkan kelas-kelas serta relasi antar kelas yang akan dirancang.

Tabel 2. 2 Simbol-simbol Class Diagram

No.	Simbol	Deskripsi
1	Kelas 	Kelas pada struktur sistem
2	Antarmuka/ <i>interface</i>  nama_interface	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3	Agregasi/ <i>aggregation</i> 	Relasi antar kelas dengan makna semuabagian (<i>whole-part</i>)
4	Asosiasi/ <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
5	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6	Kebergantungan/ <i>dependency</i> >	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)

Sumber: (Muhammad et al., 2021)

(Wicaksono & Santoso, 2022) *Class Diagram* memberikan representasi visual dari desain perangkat lunak, yang memudahkan analisis dan komunikasi antar anggota tim pengembang. Setiap kelas dalam *Class Diagram* biasanya digambarkan sebagai kotak yang terbagi menjadi tiga bagian: bagian atas berisi nama kelas, bagian tengah berisi atribut, dan bagian bawah berisi metode atau fungsi yang dapat dilakukan oleh kelas tersebut. Hubungan antar kelas, seperti asosiasi (garis), agregasi (garis dengan ujung berlian kosong), komposisi (garis dengan ujung berlian penuh), dan pewarisan (panah bersegitiga) digambarkan dengan simbol-simbol khusus yang menunjukkan sifat hubungan tersebut.

2.1.8 Activity Diagram

(Suliyanti & Putri, 2023) “*Activity* diagram adalah Gambaran proses dari aktivitas yang terjadi di dalam suatu aplikasi yang akan dirancang”. Sedangkan (Limantoro & Kristiadi, 2021) “Diagram Aktivitas (*Activity* Diagram), bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem”.

Dari kedua pengertian tersebut dapat diartikan bahwa *Activity* diagram adalah diagram yang digunakan untuk menggambarkan alur proses atau aktivitas yang terjadi dalam suatu aplikasi atau sistem.

Tabel 2. 3 Simbol-simbol *Activity* Diagram

No	Gambar	Nama	Keterangan
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antar muka saling berinteraksi satu sama lain.
2		<i>Action</i>	<i>State</i> dari sistem yang mencerminkan eksekusi dari suatu aksi.
3		<i>Initial Node</i>	Bagaimana Objek dibentuk atau diawali
4		<i>Activity Final Node</i>	Bagaimana objek di bentuk dan di hancurkan.
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.

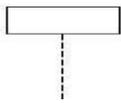
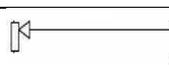
Sumber : (Muhammad et al., 2021)

2.1.9 Pengertian *Sequence* Diagram

Sequence Diagram adalah salah satu jenis diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk memvisualisasikan urutan interaksi antara objek dalam suatu sistem berdasarkan waktu. Diagram ini menggambarkan bagaimana objek-objek berkomunikasi melalui pertukaran pesan untuk menjalankan suatu fungsi atau proses tertentu dalam sistem. Setiap objek direpresentasikan oleh garis vertikal yang disebut *lifeline*, sedangkan pesan yang dipertukarkan diilustrasikan dengan panah

horizontal. (Lawecon, 2024) Dapat diartikan bahwa *Sequence diagram* adalah diagram yang digunakan untuk menggambarkan alur proses atau aktivitas yang terjadi dalam suatu aplikasi atau sistem.

Tabel 2. 4 Simbol-simbol Sequence Diagram

No	Gambar	Nama	Keterangan
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

Sumber : (Muhammad et al., 2021)

2.2 *State Of The Art*

Beberapa penelitian sebelumnya telah dilakukan terkait sistem absensi berbasis web. Penelitian oleh (Nugraha & Astutik, 2022) dari Universitas Muhammadiyah Sidoarjo mengembangkan sebuah sistem informasi absensi karyawan berbasis web. Tujuan dari sistem ini adalah untuk menggantikan sistem absensi manual yang selama ini rentan terhadap kesalahan dan manipulasi data, sehingga perusahaan dapat memperoleh data kehadiran yang lebih akurat dan efisien. Selanjutnya, penelitian oleh (Ismail et al., 2022) di Universiti Teknologi Malaysia merancang sistem absensi kelas universitas menggunakan teknologi pengenalan wajah berbasis deep learning. Sistem ini dirancang untuk meningkatkan efisiensi dan akurasi dibandingkan metode absensi tradisional seperti pencatatan manual atau penggunaan kartu identitas mahasiswa. Penggunaan teknologi ini memungkinkan verifikasi kehadiran dilakukan secara otomatis dan real-time.

Berdasarkan kedua penelitian tersebut, dapat disimpulkan bahwa baik dalam konteks perusahaan maupun institusi pendidikan, sistem absensi berbasis web

menawarkan solusi yang lebih efektif dibandingkan metode manual. Penelitian pertama menekankan pentingnya akurasi dan efisiensi dalam pencatatan kehadiran karyawan, dengan menggantikan sistem manual yang rawan kesalahan dan manipulasi. Sementara itu, penelitian kedua menunjukkan bagaimana pemanfaatan teknologi canggih seperti deep learning dan pengenalan wajah mampu mengotomatiskan proses absensi mahasiswa, sehingga menghasilkan sistem yang lebih cepat, akurat, dan minim intervensi manusia. Kedua penelitian ini sama-sama mengarah pada satu kesimpulan: penerapan teknologi berbasis web dalam sistem absensi mampu meningkatkan keandalan dan efisiensi pencatatan kehadiran dibandingkan dengan metode konvensional yang selama ini digunakan.

Di sisi lain, terdapat pula tiga penelitian lain yang menekankan pentingnya penyesuaian sistem absensi dengan kebutuhan lingkungan masing-masing. Penelitian oleh (Mansor & al., 2022) dari Universiti Utara Malaysia mengembangkan sistem pencatatan absensi dan suhu tubuh siswa sebagai bagian dari penerapan Virtual Learning Environment (VLE) selama masa pandemi. Sistem ini memungkinkan sekolah untuk memantau kehadiran sekaligus kondisi kesehatan siswa secara real-time. Sementara itu, penelitian (Hendrian, 2022) di SDIT Ibnu Khaldun Panambangan Cirebon merancang sistem informasi absensi siswa berbasis web yang ditujukan agar orang tua dapat mengakses informasi kehadiran anak mereka secara langsung dan cepat. Terakhir, (Kurnia Aji & al., 2023) dari Universitas Bhayangkara Surabaya mengembangkan sistem absensi karyawan berbasis web dengan integrasi webcam dan pelacakan lokasi untuk memverifikasi kehadiran secara lebih objektif.

Ketiga penelitian tersebut menunjukkan adanya kecenderungan pemanfaatan sistem absensi berbasis web yang lebih adaptif terhadap kebutuhan spesifik. Penelitian dari Malaysia menekankan pentingnya aspek kesehatan di sekolah selama pandemi, sedangkan penelitian Sukma Hendrian menyoroti keterlibatan orang tua dalam pemantauan kehadiran siswa. Inovasi dari Bagus Ramdana dan timnya juga membuktikan bahwa integrasi webcam dan pelacakan lokasi bisa menjadi solusi validasi kehadiran yang lebih akurat dan tidak mudah dimanipulasi. Kesamaan dari ketiganya adalah tujuan untuk menciptakan sistem yang transparan, efisien, dan sesuai dengan tantangan di lapangan.

Dengan memperhatikan berbagai penelitian sebelumnya, dapat disimpulkan



bahwa pengembangan sistem absensi berbasis web terus mengalami peningkatan baik dari sisi fitur, efisiensi, maupun adaptasi terhadap kebutuhan pengguna. Oleh karena itu, sistem yang dikembangkan dalam penelitian ini mengacu pada kelebihan-kelebihan tersebut, khususnya dalam hal efisiensi pencatatan kehadiran, transparansi data, serta kemudahan akses oleh pengguna. Diharapkan sistem ini mampu menjadi solusi absensi digital yang tidak hanya relevan dengan kebutuhan saat ini, tetapi juga fleksibel untuk pengembangan di masa depan.