

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Penelitian Terkait**

Penelitian yang dilakukan oleh Darmansah dkk (2021) dalam jurnal penelitiannya yang berjudul Perancangan Absensi Berbasis *Face Recognition* Pada Desa Sokaraja Lor Menggunakan Platform Android. penelitian ini bertujuan untuk merancang sebuah sistem absensi berbasis *face recognition* menggunakan platform Android. Sistem ini dirancang untuk memudahkan proses absensi di Kantor Desa Sokaraja Lor, yang saat ini masih menggunakan metode absensi konvensional dengan pinjer print dan pencatatan manual. Penelitian ini menggunakan metode pengumpulan data primer dan sekunder, serta analisis masalah dan analisis sistem untuk merancang sistem absensi berbasis *face recognition*. Sistem ini melibatkan pengguna (pegawai desa) dan admin, dengan tahapan input, proses, dan output yang terkait dengan verifikasi kehadiran pegawai menggunakan teknologi *face recognition*[1].

Penelitian yang dilakukan oleh Fikri Haikal dan Rama Adistya Nurcahya Pambudji (2023) dalam jurnal penelitiannya yang berjudul Sistem Akses Masuk Ruang Server Yayasan Al Muslim Menggunakan Facerecognition ESP32 Dan Notifikasi Telegram Dengan Metode Prototyping. Penelitian ini membahas tentang pengembangan sistem keamanan pintu ruang server pada Yayasan Al Muslim menggunakan

teknologi *face recognition* dengan menggunakan ESP32 dan Wemos D1 sebagai mikrokontroler utama. Sistem ini juga dilengkapi dengan notifikasi Telegram dan website untuk mengubah akses masuk dan mendaftar. Penelitian ini menghasilkan sistem yang dapat memberikan keamanan pada pintu dengan menggunakan sensor wajah yang dapat bekerja sesuai dengan fungsi dan tujuan dari sistem[2]. Penelitian yang dilakukan oleh Made Ary Wiradhi Putra (2022) dalam jurnal penelitiannya yang berjudul Sistem Deteksi Wajah Untuk Absensi Dan Akses Masuk Ruangan. Penelitian ini bertujuan untuk Mengembangkan sistem deteksi wajah menggunakan metode klasifikasi Haar yang ditemukan oleh Viola-Jones. Menggunakan ESP32-CAM dengan modul kamera OV2640 2MP untuk sistem deteksi wajah. Menguji kemampuan sistem untuk mendeteksi objek wajah manusia pada jarak dan posisi yang berbeda. Mengevaluasi nilai akurasi, presisi, dan recall dari sistem. Menganalisis hasil dan menarik kesimpulan[3].

Penelitian yang dilakukan oleh Nur Jaini (2021) dalam jurnal penelitiannya yang berjudul Sistem Manajemen Kehadiran Menggunakan Metode *Face Recognition* Berbasis Web. Penelitian ini bertujuan untuk membahas tentang sistem absensi karyawan berbasis *face recognition* menggunakan ESP32-CAM IOT. Jurnal ini membahas tentang cara kerja sistem absensi tersebut, serta teknologi-teknologi yang digunakan dalam pembuatan sistem tersebut. Jurnal ini juga membahas tentang hasil pengujian sistem dan kesimpulan dari pengujian tersebut[4].

Penelitian yang dilakukan oleh Ryan Hidayat dkk (2022) dalam jurnal penelitiannya yang berjudul Rancang Bangun Alat Absensi Karyawan menggunakan RFID dan ESP32Cam Berbasis *Internet of Things*. Penelitian ini membahas tentang rancang bangun alat absensi karyawan menggunakan RFID dan ESP32CAM berbasis *Internet of Things*. Alat ini dirancang untuk menggantikan sistem sidik jari agar meminimalisir sentuhan fisik dan lebih murah dibandingkan sistem pengenalan wajah. Data absensi akan dideteksi melalui ID Card karyawan yang telah dimasukkan ke dalam database. Alat ini menggunakan NodeMCU ESP8266 sebagai pengolah data dari RFID dan ESP32CAM untuk memotret wajah saat melakukan absensi. Selain itu, artikel ini juga membahas penggunaan LCD, I2C, buzzer, dan web server dalam alat absensi ini[5].

## **2.2 Landasan Teori**

### **2.2.1. Web site**

Website adalah fasilitas *hypertech* untuk menampilkan berupa teks, gambar, suara, animasi, dan multimedia lainnya. Website dapat kita gunakan menjadi alat promosi suatu produk maupun jasa. Kehadiran website saat ini tentu dengan melewati perkembangan yang begitu pesatnya. Website telah menjelma menjadi profil dan ‘muka’ seseorang di internet[6].

### **2.2.2. Sistem Informasi**

Sebuah sistem informasi merupakan kumpulan elemen-elemen yang saling terhubung dan berkolaborasi secara sinergis dalam rangka

mengakuisisi, memproses, menyimpan, serta menyebarluaskan informasi. Tujuan utamanya adalah untuk memfasilitasi proses pengambilan keputusan, koordinasi, pengendalian, analisis, dan visualisasi dalam lingkup sebuah organisasi[7].

### **2.2.3. Mysql**

MySQL Merupakan sebuah database server yang free, artinya kita bebas menggunakan database ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya. MySQL pertama kali dirintis oleh seorang programmer database bernama Michael Widenius . Selain database server, MySQL juga merupakan program yang dapat mengakses suatu database. Jadi MySQL adalah sebuah database yang dapat digunakan sebagai Client maupun server. Database MySQL merupakan suatu perangkat lunak database yang berbentuk database relasional atau disebut Relational Database Management System ( RDBMS ) yang menggunakan suatu bahasa permintaan yang bernama SQL (Structured Query Language ) [8].



Gambar 2. 1 Mysql

### **2.2.4. Xampp**

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (localhost), yang terdiri atas

program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public License dan bebas, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis[9].



Gambar 2. 2 Xampp

#### 2.2.5. CSS

CSS adalah singkatan dari Cascading Style Sheets, yaitu bahasa pemrograman yang digunakan untuk menentukan tampilan dan format elemen-elemen pada sebuah halaman web yang ditulis dalam bahasa markup seperti HTML. CSS membantu mengatur tata letak, warna, font, dan gaya lainnya, sehingga halaman web menjadi lebih terstruktur dan menarik secara visual. Dengan CSS, pengembang dapat membuat tampilan konsisten dan seragam di seluruh situs web, serta memisahkan konten dari presentasi, yang memudahkan pemeliharaan dan pembaruan desain web[10].



Gambar 2. 3 CSS

### 2.2.6. Visual Studio Code

Visual Studio Code adalah Software yang sangat ringan, namun kuat editor kode sumbernya yang berjalan dari desktop. Muncul dengan built-in dukungan untuk JavaScript, naskah dan Node.js dan memiliki array beragam ekstensi yang tersedia untuk bahasa lain, termasuk C ++, C # , Python, dan PHP. Editor ini adalah fitur lengkap lingkungan pengembangan terpadu (IDE) dirancang untuk pengembang yang bekerja dengan teknologi cloud yang terbuka Microsoft karena didasarkan pada gaya explorer umum, dengan panel di sebelah kiri, yang menunjukkan semua file dan folder Anda memiliki akses ke panel editor di sebelah kanan, dan menyenangkan pada mata. Ia juga memiliki fungsi yang baik, dengan intellisense dan autocomplete bekerja dengan baik untuk JSON, CSS, HTML, {kurang}, dan Node.js[11].



Gambar 2. 4 Visual studio code

### 2.2.7. Javascript

Javascript adalah bahasa yang berbentuk kumpulan skrip yang pada fungsinya berjalan pada suatu dokumen HTML, sepanjang sejarah internet bahasa ini adalah bahasa skrip pertama untuk web. Bahasa ini adalah bahasa pemrograman untuk memberikan kemampuan tambahan terhadap bahasa HTML dengan mengizinkan pengeksekusian perintah perintah di sisi user, yang artinya di sisi browser bukan di sisi server web[12].



Gambar 2. 5 Javascript

### 2.2.8. Php

PHP menurut Anhar (2010 : 3) adalah bahasa pemrograman web server-side yang bersifat open source, PHP juga merupakan script yang terintegrasi dengan HTML dan berada pada server ( server side HTML embedded script). PHP juga merupakan script yang digunakan untuk membuat halaman website yang sangat dinamis, dinamis berarti

halaman tampilan yang akan ditampilkan dibuat saat halaman itu diminta oleh client[13].



Gambar 2. 6 PHP

### 2.2.9. Metode Waterfall

Metode waterfall adalah metode yang paling banyak digunakan untuk tahap pengembangan. Metode Waterfall merupakan pendekatan SDLC paling awal yang digunakan untuk pengembangan perangkat lunak. 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).

### 2.2.10. HTML

HTML, singkatan dari *Hypertext Markup Language*, adalah salah satu bahasa pengkodean atau pemrograman yang digunakan untuk membuat halaman website yang ditampilkan pada web browser. Sebagian besar halaman yang kamu temukan di internet menggunakan HTML. HTML diciptakan oleh Sir Tim Berners-Lee pada akhir tahun 1991, meskipun tidak dirilis secara resmi saat itu. Hingga saat ini,





HTML telah mencapai versi 5 yang dirilis pada tahun 2012, HTML memiliki beberapa kelebihan, seperti memiliki banyak sumber dan komunitas yang besar, bersifat open source [14].

Gambar 2. 7 HTML

### 2.2.11. UML ( Unified Modeling Language)

UML (*Unified Modeling Language*) adalah “Salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisa & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”.UML (*Unified Modeling Language*) adalah “Sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem” [15].

Dari beberapa penjelasan teori tersebut dapat disimpulkan bahwa UML (*Unified Modeling Language*) adalah bahasa yang sering digunakan untuk membangun sebuah sistem perangkat lunak dengan melakukan penganalisaan desain dan spesifikasi dalam pemrograman berorientasi objek.

UML (*Unified Modeling Language*) memiliki diagram-diagram yang digunakan dalam pembuatan aplikasi berorientasi objek, diantaranya:


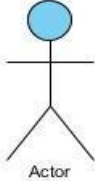

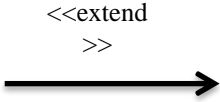
#### a. Use Case Diagram


Use Case Diagram merupakan pemodelan untuk melakukan (behavior) sistem informasi yang akan dibuat. Use case digunakan

untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Berikut adalah simbol-simbol yang ada pada diagram use case:

Tabel 2. 1 *Use case*

Simbol	Deskripsi
<p data-bbox="427 824 544 853"><i>Use case</i></p> 	<p data-bbox="837 824 1377 1003">Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal <i>frase</i> nama <i>Use Case</i></p>
<p data-bbox="427 1043 600 1072"><i>Aktor / actor</i></p> 	<p data-bbox="837 1043 1377 1361">Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informaasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal <i>frase</i> nama aktor</p>
<p data-bbox="427 1417 703 1447"><i>Asosiasi / assosiation</i></p> 	<p data-bbox="837 1417 1377 1518">Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor</p>
<p data-bbox="427 1574 651 1603"><i>Ekstensi / extend</i></p> 	<p data-bbox="837 1574 1377 1886">Relasi <i>use case</i> tambahan kesebuah <i>use case</i> dinamakan <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>intherince</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.</p>






<p>Generalisasi / <i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>
---	---

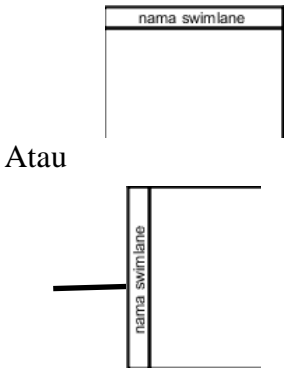
b. *Activity Diagram*

*Activity Diagram* menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Perlu diperhatikan bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Berikut adalah simbol-simbol yang ada pada diagram aktivitas:

Tabel 2. 2 *Activity Diagram*

<b>Simbol</b>	<b>Deskripsi</b>
<p>Status awal</p> 	<p>Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal</p>
<p>Aktivitas</p> 	<p>Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja</p>
<p>Percabangan / <i>decision</i></p> 	<p>Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu</p>
<p>Penggabungan / <i>join</i></p> 	<p>Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu</p>
<p>Status akhir</p> 	<p>Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir</p>

<p><i>Swimlane</i></p>  <p>Atau</p>	<p>Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi</p>
--	--

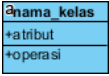
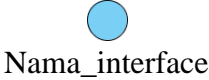

c. *Class Diagram*





Diagram kelas berfungsi untuk memvisualisasikan kerangka kerja sistem dengan mendefinisikan kelas-kelas yang akan dikonstruksi dalam pembangunan sistem. Setiap kelas terdiri dari karakteristik yang disebut atribut dan kemampuan fungsional yang disebut metode atau operasi.

Dalam diagram kelas, terdapat beberapa simbol yang digunakan untuk merepresentasikan komponen-komponen tersebut

:

Tabel 2. 3 *Class Diagram*

<b>Simbol</b>	<b>Deskripsi</b>
<p>Kelas</p> 	<p>Kelas dalam struktur sistem mirip dengan konsep antarmuka (interface) dalam pemrograman berorientasi objek.</p>
<p>Antarmuka / <i>Interface</i></p> 	<p>Hubungan antar kelas dengan makna umum, biasanya disertai dengan multiplisitas yang menunjukkan jumlah atau banyaknya objek yang terlibat dalam relasi tersebut.</p>
<p>Asosiasi / <i>Association</i></p> 	<p>Relasi antarkelas dengan makna satu kelas digunakan oleh kelas lainnya, umumnya juga disertai dengan <i>multiplisitas</i>.</p>

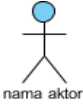
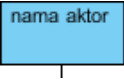


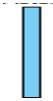

Asosiasi berarah / <i>Directed Association</i> 	relasi antara dua kelas dalam diagram kelas yang menunjukkan adanya navigasi atau koneksi satu arah dari satu kelas ke kelas lainnya
Generalisasi 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus) menggambarkan hubungan hierarki di mana satu kelas merupakan turunan atau spesialisasi dari kelas lain yang lebih umum.
Kebergantungan / <i>Dependency</i> 	Relasi antarkelas dengan makna ketergantungan menunjukkan bahwa satu kelas bergantung pada kelas lainnya untuk menjalankan fungsinya.
Agregasi / <i>Aggregation</i> 	Relasi antarkelas dengan makna keseluruhan-bagian (whole-part) menggambarkan hubungan komposisi di mana satu kelas terdiri dari beberapa kelas lain sebagai bagiannya.

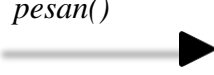

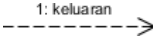
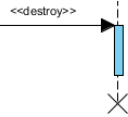
#### d. *Sequence Diagram*

Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Untuk menggambarkan diagram sekuen maka harus diketahui objek- objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Berikut adalah simbol-simbol yang ada pada diagram sekuen:

Tabel 2. 4 Squence diagram

Simbol	Deskripsi
<p>Aktor</p>  <p>Atau</p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi itu sendiri, jadi walaupun simbol dari aktor adalah orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal <i>frase</i> nama aktor</p>
<p>Garis hidup / <i>Lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek</p>
<p>Objek</p> 	<p>Menytakan objek yang berinteraksi pesan</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.</p>
<p>Pesan tipe <i>create</i></p>  <p>&lt;&lt;<i>create</i>&gt;&gt;</p>	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yaang dibuat</p>

<p>Pesan tipe <i>call</i></p> 	<p>Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri. Arah panah mengarah pada objek yang memiliki operasi / metode, karena ini memanggil operasi / metode maka operasi / metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi</p>
<p>Pesan tipe <i>send</i></p> 	<p>Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p>
<p>Pesan tipe <i>return</i></p> 	<p>Menyatakan suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian</p>
<p>Pesan tipe <i>destroy</i></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada <i>destroy</i></p>