

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian yang dilakukan Sri Mulyati pada tahun 2022 tentang sistem *fingerprint* dapat digunakan untuk absensi dengan *system gateways* juga digunakan sebagai penggerak pintu pagar menggunakan sensor *Fingerprint* untuk Identifikasi sidik jari, dikenal dengan daktiloskopi adalah ilmu yang mempelajari sidik jari untuk keperluan pengenalan kembali identitas orang dengan cara mengamati garis yang terdapat pada guratan garis jari tangan dan telapak kaki. Kita kenal menjadi ilmu sidik jari, permasalahan dalam penelitian ini meliputi perancangan sistem absensi[5].

Penelitian yang dilakukan Nova Mayasari dkk pada tahun 2022 Absensi *Fingerprint* merupakan Alat yang digunakan untuk melihat kehadiran pegawai secara komputersasi Aplikasi absensi *Fingerprint* pada *smartphone* dengan sistem operasi Android yang telah memiliki *fingerprint* pada perangkatnya dapat membantu setiap orang dalam menjalankan *social distancing* dan juga penerapan protokol kesehatan dari pemerintah. Karena dengan melakukan absensi *fingerprint* pada Android ini setiap orang dapat menghindari dari kontak fisik dan menyentuh sesuatu yang sama secara bergantian. Dengan sistem absensi yang baik maka diharapkan dapat membantu dalam mengendalikan proses penyelesaian pekerjaan sehingga didapatkan hasil yang maksimal dan sesuai dengan tujuan yang ditetapkan[6].

Penelitian yang dilakukan Asti Anugerah Rahimi dkk pada tahun 2019. Absensi (*Fingerprint*) merupakan mesin yang digunakan untuk mengetahui dan mencatat kehadiran dengan mengidentifikasi tekstur sidik jari sebagai media pengumpul data. Di berbagai instansi pemerintah maupun swasta, pegawai dari empat instansi atau perusahaan dapat memberikan data waktu dan kehadiran dengan menggunakan sidik jarinya sendiri, sehingga data waktu dan kehadiran tidak dapat dimanipulasi meskipun sidik jarinya berbeda. Berdasarkan pendapat di atas, kami menyimpulkan bahwa aplikasi absensi (sidik jari) bertujuan untuk mengetahui dan mencatat ketidakhadiran dan kehadiran pegawai atau perorangan, termasuk waktu kedatangan dan keberangkatannya, alasan ketidakhadiran atau keterangannya.[7].

Penelitian yang dilakukan Abdi Darmawan pada tahun 2016, Kehadiran tidak dapat dipisahkan dari kegiatan belajar mengajar, karena daftar hadir mempunyai banyak fungsi penting yang berkaitan dengan partisipasi siswa dalam proses belajar mengajar dan ujian. Saat ini, acara tatap muka sering dilakukan di atas kertas, dilanjutkan dengan guru memanggil siswa secara individu untuk setiap mata pelajaran. Data kehadiran siswa mungkin tidak valid, data mungkin hilang, dan mungkin memerlukan waktu lama untuk menampilkan informasi kehadiran siswa bila diperlukan. Pengawasan disiplin terhadap sekolah diperlukan karena sistem pendidikan akan berjalan lebih baik jika siswa dapat berpartisipasi dalam proses belajar mengajar. Pengawasan yang dilakukan belum tentu menjadi tanggung jawab sekolah. Kegagalan pengawasan baik pihak sekolah maupun orang tua adalah tidak adanya

mekanisme komunikasi antara sekolah dan orang tua untuk mengatur kehadiran siswa.[8].

Penelitian menurut Sari dan Yulianto pada tahun 2013 adalah alat yang digunakan sebagai pengenalan identitas, melindungi komputer dari data atau pemakaian komputer oleh orang yang tidak bertanggung jawab. Sedangkan Menurut (Suwandi, 2018) menyatakan bahwa *Software* yang menunjang untuk keperluan absensi, yang didalamnya mencakup pemasukan, 24 penyimpanan data jam masuk dan jam pulang, serta memproses data tersebut menjadi sebuah laporan yang nantinya dapat digunakan untuk pengambilan kebijakan-kebijakan yang dilakukan pimpinan[9].

Penelitian yang dilakukan oleh Axel Berkati, Licantik, Nahumi Nugrahaningsih, Ariesta Lestari, dan Felicia Sylviana pada 2024 yang berjudul “Rancang Bangun Aplikasi Absensi Guru Dan Staf TU dengan Penerapan Geolocation dan *Fingerprint* berbasis Android di SMK Gke Mandomai”. Pada android terdapat salah satu fitur yaitu *fingerprint*, dimana *fingerprint* ini yaitu teknologi geometrik untuk merekam pola sidik jari, yang kemudian disimpan untuk keperluan identifikasi. Dalam penggunaannya, sidik jari atau *fingerprint* digunakan untuk mencocokkan data sidik jari yang terekam dengan sidik jari aslinya. Dengan adanya *fingerprint* sebagai alat absensi maka data yang masuk tidak dapat dipalsukan, karena setiap orang memiliki sidik jari yang berbeda-beda, sehingga sangat tepat digunakan pada absensi[10].

2.2 Landasan Teori

2.2.1 Sistem Monitoring

Sistem monitoring merupakan suatu proses untuk mengumpulkan data dari berbagai sumber daya. Biasanya data yang dikumpulkan merupakan data yang realtime. Secara umum tujuan monitoring adalah untuk mendapatkan data-data atau pandangan agar diperoleh umpan balik bagi kebutuhan tertentu. Secara garis besar tahapan dalam sebuah sistem monitoring terbagi ke dalam tiga proses besar yaitu proses di dalam pengumpulan data monitoring, proses di dalam analisis data monitoring dan proses di dalam menampilkan data hasil monitoring[11].

2.2.2 Absensi

Absensi adalah suatu bentuk pendataan tentang ada atau tidaknya seseorang siswa atau pegawai yang menjadi bagian dari pelaporan suatu lembaga, tertata rapi, mudah dicari, dan tersedia bagi pihak-pihak yang berkepentingan kapan saja diperlukan kehadiran.[12].

2.2.3 *Fingerprint*

Fingerprint adalah alat deteksi sidik jari yang dibuat dengan *Digital Signal Processing* (DSP) untuk mengolah gambar, kalkulasi, dan kemampuan pencarian. Sensor *fingerprint* dihubungkan dengan microcontroller yang memberikan perintah atau signal ke sensor untuk mengambil gambar sidik jari, mendeteksi sidik jari, atau mencocokkan sidik jari. Pada sensor *fingerprint* terdapat memori flash yang berguna untuk menyimpan data sidik jari yang didaftarkan. Jumlah sidik jari yang

bisa disimpan tergantung seberapa besar kapasitas memori flash yang dipakai di sensor[13].

2.2.4 Framework CodeIgniter

CodeIgniter adalah sebuah *web application framework* yang digunakan untuk membangun aplikasi PHP dinamis yang dibangun menggunakan konsep *Model View Controller development pattern*. CodeIgniter menyediakan berbagai macam *library* yang dapat mempermudah dalam pengembangan dan termasuk *framework* tercepat dibandingkan dengan *framework* lainnya[14].



Gambar 2. 1 Logo CodeIgniter

(Sumber: <https://images.app.goo.gl/GJXsTkS9GHZGqYzQ9>)

2.2.5 PHP

Hypertext Preprocessor atau yang disingkat PHP, PHP merupakan script untuk pemrograman *script web server-side*, *script* yang membuat dokumen HTML secara *on the fly*, maksudnya dokumen HTML yang

dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML. PHP/FI merupakan nama awal dari PHP. PHP adalah *Personal Home Page*, FI adalah *Form Interface*. Dibuat pertama kali oleh Rasmus Lerdoff. PHP, awalnya merupakan program yang dikhususkan untuk menerima input melalui form yang ditampilkan dalam *browser web*. *Software* ini disebar dan dilisensikan sebagai perangkat lunak *Open Source*[15].



Gambar 2. 2 Logo php

(Sumber: <https://www.php.net/download-logos.php>)

2.2.6 MySQL

Structured Query Language yang sering dikenal dengan MySQL, MySQL merupakan perangkat lunak (software) gratis dibawah lisensi *GPL GNU* (General Public License). MySQL sebagai sebuah implementasi dari sistem manajemen basis data relasional *RDBMS* (Relational DataBase Management System) didistribusikan secara gratis dibawah lisensi *GPL* (General Public License). Dapat secara bebas menggunakan MySQL, namun dengan batasan tidak boleh dijadikan

produk turunan yang bersifat komersial. MySQL merupakan turunan konsep utama dalam basis data, yaitu SQL. SQL adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan input data, yang memungkinkan pengoperasian data dikerjakan dengan mudah[16].



Gambar 2. 3 Logo MySQL

(Sumber: <https://www.mysql.com/about/legal/logos.html>)

2.2.7 Visual Studio Code

Visual Studio Code merupakan sebuah teks editor ringan dan handal yang dibuat oleh *Microsoft* untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi Linux, Mac, dan Windows. Teks editor ini secara langsung mendukung bahasa pemrograman Javascript, Typescript, dan Node. Js, serta bahasa pemrograman lainnya dengan bantuan plugin yang dapat dipasang via *marketplace* Visual Studio Code seperti C++, C#, Python, Go, Java dan PHP[17].



Gambar 2. 4 Logo Visual Studio Code

(Sumber: https://www.researchgate.net/figure/Gambar-10-Logo-Visual-Studio-Code_fig1_369020845)

2.2.8 Database

Basis data adalah kumpulan file yang saling terkait dan berinteraksi.

Hubungan ini ditunjukkan dengan adanya kunci pada setiap file yang ada. Basis data mewakili kumpulan data yang digunakan dalam suatu perusahaan atau institusi. Pengolahan basis data adalah suatu metode file dalam suatu institusi yang memungkinkan file-file yang berisi informasi diorganisasikan, disortir, diambil dan disajikan dalam format laporan setiap saat sehingga dapat diolah dengan baik.

2.2.9 Java Script

Sekumpulan script yang fungsinya dijalankan pada dokumen *HTML*. *JavaScript* adalah bahasa skrip yang awalnya dikembangkan khusus untuk penggunaan di *web*. Bahasa ini digunakan sebagai bahasa pemrograman untuk memberikan fungsi tambahan pada halaman web menggunakan eksekusi perintah di sisi pengguna, yaitu di dalam browser, bukan di server web[19].



Gambar 2. 5 Logo Java Script

(Sumber: www.javilazaro.es/programacion/principios-javascript)

2.2.10 UML (Unified Modeling Language)

Unified Modelling Language atau yang disingkat UML, merupakan suatu metode dalam teknik RPL (Rekayasa Perangkat Lunak) yang berfungsi untuk menggambarkan cara kerja sistem, fungsi, alur, tujuan dan juga mekanisme kontrol sistem. *UML* dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan *UML* dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem.

Terdapat empat model *UML* yang paling sering digunakan untuk menggambarkan suatu desain sistem *Usecase diagram*, *Class diagram*, *Behavioral Statemachine diagram*, dan juga *Sequence diagram*. Teknik-teknik pemodelan *Unified Modeling Language* ini disebut juga dengan 4

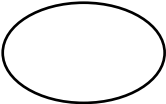



teknik dasar. Dalam proyek berorientasi objek, keempat teknik *UML* ini sangat mendominasi penggunaannya.

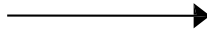
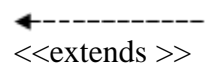

Dalam perancangan sistem terdapat *UML* yang sering digunakan sebagai berikut:

1. UseCase Diagram

Dalam pembuatan sistem informasi, *usecase* diagram merupakan pemodelan untuk behavior atau kelakuan sistem informasi. Diagram ini juga bersifat statis. Untuk simbol *UseCase* Diagram disajikan pada Tabel 2.1.

Tabel 2. 1 *UseCase Diagram*



No	Simbol	Keterangan
1.		Aksi-aksi yang ditampilkan system yang menghasilkan suatu hasil yang terukur bagi suatu actor.
2.		<i>Actor</i> entitas atau pengguna yang berinteraksi dengan sistem. Actor pada use case diagram digambarkan sebagai manusia atau objek, tergantung sistem yang dibuat.
3.		<i>Association</i> Menghubungkan antara objek sat uke objek lainnya.
4.	 <<include>>	<i>Include</i> Mengspesifikasikan <i>use case</i> sumber secara eksplisit.

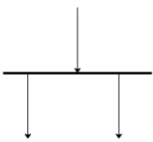

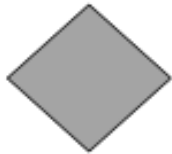
No	Simbol	Keterangan
5.		<i>Dependency</i> Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang tergantung padanya elemen yang tidak mandiri (independent).
6.		<i>Extend</i> Menspesifikasikan usecase target memperluas dari usecase sumber pada titik yang diberikan.
7.		<i>Sistem</i> Menspesifikasikan paket yang menampilkan secara terbatas

2. Activity Diagram

Activity Diagram atau Diagram Aktivitas merupakan diagram yang bersifat statis, yang menggambarkan aktivitas dari suatu sistem bisnis. Untuk simbol dari diagram aktivitas disajikan pada Tabel 2.2.

Tabel 2. 2 *Activity Digram*


No	Simbol	Keterangan
1.		<i>Start Point</i> merupakan awal dari suatu aktivitas yang peletakkanya pada pojok kiri atas.
2.		<i>End Point atau Final Node</i> merupakan gambaran akhir dari suatu aktivitas

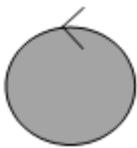
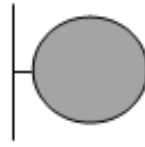

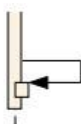


No	Simbol	Keterangan
3.		<i>Fork</i> digunakan untuk memarallelkan suatu kegiatan atau penggabungan 2 kegiatan paralel menjadi satu.
4.		<i>Activity</i> merupakan gambaran dari suatu proses
5.		<i>Decision</i> merupakan pilihan pengambilan suatu keputusan false or true.

3. Sequence Diagram

Sequence Diagram atau Diagram Urutan mendeskripsikan diagram interaksi yang mengirimkan pesan dan diterima antar objek. Untuk simbol diagram urutan disajikan pada Tabel 2.3

Tabel 2. 3 *Sequence Diagram*

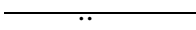
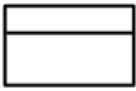
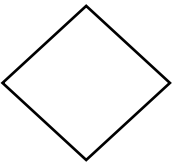




No	Simbol	Keterangan
1.		<i>Entity Class</i> merupakan bagian sistem yang membentuk deskripsi awal sistem dan berisi kumpulan kelas dalam bentuk entitas yang mendasari untuk membuat database

No	Simbol	Keterangan
2.		<i>ControlClass</i> merupakan gambaran penghubung antara <i>Boundaryclass</i> dengan suatu table
3.		<i>Boundary Class</i> merupakan gambaran dari penggambaran form.
4.		<i>message</i> menunjukkan pengiriman pesan antar class
5.		<i>Self message</i> menunjukkan pengiriman suatu pesan yang akan dikirim ke objek itu sendiri.
6.		<i>Activation</i> menggambarkan suatu objek yang melakukan sebuah aksi/eksekusi operasi
7.		<i>Lifeline</i> garis titik yang terhubung ke objek disepanjang garis lifeline memiliki aktivitas

4. Class Diagram

Diagram Class merupakan diagram yang bersifat statis, dalam diagram ini memperlihatkan himpunan kelas, antarmuka, serta relasi. Simbol dari *Class Diagram* disajikan pada Tabel 2.4.

Tabel 2. 4 *Class Diagram*

No	Simbol	Keterangan
1.		<i>Generallization</i> merupakan dimana objek descendent membagikan perilaku dan struktur data objek induknya.
2.		<i>Class</i> , adalah kumpulan objek yang saling berbagi.
3.		<i>Nary Association</i> , digunakan untuk asosiasi terhindar dengan objek lainya.
4.		<i>Collaboration</i> merupakan deskripsi urutan aksi yang ditampilkan suatu sistem yang memiliki konsekuensi terukur bagi aktor
5.		<i>Realization</i> Merupakan operasi yang valid dilakukan oleh suatu objek
6.		<i>Dependecy</i> Merupakan garis penah yang menunjukkan jika actor berinteraksi secara pasif
7.		<i>Association</i> merupakan garis yang menghubungkan objek satu dengan objek yang lain