

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian yang dilakukan oleh Edo Arribe dan Muhammad Ryandi (2023) dengan judul “Perancangan Sistem Informasi Absensi *Fingerprint* Berbasis *Website* PT. Media Andalan Nusa (Andalworks)” membahas pengembangan sistem absensi menggunakan sidik jari yang dihubungkan dengan *website*. Sistem tersebut memungkinkan data kehadiran pegawai tersimpan ke dalam *database* dan dapat diakses oleh pihak yang berwenang. Fitur laporan pada sistem tersebut membantu proses pemantauan kehadiran pegawai agar tidak lagi bergantung pada pencatatan manual. Penelitian tersebut memiliki kesamaan dengan penelitian ini karena sama-sama menggunakan *fingerprint* dan *website* sebagai media pencatatan dan pengelolaan data absensi. Perbedaannya, penelitian ini diterapkan pada lingkungan sekolah dan menambahkan fitur persetujuan siswa sebagai verifikasi kehadiran guru saat proses pembelajaran berlangsung[6].

Penelitian yang dilakukan oleh Desi Ardila dan Aninda Muliani Harahap (2023) dengan judul “Sistem Informasi Penggajian Terintegrasi dengan Fitur Absensi *Fingerprint* Berbasis *Website* pada PT. Sejahtera Mandiri Sawita II” mengembangkan sistem penggajian yang terhubung dengan data absensi *fingerprint*. Sistem tersebut menggunakan *website* untuk mengolah data karyawan, data kehadiran, lembur, BPJS, serta slip gaji. Hasil penelitian menunjukkan bahwa integrasi absensi dengan sistem informasi

dapat membantu proses pengolahan data menjadi lebih cepat dan akurat. Persamaan penelitian tersebut dengan penelitian ini terletak pada penggunaan *fingerprint* dan *website* sebagai bagian dari sistem informasi. Namun, penelitian ini tidak membahas penggajian, melainkan berfokus pada *monitoring* kehadiran guru mengajar serta validasi kehadiran melalui persetujuan siswa[7].

Penelitian yang dilakukan oleh Sri Yulianti, Nano Suyatna, dan An-Nisa Aina Nurwaida (2022) dengan judul “Rancang Bangun Sistem Informasi Kelola Absensi dengan Sidik Jari Berbasis Web di PT Kewalram Indonesia” membahas pengelolaan data kehadiran karyawan menggunakan sistem berbasis web dan *fingerprint*. Sistem tersebut dirancang untuk membantu bagian personalia dalam mencatat, menyimpan, dan mencari data kehadiran secara lebih mudah. Penelitian tersebut menunjukkan bahwa penggunaan sistem informasi berbasis web dapat membantu organisasi dalam mengurangi keterlambatan pengolahan data absensi. Penelitian ini memiliki hubungan dengan penelitian tersebut karena sama-sama mengembangkan sistem absensi berbasis *fingerprint*. Perbedaannya, sistem yang dirancang dalam penelitian ini digunakan untuk lingkungan sekolah, melibatkan empat pengguna utama, yaitu admin, guru, siswa, dan kepala sekolah, serta memiliki fitur persetujuan siswa sebagai penguat validasi kehadiran guru[8].

Berdasarkan beberapa penelitian terdahulu tersebut, dapat diketahui bahwa sistem absensi berbasis *fingerprint* dan *website* telah banyak digunakan untuk membantu pencatatan serta pengelolaan data kehadiran.

Namun, sebagian besar penelitian sebelumnya masih berfokus pada lingkungan perusahaan atau organisasi kerja. Penelitian ini memiliki fokus yang berbeda karena diterapkan pada proses *monitoring* kehadiran guru mengajar di sekolah. Selain itu, penelitian ini menambahkan fitur persetujuan siswa agar data kehadiran guru tidak hanya tercatat melalui *fingerprint*, tetapi juga dapat dikonfirmasi oleh siswa yang mengikuti kegiatan pembelajaran di kelas.

2.2 Landasan Teori

2.2.1 Sistem *Monitoring*

Sistem *monitoring* adalah sistem yang digunakan untuk memantau suatu kegiatan agar informasi mengenai kondisi atau proses yang sedang berjalan dapat diketahui oleh pihak yang membutuhkan. Dalam konteks teknologi informasi, *monitoring* tidak hanya berarti melihat data, tetapi juga membantu pengguna dalam membaca perkembangan, menemukan kendala, dan mengambil keputusan berdasarkan informasi yang tersedia. Pada penelitian ini, sistem *monitoring* digunakan untuk membantu sekolah melihat status kehadiran guru mengajar, baik guru yang hadir, belum hadir, izin, maupun tidak hadir. Dengan adanya sistem *monitoring*, kepala sekolah dan admin dapat memperoleh gambaran kondisi kehadiran guru secara lebih mudah tanpa harus menunggu laporan manual[9].

2.2.2 Sistem Absensi

Perangkat absensi merupakan sistem yang digunakan untuk mencatat kehadiran seseorang dalam suatu kegiatan atau pekerjaan. Dalam lingkungan sekolah, absensi guru menjadi salah satu data penting karena berkaitan dengan pelaksanaan jadwal mengajar dan kedisiplinan tenaga pendidik. Sistem absensi yang masih dilakukan secara manual memiliki beberapa kelemahan, seperti risiko kesalahan pencatatan, keterlambatan rekap data, dan kesulitan dalam melakukan pemantauan. Oleh karena itu, sistem absensi berbasis teknologi dapat digunakan untuk membantu pencatatan kehadiran agar data lebih mudah disimpan, diperiksa, dan digunakan sebagai bahan laporan[10].

2.2.3 *Fingerprint*

Fingerprint atau sidik jari adalah teknologi biometrik yang memanfaatkan pola unik pada jari manusia sebagai identitas seseorang. Setiap orang memiliki pola sidik jari yang berbeda, sehingga *fingerprint* dapat digunakan untuk mengenali pengguna dengan tingkat keakuratan yang baik. Dalam sistem absensi, *fingerprint* membantu memastikan bahwa proses kehadiran dilakukan oleh orang yang bersangkutan. Penggunaan *fingerprint* juga dapat mengurangi kemungkinan titip absen karena data yang tercatat berasal dari pemindaian sidik jari pengguna. Pada penelitian ini, *fingerprint* digunakan sebagai alat pencatatan awal kehadiran guru saat mengajar di kelas[11].

2.2.4 *Visual Studio Code*

Aplikasi pengedit kode sumber terbuka yang dikembangkan oleh Microsoft untuk *platform Windows, Linux, dan MacOS* [26]. *Visual Studio Code* facilitates the creation of code encompassing several programming languages, such as *C++, C#, Java, Python, PHP, and GO*. Visual Code dapat mengidentifikasi jenis bahasa pemrograman yang digunakan dan menawarkan berbagai variasi sesuai dengan kemampuan eksekusi kode [27]. *Visual Studio Code* has also been integrated into *GitHub*. Selain itu, satu komponen tambahan adalah kemampuan untuk menyertakan augmentasi, di 13 mana desainer dapat menambahkan ekstensi yang tidak tersedia dalam *Visual Studio Code*.

2.2.5 *Website*

Website adalah kumpulan halaman yang dapat diakses melalui jaringan internet atau jaringan lokal menggunakan *browser*. *Website* dapat digunakan sebagai media informasi, komunikasi, dan pengelolaan data. Dalam pengembangan sistem informasi, *website* memiliki kelebihan karena dapat diakses oleh beberapa pengguna sesuai hak akses masing-masing. Pada penelitian ini, *website* digunakan sebagai media utama untuk mengelola data guru, siswa, jadwal mengajar, absensi, izin, tugas, dan laporan *monitoring*. Dengan sistem berbasis *website*, pengguna seperti admin, guru, siswa, dan kepala sekolah dapat menggunakan fitur yang sesuai dengan kebutuhan masing-masing[12].

2.2.6 Database MySQL

Database adalah tempat penyimpanan data yang disusun secara terstruktur agar mudah dikelola dan diakses kembali. Dalam sistem informasi, *database* berfungsi sebagai pusat penyimpanan data yang dibutuhkan oleh aplikasi. *MySQL* merupakan salah satu sistem manajemen basis data yang banyak digunakan dalam pengembangan aplikasi berbasis *website*. Pada penelitian ini, *database MySQL* digunakan untuk menyimpan data pengguna, jadwal mengajar, data *fingerprint*, persetujuan siswa, izin guru, tugas, mata pelajaran, dan rekap absensi. Dengan penggunaan *database*, data yang masuk ke sistem dapat disimpan secara rapih dan digunakan kembali untuk kebutuhan laporan[13].

2.2.7 PHP

PHP adalah bahasa pemrograman yang berjalan di sisi *server* dan sering digunakan untuk membuat aplikasi berbasis *website*. *PHP* dapat digunakan untuk memproses data dari pengguna, menghubungkan aplikasi dengan *database*, serta menampilkan informasi sesuai permintaan pengguna. Dalam penelitian ini, *PHP* digunakan sebagai bahasa pemrograman untuk membangun sistem *monitoring* absensi guru. *PHP* berperan dalam mengelola proses *login*, penyimpanan data, pengambilan data dari *database*, pengajuan izin, pengiriman tugas, persetujuan siswa, dan tampilan *monitoring* kepala sekolah.

Contoh script *PHP*:

```
<?PHP
$conn
MySQLi_connect("localhost","root","", "absensi_guru");
if(!$conn){
    echo "Koneksi gagal";
}else{
    echo "Koneksi berhasil";}
```

2.2.8 Metode *Waterfall*

Metode *Waterfall* merupakan metode pengembangan perangkat lunak yang dilakukan secara bertahap. Setiap tahap dikerjakan secara berurutan, mulai dari analisis kebutuhan, perancangan, implementasi, pengujian, hingga pemeliharaan. Metode ini sesuai digunakan apabila kebutuhan sistem sudah dapat dijelaskan sejak awal dan proses pengembangan membutuhkan dokumentasi yang jelas. Pada penelitian ini, metode *Waterfall* digunakan agar proses pembuatan sistem *monitoring* absensi guru dapat berjalan secara terarah. Tahap analisis digunakan untuk memahami kebutuhan pengguna, tahap perancangan digunakan untuk menyusun alur sistem dan *database*, tahap implementasi digunakan untuk membuat aplikasi, dan tahap pengujian digunakan untuk memastikan fitur berjalan sesuai kebutuhan[14].

2.2.9 *Domain*

Domain adalah nama alamat yang digunakan untuk mengakses sebuah *website*. *Domain* berfungsi sebagai pengganti alamat IP agar pengguna lebih mudah membuka suatu halaman web. Dalam penerapan

sistem berbasis *website*, *domain* dapat membantu pengguna mengakses sistem melalui alamat yang lebih sederhana. Pada penelitian ini, pembahasan *domain* digunakan sebagai pengetahuan pendukung apabila sistem *monitoring* absensi guru akan diunggah ke *server* dan dapat diakses melalui alamat *website* tertentu[15].

2.2.10 *Hosting*

Hosting adalah layanan penyimpanan *file website* agar aplikasi dapat diakses melalui jaringan internet. *File* yang disimpan di dalam *hosting* dapat berupa *script* program, gambar, *database*, dan komponen lain yang dibutuhkan oleh *website*. Dalam sistem informasi berbasis *website*, *hosting* berperan sebagai tempat aplikasi dijalankan agar dapat digunakan oleh pengguna. Pada penelitian ini, *hosting* menjadi bagian pendukung apabila sistem *monitoring* absensi guru dikembangkan lebih lanjut dan digunakan secara *online* oleh pihak sekolah[16].

2.2.11 *UML (Unified Modeling Language)*

Unified Modeling Language (UML) merupakan bahasa pemodelan yang digunakan untuk menggambarkan rancangan sistem perangkat lunak. *UML* membantu pengembang dalam menjelaskan alur kerja, hubungan antar pengguna, proses sistem, dan struktur data yang digunakan. Dalam penelitian ini, *UML* digunakan untuk merancang sistem *monitoring* absensi guru agar alur sistem dapat dipahami sebelum aplikasi dibuat. Diagram *UML* yang digunakan meliputi *use*

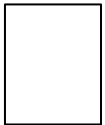
case diagram, *activity diagram*, *sequence diagram*, dan *class diagram*[17].

1. *Use case diagram*

Use case diagram digunakan untuk menggambarkan hubungan antara aktor dengan fitur yang ada di dalam sistem. Aktor dalam sistem ini terdiri dari admin, guru, siswa, dan kepala sekolah. *Activity diagram* digunakan untuk menjelaskan alur kegiatan pada setiap proses, misalnya proses *login*, absensi *fingerprint*, persetujuan siswa, pengajuan izin, dan *monitoring* kepala sekolah. *Sequence diagram* digunakan untuk menunjukkan urutan interaksi antara pengguna dan sistem. *Class diagram* digunakan untuk menggambarkan struktur data atau kelas yang digunakan dalam sistem, seperti data pengguna, jadwal mengajar, *fingerprint log*, *approval* siswa, izin guru, tugas, dan mata pelajaran

Tabel 2. 1 *Use Case Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).

NO	GAMBAR	NAMA	KETERANGAN
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</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.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>Sistem</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Detugas akhir dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
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

2. Activity diagram

Activity diagram adalah diagram yang menggambarkan alur kerja atau aktivitas suatu sistem dalam perangkat lunak.

Tabel 2. 2 *Diagram Activity*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State 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 dibentuk dan dihancurkan
5		<i>Fork node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

3. *Sequence diagram*

Sequence diagram mengilustrasikan objek-objek yang terdapat pada *Use Case* dan menggambarkan arus pesan antara satu sama lain pada Objek *Use Case*. *Sequence diagram* bersifat dinamis dan lebih banyak menampilkan aktifitas objek berdasarkan urutan waktu.

Tabel 2. 3 *Diagram Sequence*

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

NO	GAMBAR	NAMA	KETERANGAN
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

4. *Class diagram*

Class diagram adalah model statis yang mendukung tampilan data dan informasi dari keseluruhan sistem. Penggunaan *Class diagram* dikaitkan dengan struktur basis data sistem atau dapat menggantikan *ERD* pada proses penggambaran diagram rekayasa perangkat lunak yang konvensional.

Tabel 2. 4 *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Detugas akhir dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.

NO	GAMBAR	NAMA	KETERANGAN
6	----->	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7	————	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya