

BAB II

TINJAUAN PUSTAKA

2.1 Kerangka Teoritik

2.1.1 Sistem Presensi

Presensi merupakan suatu kondisi yang menandakan bahwa seseorang hadir secara fisik di suatu tempat atau kegiatan tertentu. Dalam konteks akademik maupun organisasi, presensi menjadi indikator utama untuk mengetahui keterlibatan dan kedisiplinan individu terhadap aktivitas yang dijalankan. Sistem presensi, atau yang sering disebut sistem manajemen kehadiran, adalah sebuah mekanisme yang dirancang untuk mengatur, mencatat, serta mengelola data kehadiran tersebut secara terstruktur. Sistem ini tidak hanya berfungsi sebagai pencatat hadir atau tidak hadir, tetapi juga dapat digunakan sebagai dasar evaluasi performa, penghitungan keaktifan, bahkan sebagai pendukung pengambilan keputusan manajerial. Dengan adanya sistem presensi yang terotomatisasi, proses pencatatan kehadiran menjadi lebih efisien, akurat, dan minim risiko manipulasi dibandingkan metode manual tradisional (Kadir, 2014).

2.1.2 Radio Frequency Identification (RFID)

Menurut Maryono (2005), *Radio Frequency Identification (RFID)* adalah teknologi identifikasi yang memanfaatkan transmisi gelombang radio, umumnya pada frekuensi 125 kHz, 13,56 MHz, atau 800–900 MHz, untuk mengenali secara unik suatu objek maupun individu. RFID dikategorikan sebagai salah satu teknologi *automatic data capture* yang berkembang paling cepat, karena mampu mengumpulkan informasi terkait produk, lokasi, waktu, maupun transaksi secara otomatis, cepat, dan akurat tanpa keterlibatan langsung manusia yang berisiko menimbulkan kesalahan pencatatan.

Keunggulan RFID terletak pada kemampuannya melakukan identifikasi tanpa memerlukan kontak fisik maupun garis pandang langsung (*line of sight*) seperti pada *barcode*. Teknologi ini tetap dapat berfungsi meskipun berada pada lingkungan yang tidak mendukung atau berbahaya. Selain itu, tag RFID tidak hanya berfungsi sebagai kode identitas, tetapi juga sebagai media penyimpanan data yang

dapat ditulis, diperbarui, dan dibaca kembali meskipun dalam kondisi objek bergerak.

Lebih lanjut, Maryono (2005) menjelaskan bahwa sistem RFID secara umum terdiri dari tiga komponen utama, yaitu *tag (transponder)* yang berfungsi menyimpan data identitas, *reader (interrogator)* yang mengirimkan gelombang radio dan menerima respons dari tag, serta sistem komputer atau basis data yang memproses, menyimpan, dan mengelola informasi hasil pembacaan. Integrasi ketiga komponen tersebut menjadikan RFID sebagai solusi identifikasi otomatis yang efisien, andal, dan fleksibel untuk berbagai bidang aplikasi.

2.1.3 Internet of Things (IoT)

Internet of Things (IoT) dapat dipahami sebagai suatu paradigma teknologi di mana berbagai objek fisik dilengkapi dengan sensor, aktuator, kemampuan pemrosesan, serta konektivitas jaringan, sehingga mampu berkomunikasi, bertukar data, dan bekerja sama dengan sistem lain maupun pengguna manusia. Konsep ini tidak hanya mencakup perangkat pintar individu, tetapi juga jaringan yang saling terhubung dan membentuk ekosistem digital yang dapat dimanfaatkan untuk otomatisasi, analisis data, maupun pengambilan keputusan cerdas (Boutaba, 2021).

Internet of Things (IoT) umumnya terdiri dari beberapa komponen utama yang membentuk ekosistemnya:

1. *Perception Layer* (Sensor dan Aktuator)

Tahap ini bertugas mengenali objek, menangkap data dari lingkungan sekitar, serta mengonversi fenomena fisik menjadi sinyal digital. Perangkat yang umum digunakan antara lain sensor suhu, RFID, kamera, dan aktuator seperti motor maupun relay.

2. *Network Layer* (Jaringan dan Komunikasi)

Informasi yang diperoleh sensor diteruskan melalui beragam teknologi komunikasi, seperti WiFi, Bluetooth, ZigBee, LoRa, hingga jaringan seluler (4G/5G). Lapisan ini berperan sebagai penghubung antara perangkat IoT dengan sistem pemrosesan data.

3. *Edge/Processing Layer* (Pemrosesan Data)

Data yang telah ditransmisikan kemudian diolah baik secara lokal melalui *edge computing* maupun di *server* berbasis *cloud*. Tahap pengolahan ini

meliputi penyaringan, analisis, serta integrasi dengan sistem lain agar menghasilkan informasi yang lebih bermanfaat.

4. *Application Layer* (Aplikasi dan Layanan)

Pada lapisan ini, hasil pemrosesan data diwujudkan dalam bentuk aplikasi atau layanan yang dapat digunakan oleh pengguna. Contoh penerapannya meliputi sistem pemantauan kesehatan, rumah pintar, pertanian cerdas, hingga kota cerdas.

Jadi, secara sederhana prinsip kerja IoT dapat dijelaskan sebagai berikut: “*Sense – Communicate – Process – Act*”. Perangkat IoT mendeteksi kondisi lingkungan (*sense*), mengirimkan data ke jaringan (*communicate*), data diproses di *server/cloud* (*process*), lalu hasilnya digunakan untuk memberikan respons atau tindakan tertentu (*act*) (Boutaba, 2021).

2.1.4 Website

Menurut Horton (2000), *Website* merupakan kumpulan halaman yang saling terhubung dan dapat diakses melalui jaringan internet menggunakan peramban (*browser*). Setiap halaman pada *website* dapat menyajikan berbagai jenis informasi dalam bentuk teks, gambar, audio, video, maupun animasi. Selain berfungsi sebagai media penyampaian informasi, *website* juga memungkinkan terjadinya interaksi dan komunikasi antara pengguna dengan sistem maupun antar pengguna. Kemudahan akses melalui internet menjadikan *website* sebagai media yang fleksibel untuk digunakan dalam berbagai kebutuhan, karena informasi dan layanan yang tersedia dapat diakses secara luas tanpa dibatasi oleh waktu dan lokasi tertentu.

2.1.5 RFID Mifare RC522

Radio Frequency Identification (RFID) Mifare RC522 Gambar 2.1 merupakan sebuah chip pembaca sekaligus penulis yang dirancang untuk komunikasi nirkabel pada frekuensi 13,56 MHz. Perangkat ini mendukung standar ISO/IEC 14443 A, termasuk teknologi MIFARE dan NTAG.



Gambar 2.1 Modul Reader MFRC522

Chip ini memiliki pemancar internal yang dapat langsung mengendalikan antenna pembaca/penulis untuk berinteraksi dengan kartu maupun *transponder* MIFARE tanpa perlu rangkaian tambahan. Sementara itu, bagian penerimaannya mampu bekerja secara efektif dalam menangkap, mendemodulasi, dan menerjemahkan sinyal dari perangkat yang kompatibel.

Selain itu, modul digital di dalam MFRC522 yang spesifikasinya dapat dilihat pada Tabel 2.1 berfungsi mengelola proses pembungkahan data serta melakukan pengecekan kesalahan seperti paritas dan CRC. Chip ini juga kompatibel dengan produk MF1xxS20, MF1xxS50, dan MF1xxS70, serta mendukung komunikasi nirkontak berkecepatan tinggi hingga 848 kBd pada arah pengiriman maupun penerimaan data (Semiconductors, 2016).

Tabel 2.1 Spesifikasi Mifare RFID

Parameter	Spesifikasi
Frekuensi operasi	13.56 MHz (HF, High Frequency)
Kompatibilitas standar	ISO/IEC 14443A/MIFARE
Antarmuka komunikasi	SPI (maks. 10 Mbps), I ² C, UART
Tegangan operasi	2.5 V – 3.3 V (biasanya 3.3 V)
Arus konsumsi	13–26 mA (operasi), < 80 μ A (standby)
Jarak baca	Hingga ± 5 cm(tergantung antenna dan tag)
Kecepatan transfer data	106 kbit/s
Ukuran modul	Sekitar 40 mm $\times$ 60 mm (bisa bervariasi)
Fitur tambahan	Mendukung anti-collision, CRC coprocessor, modulasi dan demodulasi terintegrasi

Sumber: (Semiconductors, 2016)

2.1.6 NodeMCU ESP8266

ESP8266 adalah mikrokontroler buatan Espressif Systems yang dirancang sebagai solusi konektivitas WiFi mandiri. Chip ini tidak hanya dapat digunakan sebagai penghubung antara mikrokontroler lain dengan jaringan WiFi, tetapi juga mampu menjalankan aplikasi secara independen. Modul NodeMCU berbasis ESP8266 sudah dilengkapi port USB dan pin yang lengkap, sehingga pengguna cukup menghubungkannya ke laptop dengan kabel micro USB untuk melakukan proses pemrograman, mirip seperti pada Arduino. Selain itu, desainnya juga dibuat ramah untuk penggunaan di *breadboard*, sehingga memudahkan pengembangan proyek (Started Getting Guide User, 2018).



Gambar 2.2 NodeMcu ESP8266

Dalam penelitian ini, NodeMCU yang digunakan adalah NodeMCU ESP8266V3 yang dapat dilihat pada Gambar 2.2. Untuk memperjelas spesifikasi dari modul NodeMCU ESP8266 V3 tersebut dapat dilihat pada Tabel 2.2 sebagai berikut.

Tabel 2.2 Spesifikasi NodeMCU ESP8266 V3

Spesifikasi	Keterangan
Processor	Tensilica 32-bit RISC CPU Xtensa LX106
Processor Speed	80 MHz – 160 MHz
Operating Voltage	3.3 V
Input Voltage	7 V – 12 V
Current Consumption	10 μ A – 170 mA
Digital I/O Pins (DIO)	16
Analog Input Pins (ADC)	1
UARTs	1
SPIs	1

Spesifikasi	Keterangan
I ² Cs	1
Flash Memory	4 MB
SRAM	64 KB
WiFi	IEEE 802.11 b/g/n
Antenna	On-board PCB antenna, gain 2 dBi
USB Port	Micro USB
Max Concurrent TCP Connections	5

Sumber : (Started Getting Guide User, 2018)

2.1.7 RFID Card Mifare 13.56MHz

Smart card Mifare Gambar 2.3 merupakan jenis kartu pintar nirkontak yang banyak digunakan pada sistem transportasi publik, manajemen akses, hingga program loyalitas pelanggan.



Gambar 2.3 RFID CARD MIFARE 13.56 MHz

Teknologi ini berbasis standar ISO/IEC 14443 Tipe A dan tersedia dalam varian dengan kapasitas memori 1 KB maupun 4 KB, serta memiliki *identifier* sepanjang 4 byte atau 7 byte. Material yang digunakan biasanya berupa PVC putih mengilap dengan kualitas cetak tinggi, sehingga memungkinkan kartu untuk dipersonalisasi atau dicetak sesuai kebutuhan aplikasi (Linton, 2010).

2.1.8 LCD 16x2

Modul LCD Gambar 2.4 dilengkapi dengan LSI *controller* yang memiliki dua register 8-bit, yaitu *instruction register* (IR) dan *data register* (DR). *Instruction register* (IR) berfungsi menyimpan kode instruksi, seperti perintah *clear display* atau *cursor shift*, serta alamat data untuk display data RAM (DDRAM) maupun *character generator* RAM (CGRAM). Register ini hanya dapat ditulis melalui mikroprosesor (MPU).



Gambar 2.4 LCD 16x2

Sementara itu, data register (DR) berfungsi sebagai penyimpan sementara data yang akan ditulis atau dibaca dari DDRAM maupun CGRAM. Proses transfer data berlangsung setelah alamat dimasukkan ke dalam IR, kemudian data akan ditulis atau dibaca melalui DR. Pemilihan antara IR dan DR dilakukan dengan memanfaatkan sinyal *register selector* (RS) (Data & Ratings, 2022).

2.1.9 Buzzer

Buzzer merupakan komponen elektronika yang digunakan sebagai indikator suara. Pada *Active Buzzer module*, bunyi yang dihasilkan berupa nada tunggal yang otomatis keluar saat modul diberi tegangan pemicu, karena di dalamnya sudah terdapat osilator internal. Modul ini umumnya menghasilkan suara dengan frekuensi sekitar 2,5 kHz. Sementara itu, untuk menghasilkan nada dengan variasi frekuensi, digunakan *Passive Buzzer module* yang membutuhkan sinyal pemicu dari luar (Rittenberry, 2012).



Gambar 2.5 Active Buzzer

2.1.10 Kabel Jumper

Kabel jumper adalah kabel penghubung yang umum dipakai pada breadboard untuk membuat koneksi sementara antar-komponen dalam rangkaian prototipe (Banzi, 2011). Kabel ini tersedia dalam berbagai jenis seperti *male-to-male*, *male-to-female*, dan *female-to-female* sesuai kebutuhan sambungan.



Gambar 2.6 Kabel Jumper

Pada proyek sistem presensi RFID berbasis IoT, kabel jumper digunakan untuk menghubungkan pin modul RFID, mikrokontroler NodeMCU ESP8266 V3, serta komponen lain seperti LCD atau buzzer, sehingga memungkinkan aliran sinyal dan daya listrik berjalan dengan baik tanpa perlu penyolderan permanen.

2.1.11 Arduino IDE

Arduino IDE (*Integrated Development Environment*) adalah perangkat lunak resmi yang disediakan oleh Arduino, berfungsi sebagai lingkungan pengembangan terpadu. IDE ini menyertakan editor teks untuk menuliskan kode program, area pesan (*message area*), konsol teks (*text console*), toolbar dengan tombol-fungsi umum, dan menu operasi secara hierarkis.



Gambar 2.7 Software Arduino IDE

Fungsi utama Arduino IDE adalah menghubungkan pengguna dengan papan Arduino: pengguna dapat menulis instruksi dalam bahasa pemrograman Arduino (berbasis Wiring), kemudian mengunggah instruksi tersebut ke mikrokontroler di papan melalui IDE (Syahwil, 2013).

2.1.12 Bahasa Pemrograman

2.1.12.1 HTML

Menurut Kadir (2023), *HyperText Markup Language* (HTML) merupakan bahasa markah yang digunakan untuk menyusun struktur dan menampilkan konten pada halaman web. File HTML umumnya disimpan menggunakan ekstensi *.htm* atau *.html*. Dalam penulisannya, HTML menggunakan elemen-elemen yang disebut tag sebagai penanda struktur dokumen. Tag ditulis dengan diawali tanda "<" dan diakhiri tanda ">", yang berfungsi untuk mengatur berbagai komponen yang ditampilkan pada halaman web.

2.1.12.2 CSS

Cascading Style Sheets (CSS) merupakan bahasa yang digunakan untuk mengatur tampilan halaman web, termasuk aspek presentasi, pemformatan, dan tata letak elemen HTML. Penggunaan CSS memungkinkan pengembang membuat desain halaman yang lebih rapi, konsisten, dan sesuai dengan kebutuhan antarmuka yang diinginkan (Kadir, 2023).

2.1.12.3 JavaScript

JavaScript berfungsi untuk mengendalikan interaksi dan perilaku elemen-elemen pada HTML. Bahasa ini menyediakan berbagai fitur yang memungkinkan *tampilan* web menjadi lebih dinamis. Beberapa contohnya adalah menampilkan jam lokal secara *real-time*, melakukan perhitungan data pada formulir, serta mengirim permintaan ke *server* secara asinkron tanpa harus memuat ulang halaman (Kadir, 2023).

2.1.12.4 PHP

PHP (*Hypertext Preprocessor*) merupakan bahasa pemrograman yang bersifat *open source* dan banyak digunakan dalam pengembangan aplikasi berbasis web. PHP dapat dijalankan pada berbagai sistem operasi, seperti Windows maupun Linux, serta umumnya tersedia dalam paket *server* lokal seperti XAMPP dan WampServer. Salah satu keunggulan PHP adalah kemampuannya untuk diintegrasikan secara langsung dengan kode HTML melalui tag `<?php ?>`, sehingga memudahkan pengembang dalam membuat halaman web yang bersifat dinamis dan interaktif (Kadir, 2023).

2.1.13 Basis Data

Basis Data atau *Database* dapat diartikan sebagai kumpulan data yang saling terintegrasi dan dikelola sedemikian rupa sehingga memudahkan proses penyimpanan, pencarian, serta manipulasi data secara cepat dan efisien. Selain memuat data utama, sebuah database juga menyimpan *metadata*, yaitu informasi yang mendeskripsikan struktur data itu sendiri. Contohnya, *metadata* dapat berupa nama kolom dan tipe data pada suatu tabel yang menjelaskan karakteristik data yang tersimpan (Raharjo, 2022).

2.1.14 XAMPP

XAMPP (*Cross-platform, Apache, MySQL, PHP/Perl*) adalah sebuah paket perangkat lunak pengembangan web yang bersifat *gratis* dan *open source*, yang menggabungkan beberapa komponen utama seperti *server web* (Apache), sistem basis data (MySQL / MariaDB), serta interpreter bahasa pemrograman seperti PHP dan Perl. Paket ini disediakan sebagai lingkungan lokal (*localhost*) agar pengembang dapat membuat, menguji, dan memanipulasi aplikasi web secara offline sebelum dipublikasikan ke *server public* (Eko Siswanto, 2021).

2.1.15 MySQL

Raharjo (2022) menjelaskan bahwa MySQL adalah perangkat lunak pengelola basis data yang menerapkan model relasional dalam penyimpanan data. MySQL memungkinkan pengguna untuk membangun basis data beserta struktur tabel yang diperlukan untuk mengorganisasi informasi. Data yang tersimpan dalam tabel-tabel tersebut selanjutnya dapat diakses, dikelola, dan dimanfaatkan oleh aplikasi sesuai kebutuhan sistem yang dikembangkan.

2.1.16 PhpMyAdmin

PhpMyAdmin merupakan aplikasi berbasis web yang digunakan untuk mengelola sistem basis data seperti MySQL maupun MariaDB melalui antarmuka grafis. Melalui phpMyAdmin, pengguna dapat melakukan berbagai aktivitas administrasi, antara lain membuat, mengubah, dan menghapus basis data, tabel, maupun kolom, serta mengeksekusi perintah SQL secara langsung. Antarmukanya yang sederhana dan mudah dipahami menjadikan phpMyAdmin populer di kalangan pengembang web, baik pemula maupun profesional, khususnya yang memanfaatkan teknologi berbasis PHP dan MySQL (Raharjo, 2022).

2.1.17 Visual Studio Code

Menurut Salamah (2021), Visual Studio Code (VS Code) merupakan editor kode sumber yang dikembangkan oleh Microsoft dengan karakteristik ringan, fleksibel, dan mendukung berbagai kebutuhan pengembangan perangkat lunak. VS Code dapat digunakan pada berbagai sistem operasi, seperti Windows, Linux, dan macOS. Secara default, aplikasi ini telah menyediakan dukungan untuk beberapa bahasa pemrograman, seperti JavaScript, TypeScript, dan Node.js. Selain itu, kemampuan VS Code dapat diperluas melalui pemasangan ekstensi yang tersedia pada marketplace resmi, sehingga mendukung berbagai bahasa pemrograman lain, seperti Python, Java, C++, C#, dan Go.

2.1.18 Web Server

Web Server merupakan perangkat lunak yang berperan untuk melayani permintaan (*request*) dari klien, biasanya melalui protokol HTTP atau HTTPS, dengan menyediakan halaman web yang diminta. Permintaan tersebut dikirimkan oleh klien melalui browser, dan *web server* akan menanggapinya dengan mengirimkan kembali (*response*) halaman web dalam format dokumen HTML. Secara umum, istilah *web server* dapat merujuk pada dua hal, yaitu sebagai perangkat keras (*hardware*) yang menjalankan layanan web, maupun sebagai perangkat lunak (*software*) yang mengelola dan memproses permintaan dari pengguna (Limbong & Sriadhi, 2021).

2.1.19 WhatsApp

Menurut James (2023), WhatsApp merupakan aplikasi komunikasi berbasis internet yang dapat digunakan pada berbagai platform perangkat. Aplikasi ini menyediakan beragam fitur komunikasi, seperti pengiriman pesan teks, pesan suara, panggilan suara, panggilan video, serta berbagi file dalam berbagai format, termasuk gambar, dokumen, dan lokasi. Dengan memanfaatkan jaringan internet, WhatsApp memungkinkan proses komunikasi berlangsung secara cepat dan efisien tanpa bergantung pada layanan pesan singkat (SMS) atau panggilan telepon konvensional. Selain itu, penerapan teknologi enkripsi *end-to-end* pada WhatsApp membantu menjaga keamanan dan kerahasiaan data yang dipertukarkan antar pengguna.

2.1.20 Wokwi

Wokwi merupakan simulator daring yang berjalan melalui peramban web dan berfungsi sebagai lingkungan virtual untuk merancang serta menguji rangkaian elektronik. Melalui platform ini, pengguna dapat menulis, mengunggah, dan menjalankan kode Arduino secara langsung tanpa membutuhkan perangkat keras fisik. Selain itu, Wokwi juga mendukung proses pembelajaran mandiri melalui panduan langkah demi langkah dan berbagai contoh proyek praktis yang membantu pengguna memahami konsep pemrograman mikrokontroler dengan lebih mudah (Qirom Qirom, Ulil Albab, 2024).

2.1.21 Efisiensi

Menurut Stair & Reynolds (2020), efisiensi merupakan kemampuan suatu proses atau aktivitas untuk mencapai hasil yang diinginkan dengan penggunaan sumber daya seminimal mungkin. Salah satu indikator efisiensi yang penting adalah waktu yang dibutuhkan untuk menyelesaikan suatu pekerjaan. Stair dan Reynolds (2020) menjelaskan bahwa penerapan teknologi informasi dapat meningkatkan efisiensi dan produktivitas organisasi melalui otomatisasi proses kerja sehingga aktivitas operasional dapat diselesaikan lebih cepat dibandingkan proses manual. Dalam konteks sistem informasi, tingkat efisiensi suatu sistem dapat dilihat dari kemampuannya dalam mempercepat penyelesaian suatu proses dan mengurangi waktu yang diperlukan pengguna dalam melakukan suatu aktivitas. Oleh karena itu, penerapan sistem informasi diharapkan mampu meningkatkan efisiensi proses kerja melalui pemanfaatan teknologi yang dapat mempercepat penyelesaian pekerjaan dibandingkan metode konvensional.

Dalam konteks perangkat lunak, menurut Pressman (2015), menjelaskan bahwa kinerja sistem berkaitan dengan kemampuan perangkat lunak dalam memberikan respons dan menyelesaikan pemrosesan dalam waktu yang sesuai. Sistem yang membutuhkan waktu pemrosesan lebih singkat untuk menghasilkan fungsi dan keluaran yang sama menunjukkan kinerja yang lebih efisien. Oleh karena itu, waktu respons dan waktu penyelesaian proses dapat digunakan sebagai indikator dalam menilai efisiensi suatu sistem.

Menurut Kanawaty (1992), pengukuran kerja atau *work measurement* merupakan penerapan teknik untuk menentukan waktu yang dibutuhkan oleh

pekerja dalam menyelesaikan suatu pekerjaan berdasarkan metode dan kondisi kerja yang telah ditetapkan. Pengukuran kerja dapat digunakan untuk menganalisis penggunaan waktu, membandingkan metode kerja, dan menilai perubahan kinerja setelah dilakukan perbaikan terhadap suatu proses. Dengan demikian, waktu penyelesaian proses dapat digunakan sebagai dasar untuk membandingkan efisiensi antara metode kerja lama dan metode kerja baru.

Salah satu metode yang dapat digunakan dalam pengukuran kerja adalah *time and motion study*. Al-moteri dkk. (2023) menjelaskan bahwa *time and motion study* dilakukan melalui pengamatan langsung dan berkelanjutan terhadap pelaksanaan suatu pekerjaan. Setiap aktivitas yang dilakukan dicatat beserta durasi waktu yang diperlukan untuk menyelesaikannya. Pencatatan waktu dapat menggunakan alat pencatat waktu, seperti stopwatch elektronik, sehingga menghasilkan data kuantitatif mengenai durasi setiap aktivitas. Metode tersebut dapat digunakan untuk mengidentifikasi aktivitas yang membutuhkan waktu paling besar dan menilai efisiensi pelaksanaan suatu proses.

Dalam membandingkan proses sebelum dan setelah penerapan teknologi, *time and motion study* dapat menggunakan desain *before-and-after observational time-motion assessment*. Radovich dkk. (2025) menerapkan desain tersebut dengan mengamati aktivitas pada periode sebelum dan setelah penerapan sistem elektronik. Durasi setiap aktivitas dijumlahkan berdasarkan hari pengamatan, kemudian rata-rata waktu pada kedua periode dibandingkan untuk mengetahui perubahan waktu setelah sistem diterapkan. Pendekatan tersebut menunjukkan bahwa penerapan sistem lama dan sistem baru tidak harus diukur pada waktu yang bersamaan, tetapi aktivitas, unit analisis, serta batas awal dan akhir proses yang dibandingkan harus ditetapkan secara setara.

Berdasarkan teori tersebut, efisiensi waktu dalam penelitian ini didefinisikan sebagai kemampuan sistem presensi RFID berbasis *Internet of Things* dalam menyelesaikan proses pencatatan dan pengelolaan data presensi dengan waktu yang lebih singkat dibandingkan metode manual. Pengukuran dilakukan dengan membandingkan durasi proses presensi manual sebagai kondisi sebelum penerapan sistem dan durasi proses presensi RFID sebagai kondisi setelah penerapan sistem. Perbandingan dilakukan terhadap jumlah data presensi dan batas

proses yang setara sehingga perubahan waktu yang dihasilkan dapat menunjukkan peningkatan efisiensi setelah sistem diterapkan.

Proses manual yang dibandingkan meliputi pemanggilan siswa, pencatatan status terlambat, sakit, izin, dan alfa ke dalam buku presensi, serta pemindahan data mingguan dari buku ke Microsoft Excel. Sementara itu, proses menggunakan RFID meliputi waktu pemrosesan transaksi kartu, pencatatan manual terhadap kondisi yang tidak diproses melalui kartu, serta pembentukan rekap presensi secara otomatis oleh sistem. Semakin kecil waktu yang diperlukan untuk memproses jumlah data presensi yang sama, semakin tinggi tingkat efisiensi waktu yang dihasilkan.

2.1.22 Real-time

Konsep *real-time* merujuk pada pemrosesan data yang dilakukan sesuai dengan kebutuhan waktu dari proses atau peristiwa yang terjadi di luar sistem komputer. ISO/IEC 2382 (2015), mendefinisikan *real-time* sebagai pemrosesan data oleh komputer yang berkaitan dengan proses di luar komputer dan dilakukan sesuai dengan persyaratan waktu yang ditetapkan oleh proses tersebut. Sistem *real-time* merupakan sistem yang pemrosesannya memenuhi kebutuhan waktu tersebut. ISO/IEC 2382:2015 masih berstatus berlaku dan telah dikonfirmasi kembali pada tahun 2025.

Secara umum, sistem *real-time* diklasifikasikan menjadi beberapa jenis, yaitu *hard real-time*, *soft real-time*, dan *near real-time*. *Hard real-time* menuntut respons tanpa toleransi keterlambatan, sedangkan *soft real-time* masih memberikan toleransi terhadap keterlambatan dalam batas tertentu. Adapun *near real-time* merupakan pendekatan yang banyak digunakan pada sistem modern, khususnya *Internet of Things* (IoT), di mana data diproses dan diperbarui dengan jeda waktu yang sangat singkat namun masih dalam batas toleransi tertentu.

Kopetz (2011), menjelaskan bahwa karakteristik utama sistem *real-time* terletak pada kemampuannya memenuhi spesifikasi waktu yang ditetapkan sesuai dengan kebutuhan aplikasi. Dengan demikian, batas waktu pada sistem *real-time* tidak selalu sama untuk setiap jenis aplikasi, tetapi ditentukan berdasarkan fungsi, tingkat kepentingan informasi, dan konsekuensi apabila informasi terlambat diterima.

Sistem presensi yang dikembangkan pada penelitian ini tidak termasuk dalam kategori *hard real-time* karena keterlambatan informasi tidak menyebabkan kegagalan yang membahayakan keselamatan atau membuat keseluruhan sistem tidak dapat digunakan. Sistem lebih sesuai dikategorikan sebagai *soft real-time* dengan pendekatan *near real-time*, yaitu informasi diproses dan disampaikan secara otomatis dalam rentang waktu yang masih mempertahankan relevansinya bagi pengguna. Istilah *near real-time* pada penelitian ini digunakan sebagai definisi operasional dan bukan sebagai klasifikasi formal dalam ISO/IEC 2382

Dalam sistem presensi berbasis IoT, relevansi waktu berkaitan dengan kemampuan sistem dalam menyajikan informasi kehadiran setelah siswa melakukan presensi. Informasi tersebut perlu tersedia dalam waktu yang mendukung pemantauan oleh admin, wali kelas/guru, dan orang tua. Waktu penyampaian informasi dapat dipengaruhi oleh proses pembacaan kartu, pengiriman data melalui jaringan, pemrosesan pada *server*, pembaruan *dashboard*, serta layanan pihak ketiga yang digunakan untuk mengirimkan notifikasi.

Pada aplikasi berbasis *website*, pembaruan informasi dapat dilakukan menggunakan mekanisme *polling*, yaitu klien menjalankan permintaan data kepada server secara berkala. Fungsi *setInterval()* pada JavaScript dapat digunakan untuk menjadwalkan pemanggilan fungsi secara berulang berdasarkan interval yang ditentukan. Namun, interval tersebut merupakan waktu penjadwalan pemanggilan fungsi dan tidak menjamin bahwa data akan selesai ditampilkan tepat pada waktu yang sama karena dapat dipengaruhi oleh pemrosesan *server*, jaringan, dan *browser*.

Selain melalui *dashboard*, penyampaian informasi dapat dilakukan melalui layanan notifikasi pihak ketiga. Pengukuran waktu notifikasi dilakukan menggunakan pendekatan *end-to-end latency*, yaitu selisih antara waktu sistem mengirimkan notifikasi dan waktu pesan diterima pada perangkat pengguna. Pengukuran tersebut mencakup seluruh jalur pengiriman, mulai dari aplikasi presensi, layanan API, platform pesan, jaringan, hingga perangkat penerima.

Berdasarkan uraian tersebut, *real-time* pada penelitian ini didefinisikan sebagai kemampuan sistem untuk:

1. memperbarui data presensi pada dashboard secara otomatis dan berkala tanpa memerlukan *refresh* manual;
2. menampilkan informasi yang sesuai dengan data presensi pada basis data; dan
3. mengirimkan notifikasi secara otomatis kepada orang tua serta mencatat waktu penerimaan pesan sebagai latensi *end-to-end*.

Dengan demikian, pengujian *real-time* dalam penelitian ini tidak ditujukan untuk membuktikan jaminan waktu yang bersifat mutlak, tetapi untuk menilai ketepatan waktu dan otomatisasi penyampaian informasi presensi dalam konteks operasional sekolah.

2.1.23 Efektivitas

Menurut Stair dan Reynolds (2018), efektivitas merupakan kemampuan suatu sistem dalam membantu organisasi mencapai tujuan yang telah ditetapkan melalui penyediaan informasi yang berguna dan mendukung pelaksanaan proses bisnis. Suatu sistem informasi dikatakan efektif apabila mampu menghasilkan informasi yang dibutuhkan pengguna untuk mendukung pelaksanaan aktivitas dan pengambilan keputusan secara lebih baik dibandingkan metode yang digunakan sebelumnya.

Dalam konteks sistem informasi, efektivitas tidak hanya berkaitan dengan kemampuan sistem dalam menghasilkan informasi, tetapi juga berkaitan dengan kontribusi sistem terhadap peningkatan kinerja proses kerja. Penerapan sistem informasi yang efektif diharapkan dapat membantu pengguna memperoleh informasi yang dibutuhkan secara lebih cepat, lebih lengkap, serta mendukung penyederhanaan proses kerja melalui pemanfaatan teknologi (Stair & Reynolds, 2018).

Menurut Harrington (1991), perbaikan proses bisnis dapat dilakukan dengan menghilangkan aktivitas yang tidak bernilai tambah, menyederhanakan tahapan kerja, dan mengurangi pekerjaan yang tidak diperlukan. Oleh karena itu, pengurangan aktivitas manual dapat digunakan sebagai indikator untuk menilai keberhasilan sistem dalam memperbaiki proses administrasi presensi.

Berdasarkan konsep efektivitas sistem informasi yang dikemukakan oleh Stair dan Reynolds (2018) dan Harrington (1991), penelitian ini mengukur

efektivitas sistem presensi RFID berdasarkan kemampuan sistem dalam mendukung pengelolaan data presensi siswa, dan dirumuskan dalam indikator sebagai berikut:

1. Ketersediaan informasi, yaitu kemampuan sistem dalam menyediakan informasi presensi siswa sehingga dapat diakses kapan saja oleh pihak yang membutuhkan (National Institute of Standards and Technology, 2020).
2. Kelengkapan data, yaitu kemampuan sistem dalam menghasilkan data presensi yang lebih lengkap sehingga dapat mendukung kebutuhan administrasi sekolah.
3. Pengurangan aktivitas manual, yaitu kemampuan sistem dalam menyederhanakan proses kerja melalui pengurangan tahapan atau aktivitas yang sebelumnya dilakukan secara manual.

Melalui ketiga indikator tersebut, efektivitas sistem dapat dinilai berdasarkan sejauh mana sistem presensi RFID mampu meningkatkan kualitas pengelolaan data presensi dibandingkan metode presensi manual yang digunakan sebelumnya.

2.1.24 Usability

Menurut Nielsen (1993), *usability* merupakan atribut kualitas yang digunakan untuk menilai seberapa mudah suatu sistem atau antarmuka digunakan oleh pengguna. Nielsen menjelaskan bahwa *usability* terdiri dari lima komponen utama, yaitu *learnability*, *efficiency*, *memorability*, *errors*, dan *satisfaction*. *Learnability* menunjukkan tingkat kemudahan pengguna dalam mempelajari penggunaan sistem saat pertama kali digunakan.

Salah satu metode yang secara luas digunakan untuk mengukur *usability* adalah *System Usability Scale* (SUS) yang dikembangkan oleh John Brooke pada tahun 1998. Menurut Brooke (1998), *System Usability Scale* (SUS) merupakan instrumen evaluasi yang digunakan untuk mengukur *usability* suatu sistem berdasarkan persepsi pengguna setelah berinteraksi dengan sistem tersebut. SUS terdiri dari sepuluh pernyataan yang dinilai menggunakan skala Likert lima poin dan dapat digunakan untuk memberikan penilaian secara cepat terhadap *usability* suatu sistem. Hasil pengukuran SUS dapat digunakan untuk menggambarkan sejauh mana sistem dapat diterima dan digunakan dengan mudah oleh pengguna.

SUS menggunakan sepuluh pernyataan sebagai instrumen penilaian yang diukur dengan skala Likert lima tingkat, mulai dari sangat tidak setuju hingga sangat setuju. Setiap pernyataan disusun secara bergantian antara bentuk positif dan negatif untuk membantu mengurangi kemungkinan terjadinya bias jawaban dari responden. Penggunaan SUS dalam berbagai penelitian menunjukkan bahwa instrumen ini memiliki tingkat keandalan dan validitas yang baik dalam mengevaluasi usability suatu sistem.

Menurut Lewis & Sauro (2017), SUS telah menjadi metode yang banyak dijadikan acuan dalam pengukuran usability karena mampu memberikan penilaian umum mengenai kemudahan penggunaan sistem secara cepat dan sederhana. Meskipun jumlah pertanyaannya terbatas, hasil yang diperoleh tetap mampu merepresentasikan persepsi pengguna terhadap usability sistem yang diuji.

Perhitungan skor SUS dilakukan melalui beberapa tahapan. Untuk item bernomor ganjil, skor kontribusi dihitung dengan rumus:

$$Skor = Nilai Respons - 1$$

Sedangkan untuk item bernomor genap, skor kontribusi dihitung dengan rumus:

$$Skor = 5 - Nilai Respons$$

Seluruh skor kontribusi kemudian dijumlahkan dengan nilai maksimum 40, dan hasilnya dikalikan dengan faktor 2,5 untuk menghasilkan skor akhir dalam rentang 0–100.

$$Skor SUS = (Total Skor Kontribusi) \times 2,5$$

Rumus ini digunakan karena setiap item pada SUS memiliki bobot yang sama dan dikonversi ke skala standar 0–100 agar memudahkan interpretasi hasil serta perbandingan antar sistem.

2.1.25 Flowchart

Menurut Ladjamudin (2013), flowchart merupakan diagram yang digunakan untuk menggambarkan urutan langkah dalam suatu prosedur atau proses kerja dengan memanfaatkan simbol-simbol tertentu yang dihubungkan oleh garis alir. Diagram ini berfungsi untuk memperjelas alur kegiatan, proses pengambilan

keputusan, serta keterkaitan antar aktivitas dalam suatu sistem sehingga lebih mudah dipahami.

Gery B. Shelly (2011) menjelaskan bahwa flowchart banyak dimanfaatkan pada tahap analisis dan perancangan sistem informasi karena mampu menggambarkan logika dan aliran proses secara visual sebelum sistem diimplementasikan ke dalam bentuk program. Dengan adanya flowchart, pengembang dapat memahami dan merancang proses sistem secara lebih terstruktur. Pada penelitian ini digunakan beberapa symbol flowchart sebagaimana ditunjukkan pada Tabel 2.3. Setiap simbol memiliki fungsi yang berbeda-beda, antara lain:

1. *Oval (Terminator)*

Digunakan untuk menyatakan awal dan akhir sebuah flowchart. Setiap alur proses idealnya memiliki satu simbol mulai (*start*) dan minimal satu simbol selesai (*end*).

2. *Parallelogram (Input/Output)*

Menyatakan kegiatan pemasukan data (*input*) ke dalam sistem maupun kegiatan menampilkan keluaran (*output*), seperti membaca kartu, memasukkan data formulir, atau menampilkan laporan pada layar.

3. *Rectangle (Process)*

Melambangkan proses atau langkah pengolahan yang dilakukan oleh sistem, misalnya perhitungan, penyimpanan data, validasi, atau manipulasi data lainnya.

4. *Diamond (Decision)*

Digunakan untuk pengambilan keputusan yang menghasilkan dua atau lebih cabang alur (misalnya “ya/tidak” atau beberapa alternatif kondisi).

5. *Arrows (Flow Line)*

Menunjukkan arah aliran logika antar simbol. Panah menghubungkan satu langkah proses ke langkah berikutnya sesuai urutan yang diinginkan.

6. *Circle (On Page Connector)*

Menyatakan penghubung alur pada halaman yang sama ketika diagram cukup kompleks sehingga alur perlu dipotong menjadi beberapa bagian agar tidak saling tumpang tindih.

7. *Off Page Connector*

Melambangkan penghubung alur antar halaman berbeda. Simbol ini digunakan jika flowchart tidak cukup ditampilkan dalam satu halaman sehingga memerlukan lanjutan pada halaman berikutnya.

8. *Predefined Process / Function*

Menyatakan proses yang sudah didefinisikan sebelumnya atau sub-proses yang detailnya digambarkan pada diagram terpisah. Simbol ini memudahkan penyederhanaan flowchart yang kompleks.

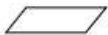
9. *Preprocessor*

Digunakan untuk kegiatan pendahuluan atau proses persiapan sebelum proses utama dilakukan, misalnya inisialisasi variabel atau penyiapan perangkat.

10. *Comments*

Menunjukkan catatan atau keterangan tambahan pada bagian tertentu dari flowchart untuk menjelaskan proses yang dianggap penting atau tidak cukup jelas hanya dengan simbol.

Tabel 2.3 Simbol dan Fungsi *Flowchart*

Symbol Name	Symbol	function
Oval		Used to represent start and end of flowchart
Parallelogram		Used for input and output operation
Rectangle		Processing: Used for arithmetic operations and data-manipulations
Diamond		Decision making. Used to represent the operation in which there are two/three alternatives, true and false etc
Arrows		Flow line Used to indicate the flow of logic by connecting symbols
Circle		Page Connector
		Off Page Connector
		Predefined Process /Function Used to represent a group of statements performing one processing task.
		Preprocessor
		Comments

Dengan penggunaan simbol-simbol tersebut, flowchart yang dibuat dalam penelitian ini diharapkan mampu menggambarkan alur proses sistem secara jelas, konsisten, dan mudah dipahami oleh semua pihak yang terlibat.

2.1.26 Use Case Diagram

Menurut Fowler (2004), *use case diagram* merupakan salah satu jenis diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan kebutuhan fungsional suatu sistem berdasarkan interaksi antara pengguna dan sistem. Diagram ini menunjukkan aktor yang terlibat serta berbagai fungsi atau layanan yang dapat diakses oleh masing-masing aktor dalam sistem.

Nugroho (2009) menjelaskan bahwa *use case diagram* berfungsi untuk memperlihatkan hubungan antara aktor dan fitur yang tersedia dalam sistem, sehingga dapat digunakan untuk mengidentifikasi siapa saja pengguna sistem dan aktivitas yang dapat mereka lakukan. Oleh karena itu, *use case diagram* sering dimanfaatkan pada tahap analisis kebutuhan untuk memberikan gambaran awal mengenai fungsi-fungsi yang harus disediakan oleh sistem.

Kegunaan *use case diagram* dalam penerapan maupun implementasi sistem adalah (Pressman, 2015):

1. Mengkomunikasikan kebutuhan sistem antara analis, pengembang, dan pemilik kepentingan secara sederhana dalam bentuk skenario fungsional.
2. Membantu mengidentifikasi batasan sistem, aktor yang terlibat, dan fungsi yang harus disediakan agar kebutuhan pengguna dapat terpenuhi.
3. Menjadi dasar penyusunan spesifikasi kebutuhan fungsional dan skenario pengujian, karena setiap *use case* dapat diuraikan menjadi langkah-langkah interaksi yang lebih detail.

Simbol-simbol yang digunakan dalam *use case diagram* pada penelitian ini ditunjukkan pada Tabel 2.4, dengan penjelasan sebagai berikut:

1. *Actor*

Mewakili peran eksternal yang berinteraksi dengan sistem, bisa berupa pengguna manusia, sistem lain, atau perangkat eksternal. Aktor tidak selalu satu orang, tetapi dapat berupa kelompok atau entitas yang memiliki tujuan yang sama terhadap sistem.

2. *Dependency*

Menunjukkan ketergantungan antara elemen dalam *use case diagram*. Perubahan pada satu elemen dapat mempengaruhi elemen lain yang bergantung padanya.

3. *Generalization*

Menyatakan hubungan pewarisan (*inheritance*) antara aktor atau antara *use case*. Aktor anak mewarisi perilaku aktor induk; demikian pula, *use case* khusus mewarisi perilaku *use case* yang lebih umum.

4. *Include*

Menunjukkan bahwa suatu *use case* selalu menyertakan perilaku *use case* lain secara eksplisit. *Use case* yang di-include biasanya menggambarkan fungsi umum yang dipakai berulang oleh beberapa *use case* lain, misalnya “Validasi Login”.

5. *Extend*

Menggambarkan bahwa *use case* target memperluas perilaku *use case* lain pada kondisi tertentu. *Use case* yang meng-extend hanya dieksekusi bila kondisi pemicu terpenuhi, sehingga bersifat opsional.

6. *Association*

Menunjukkan hubungan komunikasi antara aktor dan *use case*, yaitu aktor mana yang terlibat pada suatu fungsi tertentu. Garis asosiasi mengindikasikan bahwa aktor berpartisipasi dalam pelaksanaan *use case* tersebut.

7. *System (Boundary)*

Biasanya digambarkan sebagai persegi panjang yang membungkus seluruh *use case*. Simbol ini menunjukkan batas sistem (*system boundary*), sehingga terlihat jelas mana fungsi yang berada di dalam sistem dan mana entitas yang berada di luar sistem.

8. *Use Case*

Digambarkan dalam bentuk *elips* dan merepresentasikan fungsionalitas atau layanan yang diberikan sistem kepada aktor, misalnya “Mengelola Data Presensi” atau “Mencetak Laporan”. Setiap *use case* menggambarkan urutan aksi yang menghasilkan keluaran bernilai bagi aktor.

9. Collaboration

Menunjukkan interaksi kerja sama antara beberapa elemen (aktor maupun *use case*) untuk mencapai perilaku yang lebih kompleks. Simbol ini menekankan adanya sinergi antar elemen.

10. Note

Digunakan untuk menambahkan catatan atau penjelasan tambahan terhadap elemen tertentu pada diagram, misalnya asumsi, batasan, atau keterangan khusus yang perlu diberi penjelasan tekstual.

Tabel 2.4 Simbol dan Keterangan *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 (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
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 <i>eksplisit</i> .
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>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 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

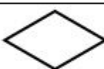
Melalui pemodelan dengan *use case diagram*, kebutuhan fungsional sistem yang dikembangkan dalam penelitian ini dapat terdokumentasi dengan baik, mudah dipahami oleh pemangku kepentingan, dan menjadi dasar dalam perancangan diagram UML lainnya maupun implementasi sistem.

2.1.27 Activity Diagram

Activity diagram merupakan salah satu diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk memvisualisasikan alur aktivitas atau proses yang terjadi dalam suatu sistem. Diagram ini menggambarkan urutan aktivitas yang dilakukan mulai dari proses awal hingga proses berakhir, termasuk keputusan, percabangan, serta aliran kontrol yang terjadi selama proses berlangsung.

Menurut Booch, James Rumbaugh dan Ivar Jacobson (2005), *activity diagram* digunakan untuk memodelkan perilaku dinamis suatu sistem dengan menunjukkan hubungan antar aktivitas, kondisi yang memengaruhi jalannya proses, serta transisi yang menghubungkan setiap aktivitas secara terstruktur. Oleh karena itu, *activity diagram* sering dimanfaatkan untuk menggambarkan proses bisnis maupun alur kerja sistem agar lebih mudah dipahami dan dianalisis. Elemen pada *Activity diagram* dapat dilihat pada Tabel 2.5 dibawah ini:

Tabel 2.5 Simbol Elemen *Activity Diagram*

Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / Join	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

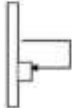
Dengan demikian, *activity diagram* digunakan untuk memberikan gambaran menyeluruh mengenai alur proses dalam sistem, sehingga memudahkan dalam analisis, perancangan, dan komunikasi sistem.

2.1.28 Sequence Diagram

Sequence diagram merupakan salah satu diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan interaksi antar objek dalam suatu sistem berdasarkan urutan waktu. Diagram ini menunjukkan bagaimana objek saling bertukar pesan untuk menjalankan suatu proses atau fungsi tertentu sehingga alur komunikasi dalam sistem dapat dipahami dengan lebih jelas.

Menurut Grady Booch et al. (2005), sequence diagram digunakan untuk memodelkan perilaku dinamis sistem dengan menampilkan urutan pengiriman pesan antar objek secara kronologis. Melalui diagram ini, hubungan antar objek dan aliran proses yang terjadi selama suatu fungsi dijalankan dapat divisualisasikan secara terstruktur. Elemen utama pada sequence diagram dapat dilihat pada Tabel 2.6 berikut:

Tabel 2.6 Simbol Elemen Sequence Diagram

Gambar	Nama	Keterangan
	Entity Class	Gambaran sistem sebagai landasan dalam menyusun basis data
	Boundary Class	Menangani komunikasi antar lingkungan sistem
	Control Class	Bertanggung jawab terhadap kelas-kelas terhadap objek yang berisi logika
	Recursive	Pesan untuk dirinya
	Activation	Mewakili proses durasi aktivasi sebuah operasi
	Life Line	Komponen yang digambarkan garis putus terhubung dengan objek

Dengan demikian, sequence diagram digunakan untuk menggambarkan proses komunikasi dalam sistem secara kronologis sehingga memudahkan analisis logika sistem dan implementasi program.

2.1.29 *Entity Relationship Diagram (ERD)*

Entity Relationship Diagram (ERD) merupakan model yang digunakan untuk menggambarkan struktur data dalam suatu sistem secara konseptual. Diagram ini menunjukkan entitas, atribut yang dimiliki oleh setiap entitas, serta hubungan yang terbentuk di antara entitas tersebut. Menurut Thomas Connolly & Begg (2015), ERD digunakan untuk memodelkan kebutuhan data suatu organisasi dalam bentuk diagram yang terstruktur sebelum diterapkan pada sistem basis data relasional.

ERD membantu dalam mengidentifikasi komponen utama dalam database, yaitu:

1. Entitas (*entity*), yang merepresentasikan objek atau konsep seperti siswa atau kartu RFID
2. Atribut (*attribute*), yang merupakan karakteristik dari entitas, seperti ID atau nama
3. Relasi (*relationship*), yaitu hubungan antar entitas, seperti hubungan antara siswa dan data presensi

Selain itu, *ERD* juga digunakan untuk menggambarkan jenis relasi antar entitas, seperti *one-to-one*, *one-to-many*, dan *many-to-many*, yang berperan penting dalam menentukan struktur tabel pada database relasional. Dengan adanya ERD, perancangan *database* menjadi lebih sistematis, mengurangi redundansi data, serta mempermudah proses implementasi dan pengembangan sistem.

2.1.30 *Diagram Swimlane*

Diagram Swimlane merupakan salah satu jenis diagram alur yang digunakan untuk memvisualisasikan proses bisnis dengan membagi aktivitas ke dalam beberapa jalur (*lane*) berdasarkan pihak atau aktor yang bertanggung jawab terhadap aktivitas tersebut. Diagram ini membantu menunjukkan hubungan antar aktor, aliran informasi, serta urutan aktivitas yang terjadi dalam suatu proses bisnis. Setiap *lane* mewakili individu, departemen, atau sistem yang terlibat sehingga pembagian tugas dan tanggung jawab dapat terlihat secara jelas.

Menurut Rummier dan Brache (2013), Swimlane Diagram digunakan untuk memetakan proses bisnis lintas fungsi (*cross-functional process*) dengan tujuan *memperjelas* interaksi antar pihak yang terlibat dalam suatu proses. Diagram ini mampu membantu mengidentifikasi alur kerja, perpindahan informasi, potensi hambatan proses, serta peluang perbaikan yang dapat dilakukan pada suatu sistem. Oleh karena itu, Diagram Swimlane banyak digunakan dalam analisis dan perancangan sistem informasi untuk menggambarkan proses yang sedang berjalan maupun proses yang diusulkan.

Pada penelitian ini, Diagram Swimlane digunakan untuk memodelkan proses presensi siswa yang melibatkan beberapa aktor, yaitu siswa, sistem RFID, admin tata usaha, dan orang tua. Penggunaan diagram ini bertujuan untuk menggambarkan secara rinci alur proses presensi, pertukaran informasi antar aktor, serta tanggung jawab masing-masing pihak dalam sistem presensi berbasis RFID dan *Internet of Things* (IoT) yang dikembangkan.

2.1.31 Pengujian Beban Pengguna (*Load Testing*)

Pengujian kinerja atau *performance testing* merupakan pengujian nonfungsional yang dilakukan untuk mengetahui tingkat responsivitas dan kemampuan sistem ketika menerima beban tertentu. Menurut ISTQB (2018), pengujian kinerja digunakan untuk mengevaluasi perilaku sistem pada volume beban yang berbeda. Aspek yang dapat dinilai meliputi perilaku waktu (*time behavior*), pemanfaatan sumber daya, dan kapasitas sistem. Pengukuran perilaku waktu berhubungan dengan kemampuan sistem dalam memberikan respons terhadap masukan pengguna pada kondisi pengujian yang telah ditentukan.

Salah satu jenis pengujian kinerja adalah *load testing*. ISTQB (2018), mendefinisikan *load testing* sebagai pengujian yang berfokus pada kemampuan sistem dalam menangani peningkatan beban realistis yang diperkirakan, yang dihasilkan oleh sejumlah pengguna atau proses konkuren secara terkendali. Pengujian ini berbeda dengan *stress testing* yang memberikan beban melampaui batas operasional sistem dan *spike testing* yang memberikan lonjakan beban puncak secara tiba-tiba.

Pengguna konkuren atau *concurrent user* merupakan sejumlah pengguna yang menjalankan sesi atau aktivitas pada sistem dalam waktu yang bersamaan atau

berdekatan. Pada perangkat pengujian beban, pengguna konkuren umumnya direpresentasikan melalui pengguna virtual atau *execution thread*. Jumlah pengguna saja belum sepenuhnya menggambarkan besar beban karena aktivitas yang dilakukan dan frekuensi permintaan juga memengaruhi beban yang diterima sistem. Oleh karena itu, skenario aktivitas pengguna harus ditetapkan secara konsisten pada setiap tingkat pengujian (ISTQB, 2018).

Pengujian beban dapat dilaksanakan secara bertingkat dengan meningkatkan jumlah pengguna virtual pada setiap skenario. ISTQB (2018) menyarankan penggunaan kondisi beban inkremental untuk mengamati pengaruh peningkatan beban terhadap respons sistem. Melalui pendekatan tersebut, perubahan kinerja dapat dibandingkan pada setiap tingkat beban sehingga kecenderungan penurunan atau peningkatan performa sistem dapat diketahui.

Parameter yang umum digunakan dalam pengujian beban meliputi waktu respons, *throughput*, dan tingkat kesalahan. Waktu respons menunjukkan waktu yang diperlukan sistem untuk menerima, memproses, dan memberikan respons terhadap permintaan. *Throughput* menunjukkan jumlah transaksi atau permintaan yang dapat diproses dalam satuan waktu, sedangkan tingkat kesalahan menunjukkan persentase permintaan yang gagal diproses oleh sistem.

Apache JMeter merupakan perangkat lunak yang dapat digunakan untuk mensimulasikan akses pengguna virtual terhadap aplikasi web. Pada JMeter, jumlah pengguna ditentukan melalui *number of threads*, sedangkan *ramp-up period* menentukan waktu yang diperlukan untuk mengaktifkan seluruh pengguna virtual. Beberapa *thread* yang dijalankan secara independen digunakan untuk mensimulasikan koneksi pengguna konkuren terhadap aplikasi server.

Penentuan tingkatan pengguna virtual dapat disesuaikan dengan kebutuhan operasional sistem dan jumlah pengguna potensial. Xu (2024) melakukan pengujian terhadap sistem informasi manajemen mahasiswa menggunakan 20, 40, 60, 80, dan 100 pengguna konkuren. Tingkatan tersebut digunakan untuk mengamati hubungan antara peningkatan jumlah pengguna dan perubahan waktu respons sistem. Dengan demikian, penggunaan tingkatan 20 sampai 100 pengguna dapat dijadikan sebagai referensi empiris, tetapi bukan ketentuan baku yang harus diterapkan pada seluruh sistem.

2.1.32 CIA TRIAD

CIA Triad merupakan konsep dasar keamanan informasi yang terdiri atas tiga aspek utama, yaitu *confidentiality*, *integrity*, dan *availability*. Ketiga aspek tersebut digunakan untuk menjaga agar informasi terlindungi dari akses, perubahan, penghapusan, maupun gangguan yang tidak sesuai dengan kewenangan. NIST menjelaskan bahwa keamanan informasi dilakukan dengan melindungi informasi dan sistem dari akses, penggunaan, pengungkapan, gangguan, perubahan, atau penghancuran yang tidak sah guna mempertahankan kerahasiaan, integritas, dan ketersediaan informasi (National Institute of Standards and Technology, 2020).

1. *Confidentiality* (Kerahasiaan)

Confidentiality merupakan upaya mempertahankan pembatasan akses dan pengungkapan informasi agar hanya dapat dilakukan oleh pihak yang memiliki kewenangan. Aspek ini bertujuan melindungi informasi pribadi maupun informasi lain yang tidak boleh diakses secara bebas. Penerapannya dapat dilakukan melalui autentikasi, pembagian hak akses, dan pembatasan informasi berdasarkan peran pengguna.

2. *Integrity* (Integritas)

Integrity berkaitan dengan perlindungan informasi dari perubahan atau penghapusan yang tidak sah. Aspek ini memastikan bahwa informasi tetap akurat, lengkap, konsisten, dan dapat dipercaya selama disimpan, diproses, maupun dikirimkan. Integritas juga mencakup kemampuan untuk memastikan keaslian sumber dan mencegah manipulasi data oleh pihak yang tidak berwenang.

3. *Availability* (Ketersediaan)

Availability merupakan upaya memastikan bahwa informasi dan layanan dapat diakses secara tepat waktu dan andal oleh pengguna yang berwenang ketika dibutuhkan. Ketersediaan dapat terganggu oleh kerusakan perangkat, gangguan jaringan, kegagalan *server*, maupun kesalahan sistem. Oleh karena itu, sistem perlu menyediakan mekanisme yang mendukung keberlangsungan layanan dan mengurangi risiko kehilangan akses terhadap informasi.

Berdasarkan konsep tersebut, CIA Triad dapat digunakan sebagai landasan untuk meninjau perlindungan dan pengelolaan informasi dalam suatu sistem. Dalam

penelitian ini, aspek *availability* digunakan sebagai teori pendukung dalam menilai ketersediaan informasi kehadiran bagi pengguna yang berwenang. Sementara itu, aspek *confidentiality* dan *integrity* tetap digunakan untuk menjelaskan perlindungan akses dan keutuhan data pada sistem. Penerapan ketiga aspek tersebut dijelaskan lebih lanjut pada bagian pembahasan hasil penelitian.

2.2 Metode Pengembangan Perangkat Lunak

Penelitian ini menggunakan metode pengembangan perangkat lunak prototipe (*prototyping model*). Menurut Pressman (2015), model prototipe adalah salah satu pendekatan rekayasa perangkat lunak yang memungkinkan pengembang membangun rancangan awal sistem (*prototype*), kemudian melakukan pengujian dan diskusi bersama pengguna hingga sistem akhir terbentuk sesuai dengan kebutuhan. Model ini dinilai tepat ketika penelitian menuntut adanya interaksi yang intensif antara pengguna dengan pengembang, sebab kebutuhan sistem dapat mengalami perubahan maupun penyempurnaan sepanjang proses pengembangan.

Adapun tahapan dalam metode prototipe menurut Pressman (2015) meliputi:

1. *Communication* (Tahap Komunikasi): Tahap pengumpulan kebutuhan dengan dilakukan komunikasi dengan pengguna.
2. *Quick Plan* (Tahap Perencanaan Cepat): perencanaan cepat untuk membuat prototype, seperti ruang lingkup prototype, prioritas fitur, jadwal singkat, sumber daya, dan rencana iterasi.
3. *Prototype design* (desain prototipe): membuat rancangan sederhana yang dapat divisualisasikan kepada pengguna.
4. *Construction Phase* (Tahap Konstruksi): merealisasikan prototipe baik berupa perangkat keras maupun perangkat lunak.
5. *Deployment, Delivery & Feedback* (Tahap Uji dan Evaluasi): Prototype diujikan atau didemokan ke pengguna untuk mendapatkan *feedback*. Hasil evaluasi ini jadi dasar perbaikan iteratif (ulang lagi ke tahap plan atau design sesuai kebutuhan).

Kelebihan metode prototipe terletak pada sifatnya yang interaktif, mudah dipahami oleh pengguna, serta mampu mempercepat proses identifikasi kebutuhan sistem. Oleh karena itu, pendekatan ini dianggap sesuai untuk penelitian yang berfokus pada pengembangan sistem presensi berbasis IoT dan RFID.

2.3 Penelitian Relevan

Hidayat dkk. (2025) Penelitian ini mengembangkan sistem informasi presensi siswa berbasis RFID yang terintegrasi dengan WhatsApp *Gateway* di MAS Al-Huda Kangayan. Sistem ini mencatat kehadiran secara otomatis saat siswa menempelkan kartu RFID, dan secara langsung mengirimkan notifikasi kehadiran ke orang tua melalui WhatsApp. Hasil pengujian menunjukkan peningkatan efisiensi waktu hingga 83% dibandingkan metode manual. Sistem ini juga dirancang dengan antarmuka web yang memungkinkan admin tata usaha dan guru mengelola data presensi. Namun, akses orang tua masih terbatas pada notifikasi satu arah, tanpa adanya *dashboard* khusus untuk pemantauan yang lebih interaktif.

Akbar dkk. (2025) Penelitian ini merancang sistem presensi dosen berbasis RFID dan IoT yang terintegrasi dengan kamera ESP32-CAM dan notifikasi Telegram. Sistem ini mencatat kehadiran secara otomatis, mengambil foto pengguna sebagai verifikasi visual, dan mengirimkan notifikasi ke administrator. Data kehadiran disimpan dalam *database web* dan dapat diakses secara *real-time*. Sistem ini juga dilengkapi dengan sensor ultrasonik untuk mendeteksi keberadaan pengguna di depan pintu. Penelitian ini menekankan aspek keamanan dan transparansi, namun fokusnya masih terbatas pada lingkungan perguruan tinggi dan belum mencakup keterlibatan orang tua dalam konteks pendidikan dasar atau menengah.

Harum dkk. (2023) merancang sistem presensi berbasis RFID yang dilengkapi dengan modul pencegahan penyakit menular menggunakan sensor suhu tubuh. Sistem ini menggunakan arsitektur IoT tiga lapis dan protokol MQTT untuk transmisi data *real-time*. Selain mencatat kehadiran, sistem ini juga mendeteksi gejala demam sebagai upaya pencegahan penyebaran penyakit di lingkungan sekolah. Penelitian ini menekankan pentingnya integrasi data kesehatan dalam sistem presensi, namun belum mengembangkan fitur *multi-user* untuk orang tua.

Srujan dkk. (2023) Dalam jurnal “*Sustainable and No Contact Attendance System (RFID Based)*”, peneliti mengusulkan sistem presensi tanpa kontak berbasis RFID yang dilengkapi dengan layar LCD dan mekanisme pembuka pintu otomatis. Sistem ini dirancang untuk mengurangi kontak fisik dan meningkatkan keamanan selama pandemi COVID-19. Data kehadiran dicatat secara otomatis dan

ditampilkan secara *real-time*. Sistem ini juga mendukung integrasi dengan Excel untuk pencatatan waktu masuk dan keluar. Penelitian ini menekankan keberlanjutan, efisiensi, dan keamanan, namun belum mengembangkan fitur akses *multi-user* atau *dashboard* berbasis web untuk orang tua.

Sumarsono & Mayasari, (2025) mengembangkan sistem presensi dan monitoring keamanan berbasis *Internet of Things* (IoT), RFID, *webserver*, dan *Android* dengan memanfaatkan komponen *open-source*. Sistem yang dikembangkan mampu mencatat kehadiran secara otomatis menggunakan RFID, menyimpan data ke dalam basis data MySQL, menampilkan rekap presensi melalui website, serta mengirimkan notifikasi secara *real-time* ke aplikasi Android. Selain fungsi presensi, penelitian ini juga mengintegrasikan fitur monitoring keamanan, seperti deteksi suhu tubuh, deteksi pergerakan manusia, dan deteksi kebakaran melalui Telegram Bot. Hasil penelitian menunjukkan bahwa seluruh fungsi sistem berjalan dengan baik dan mampu mendukung proses monitoring secara efektif, efisien, responsif, dan *real-time*. Namun, penelitian ini masih berfokus pada lingkungan usaha (*Small and Medium Enterprises/SMEs*) serta belum menyediakan *dashboard website multi-user* yang melibatkan orang tua sebagai pengguna sistem, sehingga keterlibatan orang tua dalam pemantauan kehadiran belum menjadi perhatian utama.

Kelima penelitian di atas menunjukkan bahwa penerapan RFID pada sistem presensi sekolah berbasis IoT mampu meningkatkan efisiensi, akurasi dan keamanan dalam pencatatan internal dan belum mengakomodasi akses *multi-user*, khususnya bagi orang tua siswa. Oleh karena itu, penelitian ini menawarkan kebaruan berupa pengembangan sistem presensi RFID yang terintegrasi dengan *dashboard website multi-user*, yang memungkinkan admin tata usaha, guru dan orang tua untuk mengakses data kehadiran siswa secara *real-time* sesuai dengan hak akses masing-masing. Hal ini diharapkan dapat meningkatkan transparansi, keterlibatan orang tua, dan efektivitas manajemen kehadiran di sekolah.

2.4 Kerangka berfikir

Permasalahan utama dalam proses pencatatan kehadiran siswa di SMP Islam Al-Azhar Bukittinggi adalah masih digunakannya metode manual berbasis buku absensi fisik, yang menimbulkan berbagai kendala seperti ketidakefisienan waktu,

ketergantungan pada ketelitian guru, potensi kesalahan pencatatan, serta keterlambatan dalam rekapitulasi data kehadiran. Kondisi tersebut berdampak pada kurang optimalnya pengelolaan data kehadiran dan efektivitas administrasi sekolah.

Sebagai upaya modernisasi, sekolah sempat menerapkan metode presensi berbasis *fingerprint*. Namun, metode ini belum memberikan hasil yang optimal karena memiliki sejumlah keterbatasan yang menunjukkan perlunya alternatif teknologi yang lebih sesuai untuk lingkungan sekolah.

Seiring dengan perkembangan teknologi, penerapan *Radio Frequency Identification* (RFID) yang terintegrasi dengan *Internet of Things* (IoT) menjadi solusi yang lebih relevan. Teknologi RFID memungkinkan proses pencatatan kehadiran dilakukan secara otomatis, cepat, dan tanpa kontak fisik, sehingga mampu mengurangi kesalahan pencatatan serta meningkatkan efisiensi pengolahan data kehadiran. Data hasil pembacaan RFID dapat langsung dikirimkan ke sistem berbasis IoT untuk disimpan dan dikelola secara terpusat.

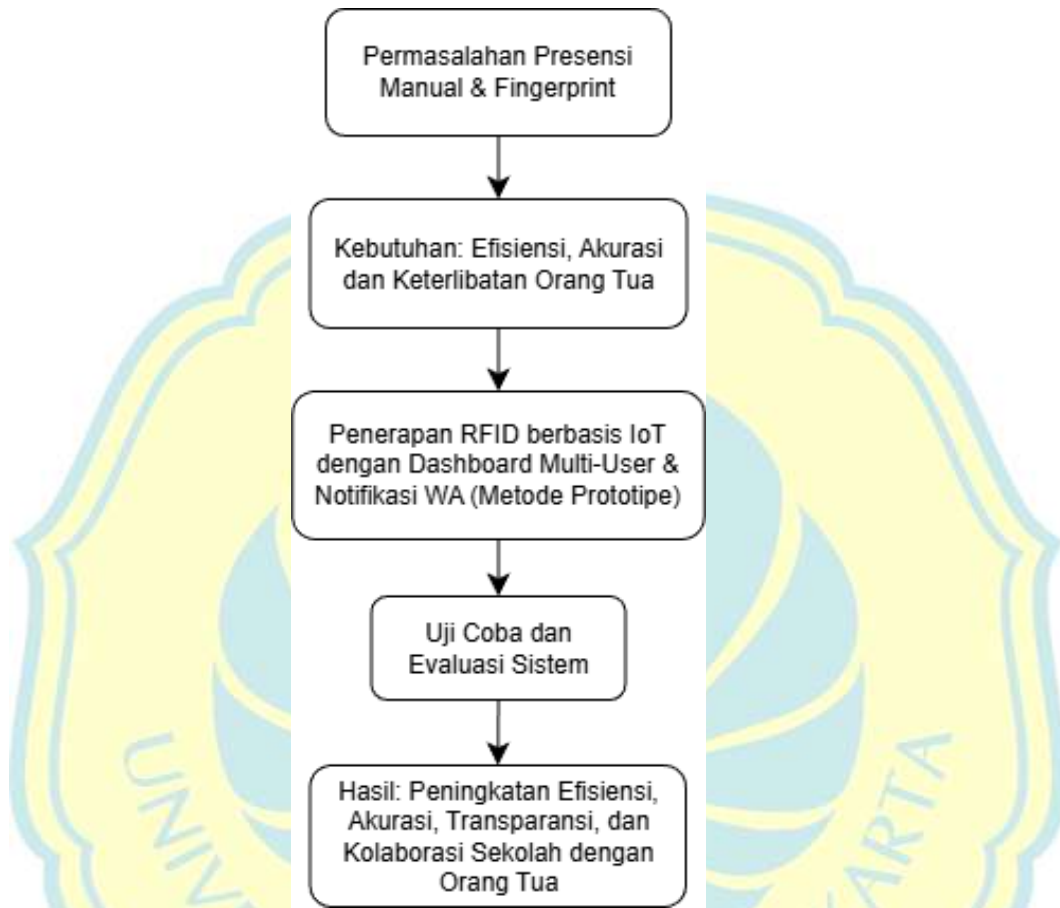
Meskipun penerapan RFID dalam pencatatan kehadiran telah banyak dilakukan, penelitian-penelitian sebelumnya umumnya masih berfokus pada kebutuhan internal sekolah, seperti pengelolaan oleh admin tata usaha dan guru. Keterlibatan orang tua dalam pemantauan kehadiran siswa masih terbatas, biasanya hanya melalui notifikasi satu arah menggunakan aplikasi pesan instan, tanpa didukung oleh media pemantauan yang lebih komprehensif dan interaktif.

Berdasarkan kondisi tersebut, penelitian ini berfokus pada penerapan teknologi RFID pada pencatatan kehadiran siswa berbasis IoT yang terintegrasi dengan *dashboard website multi-user* dan notifikasi WhatsApp otomatis. *Dashboard website* dirancang untuk digunakan oleh tiga jenis pengguna, yaitu admin tata usaha sebagai pengelola data, guru sebagai pihak yang memantau kehadiran siswa, dan orang tua sebagai pihak yang memantau kehadiran anak secara *real-time* sesuai dengan hak akses masing-masing. Melalui penerapan ini, diharapkan:

- a) Proses pencatatan kehadiran siswa menjadi lebih efisien, akurat, dan minim kesalahan.
- b) Guru dan pihak sekolah lebih mudah dalam mengelola dan merekap data kehadiran siswa.
- c) Orang tua dapat terlibat secara langsung dalam memantau kehadiran anak

melalui *dashboard website* dan notifikasi WhatsApp.
d) Pengelolaan administrasi kehadiran siswa menjadi lebih terstruktur, transparan, dan akuntabel.

Secara visual, kerangka berpikir penelitian ini digambarkan pada Gambar 2.8.



Gambar 2.8 Kerangka Berfikir

Berdasarkan uraian kerangka berpikir tersebut, dapat disimpulkan bahwa penerapan RFID berbasis IoT yang didukung oleh *dashboard website multi-user* dan notifikasi WhatsApp merupakan pendekatan yang tepat untuk mengatasi permasalahan pencatatan kehadiran manual maupun keterbatasan metode fingerprint di SMP Islam Al-Azhar Bukittinggi. Penerapan ini diharapkan mampu meningkatkan efisiensi pengolahan data kehadiran, menyederhanakan alur administrasi, serta memperkuat keterlibatan orang tua dalam proses pemantauan kehadiran siswa. Dengan demikian, penelitian ini tidak berfokus pada perancangan sistem presensi secara luas, melainkan pada penerapan teknologi RFID sebagai solusi pendukung pengelolaan kehadiran siswa yang lebih efektif dan relevan dengan kebutuhan sekolah.