

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Teori Terkait

Penelitian terdahulu yang berkaitan dengan sistem *monitoring* dan kontrol berbasis *web*. Uraian mencakup jenis data yang digunakan, metode validasi, penelitian terkait, metrik evaluasi, serta hasil evaluasi pengembangan yang bertujuan untuk meningkatkan kinerja, fitur, dan keamanan sistem. Ringkasan penelitian terkait disajikan pada Table 2.1.

Tabel 2. 1. Teori Terkait

| Penulis dan Tahun                                         | Judul                                                                                                                                      | Validasi                                                                                                           | Penelitian Terkait                                                                                                                                     | Metrik Evaluasi                                                                  | Hasil Evaluasi                                                                         |
|-----------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|
| Waryanto, Mursalim, Tresia Aprilia, Muhammad Khozin(2025) | Implementasi Ganda Modul ESP32 untuk Kontrol Pintu Otomatis dengan Monitoring Realtime dan Autolock Terjadwal melalui Web dan Chatbot[31]. | Implementasi dual modul ESP32 (Master & Slave) untuk pembagian tugas sistem dan divalidasi dengan metode Black-box | Mengatasi keterbatasan sistem sebelumnya yang hanya menggunakan satu mikrokontroler tanpa cadangan baterai dan belum memiliki fitur autolock terjadwal | Persentase keberhasilan sub-pengujian, rata-rata latensi, dan ketahanan baterai. | 86% sub-pengujian sukses, rata-rata latensi 16,3 detik, baterai tahan hingga 32,5 jam. |

| Penulis dan Tahun          | Judul                                                                                                                        | Validasi                                                                                                     | Penelitian Terkait                                                                                                                              | Metrik Evaluasi                                                         | Hasil Evaluasi                                                           |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------|--------------------------------------------------------------------------|
| M. W. Aryadi et al. (2025) | <i>Smart Lock System</i> Berbasis IoT Menggunakan ESP32 untuk Keamanan Akses Pusat Data (Studi Kasus: UPA TIK Undiksha)[29]. | Penggunaan protokol MQTT untuk komunikasi dua arah dan pendaftaran pengguna dinamis via bot Telegram         | Mengatasi kelemahan sistem stand-alone yang belum mampu mendeteksi akses ilegal dan pendaftaran user yang masih harus melalui pemrograman ulang | Rata-rata latensi (detik) dan tingkat keberhasilan autentikasi (%)      | Rata-rata latensi 4 detik dan tingkat keberhasilan autentikasi 100%      |
| R. H. Tidar et al. (2025)  | Design and development of an IoT-based archive room security system integrating RFID and fingerprint authentication [26].    | Penggabungan autentikasi ganda (RFID dan sidik jari) yang terintegrasi dengan pencatatan log berbasis cloud. | Banyak sistem keamanan arsip militer sebelumnya hanya mengandalkan metode tunggal atau kurang dalam komunikasi real-time.                       | Akurasi (%), waktu respon (detik), False Rejection Rate (FRR), dan FAR. | Akurasi gabungan 90,6%, waktu respon RFID 4,26 detik, FRR 8%, dan FAR 0% |
| Nurhafidzah et al. (2023)  | Implementasi Face Recognition, Sensor Getar dan Password sebagai Pengaman Ganda pada Smart Door Lock Berbasis IoT[3].        | Penggabungan Face Recognition, Keypad, dan Sensor Getar yang terintegrasi dengan Telegram.                   | Meningkatkan keamanan pintu rumah agar dapat memberikan notifikasi berupa foto dan pesan <i>real-time</i> saat ada orang asing terdeteksi.      | Tingkat akurasi pengenalan wajah (%) dan tingkat kesalahan (error).     | Akurasi pengenalan wajah 85,7% dengan tingkat error 14,3%.               |

| Penulis dan Tahun        | Judul                                                                                 | Validasi                                                                                  | Penelitian Terkait                                                                                       | Metrik Evaluasi                               | Hasil Evaluasi                                           |
|--------------------------|---------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|-----------------------------------------------|----------------------------------------------------------|
| T. Sutikno et al. (2024) | Fingerprint based smart door lock system using Arduino and smartphone application[11] | Pembuatan perangkat kunci nirkabel berbasis sidik jari dengan kontrol aplikasi smartphone | Memberikan solusi keamanan modern yang terjangkau untuk menekan angka kriminalitas yang terus meningkat. | Rata-rata waktu respon (latensi) dalam detik. | Rata-rata waktu respon autentikasi sidik jari 1,54 detik |

## 2.2 Landasan Teoreti

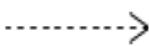
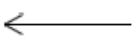
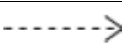
### 2.2.1 Unified Modeling Language (UML)

Sebelum sebuah sistem perangkat lunak dibangun, diperlukan sebuah alat untuk merancang dan mendokumentasikan sistemnya terlebih dahulu agar pengembangan berjalan terarah. Alat yang umum digunakan keperluan tersebut adalah *unified modeling language* (UML), yaitu bahasa pemodelan visual yang telah menjadi standar industri dalam rekayasa perangkat lunak. Pada penelitian ini, UML digunakan untuk menggambarkan sistem dari berbagai sudut pandang melalui empat jenis diagram, yaitu *Use Case Diagram*, *Sequence Diagram*, *Class Diagram* dan *Activity Diagram*, yang secara bersama-sama memepersentasikan fungsionalitas, alur proses, struktur data, dan interaksi antara komponen sistem secara menyeluruh.

#### 1. Use Case Diagram

Ketika merancang sebuah sistem, hal pertama yang perlu diperjelas adalah siapa saja yang menggunakan sistem tersebut dan apa saja yang bisa mereka lakukan. Untuk menjawab hal itu, digunakan diagram *Use Case* yang menggambarakan interaksi antara aktor dengan fungsionalitas sistem tanpa menjelaskan detail teknisnya, sehingga batasan dan hak akses setiap peran dalam sistem dapat terdefinisi dengan jelas sebagaimana ditunjukkan pada Table 2.2.

Tabel 2. 2. Use Case Diagram

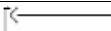
| NO | Simbol                                                                              | Nama           | Keterangan                                                                                                   |
|----|-------------------------------------------------------------------------------------|----------------|--------------------------------------------------------------------------------------------------------------|
| 1. |  | Actor          | Menggambarkan pengguna atau pihak eksternal yang berinteraksi langsung dengan sistem                         |
| 2. |  | Dependency     | Menggambarkan Ketergantungan antar elemen, dimana perubahan pada satu elemen akan berdampak pada elemen lain |
| 3. |  | Generalization | Menggambarkan hubungan perwarisan dimana elemen anak mewarisi sifat dan perilaku dari elemen induknya        |
| 4. |  | Include        | Menggambarkan bahwa satuan use case selalu memanggil usecase lain                                            |

| NO  | Simbol                                                                              | Nama          | Keterangan                                                                                                         |
|-----|-------------------------------------------------------------------------------------|---------------|--------------------------------------------------------------------------------------------------------------------|
|     |                                                                                     |               | sebagai bagian dari prosesnya                                                                                      |
| 5.  |    | Extend        | Menggambarkan bahwa suatu use case dapat menambahkan perilaku tambahan pada use case dalam sistem                  |
| 6.  |    | Association   | Menggambarkan garis penghubung yang menunjukan relasi antara actor dengan use case dalam sistem                    |
| 7.  |  | System        | Menggambarkan batas ruang lingkup sistem yang membedakan bagian dalam dan luar sistem                              |
| 8.  |  | Use Case      | Menggambarkan fungsi atau layanan spesifik yang disediakan sistem dan dapat digunakan oleh aktor                   |
| 9.  |  | Collaboration | Menggambarkan Kerjasama antar elemen sistem untuk menghasilkan perilaku yang tidak dapat dicapai secara individual |
| 10. |  | Note          | Menggambarkan keterangan atau komentar tambahan yang memperjelas elemen dalam diagram                              |

## 2. Sequence Diagram

Setelah fungsionalitas sistem terdefinisi, langkah berikut adalah memahami bagaimana setiap proses berjalan secara kronologi antar komponen. Untuk itu digunakan diagram *sequence* yang menggambarkan urutan interaksi antar komponen berdasarkan dimensi waktu khusus seperti autentifikasi penggunaan, verifikasi *email*, dan *control* akses pintu dari jarak jauh sebagaimana ditunjukkan pada Table 2.3.

Tabel 2. 3. Sequence Diagram

| NO | Simbol                                                                              | Nama           | Keterangan                                                                                                               |
|----|-------------------------------------------------------------------------------------|----------------|--------------------------------------------------------------------------------------------------------------------------|
| 1. |  | Life Line      | Objek entity, antarmuka yang saling berinteraksi                                                                         |
| 2. |  | Message        | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi                   |
| 3. |  | Message        | Spesifikasi dari komunikasi antar objek yang memuat informasi objek yang memuat informasi tentang aktifitas yang terjadi |
| 4. |  | Entity Class   | Gambar sistem sebagai landasan dalam menyusun                                                                            |
| 5. |  | Boundary Class | Menangani komunikasi antar lingkungan sistem                                                                             |
| 6. |  | Control Class  | Bertanggung jawab terhadap kelas-kelas terhadap                                                                          |

| NO | Simbol                                                                            | Nama      | Keterangan          |
|----|-----------------------------------------------------------------------------------|-----------|---------------------|
| 7. |  | Recursive | Pesan untuk dirinya |

### 3. Class Diagram

Selain alur prosesnya, struktur data yang menopang sistem juga perlu dirancang secara eksplisit sebelum implementasi dilakukan. Diagram kelas hadir untuk kebutuhan tersebut dengan menggambarkan kelas, atribut metode, serta relasi antar entitas seperti pengguna perangkat, log akses, dan sidik jari yang menjadi fondasi basis data sistem sebagaimana ditunjukkan pada Table 2.4.

Tabel 2. 4. Class Diagram

| NO | Simbol                                                                              | Nama           | Keterangan                                                             |
|----|-------------------------------------------------------------------------------------|----------------|------------------------------------------------------------------------|
| 1. |  | Generalization | Hubungan dimana objek anak (descendent)                                |
| 2. |  | Class          | Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama |
| 3. |  | Association    | Apa yang menghubungkan antara objek satu dengan objek lain             |

### 4. Activity Diagram

Untuk memastikan tiap alur aktivitas pengguna dalam sistem berjalan sesuai logika yang diharapkan, digunakan

diagram aktivitas yang menggambarkan proses bisnis secara sekuensial maupun bercabang. Diagram ini memodelkan aktivitas seperti proses masuk ke sistem pembukaan pintu berdasarkan verifikasi lokal, dan pendaftaran sidik jari yang disertai verifikasi *email* sebagai lapisan keamanan tambahan sebagaimana ditunjukkan pada Table 2.5.

Tabel 2. 5. Activity Diagram

| No | Simbol                                                                              | Nama                | keterangan                                                                                |
|----|-------------------------------------------------------------------------------------|---------------------|-------------------------------------------------------------------------------------------|
| 1. |    | Activity            | Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain |
| 2. |  | Action              | State dari sistem yang menceminkan eksekusi dari suatu aksi                               |
| 3. |  | Initial Node        | Bagaimana objek dibentuk atau diawali                                                     |
| 4. |  | Activity Final Node | Bagaimana objek dibentuk dan dihancurkan                                                  |
| 5. |  | Fork Node           | Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran                      |

### 2.2.2 Mikrokontroler

Pusat kendali utama dalam sistem ini menggunakan ESP32 yang mengintegrasikan seluruh *sensor* dan *akuator* secara terpadu. Perangkat ESP32 dimanfaatkan karena telah dilengkapi konektivitas

Wi-Fi dan Bluetooth internal untuk mendukung komunikasi nirkabel dan pengiriman data ke *server* atau *platform cloud* secara efisien[42], [43], [44], [45]. Modul tersebut bertugas mengolah data dari *sensor* autentikasi serta memberikan perintah penguncian pintu secara responsif dan *real-time*[46].

### 2.2.3 MQTT

Sistem ini memanfaatkan *Message Queuing Telemetry Transport* (MQTT) sebagai protokol komunikasi yang ringan dan efisien, sehingga efektif diterapkan pada perangkat IoT seperti ESP32[47]. MQTT digunakan untuk mengirimkan data status sistem dan hasil pembacaan *sensor* secara *real-time* melalui mekanisme *publish/subscribe*[48]. Dalam arsitektur ini, ESP32 berperan sebagai *publisher* yang mengirimkan data ke *broker* sebagai perantara komunikasi[49]. Selanjutnya, data tersebut diteruskan dan disinkronkan ke layanan *cloud* seperti *Firebase* untuk disimpan dan dikelola secara daring[50]. Data yang tersimpan kemudian ditampilkan pada *dashboard* monitoring, sehingga informasi kondisi sistem dapat dipantau secara responsif dan *real-time* tanpa perlu melakukan penyegaran halaman secara manual[51].

### 2.2.4 Firebase

Layanan *Firebase* berfungsi sebagai infrastruktur backend berbasis cloud. *Platform* ini mengintegrasikan penyimpanan data secara *real-time* guna menjamin sinkronisasi otomatis antara

perangkat keras, *server*, dan *dashboard* monitoring secara efisien[52]. Melalui transmisi protokol MQTT, setiap informasi dipreservasi secara instan sehingga pembaruan data pada antarmuka web dapat diakses tanpa perlu melakukan penyegaran halaman. Pemilihan teknologi ini didasari oleh kapabilitasnya dalam menyediakan sistem pemantauan berbasis *cloud* yang stabil, responsif, serta mudah diintegrasikan[53].

#### 2.2.5 Laravel

Penelitian ini mengimplementasikan *framework* berbasis Laravel. *Platform* tersebut berfungsi sebagai infrastruktur manajemen data terpusat yang mengukung arsitektur Model, View, Controller guna menjamin pengorganisasian kode secara sistematis dan efisien. Pemanfaatan *endpoint* RESTful API memungkinkan sistem menerima transmisi data *sensor* berformat JSON serta berkas gambar dari modul *ESP32* secara optimal[54]. Selain mengelola persistensi basis data, *platform* ini menyajikan visualisasi pemantauan *real-time* melalui *dashboard web* yang mendukung aspek skalabilitas untuk pengembangan fitur di masa depan[55].

#### 2.2.6 Groq Cloud API

Platform Kecerdasan buatan berbasis *cloud* yang menyediakan layanan inferensi dengan latensi rendah adalah Groq, yang diakses mulai antar muka Groq Cloud API. Keunggulan platform ini terletak pada kecepatan pemrosesan yang sangat tinggi di bandingkan

platformAI sejenis, sehingga mampu menghasilkan respons analisis secara *real-time* dengan rata-rata latensi sekitar 2 detik[56]. Integrasi Groq ke dalam sistem web dilakukan melalui pengiriman HTTP *request* menggunakan metode *POST endpoint* yang kompatibel dengan standar OpenAI API, dengan muatan berformat JSON[33]. Dalam proses ini diterapkan strategi *prompt engineering*, yaitu mendefinisikan peran sistem sebagai instruksi perilaku model dan role user yang membuat data kontekstual seperti log keamanan, guna menghasilkan analisis serta rekomendasi yang dapat langsung ditindaklanjuti oleh pengguna melalui *dashboard* aplikasi.

#### 2.2.7 Verifikasi Email

Keamanan sistem berbasis web modern memerlukan maksimisasi autentikasi berlapis untuk melindungi operasi yang bersifat sensitif[57]. Salah satu pendekatan yang umum diterapkan adalah verifikasi *email* menggunakan *One-Time Password (OTP)*, yaitu kode unik berbatas waktu yang hanya berlaku untuk satu kali penggunaan mencegah penyalahgunaan akses[58]. Mekanisme ini merupakan bagian dari konsep *Two-Factor Authentication (2FA)* yang menambahkan lapisan[59]. Verifikasi *email* diterapkan sebagai syarat wajib sebelum penggunaan dapat menambahkan data sidik jari baru, dengan token dikirim melalui layanan Laravel Mail ke alamat *email* terdaftar sebagai konfirmasi identitas pengguna[60].

### 2.2.8 MySQL

Penelitian ini menyimpan data user secara permanen menggunakan MySQL merupakan sistem manajemen basis data relasional *Relational Database Management System* (RDBMS) yang bersifat *open-source* dan banyak digunakan dalam pengembangan aplikasi berbasis web[61]. Pada penelitian ini, MySQL digunakan sebagai penyimpanan utama data pengguna, termasuk informasi akun, data sidik jari, dan log akses yang dikelola melalui *framework* Laravel[62]. Pemilihan MySQL didasari oleh kemampuannya dalam menangani query secara efisien, kemudahan integrasinya dengan Laravel melalui *Eloquent ORM*, serta kestabilannya dalam menyimpan data relasional secara terstruktur. Berbeda dengan Firebase Realtime Database yang digunakan untuk sinkronisasi data perangkat secara *real-time*, MySQL berperan sebagai penyimpanan data yang bersifat persisten dan teraudit, sehingga keduanya saling melengkapi dalam arsitektur sistem yang dikembangkan.