

## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Teori Terkait

Penelitian sebelumnya yang dilakukan oleh Nurhayati Djabir (2022) membahas *prototype* pintu gerbang otomatis berbasis perintah suara menggunakan Arduino dan aplikasi *Bluetooth Voice*. Sistem ini menggunakan *Arduino UNO*, modul Bluetooth *HC-05*, motor stepper, dan driver motor untuk membuka dan menutup gerbang. Hasil pengujian menunjukkan sistem dapat merespon perintah suara secara efektif pada jarak 10–60 cm dalam kondisi hening, sehingga mempermudah kontrol gerbang tanpa interaksi manual[6].

Penelitian oleh Wayan Raditya (2022) membahas sistem keamanan gerbang rumah berbasis Telegram menggunakan ESP8266. Sistem ini menggunakan sensor *PIR* untuk mendeteksi orang, sensor magnet untuk memantau kondisi gerbang, dan lampu sorot sebagai alarm. ESP8266 berfungsi sebagai pengendali utama dan mengirim notifikasi ke Telegram, sehingga keamanan rumah lebih efektif dan dapat dikontrol secara jarak jauh melalui *IoT*[7].

Penelitian oleh Toni Sukendar (2022) membahas perancangan pintu gerbang otomatis berbasis Arduino yang dikontrol melalui smartphone Android menggunakan koneksi *Bluetooth HC-05*. Sistem ini memanfaatkan *Arduino Uno*, motor *DC/servo*, *buzzer*, *limit switch*, dan *LED* indikator untuk membuka dan menutup gerbang secara otomatis. Pengujian menunjukkan

*Bluetooth* memungkinkan kontrol dari jarak hingga 15 meter di ruang terbuka dan 10 meter di ruang tertutup, sehingga memudahkan pengguna tanpa perlu mengoperasikan gerbang secara manual. Sistem ini juga menggunakan pemrograman Arduino berbasis bahasa C untuk mengatur semua fungsi perangkat keras[8].

Penelitian oleh Bayu Hafit Setiawan & Erfian Junianto (2024) membahas sistem pengendalian pintu gerbang otomatis berbasis *IoT* menggunakan *NodeMCU* ESP8266 dan aplikasi *Blynk*. Sistem ini menggunakan *NodeMCU* ESP8266, motor stepper 28BYJ-48, sensor infrared, dan aplikasi *Blynk* untuk memungkinkan kontrol gerbang dari jarak jauh melalui jaringan internet. Sensor infrared mendeteksi gerakan atau jarak, sedangkan motor stepper dikontrol oleh *NodeMCU* untuk membuka dan menutup gerbang secara presisi. Aplikasi *Blynk* berfungsi sebagai antarmuka pengguna untuk mengoperasikan gerbang secara *real-time*, baik membuka maupun menutupnya. Pengujian menunjukkan bahwa sistem ini memiliki waktu respons rata-rata sekitar 1 detik, efisien, dan dapat memudahkan pengguna dalam mengontrol pintu gerbang dari jarak jauh tanpa interaksi fisik langsung[9].

Febriansyah & Yunanda (2025) mengembangkan sistem keamanan gerbang otomatis berbasis *IoT* dengan teknologi *keyless* untuk perumahan. Sistem menggunakan *RFID* untuk autentikasi, ESP32 sebagai mikrokontroler, sensor proximity, servo motor, dan *LED* indikator, serta notifikasi *real-time* melalui Telegram. Pengguna cukup mendekatkan kartu

*RFID* yang terdaftar, gerbang akan terbuka otomatis, dan aktivitas tercatat dalam sistem. Pengujian black-box menunjukkan sistem bekerja sesuai harapan: hanya kartu terdaftar yang dapat membuka gerbang, *servo* dan sensor *proximity* berfungsi akurat, dan notifikasi dikirim secara *real-time*. Sistem ini meningkatkan keamanan, efisiensi, dan kenyamanan bagi penghuni perumahan[10].

### **Sintesis dan Posisi Penelitian**

Berdasarkan kelima penelitian terdahulu yang telah dijabarkan, perlu ditetapkan rujukan utama dan penelitian pembandingan untuk mempertegas posisi serta nilai kebaruan (*novelty*) dari sistem yang dikembangkan pada penelitian ini. Rujukan utama pada penelitian ini adalah Febriansyah & Yunanda (2025) serta Setiawan & Junianto (2024). Hal ini dikarenakan adanya kesamaan fundamental pada arsitektur sistem, yaitu penggunaan konsep *Internet of Things* (IoT), pemanfaatan mikrokontroler canggih keluarga ESP, serta penggunaan metode akses non-fisik (RFID) dan media notifikasi jarak jauh.

Sementara itu, penelitian pembandingan yang digunakan adalah penelitian oleh Djabir (2022), Raditya, et al. (2022), dan Sukendar, et al. (2022) untuk mengevaluasi efektivitas media komunikasi data dan penggerak mekanis. Ketiga penelitian tersebut menunjukkan batas komparasi, seperti penggunaan Bluetooth yang membatasi jarak kontrol secara signifikan, atau penggunaan mikrokontroler dengan spesifikasi konektivitas dasar seperti Arduino UNO.

### **Kebaruan Penelitian (*Novelty*)**

Perbedaan mendasar sekaligus kebaruan dari penelitian ini dibandingkan rujukan di atas terletak pada integrasi sistem kontrol menggunakan *dashboard* berbasis website dengan *framework* Laravel untuk pemantauan terpusat, penggunaan motor DC untuk efisiensi pergerakan prototipe gerbang, yang dikombinasikan secara utuh dengan sistem autentikasi *RFID* luring, kendali manual ruangan melalui *push button*, serta bot notifikasi Telegram untuk peringatan dan pemantauan jarak jauh. Implementasi seluruh elemen ini dalam satu kesatuan sistem yang komprehensif belum ditemukan pada penelitian-penelitian sebelumnya.

Penelitian terdahulu yang menjadi acuan pada bagian ini berkaitan erat dengan sistem kendali pintu gerbang otomatis, keamanan akses berbasis *Internet of Things* (IoT), serta integrasi sistem pemantauan (*monitoring*) berbasis web. Uraian berikut mencakup metode dan fitur utama yang digunakan, kesenjangan masalah (*research gap*) yang diselesaikan, metrik evaluasi pengujian, serta hasil evaluasi kuantitatif dari masing-masing penelitian yang bertujuan untuk meningkatkan kinerja, efisiensi, dan keamanan sistem. Ringkasan dari evaluasi penelitian terkait tersebut disajikan pada Tabel 2.1.

Tabel 2. 1 Teori Terkait

| Penulis dan Tahun            | Judul                                                                                                                 | Metode Utama                                                                                                           | Hasil Evaluasi                                                                                                                           | Research Gap                                                                                                                                             |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nurhayati Djibir (2022)      | <i>Prototype Pintu Gerbang Otomatis dengan Perintah Suara Menggunakan Mikrokontroler dan Aplikasi Bluetooth Voice</i> | Penggunaan perintah suara via aplikasi <i>Bluetooth Voice</i> , mikrokontroler Arduino UNO, dan motor <i>stepper</i> . | Modul merespons perintah suara dengan efektif pada jarak 10–60 cm. Tingkat keberhasilan 95% (kata "buka") dan 90% (kata "tutup").        | Sistem masih berbasis koneksi lokal ( <i>Bluetooth</i> ), belum memiliki manajemen <i>user</i> , dan rentan terhadap kebisingan suara di sekitar.        |
| Wayan Raditya, et al. (2022) | Penerapan Sistem Keamanan Gerbang Rumah Berbasis Telegram Menggunakan ESP8266                                         | Penggunaan sensor <i>PIR</i> , sensor magnet, lampu sorot, dan ESP8266 untuk notifikasi keamanan via bot Telegram.     | Alat layak beroperasi stabil dalam pengujian selama 10 hari. Jangkauan kontrol berhasil mencapai 70 meter.                               | Belum memiliki antarmuka <i>website</i> terpusat untuk <i>dashboard</i> dan manajemen <i>user</i> , serta tidak memiliki akses autentikasi <i>RFID</i> . |
| Toni Sukendar, et al. (2022) | Perancangan Pintu Gerbang Otomatis Menggunakan Arduino dengan Koneksi <i>Bluetooth</i> Berbasis Android               | Kontrol gerbang via antarmuka tombol di <i>smartphone</i> Android menggunakan <i>Bluetooth</i> HC-05 dan Arduino Uno.  | Sistem berhasil mengotomatisasi buka-tutup gerbang secara nirkabel dengan jangkauan maksimal 15 meter (terbuka) dan 10 meter (tertutup). | Kontrol terbatas pada area jangkauan <i>Bluetooth</i> , tidak dapat dipantau dari luar jangkauan, serta tidak memiliki fitur peringatan bahaya (alarm).  |

| Penulis dan Tahun                            | Judul                                                                                                                             | Metode Utama                                                                                                                                           | Hasil Evaluasi                                                                                                                     | Research Gap                                                                                                                                                  |
|----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Bayu Hafit Setiawan & Erfian Junianto (2024) | Sistem Pengendalian Pintu Gerbang Otomatis Berbasis IoT Menggunakan Aplikasi Blynk                                                | Pemanfaatan NodeMCU ESP8266, sensor <i>infrared</i> , dan aplikasi Blynk sebagai antarmuka kendali jarak jauh via internet.                            | Waktu respons eksekusi perintah dari aplikasi ke perangkat keras tergolong sangat cepat dengan rata-rata 1 detik.                  | Bergantung sepenuhnya pada <i>platform</i> pihak ketiga (Blynk), sulit untuk kustomisasi fitur keamanan spesifik seperti <i>monitoring box</i> .              |
| Febriansyah & Yunanda (2025)                 | Simulasi Sistem Keamanan dan Kendali Gerbang Otomatis pada Portal Perumahan Menggunakan Teknologi Tanpa Kunci Berbasis <i>IoT</i> | Autentikasi akses <i>RFID</i> (keyless), sensor <i>proximity</i> , ESP32, motor servo, dan <i>log</i> notifikasi ke Telegram (diuji dengan Black-box). | Sistem berhasil menolak kartu tidak terdaftar dengan akurasi 100% dan log aktivitas tercatat di Telegram secara <i>real-time</i> . | Belum ada integrasi <i>website</i> terpusat untuk manajemen <i>user</i> (CRUD) dan belum ada sensor deteksi sabotase kotak kontrol ( <i>monitoring box</i> ). |

## 2.2. Landasan Teori

### 2.2.1. Pintu Gerbang

Pintu gerbang adalah titik akses utama yang berfungsi untuk mengatur dan mengontrol pergerakan orang atau kendaraan masuk serta keluar dari suatu area yang dibatasi, seperti pekarangan rumah, kompleks perumahan, atau fasilitas khusus[11].



Gambar 2. 1 Pintu Gerbang

### 2.2.2. Visual Studio Code

*Visual Studio Code* merupakan perangkat lunak editor kode sumber yang ringan namun memiliki kemampuan yang andal dan berjalan pada sistem desktop. Aplikasi ini digunakan dalam proses pengembangan program karena memiliki fitur yang mendukung penulisan dan pengelolaan kode secara efisien *Visual Studio Code* mendukung berbagai bahasa pemrograman, di antaranya *JavaScript*, *HTML*, *CSS*, *PHP*, *Python*, *C++*, serta bahasa lainnya[12].



Gambar 2. 2 *Visual Studio Code*

### 2.2.3. Laravel

*Laravel* merupakan salah satu *framework* pengembangan web yang berbasis bahasa pemrograman *PHP* dan digunakan untuk

mendukung proses pembuatan aplikasi *website*. Framework ini dilengkapi dengan berbagai fitur modern yang membantu pengembang dalam membangun aplikasi secara lebih efektif dan efisien. Fitur-fitur tersebut meliputi *Bundles*, *Eloquent ORM* (Object-Relational Mapping), *Query Builder*, *Resource Controller*, *Blade*, *Migration*, *Middleware*, serta *Automatic Pagination*. Selain itu, *Laravel* memiliki sejumlah keunggulan, antara lain dukungan *Command Line Interface* (CLI) melalui *Artisan*, pemanfaatan *package manager PHP Composer*, penulisan kode program yang lebih ringkas, mudah dipahami, dan bersifat ekspresif[13].



Gambar 2. 3 *Laravel*

#### 2.2.4. Database

*Database* merupakan kumpulan data atau informasi yang disimpan secara terstruktur di dalam sistem komputer, sehingga dapat dikelola dan diakses oleh program komputer untuk keperluan pengambilan informasi. Istilah *basis data* berasal dari bidang ilmu komputer, dan pada awalnya digunakan untuk merujuk pada penyimpanan data secara elektronik. Namun, seiring perkembangan waktu, konsep basis data meluas hingga mencakup sistem pencatatan non-elektronik[14].



Gambar 2. 4 Database

#### 2.2.5. Mysql

*MySQL* merupakan salah satu sistem manajemen basis data server yang paling banyak digunakan. Sistem ini memanfaatkan bahasa *Structured Query Language* (SQL) sebagai sarana untuk mengakses dan mengelola basis data. *MySQL* tersedia dalam *lisensi Free and Open Source Software* (FOSS) *License Exception*, serta disediakan pula dalam versi berlisensi komersial[15].



Gambar 2. 5 Mysql

#### 2.2.6. Tailwind CSS

*Tailwind CSS* merupakan *framework CSS* yang memberikan kemudahan kepada pengembang dalam membuat antarmuka pengguna (user interface) dengan menggunakan kelas-kelas langsung

pada elemen *HTML*, sehingga dapat membantu proses perancangan tampilan website menjadi lebih cepat dan responsif[16].



Gambar 2. 6 *Tailwind CSS*

#### 2.2.7. **PHP**

*PHP* merupakan bahasa pemrograman yang bersifat multiplatform, sehingga dapat dijalankan pada berbagai jenis perangkat dan sistem operasi, seperti *Linux*, *Unix*, *macOS*, dan *Windows*. Selain itu, *PHP* dapat dieksekusi secara *runtime* melalui konsol serta memiliki kemampuan untuk menjalankan berbagai perintah sistem lainnya[17].

#### 2.2.8. **Website**

*Website* adalah kumpulan halaman informasi yang disajikan melalui jaringan internet, sehingga dapat diakses kapan saja dan dari mana saja selama pengguna memiliki koneksi internet. *Website* berfungsi sebagai media penyampaian informasi, komunikasi, maupun interaksi dengan pengguna. Dalam konteks pengembangan sistem informasi, pembuatan *website* merupakan bagian dari proses

pengembangan sistem (system development) yang mencakup perancangan, implementasi, pengujian, dan pemeliharaan agar *website* dapat berjalan secara optimal serta memenuhi kebutuhan penggunaannya[18].

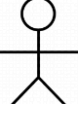
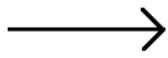
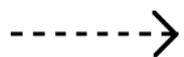
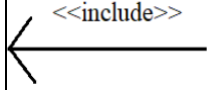
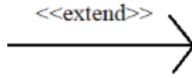
### 2.2.9. UML (Unified Modeling Language)

*Unified Modeling Language* (UML) merupakan bahasa pemodelan perangkat lunak yang telah distandardisasi dan digunakan sebagai sarana untuk menyusun cetak biru (blueprint) dalam pengembangan perangkat lunak (Pressman). *UML* dapat dimanfaatkan untuk melakukan visualisasi, spesifikasi, konstruksi, serta dokumentasi berbagai komponen atau bagian dari suatu sistem perangkat lunak[19]. Penggunaan *UML* membantu pengembang dan pemangku kepentingan dalam memahami struktur serta perilaku sistem secara lebih jelas sebelum tahap implementasi dilakukan. Beberapa diagram *UML* yang umum digunakan antara lain:

#### 1. *Use Case Diagram*

Diagram ini menggambarkan kumpulan *use case* beserta aktor yang terlibat di dalamnya. Diagram tersebut berperan penting dalam mengorganisasi serta memodelkan perilaku sistem yang diharapkan dan dibutuhkan oleh pengguna atau pihak manusia yang berinteraksi dengan sistem.

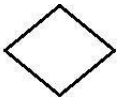
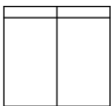
Tabel 2. 2 *UseCase Diagram*

| Simbol                                                                              | Nama               | Keterangan                                                                                                                     |
|-------------------------------------------------------------------------------------|--------------------|--------------------------------------------------------------------------------------------------------------------------------|
|    | <i>Actor</i>       | Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i> .                                |
|    | <i>Use case</i>    | Abstraksi dan interaksi antara sistem dan aktor.                                                                               |
|    | <i>Association</i> | Abstraksi dari penghubung antara aktor dan <i>use case</i> .                                                                   |
|    | Generalisasi       | Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i> .                                             |
|   | <i>include</i>     | Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.                      |
|  | <i>extend</i>      | Menunjukkan bahwa suatu <i>usecase</i> merupakan tambahan fungsional dari <i>usecase</i> lainnya jika suatu kondisi terpenuhi. |

## 2. Activity Diagram

*Activity Diagram* merupakan salah satu diagram dalam *UML* yang digunakan untuk menggambarkan perilaku dinamis suatu sistem atau bagian dari sistem. Diagram ini menunjukkan alur kontrol aktivitas melalui rangkaian aksi atau kegiatan yang dilakukan oleh sistem.

Tabel 2. 3 *Activity Diagram*

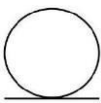
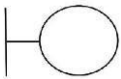
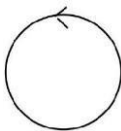
| Simbol                                                                              | Nama                             | Keterangan                                                                                                                                                                     |
|-------------------------------------------------------------------------------------|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <i>Initial Node</i>              | Sebuah diagram aktivitas memiliki sebuah status awal.                                                                                                                          |
|    | <i>Action</i>                    | State dari sistem yang mencerminkan eksekusi dari suatu aksi.                                                                                                                  |
|    | <i>Activity</i>                  | Memperlihatkan bagaimana masing masing kelas antarmuka saling berinteraksi satu sama lain.                                                                                     |
|    | <i>Decision</i>                  | Percabangan dimana ada pilihan aktivitas yang lebih dari satu.                                                                                                                 |
|   | Penggabungan<br>atau <i>Join</i> | Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.                                                                                            |
|  | <i>Activity Final Node</i>       | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.                                                                                     |
|  | <i>Vertical Swimlane</i>         | metode pemodelan proses yang membagi aktivitas berdasarkan pihak atau aktor yang bertanggung jawab, sehingga alur proses menjadi lebih jelas, terstruktur, dan mudah dipahami. |

### 3. *Sequence Diagram*

*Sequence Diagram* digunakan untuk menggambarkan interaksi antar objek di dalam maupun di sekitar sistem, termasuk pengguna dan komponen antarmuka, dalam bentuk pertukaran pesan (message) yang tersusun berdasarkan urutan waktu. Diagram ini umumnya dimanfaatkan untuk menjelaskan skenario

atau tahapan proses yang terjadi sebagai respons terhadap suatu peristiwa (event) tertentu hingga menghasilkan keluaran yang diharapkan.

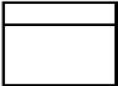
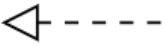
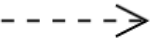
Tabel 2. 4 *Sequence Diagram*

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

#### 4. *Class Diagram*

*Class Diagram* merupakan diagram yang digunakan untuk menggambarkan struktur sistem, termasuk deskripsi kelas (class), paket (package), serta objek-objek beserta hubungan yang terjadi di antaranya, seperti pewarisan (inheritance), asosiasi, dan hubungan lainnya.

Tabel 2. 5 *Class Diagram*

| Simbol                                                                              | Nama                    | Keterangan                                                                                                                                   |
|-------------------------------------------------------------------------------------|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
|    | <i>Generalization</i>   | Hubungan dimana objek anak berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk.                                    |
|    | <i>Nary Association</i> | Upaya untuk menghindari asosiasi dengan lebih 2 objek.                                                                                       |
|    | <i>Class</i>            | Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.                                                                      |
|   | <i>Collaboration</i>    | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .                 |
|  | <i>Realization</i>      | Operasi yang benar-benar dilakukan oleh suatu objek.                                                                                         |
|  | <i>Dependency</i>       | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri. |
|  | <i>Association</i>      | Apa yang menghubungkan antara objek satu dengan objek lainnya.                                                                               |