

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Dalam penyusunan tugas akhir ini, penulis sedikit banyak terinspirasi dan referensi dari penelitian-penelitian sebelumnya yang berkaitan dengan sistem manajemen kunci pintu rumah otomatis berbasis *website*. Berikut ini penelitian terdahulu yang berhubungan dengan tugas akhir ini antara lain:

Penelitian yang dilakukan oleh Sucipto et al. (2024) berjudul "Penerapan Sistem Keamanan Otomatis Kunci Pintu Rumah Dengan *Microcontroller ESP32* Berbasis *Website*" mengembangkan sistem keamanan pintu otomatis yang menggabungkan *mikrokontroler ESP32* dengan *platform website*. Sistem ini memungkinkan pengguna untuk memantau dan mengendalikan akses pintu secara *remote* melalui perangkat seluler yang terhubung ke internet. Dengan integrasi logika fuzzy Mamdani, sistem ini dapat memberikan respons cepat terhadap situasi keamanan rumah, serta meningkatkan fleksibilitas akses dan pemantauan aktivitas rumah[3].

Penelitian yang dilakukan oleh Rahmadhan (2020). Yang berjudul "Rancang Bangun Akses Kunci Pintu Gerbang Indekos Menggunakan E-KTP (Elektronik Kartu Tanda Penduduk) Berbasis Mikrokontroler", penelitian ini bertujuan untuk merancang sistem keamanan pada pintu gerbang indekos. Penelitian ini menghasilkan suatu alat yang dapat

meningkatkan keamanan pada akses gerbang masuk kos dengan menggunakan E-KTP untuk akses masuknya. Alat ini dirancang layaknya fitur *NFC* (*Near Field Communication*), dengan memanfaatkan sensor *RFID* untuk memindai E-KTP, E-KTP tersebut sebagai alat ganti *NFC* untuk mengakses *Solenoid DoorLock* agar pintu gerbang kos dapat terbuka, terdapat Buzzer sebagai notifikasi peringatan dan *LCD* sebagai alat monitoring jika autentikasi tidak berhasil[4].

Penelitian yang dilakukan oleh Ulandari et al. (2023) dalam jurnal "Rancang Bangun *Prototype* Sistem Monitoring Keamanan Rumah Menggunakan *NodeMCU ESP32* dengan Multisensor Berbasis *Website*" memanfaatkan *NodeMCU ESP32* sebagai pusat kendali yang terintegrasi dengan berbagai sensor seperti sensor ultrasonik, PIR, api, dan gas. Sistem ini tidak hanya mendeteksi potensi ancaman keamanan di sekitar rumah tetapi juga memberikan notifikasi melalui aplikasi Telegram dan menyediakan akses monitoring melalui *website*, sehingga pengguna dapat dengan mudah memantau kondisi rumah secara real-time[5]

2.2 Landasan Teori

2.2.1 Sistem Monitoring

Monitoring adalah proses rutin pengumpulan data dan pengukuran kemajuan atas objektif program. Memantau perubahan yang fokus pada proses dan keluaran. Monitoring akan memberikan informasi tentang status dan kecenderungan bahwa

pengukuran dan evaluasi yang disediakan berulang kali dari waktu ke waktu, pemantauan umumnya dilakukan untuk tujuan tertentu, untuk memeriksa proses terhadap suatu objek atau untuk mengevaluasi kondisi atau kemajuan menuju tujuan hasil manajemen atas efek tindakan dari beberapa jenis tindakan untuk mempertahankan manajemen yang sedang berjalan[6]. Pemantauan yang dapat dijelaskan sebagai kesadaran (awareness) tentang apa yang ingin diketahui, pemantauan berkadar tingkat tinggi dilakukan agar dapat membuat pengukuran melalui waktu yang menunjukkan pergerakan ke arah tujuan atau malah menjauh dari tujuan (Saputro, 2018)[7].

2.2.2 Website

Website merupakan kumpulan informasi yang terdiri dari halaman *web* yang saling terhubung satu sama lain yang disediakan secara perorangan, kelompok, atau pun organisasi. Situs *web* yang baik menampilkan visual yang menarik dan berfungsi sesuai dengan kebutuhan pengguna[8]

2.2.3 PHP

PHP merupakan bahasa pemrograman berbasis *web* yang memiliki kemampuan untuk memproses data dinamis. *PHP* dikatakan sebagai sebuah *server-side embedded script language* artinya sintaks-sintaks dan perintah yang kita berikan akan sepenuhnya dijalankan oleh server tetapi disertakan pada halaman

HTML biasa. Aplikasi-aplikasi yang dibangun oleh *PHP* pada umumnya akan memberikan hasil pada *web browser* tetapi prosesnya secara keseluruhan dijalankan di server[9].



Gambar 2. 1 Logo PHP.(PHP Development Team. 2025).

Src: <https://www.php.net/images/logos/php-logo.svg>

2.2.4 *CodeIgniter*

CodeIgniter adalah komponen pemrograman yang telah siap digunakan kapan saja, sehingga memudahkan programmer tidak membuat skrip yang sama dalam tugas yang sama,[10] *framework php* yang bersifat *open source* dan menggunakan metode MVC (*Model, View, Controller*) untuk memudahkan developer atau programmer dalam membangun sebuah aplikasi berbasis web tanpa harus membuatnya dari awal.



Gambar 2. 2 Logo CodeIgniter. Src:

https://codeigniter.com/user_guide/images/ci-logo.png

2.2.5 *MySQL*

Structured Query Language yang sering dikenal dengan *MySQL*, adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Dimana setiap orang bebas untuk menggunakan *MySQL*, namun tidak boleh dijadikan produk turunan yang bersifat komersial. *MySQL* sebenarnya merupakan turunan salah satu konsep utama dalam *database* sejak lama, yaitu *SQL (Structured Query Language)*[11].



Gambar 2. 3 Logo MySQL. Src:

<https://www.mysql.com/common/logos/mysql-logo-72x30.png>

2.2.6 *Visual Studio Code*

Visual Visual Studio Code (VS Code) adalah editor kode sumber (*source code editor*) yang dikembangkan oleh *Microsoft*. *VS Code* banyak digunakan oleh programmer untuk menulis, mengedit, dan menjalankan berbagai bahasa pemrograman seperti *Python*, *JavaScript*, *PHP*, *Java*, *C++*, dan lainnya[12]



Gambar 2. 4 Logo Visual Studio Code

Src: <https://share.google/images/gSP4JVOQfpuM5a1SK>

2.2.7 *Database*

Database atau basis data adalah kumpulan data yang terstruktur dan terorganisir secara sistematis agar dapat diakses, dikelola, dan diperbarui dengan mudah. *Database* digunakan untuk menyimpan informasi dalam berbagai bentuk seperti data pengguna, transaksi, produk, atau informasi lainnya[13].

2.2.8 *Java Script*

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



Gambar 2. 5 Logo Java Script

Src: <https://share.google/images/ltFrhOUAgcOvMLofT>

2.2.9 UML

Unified Modelling Language atau yang disingkat *UML*, adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan *desain*, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Sukanto & Shalahudin, 2018)[14]. Dalam proyek berorientasi objek, keempat teknik *UML* ini sangat mendominasi penggunaannya.

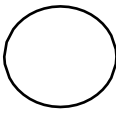
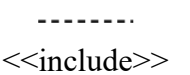
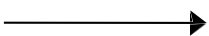
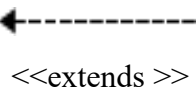
Dalam Perancangan sistem terdapat *UML* yang sering digunakan sebagai berikut:

1. *Use Case Diagram*

Use Case Diagram adalah deskripsi tentang bagaimana pengguna (aktor) berinteraksi dengan sistem untuk mencapai suatu tujuan tertentu. Dalam pengembangan perangkat lunak,

Use Case digunakan untuk memahami kebutuhan sistem dan menggambarkan bagaimana suatu fitur akan digunakan[15].

Tabel 2. 1 *Use Case Diagram*

No	Simbol	Keterangan
1.		Use case merupakan deskripsi fungsional yang telah disediakan oleh sistem sebagai entitas yang menghasilkan hasil yang terukur untuk suatu <i>actor</i> .
2.		<i>Actor</i> merupakan himpunan peran untuk berinteraksi dengan <i>use case</i>
3.		<i>Association</i> merupakan garis yang menghubungkan objek satu dengan objek yang lain
4.		<i>Include</i> merupakan gambaran jika <i>usecase</i> dipanggil oleh <i>usecase</i> lain.
5.		<i>Dependecy</i> merupakan garis panah yang menunjukan jika <i>actor</i> berinteraksi secara pasif
6.		<i>Extend</i> merupakan gambar jika memperluas <i>usecase</i> target.

2. Activity Diagram

Activity Diagram atau Diagram aktivitas merupakan diagram yang bersifat statis, yang menggambarkan aktivitas dari suatu sistem bisnis.

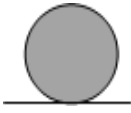
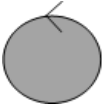
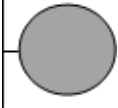
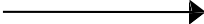
Tabel 2. 2 *Activity Diagram*

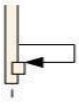
No	Simbol	Keterangan
1.		<i>End point</i> atau <i>final Node</i> merupakan gambaran akhir dari suatu aktivitas
2.		<i>Start Point</i> merupakan awal dari suatu aktivitas yang peletakkannya pada pojok kiri atas
3.		<i>Fork</i> atau <i>join</i> digunakan untuk memarallelkan suatu kegiatan atau penggabungan 2 kegiatan parallel menjadi satu
4.		<i>Activity</i> merupakan gambaran dari suatu proses
5.		<i>Decision</i> merupakan pilihan pengambilan suatu keputusan <i>false or true</i> .

3. *Sequence Diagram*

Sequence Diagram atau Diagram Urutan mendeskripsikan diagram interaksi yang mengirimkan pesan dan diterima antar objek.

Tabel 2. 3 *Sequence Diagram*

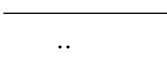
No	Simbol	Keterangan
1.		<i>Entity Class</i> merupakan bagian sistem yang membentuk deskripsi awal sistem dan berisi kumpulan kelas dalam bentuk entitas yang mendasari untuk membuat <i>database</i>
2.		<i>ControlClass</i> merupakan gambaran penghubung antara <i>Boundaryclass</i> dengan suatu <i>table</i>
3.		<i>Boundary Class</i> merupakan gambaran dari penggambaran <i>table</i>
4.		Pesen atau <i>message</i> menunjukkan pengiriman pesen antar <i>class</i>

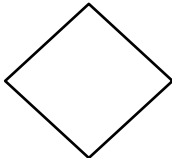
No	Simbol	Keterangan
5.		<i>Self message</i> menunjukkan pengiriman suatu pesan yang akan dikirim ke objek itu sendiri.
6.		<i>Activation</i> menggambarkan suatu objek yang melakukan sebuah aksi/eksekusi operasi
7.		<i>Lifeline</i> garis titik yang terhubung ke objek sepanjang garis <i>lifeline</i> memiliki aktivitas

4. *Class Diagram*

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

Tabel 2. 4 Class Diagram

No	Simbol	Keterangan
1.		Generallization, merupakan dimana objek descendent membagikan perilaku dan struktur data objek induknya

No	Simbol	Keterangan
2.		Class, adalah kumpulan objek yang saling berbagi.
3.		Nary Association, digunakan untuk asosiasi terhindar dengan objek lainya.
4.		Collaboration merupakan deskripsi urutan aksi yang ditampilkan suatu sistem yang memiliki konsekuensi terukur bagi actor
5.		Merupakan operasi yang valid dilakukan oleh suatu objek
6.		Merupakan garis penah yang menunjukkan jika actor berinteraksi secara pasif
7.		Association merupakan garis yang menghubungkan objek satu dengan objek