

BAB II

TINJAUAN PUSTAKA

2.1. Penelitian Terkait

Penelitian yang telah dilakukan oleh Muhamad Nizam, Haris Yuana, dan Zunita Wulansari, 2022, yang berjudul “Mikrokontroler ESP 32 Sebagai Alat Monitoring Pintu Berbasis *Web*” Pada penelitian ini peneliti menggunakan alat monitoring pada pintu menggunakan ESP32 dengan tujuan dapat menjadi solusi dalam upaya pemantauan dan sistem keamanan dalam suatu ruangan. Penelitian ini menggunakan metode pengumpulan data studi literatur dimana peneliti mengambil bahan penelitian dari jurnal ilmiah. Alat ini menggunakan ESP32 *Magnetic Door Switch, Buzzer, Power Supply*, sedangkan software terdiri dari *Arduino IDE, XAMPP, dan Telegram*. Berdasarkan pengamatan diperoleh hasil pengujian alat bekerja dengan baik mulai dari pembacaan sensor sampai pengiriman notifikasi yang berupa pesan ke pengguna telegram. Dari penelitian ini dapat digunakan sebagai alat monitoring untuk tujuan kemanan pada ruangan atau kamar[3].

Penelitian yang telah dilakukan oleh Nelly Khairani Daulay dan M Nur Alamsyah, 2019, yang berjudul “Monitoring Sistem Keamanan Pintu Menggunakan RFID dan *Fingerprint* Berbasis *Web* dan *Database*” Dalam penelitian ini, kunci pintu mekanik akan tergantikan dengan penggunaan kartu RFID dan pengenalan biometrik, dalam hal ini adalah

pengenalan sidik jari. Dengan Menggunakan kartu RFID dan sidik jari sebagai pengganti kunci pintu. Kartu RFID dan sidik jari akan dibaca oleh modul mikrokontroler untuk dapat dicocokkan dengan *database*. Informasi kehadiran dikirimkan ke *web server* sehingga dapat diakses dari mana saja. Untuk menggerakkan pintu menggunakan solenoid *doorlock*. Jika sidik jari dan kartu tidak cocok dengan data yang disimpan dalam program, tampilan layar sidik jari akan muncul kartu anda salah. Jika sidik jari dan kartu cocok dengan data yang disimpan dalam program, pintu akan terbuka[4].

Penelitian yang telah dilakukan oleh Kristomson H, Rosalia H Subrata, dan Ferrianto Gozali 2018, yang berjudul “Sistem Keamanan Ruangan Berbasis *Internet of Things* Dengan Menggunakan Aplikasi *Android*” Skripsi ini akan dirancang suatu sistem keamanan ruangan berbasis IoT dengan menggunakan Aplikasi *Android*, sistem ini bekerja menggunakan beberapa module diantaranya modul RFID, modul kamera, dan solenoid *door lock*, Kartu RFID pada modul RFID berfungsi sebagai identifikasi orang yang ingin masuk ke dalam ruangan tersebut. Data yang diperoleh dari kedua modul tersebut nantinya akan dikirimkan ke dalam Aplikasi *Android*. Admin dari ruangan tersebut dapat melakukan proses verifikasi untuk mengizinkan atau menolak orang yang ingin memasuki ruangan tersebut. Hasil pengujian sistem ini mungkin dapat bekerja dengan baik pada saat kecepatan *internet* akses

cepat dan stabil, namun kurang optimal pada saat kecepatan *internet* akses lambat.[5].

2.2. Landasan Teori

2.2.1. *Visual Studio Code*

Visual Studio Code ialah sebuah kode editor yang tidak berat, *Visual Code* dapat berjalan di *desktop* dan tersedia untuk *Windows*, *macOS*, dan *Linux*. Ini memiliki dukungan bawaan untuk *JavaScript*, *TypeScript* dan *Node.js*. *Visual Studio Code* mencakup ekosistem ekstensi yang kaya untuk bahasa lain seperti *C++*, *C#*, *Java*, *Python*, *PHP*, *Go* dan *runtime* seperti *.Net* dan *Unity*.[6].



Gambar 2. 1 Logo *Visual Studio Code*

2.2.2. Sistem Informasi

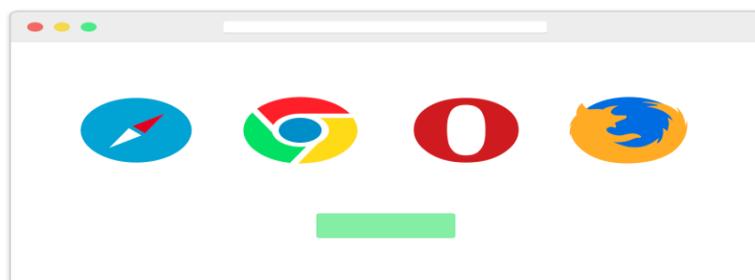
Sistem informasi adalah suatu proses pengumpulan data (*input*), pengolahan, penyimpanan, analisis, dan penyebaran informasi (*output*) untuk tujuan tertentu. Sederhanannya sebuah sistem informasi akan menerima *input* berupa data, data ini akan disimpan (*storage*) dan kemudian akan diproses untuk menghasilkan *output*[7].



Gambar 2. 2 Sistem Informasi

2.2.3. *Web Browser*

Menurut Shalahudin dan Rosa, browser adalah perangkat lunak yang berjalan di komputer pengguna yang menampilkan dokumen atau data *web* yang diambil dari *server web*. *Browser* adalah perantara yang paling umum digunakan antara pengguna dan *server web*. *Server web* sendiri adalah kumpulan jaringan yang berisi dokumen dan tersambung atau saling terhubung satu dengan yang lain, yang dikenal sebagai *www (World Wide Web)*[8].

Gambar 2. 3 *Web Browser*

2.2.4. *Website*

Website atau situs *web* dapat didefinisikan sebagai kumpulan halaman-halaman yang menampilkan teks, gambar bergerak, gambar bergerak, animasi, suara, atau kombinasi dari

semua itu, bersifat statis maupun dinamis yang saling terhubung, masing-masing dihubungkan dengan jaringan-jaringan halaman[9].



Gambar 2. 4 Website

2.2.5. *Bootstrap*

Bootstrap adalah *framework* yang bagus dan luar biasa yang mengedepankan tampilan yang bisa digunakan di berbagai *device*. Guna mempercepat dan mempermudah pengembangan *website*. *Bootstrap* adalah kerangka kerja untuk membuat desain *web* responsif. Artinya tampilan *web* yang dibuat dengan *bootstrap* akan menyesuaikan ukuran layar dari *browser* yang kita gunakan baik *desktop*, *tablet* ataupun *mobile device*. Dengan *bootstrap* kita bisa membangun *web* dinamis atau statis[10].



Gambar 2. 5 Logo *Bootstrap*

2.2.6. XAMPP

XAMPP adalah paket PHP *open source*, XAMPP menggabungkan beberapa paket perangkat lunak yang berbeda ke dalam satu paket. Paket yang disediakan antara lain *Apache*, *MySQL*, *PHP*, *FilZilla FTP Server*, *PHP MyAdmin* dan lain-lain[11].



Gambar 2. 6 Logo XAMPP

2.2.7. MySQL

MySQL merupakan *software* RDMS atau *server database* yang dapat mengelola database dengan cepat, dapat menampung data dalam jumlah yang besar, dapat diakses oleh banyak *user* (*multi-user*), dan melakukan suatu proses secara sinkron atau bersamaan (*multi-user threaded*)[12].



Gambar 2. 7 Logo MySQL

2.2.8. Database

Database (basis data) merupakan pengorganisasian sekumpulan data yang saling berkaitan sehingga memudahkan aktivitas untuk memperoleh informasi basis data (*database*) dimaksudkan untuk mengatasi problem pada sistem yang memakai pendekatan berbasis berkas. Untuk mengelola suatu basis data (*database*) diperlukan perangkat lunak yang disebut *Database Management System* (DBMS)[13].



Gambar 2. 8 Ilustrasi *Database*

2.2.9. HTML

HTML (*Hyper Text Markup Language*) adalah sebuah bahasa *formatting* yang digunakan untuk membuat sebuah halaman *website*. Dalam dunia pemrograman berbasis *website*. HTML menjadi pondasi dasar pada halaman *website*. File HTML disimpan dengan *tag .html (dot html)*. File tersebut dapat diakses menggunakan *web browser*[14].



Gambar 2. 9 Logo HTML

2.2.10. PHP (*Hypertext Preprocessor*)

PHP (*Hypertext Preprocessor*) merupakan bahasa pemrograman berbasis *website* bersifat *server-side*, yang berarti kode PHP diproses seluruhnya di dalam *web server*. PHP adalah *server-side embedded script language* yang berarti bahwa semua *syntax* tertulis dan perintah program dijalankan sepenuhnya oleh *server*, tetapi dapat ditambahkan ke halaman HTML biasa[15].



Gambar 2. 10 Logo PHP

2.2.11. RFID

RFID (*Radio Frequency Identification*) adalah kombinasi antara frekuensi radio yang berbasis teknologi dan teknologi *microchip*. Informasi terkandung di dalam *tag microchip* dan ditempelkan pada bahan Pustaka dapat dibaca menggunakan

teknologi frekuensi radio. Sebuah alat pembaca (alias sensor, pemindai, atau *interrogator*) mencari *antenna* pada tag dan mengambil informasi dari microchip dalam perangkat RFID. *Chip* RFID menjadi bagian yang sangat penting, karena *chip* tersebut dapat ditambahkan pada setiap jenis dokumen dan dapat dibaca dan diperbarui dari kejauhan[16].



Gambar 2. 11 RFID dan RFID Reader

2.2.12. CodeIgniter

CodeIgniter adalah sebuah *open source* berupa *framework* PHP dengan *model* MVC (*Model, View, Controller*) untuk membangun *website* dinamis dengan cepat dan mudah. *CodeIgniter* ini memungkinkan para pengembang untuk menggunakan *framework* secara parsial atau secara keseluruhan. Artinya bahwa *CodeIgniter* masih memberi kebebasan kepada para pengembang untuk menulis bagian-bagian kode tertentu di dalam aplikasi menggunakan cara konvensional atau *syntax* umum didalam PHP, tidak harus menggunakan aturan penulisan kode di *CodeIgniter*[17].

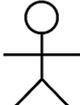
2.2.13. UML (*Unified Modelling Language*)

UML (*Unified Modelling Language*) merupakan peralatan yang dapat digunakan atau dipilih oleh analis untuk merancang sistem yang nantinya akan diterapkan oleh pengembangan *website*. Penggunaan UML ini diperlukan dan merupakan hal yang sangat fundamental untuk digunakan pada bidang *website*[18]. Diagram yang digunakan UML adalah:

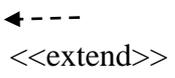
1. *Use Case Diagram*

Use Case Diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Sebuah *Use Case* mempresentasikan sebuah interaksi antara *actor* dengan sistem.

Tabel 2. 1 *Use Case diagram*

No	Simbol	Pengertian	Keterangan
1.		<i>Use Case</i>	Menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja.
2.		<i>Actor/Aktor</i>	<i>Abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga

No	Simbol	Pengertian	Keterangan
			kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>Use Case</i> , tetapi tidak memiliki kontrol terhadap <i>use case</i> .
3.	_____	<i>Association/</i> <i>Asosiasi</i>	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.
4.	→	<i>Generalization/</i> <i>Generalisasi</i>	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk Mengindikasikan bila aktor berinteraksi secara pasif dengan sistem
5.	---> <<include>>	<i>Include/Uses</i>	Merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.

No	Simbol	Pengertian	Keterangan
6.		<i>Extend/Ekstensi</i>	Merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

2. *Activity Diagram* (Diagram Aktivitas)

Activity Diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir.

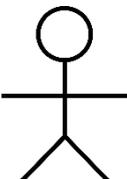
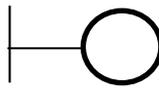
Tabel 2. 2 *Activity Diagram*

No	Simbol	Pengertian	Keterangan
1.		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2.		<i>Activity Final Mode</i>	bagaimana objek dibentuk dan dihancurkan.
3.		<i>Initial Node</i>	bagaimana objek dibentuk atau diawali.
4.		<i>Fork Node</i>	Satu aliran yang tahap tertentu berubah menjadi beberapa aliran.

3. *Sequence Diagram* (Diagram Urutan)

Sequence Diagram menggambarkan jalannya urutan interaksi dari objek di dalam sekitar sistem (termasuk pengguna, *display*, dan sebagainya).

Tabel 2. 3 *Sequence Diagram*

No	Simbol	Pengertian	Keterangan
1.		Aktor	Menggambarkan orang yang sedang berinteraksi dengan sistem.
2.		<i>Entity Class</i>	Menggambarkan penghubung antara boundary dengan tabel.
3.		<i>Asynchronous Message</i>	Menggambarkan pengiriman pesan.
4.		<i>Control Class</i>	Menggambarkan penghubung antara <i>boundary</i> dengan tabel.
5.		<i>Boundary Class</i>	Menggambarkan sebuah gambaran dari <i>foem</i> .

No	Simbol	Pengertian	Keterangan
6.		<i>A Focus Of Control & A Life Line</i>	Menggambarkan penghubung antara <i>boundary</i> dengan tabel.

4. *Class Diagram* (Diagram Kelas)

Class Diagram adalah spesifikasi yang ketika diwujudkan, menciptakan sebuah objek dan merupakan inti dari pengembangan desain berorientasi objek. *Class Diagram* menggambarkan struktur dan deskripsi kelas, paket, dan objek serta hubungannya, seperti keamanan, warisan, asosiasi, dan lain-lain.