

BAB II

TINJAUAN PUSTAKA

2.1. Penelitian terkait

Penelitian terdahulu yang berjudul “Rancang Bangun Sistem Keamanan Kunci Kontak Sepeda Motor Berbasis Android” merancang sistem keamanan kendaraan menggunakan mikrokontroler Arduino Uno, model Bluetooth HC-06, buzzer, dan aplikasi Android. Sistem ini memungkinkan pengguna mengendalikan alarm dan pengapian kendaraan secara nirkabel melalui koneksi Bluetooth. Namun, sistem memiliki keterbatasan pada jangkauan komunikasi yang hanya mencapai sekitar 10 meter. Ketika jarak antara smartphone dan kendaraan melebihi batas tersebut, koneksi tidak dapat terjalin sehingga fungsi pengendalian jarak jauh tidak dapat digunakan[8].

Penelitian terdahulu yang berjudul “Sistem Keamanan Kendaraan Bermotor Berbasis Bluetooth Menggunakan Arduino Uno” merancang sistem keamanan sepeda motor berbasis Bluetooth menggunakan mikrokontroler Arduino Uno. Sistem ini menghubungkan smartphone dan rangkaian elektronik yang mengatur relay untuk mengaktifkan atau memutus sistem pengapian motor. Namun, sistem memiliki keterbatasan jarak komunikasi Bluetooth relatif pendek sehingga pengendalian jarak jauh tidak dapat dilakukan dan koneksi rentan terganggu oleh jarak maupun adanya penghalang[9].

Penelitian terdahulu yang berjudul “Rancang Bangun Sistem Kontrol dan Monitoring Keamanan Sepeda Motor Berbasis IoT dengan Modul GPS Neo-6M dan Sensor Getar SW-420” merancang sistem keamanan sepeda motor berbasis IoT yang menggunakan ESP32 WROOM, modul GPS Neo-6M, dan sensor getar SW-420. Sistem ini mampu mendeteksi gangguan fisik, pemantauan lokasi secara real-time, serta mengirimkan notifikasi melalui aplikasi mobile RIDER. Namun, performa pengiriman data masih dipengaruhi kondisi cuaca dan kualitas jaringan internet. Selain itu, akurasi GPS memiliki galat sekitar 24–27 meter dan delay sistem rata-rata sekitar 1 detik[10].

2.2. Landasan teori

2.2.1. Sistem Kontrol

Sistem kontrol merupakan mekanisme yang mengatur suatu proses agar berjalan sesuai nilai atau kondisi yang telah ditentukan[11]. Sistem kontrol biasanya terdiri dari sensor sebagai pendeteksi kondisi, mikrokontroler sebagai pengolah data, dan aktuator sebagai pelaksana. Pada sistem kontrol ini menghubungkan aplikasi untuk mengirimkan perintah, ESP32 sebagai pusat pengolahan informasi, dan aktuator berupa relay serta sirene yang berperan sebagai hasil dari aksi. Sistem ini dirancang untuk mengaktifkan mode siaga (Lock), menonaktifkan sistem keamanan (Unlock), serta memutus aliran pengapian mesin secara darurat melalui aplikasi saat bahaya terdeteksi.

2.2.2. Sistem Monitoring

Sistem Monitoring merupakan proses pengamatan data terhadap suatu situasi atau keadaan untuk mendukung pengambilan keputusan berdasarkan hasil yang diperoleh[12]. Dalam sistem ini, pemantauan digunakan untuk memantau lokasi sepeda motor dalam waktu nyata melalui data dari modul GPS yang ditampilkan pada peta digital di aplikasi Android. Selain itu, Sistem juga mencatat riwayat insiden mencurigakan beserta informasi waktu dan lokasi koordinat yang disimpan dalam basis data.

2.2.3. Smartphone

Smartphone adalah perangkat seluler yang memiliki kemampuan komunikasi yang mendukung berbagai layanan digital. Dengan tersedianya layanan akses data, memungkinkan smartphone terhubung ke internet kapan saja dan di mana saja. Dalam Internet of Things (IoT), smartphone berperan sebagai antarmuka pengguna untuk memantau data dari perangkat keras secara real-time serta mengirimkan instruksi kontrol melalui jaringan internet[13].

2.2.4. Android

Android adalah sistem operasi turunan Linux yang dirancang untuk perangkat mobile, termasuk smartphone dan tablet[14]. Android menyediakan Application Programming Interface (API) yang mendukung integrasi dengan berbagai modul hardware eksternal, termasuk perangkat IoT. Hal ini menjadikan Android cocok

digunakan sebagai platform kendali sistem keamanan elektronik. Selain itu, Android mendukung berbagai protokol komunikasi nirkabel, sehingga populer dalam pengembangan aplikasi kontrol.



Gambar 2. 1 Android

(sumber: <https://id.androidayuda.com>)

2.2.5. Visual Studio Code

Visual Studio Code (VS Code) adalah IDE yang dikembangkan oleh Microsoft dan dikenal karena ringan dan memiliki kinerja yang cepat, sehingga mampu mengelola proyek berukuran besar tanpa memerlukan banyak sumber daya sistem[15]. VS Code mendukung berbagai ekstensi yang membantu proses pengembangan aplikasi, termasuk framework Flutter dan bahasa pemrograman Dart, yang memudahkan pengembang dalam penulisan kode, proses debugging, serta integrasi dengan layanan backend seperti Firebase.



Gambar 2. 2 Visual Studio Code

(sumber: <https://apps.microsoft.com>)

2.2.6. Flutter

Flutter adalah framework antarmuka open-source dari Google untuk membangun aplikasi mobile lintas platform[16]. Flutter menggunakan bahasa pemrograman Dart, yang mendukung performa tinggi dan kemampuan just-in-time (JIT) dan ahead-of-time (AOT) compilation sehingga memungkinkan proses pengembangan yang lebih cepat melalui hot reload serta performa native ketika aplikasi dijalankan.



Gambar 2. 3 Flutter

(sumber: <https://mobiverry.com>)

2.2.7. Firebase Realtime

Firebase Real-Time Database merupakan layanan basis data nonSQL yang mampu menyimpan berbagai jenis data seperti String, Long, dan Boolean[17]. Firebase menyimpan data dalam format JSON (JavaScript Object Notation) yang lebih fleksibel untuk

pengembangan aplikasi mobile. Setiap data koordinat GPS atau status sensor yang dikirim oleh ESP32 melalui jaringan GPRS disimpan di Firebase, kemudian diperbarui secara otomatis pada aplikasi Android. Penggunaan layanan ini mendukung pertukaran data secara real-time sehingga notifikasi peringatan dan perintah kendali dapat di proses dengan cepat.



Gambar 2. 4 Firebase

(sumber: <https://www.exabytes.co.id>)

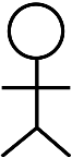
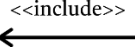
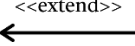
2.2.8. UML (United Modelling Language)

UML (Unified Modeling Language) merupakan bahasa pemodelan visual standar yang digunakan untuk merancang, menggambarkan, mengembangkan, dan mendokumentasikan sistem perangkat lunak berorientasi objek, mencakup aspek struktur maupun perilakunya[18].

1. Use case Diagram

Pada bagian ini Use Case Diagram adalah diagram yang menampilkan interaksi sistem dan pengguna. Diagram ini menunjukkan layanan atau fungsi yang tersedia dalam sistem serta pihak yang terlibat dalam memanfaatkannya.

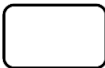
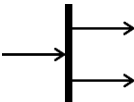
Tabel 2. 1 Use Case Diagram

No	Gambar	Simbol	Keterangan
1.		Aktor	Mewakili peran manusia, sistem lain, atau perangkat saat berinteraksi dengan use case.
2.		Use Case	Abstraksi dan interaksi antara <i>system</i> dan actor.
3.		<i>Association</i>	Abstrak dari penghubung antar <i>actor</i> dengan <i>use case</i>
4.		Generalisasi	Menjelaskan bentuk khusus dari seorang aktor agar dapat terlibat dalam sebuah use case.
5.		<i>Include</i>	Menjelaskan bahwa sebuah use case sepenuhnya merupakan bagian dari fungsionalitas use case lain.
6.		<i>Extend</i>	Menggambarkan bahwa sebuah use case akan menambahkan fungsi tertentu pada use case lain apabila kondisi tertentu terpenuhi.

2. Activity Diagram

Pada bagian ini Activity Diagram memvisualisasikan alur kegiatan dalam suatu sistem atau proses bisnis, menunjukkan urutan tahapan yang berlangsung dari awal hingga akhir.

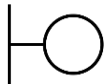
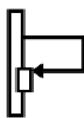
Tabel 2. 2 Activity Diagram

No	Gambar	Simbol	Keterangan
1.		Initial node	Menandai titik dimulainya suatu aktivitas yang akan dilaksanakan.
2.		Activity	Menunjukkan cara setiap kelas antarmuka saling berinteraksi satu sama lain.
3.		Decision	Menjelaskan pilihan atau langkah yang perlu ditentukan ketika berada pada kondisi tertentu.
4.		Fork node	Satu alur proses pada suatu tahap yang kemudian bercabang menjadi beberapa alur.
5.		Fork Join Node	Berfungsi untuk menunjukkan aktivitas yang berlangsung secara paralel atau untuk menggabungkan dua aktivitas paralel menjadi satu alur.
6.		Activity Final Node	Menandai titik berakhirnya aktivitas yang telah dilaksanakan.

3. Sequence Diagram

Pada bagian ini Sequence Diagram mempresentasikan interaksi antarobjek dalam suatu use case dengan menampilkan rentang waktu keberadaan objek serta aliran pesan yang dikirim dan diterimanya.

Tabel 2. 3 Sequence Diagram

No	Gambar	Simbol	Keterangan
1.		Entity Class	Menunjukkan titik awal sistem yang menjadi dasar dalam penyusunan basis data.
2.		Boundary Class	Menunjukkan interaksi atau pertukaran informasi antara sistem, seperti melalui form.
3.		Control Class	Berfungsi sebagai penghubung antara boundary dan tabel.
4.		Recursive	Mengirimkan pesan yang ditujukan kembali ke dirinya sendiri.
5.		Activation	Menggambarkan lamanya waktu Suatu proses dijalankan.
6.		Life Line	Mengaitkan setiap objek yang memiliki aktivitas
7.		Message	Berfungsi untuk mengirim pesan ke setiap kelas.

4. Class Diagram

Class Diagram merupakan diagram yang menggambarkan struktur sistem melalui kumpulan kelas beserta hubungan antar kelas yang ada di dalamnya. Diagram ini menjelaskan atribut, operasi (method), serta peran masing-masing kelas dalam membentuk perilaku sistem. Selain itu, relasi antar kelas disertai informasi mengenai jumlah keterlibatan elemen dalam hubungan tersebut yang dinyatakan melalui multiplicity atau cardinality.

Tabel 2. 4 Class Diagram

No	Gambar	Simbol	Keterangan
1.		Entity Class	Menunjukkan titik awal sistem yang menjadi dasar pembuatan basis data.
2.		Boundary Class	Menunjukkan interaksi antara sistem, misalnya melalui form.
3.		Control Class	Berperan sebagai penghubung antara boundary dan tabel.
4.		Recursive	Mengirim pesan yang ditujukan kembali kepada dirinya sendiri.
5.		Activation	Mewakili proses durasi aktivasi sebuah proses
6.		Life Line	Menghubungkan setiap objek yang memiliki aktivasi.