

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian terkait digunakan sebagai dasar dalam pengembangan sistem *monitoring smartlocker* berbasis Internet of Things (*IoT*). Penelitian oleh Pradana dan Wiharto (2020) membahas rancang bangun *smartlocker* menggunakan *RFID* berbasis *Arduino Uno*. Sistem tersebut mampu meningkatkan keamanan serta melakukan pencatatan akses secara digital secara otomatis[7]. Penelitian ini menggunakan *Arduino Uno* dan modul *RFID* sebagai metode autentikasi pengguna. Hasil penelitian menunjukkan bahwa sistem dapat merekam aktivitas akses pengguna secara otomatis sehingga dapat meningkatkan keamanan penggunaan loker. Penelitian tersebut menjadi referensi dalam penerapan sistem autentikasi pada *smartlocker*.

Penelitian selanjutnya oleh Syahputra dan Ananda (2022) merancang *smartlocker* berbasis *fingerprint* menggunakan *ESP32*. Berdasarkan hasil penelitian, autentikasi sidik jari dinilai lebih aman karena bersifat unik dan tidak dapat dipindah tangankan seperti kartu atau PIN[6].

Penelitian oleh Fauzi dan Minggu (2024) mengembangkan sistem monitoring berbasis *IoT* dengan memanfaatkan *Firestore* sebagai *database realtime*. Sistem yang dibuat mampu mengirimkan data aktivitas secara

langsung ke aplikasi sehingga pengguna dapat melakukan pemantauan dari jarak jauh[8].

Penelitian yang dilakukan oleh Mardian dkk. (2022) membahas rancang bangun kunci loker otomatis berbasis *Raspberry Pi* dan *RFID*[9]. Sistem tersebut dirancang untuk meningkatkan efisiensi dan keamanan penggunaan loker melalui pengendalian akses otomatis. Penelitian ini menjadi referensi dalam mekanisme pengendalian akses pada sistem *smartlocker*.

Selain itu, penelitian oleh Zhang et al. (2021) menunjukkan bahwa penerapan *IoT* dalam sistem keamanan mampu meningkatkan efisiensi operasional dan akurasi *monitoring* karena perangkat dapat berkomunikasi secara otomatis melalui jaringan internet[10].

Penelitian lainnya dilakukan oleh Prayoga (2023) mengenai sistem *monitoring* dan pengelolaan berbasis *IoT* dan *cloud* menggunakan *Firebase*[11].

Berdasarkan penelitian tersebut, sistem *smartlocker* terus berkembang dari sistem konvensional menjadi sistem berbasis *IoT* yang terintegrasi dengan aplikasi *mobile* dan layanan *cloud*.

2.2 Landasan Teori

2.2.1 *Internet of Things (IoT)*

Internet of Things (IoT) merupakan konsep yang memungkinkan perangkat fisik terhubung ke jaringan internet untuk saling bertukar data tanpa campur tangan manusia secara langsung[1].

Dalam penerapannya, *IoT* memungkinkan perangkat seperti mikrokontroler untuk mengirimkan data secara *realtime* ke server. Hal ini menjadikan sistem *monitoring* dapat dilakukan secara jarak jauh dan terpusat, sehingga meningkatkan efisiensi serta kemudahan dalam pengawasan sistem[10].



Gambar 2. 1 *Internet of Things (IoT)*

Sumber: <https://kineis.com/en/iot-explained-simply/>

2.2.2 *Smartlocker*

Smartlocker merupakan sistem loker modern yang menggunakan teknologi elektronik sebagai pengganti kunci manual. Sistem ini dirancang untuk meningkatkan keamanan serta mempermudah pengelolaan akses pengguna[2].

Smartlocker merupakan sistem loker cerdas yang menggunakan perangkat elektronik seperti mikrokontroler, sensor autentikasi, dan

aktuator untuk mengelola akses secara otomatis tanpa menggunakan kunci manual. Sistem ini menawarkan tingkat keamanan yang lebih tinggi karena proses autentikasi dikendalikan secara digital sehingga mengurangi risiko kehilangan atau duplikasi kunci.

Tidak seperti loker konvensional, *smartlocker* mampu melakukan:

1. autentikasi pengguna (/ *RFID* / *QR code*)
2. pencatatan aktivitas penggunaan (*logging*)
3. monitoring status loker
4. integrasi *IoT* untuk kontrol jarak jauh dan notifikasi

Dengan adanya *smartlocker*, setiap aktivitas penggunaan loker dapat tercatat secara otomatis dan status loker dapat diketahui secara realtime, baik oleh pengguna maupun administrator[5].



Gambar 2. 2 *Smartlocker*

Sumber: <https://www.marcopolosicurezza.it/>

2.2.3 Sistem Autentikasi Sidik Jari

Autentikasi sidik jari adalah metode identifikasi biometrik yang memanfaatkan pola unik pada sidik jari manusia sebagai identitas pengguna. Setiap individu memiliki pola sidik jari yang berbeda, sehingga metode ini memiliki tingkat keamanan yang tinggi[4].

Penggunaan *fingerprint* dalam sistem keamanan dinilai lebih efektif dibandingkan metode konvensional seperti PIN atau kartu akses, karena tidak dapat dipinjamkan maupun diduplikasi dengan mudah[3].



Gambar 2. 3 Sistem Autentikasi Sidik Jari

Sumber : https://idn.freepik.com/foto-premium/login-pengusaha-dengan-teknologi-pemindaian-sidik-jari-sidik-jari-untuk-mengidentifikasi-pribadi-sistem-keamanan_6167114.htm

2.2.4 Flutter sebagai Framework Mobile

Flutter merupakan *framework open-source* yang digunakan untuk mengembangkan aplikasi mobile dengan satu basis kode. *Framework* ini menawarkan performa yang baik serta tampilan antarmuka yang responsif dan menarik[12].

Selain itu, Flutter mendukung integrasi dengan layanan backend seperti *Firebase*, sehingga memudahkan dalam pengembangan aplikasi *monitoring* berbasis *realtime*[13].



Gambar 2. 4 Flutter

Sumber: <https://www.igmguru.com/digital-marketing-programming/flutter-training>

2.2.5 Sistem Monitoring

Sistem *Monitoring realtime* adalah mekanisme pemantauan kondisi perangkat atau sistem yang datanya diperbarui secara langsung tanpa jeda waktu yang panjang. Dalam konteks *smartlocker*, *monitoring realtime* mencakup:

- a. status loker (terbuka, tertutup, terkunci)
- b. aktivitas pengguna (berhasil/gagal autentikasi)
- c. notifikasi peringatan jika terjadi akses ilegal
- d. pembaruan data otomatis tanpa *refresh* aplikasi

Sistem *monitoring realtime* merupakan sistem yang mampu menampilkan data secara langsung tanpa adanya jeda waktu yang signifikan[11].

Dalam sistem *smartlocker*, *monitoring realtime* digunakan untuk mengetahui status loker, aktivitas pengguna, serta riwayat penggunaan secara langsung, sehingga mempermudah proses pengawasan dan pengambilan keputusan[8].

2.2.6 Sistem Notifikasi pada Aplikasi Mobile

Sistem notifikasi pada aplikasi *mobile* merupakan mekanisme penyampaian informasi kepada pengguna ketika terjadi suatu aktivitas atau perubahan data pada sistem. Notifikasi digunakan untuk memberikan informasi secara cepat sehingga pengguna dapat segera mengetahui kondisi atau peristiwa yang terjadi tanpa harus melakukan pengecekan aplikasi secara terus-menerus. Penerapan sistem notifikasi pada aplikasi *mobile* dapat meningkatkan efektivitas penyampaian informasi dan membantu pengguna dalam merespons suatu kejadian dengan lebih cepat[14].

Pada penelitian ini, sistem notifikasi diterapkan menggunakan *Flutter Local Notifications* yang berfungsi untuk menampilkan notifikasi lokal pada aplikasi ketika aplikasi mendeteksi perubahan data akses gagal pada *Firebase Database*. Dengan demikian, pengguna dapat memperoleh informasi mengenai percobaan akses yang tidak berhasil secara langsung melalui aplikasi tanpa menggunakan layanan *Firebase Cloud Messaging (FCM)*.

2.2.7 Sistem Backend Berbasis Web

Sistem backend berbasis *web* berfungsi sebagai pusat pengolahan dan penyimpanan data dalam sistem *IoT*. *Backend* menerima data dari perangkat, menyimpannya ke *database*, dan menampilkannya pada *dashboard* admin maupun aplikasi pengguna[8].

Dengan adanya *backend*, administrator dapat melakukan monitoring seluruh sistem secara terpusat, termasuk melihat status semua loker dan aktivitas pengguna[11].

2.2.8 Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan bahasa pemodelan yang digunakan untuk menggambarkan sistem perangkat lunak secara visual. *UML* membantu dalam memahami alur sistem serta hubungan antar komponen yang ada di dalamnya[15].

Dalam penelitian ini, *UML* digunakan untuk merancang sistem melalui beberapa diagram seperti *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram*.

2.2.9 Node Js

Node.js merupakan lingkungan *runtime JavaScript* yang digunakan untuk menjalankan kode *JavaScript* di sisi server. *Node.js* memungkinkan proses pengolahan data dan komunikasi antara *website* dengan *database* dilakukan secara lebih cepat dan efisien. Pada penelitian ini, *Node.js* digunakan sebagai server lokal untuk menjalankan *website monitoring smartlocker*[16]. *Node.js* mendukung proses pengembangan *website* dengan memanfaatkan *JavaScript* sebagai dasar pemrograman sehingga komunikasi data dan pengolahan sistem dapat dilakukan secara lebih efisien. Selain itu, *Node.js* juga menyediakan berbagai layanan seperti pengelolaan *HTTP* dan

pembuatan *RESTful API* yang membantu proses pertukaran data pada aplikasi *web*[17].

2.2.10 JSON

JSON (JavaScript Object Notation) merupakan format pertukaran data yang bersifat ringan dan mudah dipahami baik oleh manusia maupun sistem komputer. *JSON* banyak digunakan dalam pengembangan aplikasi berbasis web karena mampu menyimpan dan mengirim data secara terstruktur dengan proses yang lebih efisien. Selain itu, *JSON* juga mendukung proses pertukaran data antar sistem sehingga mempermudah komunikasi antara server dan *client* pada aplikasi *website*. Dalam penelitian ini, *JSON* digunakan sebagai media penyimpanan data pengguna serta data *monitoring smartlocker* pada *website* admin[18].

2.2.11 HTML

HTML (HyperText Markup Language) merupakan bahasa markup yang digunakan untuk membangun struktur dasar halaman *website*. *HTML* berfungsi untuk menampilkan berbagai komponen pada halaman *web* seperti teks, tabel, gambar, tombol, dan tampilan *dashboard monitoring*. Dalam penelitian ini, *HTML* digunakan untuk membangun tampilan *website* admin pada sistem monitoring *smartlocker* sehingga administrator dapat melihat status loker secara *realtime*[19].

2.2.12 *JavaScript*

JavaScript merupakan bahasa pemrograman yang digunakan untuk membuat *website* menjadi lebih interaktif dan dinamis. *JavaScript* mampu menjalankan proses pengolahan data secara langsung pada halaman website sehingga informasi dapat diperbarui tanpa harus memuat ulang halaman. Pada penelitian ini, *JavaScript* digunakan untuk mengambil dan menampilkan data *monitoring* loker secara *realtime* pada *website* admin.