Implementation of Finite State Machine in a Goods Storage System Based on ESP8266 with Fingerprint Security Authentication on Android Devices via Application

Sri Wahyuni Dali[1], Putri Elfa Mas'udia[2], Lutfi Kurniawan[3]
[1,3] Digital Telecommunication Network Study Program, Department of Electrical Engineering, State Polytechnic of Malang, 65141, Indonesia.
[2] Telecommunication Engineering Study Program, Department of Electrical Engineering, State Polytechnic of Malang, 65141, Indonesia.

[1]sri.wahyuni@polinema.ac.id , [2]putri.elfa@polinema.ac.id, [3]2141160005@student.polinema.ac.id

***Abstract*— The absence of secure storage facilities at SMK Hidayatul Ummah, Gresik, is a problem because students still use ordinary cabinets that can be accessed by anyone without key security. To overcome this, this study developed a smart locker system based on the Internet of Things (IoT) that is able to control the locker door and authenticate users through fingerprint scanning from an Android application. This system is designed using an ESP8266 microcontroller and a Finite State Machine (FSM) algorithm to set the locker status in the conditions "Available", "Reserved", and "Used." User data and locker status are managed in real-time using Firebase as a cloud database. The test results show an average system response time of 1.7 seconds, which is still in the acceptable category for a real-time system. The fingerprint feature in the application also provides an additional layer of security, ensuring that only authorized users can access the locker. Visualization of locker status through color indicators-green for "Available," yellow for "Reserved," and red for "Used"-is considered effective in conveying information directly and intuitively. Thus, this system is able to become a safe, efficient storage solution that supports the implementation of technology in educational environments.**

***Keywords*— Android Application, Fingerprint Authentication, Finite State Machine, Internet of Things (IoT), Locker**

I. INTRODUCTION

Lockers serve as secure storage facilities for personal or valuable items and are widely used in schools, universities, gyms, and places of worship [1]. They offer a practical solution for safeguarding personal belongings while individuals engage in other activities. At SMK Hidayatul Ummah, Gresik, East Java, no dedicated secure storage is available for students to store their items during religious activities in the school mosque. The current storage method relies on open cabinets accessible to anyone, lacking individual locks and proper organization. This condition poses a high risk of theft or loss and underlines the urgent need for a safer and more efficient storage solution [2].

Technological advancements in microcontrollers and the Internet of Things (IoT) have opened opportunities for developing smart locker systems that are more secure, flexible, and user-friendly [3]. Previous research has introduced various approaches to improve locker security. The study in proposed a smart locker integrating IoT, One-Time Password (OTP) authentication, and face detection, where users request an OTP via email valid for a limited time, and face detection ensures that only one authorized user is present. The system also includes environmental temperature monitoring with alerts if the threshold is exceeded [4]. Another study designed an IoT-based security and monitoring system using RFID for identity verification, ESP32-CAM for capturing images inside the locker, an;d Telegram for remote notifications, combined with a solenoid lock to automatically open after successful authentication [5]. In a locker security system was developed using fingerprint, touch, and ultrasonic sensors connected via the Blynk application. Arduino Uno controlled the sensors, while ESP32 Devkit V1 provided internet connectivity. The system allowed fingerprint-based locker access, touch-based condition control, and ultrasonic detection of stored items [6].

While these approaches enhance security, they often require additional dedicated hardware such as cameras, RFID cards, or external fingerprint sensors, which may be less practical in environments with limited resources [7]. In contrast, this study develops a smart locker system that leverages the built-in fingerprint authentication feature of Android devices, eliminating the need for extra biometric hardware [8]. This method enhances convenience by allowing users to unlock lockers directly via a smartphone application, reducing the risk of lost keys or RFID cards and ensuring accurate user verification [9].

The proposed system also integrates IoT connectivity, enabling remote access and real-time monitoring of locker status via a color-coded interface (e.g., green for available, yellow for reserved, red for in use). Furthermore, the Finite State Machine (FSM) algorithm is implemented to manage locker states (*Available*, *Reserved*, *In Use*) and their transitions efficiently [10]. This structured approach ensures that the

locker access process is well-organized, responsive to authentication outcomes, and adaptable to multiple usage scenarios [11].

The objectives of this research are twofold: (1) to design and implement a smart locker system utilizing Android-based fingerprint authentication for secure storage at SMK Hidayatul Ummah, and (2) to apply the FSM algorithm to optimize locker access management and state transitions.

II. METHOD

This study applies an experimental research approach to design and implement a smart locker system with Android-based fingerprint authentication and the Finite State Machine (FSM) algorithm for efficient access control. The experiment tested how the authentication process and FSM logic can enhance locker security and management efficiency in SMK Hidayatul Ummah, Gresik, East Java.

A. System Design

The smart locker system consists of three main parts: input, process, and output. The Android application serves as the main user interface, enabling booking, unlocking, and status monitoring. Built-in fingerprint authentication ensures only verified users can access lockers [12]. A 12V DC power adapter supplies all components, including the ESP8266 microcontroller, solenoid lock, and other electronics. A 4×4 keypad provides emergency access, and MC-38 magnetic sensors detect forced openings.

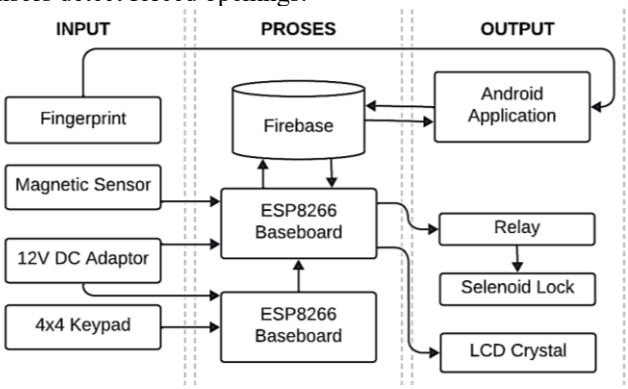


Figure 1. System Blok Diagram

The ESP8266 acts as the control unit, receiving commands from the Android app via the internet and communicating with Firebase for real-time status synchronization [13]. Upon successful authentication, the ESP8266 activates the relay to control the solenoid lock.

B. Hardware Circuit

The hardware circuit is designed to control and monitor a smart locker system using the ESP8266 Base Board as the main controller. Each locker is equipped with a solenoid lock that is activated through a 5V relay module, allowing the ESP8266 to securely manage the locking and unlocking mechanism. A 16×2 LCD module is connected to display real-time locker status, such as locked, unlocked, or intrusion detected. For backup access, a 4×4 keypad is provided to enable users to enter an emergency PIN code manually. To enhance security, MC-38 magnetic sensors are installed on each locker door to detect unauthorized or forced access. All components are powered by a 12V DC adapter, which supplies stable voltage to the system. In the event of a security breach, the ESP8266 immediately transmits an alert through Firebase Cloud Messaging, ensuring that notifications are delivered in real time to the user's Android application. This integration provides a reliable and responsive security mechanism for the locker system.

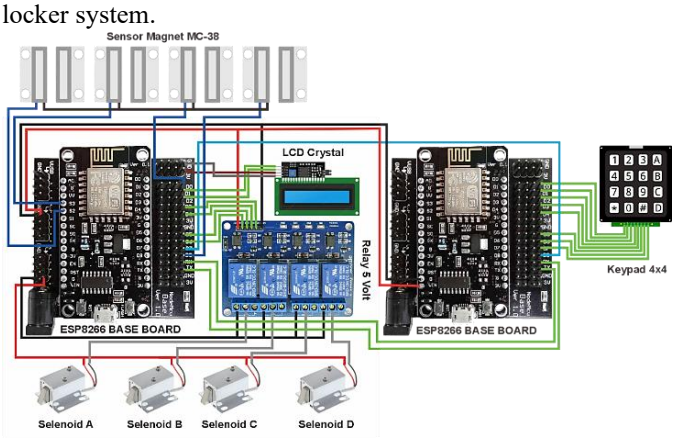


Figure 2. System Blok Diagram

C. System Flowchart

The system starts with the ESP8266 connecting to Wi-Fi. If successful, users access the locker through the Android app. In case of connection failure, authorized personnel can use the keypad with an emergency PIN [14]. Users can reserve, open, or release lockers via fingerprint authentication, with FSM handling state transitions between *Available*, *Reserved*, and *In Use*.

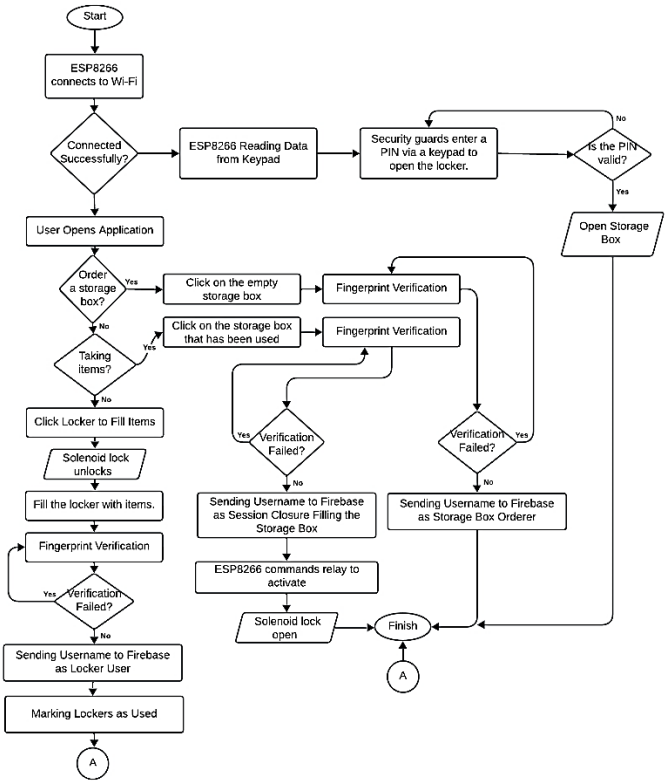


Figure 3. System Flowchart

D. Data Flow Diagram

The Data Flow Diagram (DFD) shows how users, the system, and lockers interact. The Android app uses the device's fingerprint API for authentication without storing biometric data. Firebase stores user and locker status, while the FSM decides locker access based on verification results.

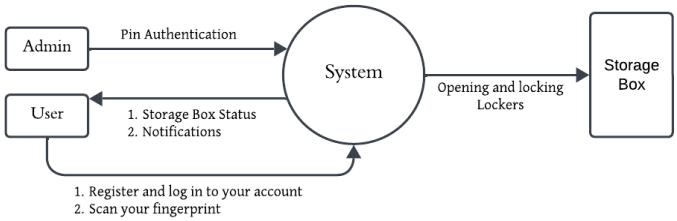


Figure 4. DFD Level 0

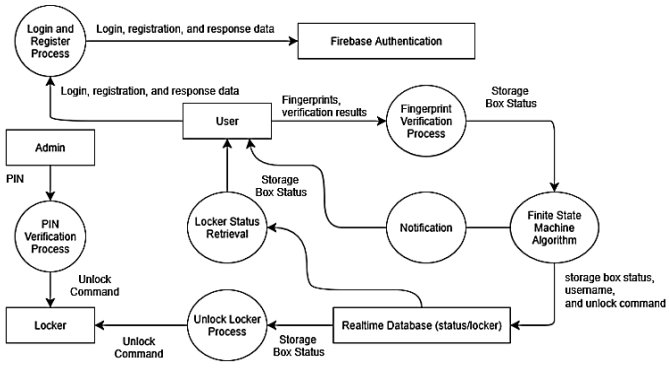


Figure 5. DFD Level 1

E. Application Design

The app consists of three main pages: Login, Register, and Locker Dashboard. The dashboard visually represents locker status and allows user interaction for booking or releasing lockers.

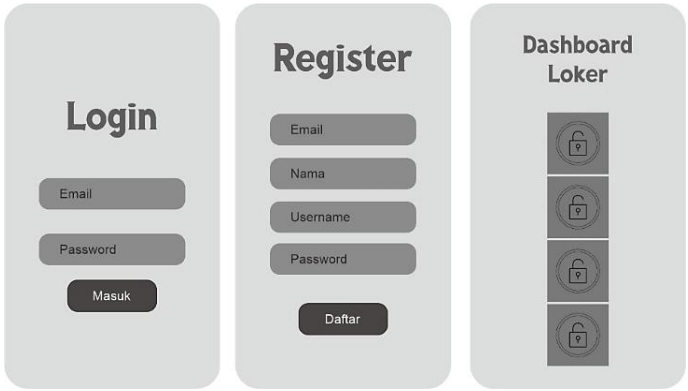


Figure 6. Application UI Design

F. Finite State Machine (FSM) Implementation

The Finite State Machine (FSM) is implemented to manage the operational states of the locker system. The FSM defines three main states: Available, Reserved, and In Use. Transitions between these states are triggered by specific events such as booking a locker, canceling a reservation, starting usage, temporarily opening, or releasing the locker. By applying this structured approach, the system ensures efficient state management, minimizes conflicts, and prevents unauthorized or simultaneous use of the lockers. This mechanism improves reliability and provides a clear operational flow for user interactions with the smart locker system [15].

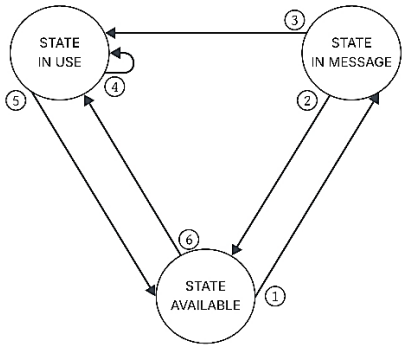


Figure 7. FSM Design

G. Tools and Materials

The tools and materials used in the development of the smart locker system are summarized in Table 1. These include both hardware and software components, each serving a specific role in building a secure IoT-based locker system.

Table 1. Tools and Materials

Category	Tool/Material	Description
Hardware	ESP8266	Wi-Fi module functioning as the main microcontroller for controlling devices and communicating with Firebase.

Category	Tool/Material	Description
	Solenoid Lock	Electrically controlled lock for securing and unlocking the locker.
	Relay	Electronic switch enabling ESP8266 to control the solenoid lock without risk of damage.
	LCD I2C	Displays locker status informationer.
	Keypad 4x4	PIN input device for emergency access
	Baseboard	Component mounting board for stability and simplified wiring.
	12 DC Adapter	Power supply for the solenoid lock and relay, ensuring stable system operation.
	Magnetic Sensor MC-38	Detects and alerts forced door openings.
Software	Android Studio	IDE for developing the Android application as the user interface for the locker system.
	Firebase	Cloud platform used for user authentication and real-time locker status storage.

H. Testing Parameters

The parameters used to evaluate the performance and reliability of the IoT-based smart locker system are shown in Table 2. These parameters assess both technical and user-experience aspects of the system.

Table 2. Testing Parameter

Parameter	Decription	Success Indicator
Locker Status Transition	Ensures each locker can only change state according to predefined FSM rules.	All state transitions follow FSM rules without errors.
Response Time	Measures the time required to unlock or lock the locker after receiving a command.	Response time ≤ 2 seconds.
Application Security	Evaluates the security level of the Android application through user authentication mechanisms.	Successful authentication using user email and fingerprint.
Locker Status Visualization	Displays locker status in real-time in the Android application with clear color indicators.	Status displayed with colors: green (available), red (in use), yellow (reserved).

III. RESULTS AND DISCUSSION

The research was conducted at SMK Hidayatul Ummah, located at Jl. Raya Balongpanggang, Tamping, Pucung, Balongpanggang District, Gresik Regency, East Java, Indonesia. The site was selected for its supportive educational environment for IoT-based technology applications. The development process started in February 2025, from system design to hardware assembly and application development, with final testing conducted on April 30, 2025.

A. Implementation Results

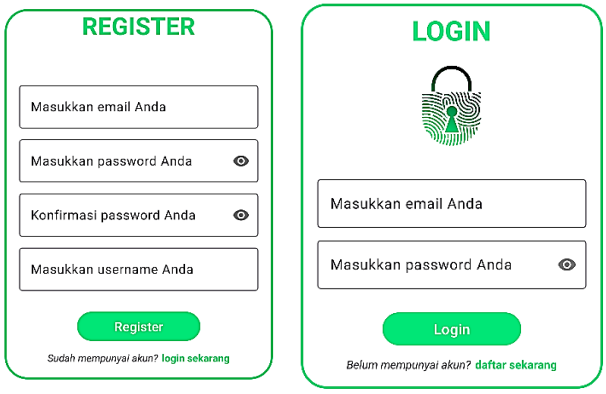
The hardware system consisted of an ESP8266 microcontroller as the core controller, a solenoid lock mechanism controlled via a relay module, an LCD I2C 16×2 display for locker status, and a 4×4 keypad for PIN entry. The magnetic sensor (MC-38) was integrated for forced-door detection, and all electronic components were housed within a 20×15×10 cm control box equipped with ventilation and organized wiring. The locker unit itself was constructed from MDF wood, measuring 180×60×40 cm, with four independent compartments, each fitted with stainless steel handles and an integrated solenoid locking system.

Figure 9. Locker Physical Design

The Android application was developed using Android Studio, with Firebase Authentication handling user login and Firestore managing real-time locker status data. Upon launching, the splash screen as shown in Figure 11 checks login credentials stored via Shared Preferences to determine whether the user should be directed to the dashboard or login page.

Figure 10. Splash Screen

New users register by entering an email, password, and username, where authentication credentials are stored in Firebase Authentication, and additional user data is stored in Firestore. Returning users can log in through the login page, with incorrect credentials triggering an error notification.



(a) (b)

Figure 11. (a) Register Page and (b) Login Page

The dashboard as shown in Figure 13 displays four interactive locker entries, each with a color-coded status: green for available, yellow for reserved, and red for in-use. Access to a locker requires fingerprint authentication via the Android device, ensuring that only authorized users can unlock or operate the locker.

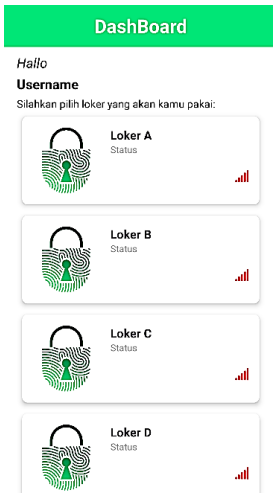


Figure 12. Dashboard Screen

B. Response Time Testing

The response time test measured the duration from fingerprint verification to the solenoid lock's activation. Testing was performed 10 times under stable network conditions, with measurements recorded manually using a stopwatch. The success criterion was a response time of ≤ 2 seconds.

Table 3 presents the recorded times, which ranged from 1.5 to 2.3 seconds, with an average of 1.7 seconds. Only two trials slightly exceeded the 2-second limit, likely due to network latency or sensor response delays.

Table 3. Response Time Testing Results

Trial No-	Action	Measuring Tool	Response (seconds)
1	Fingerprint Verification	Stopwatch	1.7
2	Fingerprint Verification	Stopwatch	1.6
3	Fingerprint Verification	Stopwatch	1.8
4	Fingerprint Verification	Stopwatch	2.3
5	Fingerprint Verification	Stopwatch	1.6
6	Fingerprint Verification	Stopwatch	1.5
7	Fingerprint Verification	Stopwatch	2.1
8	Fingerprint Verification	Stopwatch	1.7
9	Fingerprint Verification	Stopwatch	1.6
10	Fingerprint Verification	Stopwatch	1.9
	Average Response		**1.7**

The results confirm that the system operates in near real-time, with most trials meeting the ≤ 2 seconds criterion. The average response time of 1.7 seconds indicates efficient integration between fingerprint authentication, Firebase cloud services, and ESP8266 hardware. The minimal delay observed in two trials suggests that network stability and sensor processing speed are critical factors in maintaining consistent performance

C. Locker Status Transition Testing

ocker status transition testing was carried out to verify that the system manages state changes correctly according to user actions. The locker status logic is based on the Finite State Machine (FSM) approach, ensuring that only valid transitions occur, preventing invalid or skipped states.

Table 4. Locker Status Transition Test Results

No	Current State	Event (Trigger)	Next State	Actual Result	Validation
1	Available	Book	Reserved	Reserved	Valid
2	Available	Start Using	In Use	In Use	Valid
3	Reserved	Cancel Reservation	Available	Available	Valid
4	Reserved	Start Using	In Use	In Use	Valid
5	Reserved	Book by Another User	Reserved	Reserved	Valid
6	Reserved	Start Using by Another User	Reserved	Reserved	Valid
7	In Use	Temporary Open	In Use	In Use	Valid
8	In Use	End Usage	Available	Available	Valid
9	In Use	Book by Another User	In Use	In Use	Valid
10	In Use	Start Using by Another User	In Use	In Use	Valid
11	Already Booked	Book Another Locker	Reserved	Reserved	Valid

The test results, summarized in Table 4, show that all user actions (booking, using, canceling, and unauthorized access attempts) produced correct and expected state changes in accordance with the FSM design depicted. This confirms that multi-user locker usage can be managed without state conflicts.

The FSM successfully enforced valid state transitions and access control. For example, a "Booking" action from the Available state correctly changed the status to Reserved, while unauthorized attempts to access a Reserved or In-Use locker were denied. The transition logic also allowed flexibility for direct usage without prior booking. The FSM functions tested are detailed in Table 5.

Table 5. Function Metode Finite State Machine

Function	Initial State	Final State	Biometric Verification	Description
Transition from Available to In Use	Available	In Use	Yes	Changes locker status to occupied after fingerprint verification.
Transition from Available to Reserved	Available	Reserved	Yes	Changes locker status to reserved if fingerprint verification is successful

Function	Initial State	Final State	Biometric Verification	Description
Transition from Reserved to In Use	Available	In Use	Yes	Changes locker status to occupied after verification of the reserving user.
Temporary Open while In Use	In Use	In Use	Yes	Allows temporary opening of the locker without changing its status.
Transition from In Use to Available	In Use	Available	Yes	Ends locker usage and changes status back to available.
Transition from Reserved to Available	Reserved	Available	Yes	Cancels reservation and restores locker status to available.

Documentation of the status interface during testing is shown in:

- Empty Storage Box – Displays available lockers with options to "Use Locker" or "Book Locker."

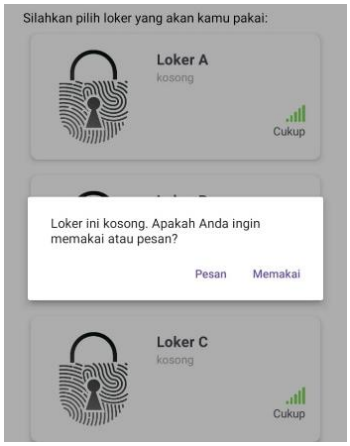


Figure 13. Empty Storage Box

- Storage Box Reserved – Shows lockers in Reserved status, restricting access to non-owners.

Figure 14. Storage Box Reserved

- Used Storage Box – Indicates active usage, with options for temporary opening or ending the session.

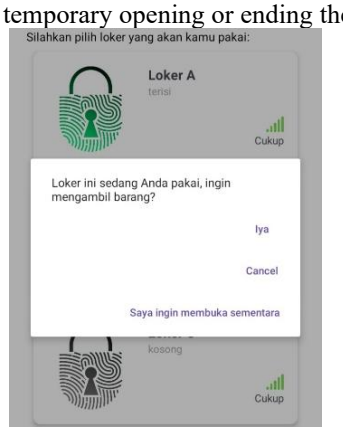


Figure 15. Storage Box Reserved

D. Application Security Testing

The application's security was tested using two authentication layers: email-password login and fingerprint verification.

Security testing confirmed that the system accurately distinguished between authorized and unauthorized access attempts. Login with valid credentials granted access, while invalid credentials triggered error notifications. Fingerprint verification further ensured that only registered users could unlock lockers. The outcomes are summarized in Table 6.

Table 6. Application Security Testing Results

No	Testing Scenario	Step Performed	Expected Result	Actual Result	Outcome
1	Login with valid email and password	*Enter correct credentials*	Successfully logged into the application	Success	Application displays the main dashboard
2	Login with invalid email or password	*Enter incorrect credentials*	Access denied, error notification appears	Access Denied	Notification "Incorrect Email or Password" displayed
3	Unlock locker with registered fingerprint	Place registered fingerprint on the device	Locker opens after validation	Locker Opens	Solenoid activated and locker door opened
4	Unlock locker with unregistered fingerprint	Place fingerprint from another/unregistered user	Access denied, no response	Access Denied	No response, failure notification displayed
5	Use more than 1 device to log in with 1 account	*Enter credentials on multiple devices*	Warning appears: only 1 device login allowed	*Login Failed*	Warning displayed: account can only be logged in on one device
6	Login to an account registered on Device A using Device B	Access Locker	Warning appears: transactions not allowed on different device	Cannot Transact	Warning displayed: transactions not allowed on a different device

The two-layer security mechanism (Firebase Authentication + fingerprint) proved effective in preventing unauthorized access. Even if account credentials were compromised, the fingerprint verification acted as a secondary safeguard, significantly enhancing overall system security.

E. Locker Status Visualization Testing

This test verified that locker statuses are displayed in the Android application with the correct color indicators: green (Available), yellow (Reserved), and red (In Use).

Table 7. Results of Locker Status Visualization Testing

Locker Status	Color	Description	Testing Result
Available	Green	Locker is free to use	The application correctly displays green, indicating that the locker is available.
Reserved	Yellow	Locker is reserved by a user	The application displays yellow, indicating that the locker is currently reserved.
In Use	Red	Locker is actively being used	The application correctly displays red, indicating that the locker is actively in use.

As shown in Table 7, the application correctly displayed each status with the expected color in real-time, ensuring quick and intuitive status recognition by users.

The color-coded visualization enhanced user experience by providing an immediate, text-free understanding of locker status. Consistent use of green, yellow, and red indicators made the interface intuitive and effective for real-time decision-making.

IV. CONCLUSION

Based on the design, implementation, and testing that have been carried out, it can be concluded that the smart locker system using ESP8266 and Android fingerprint authentication was successfully developed. The system integrates ESP8266, Firebase Realtime Database, and an Android application to provide secure and efficient locker access with an average response time of 1.7 seconds. The application also manages locker status transitions effectively through the Finite State Machine (FSM) approach, ensuring accurate and reliable status management between Available, Reserved, and In Use.

Overall, this smart locker system has been proven feasible for implementation, particularly in school environments, as a modern and secure storage solution. For future development, additional features such as two-factor authentication, real-time notifications, and an admin dashboard are recommended to further enhance security, usability, and system management.

ACKNOWLEDGEMENTS

The authors would like to express their sincere gratitude to State Polytechnic of Malang for the academic guidance and support throughout the research process. The authors also extend appreciation to SMK Hidayatul Ummah, Gresik, for providing the facilities and assistance during the system implementation and testing phase.

REFERENCES

[1] F. Rozy and I. Fahruzi, "Locker Security System Using PN532 RFID/NFC Smart Card," *Jurnal Integrasi*, vol. 14, pp. 114–121, 2022.

[2] M. N. Mostakim, R. Sarkar, and M. A. Hossain, "Smart Locker: IoT-Based Intelligent Locker with Password Protection and Face Detection Approach," *Wireless and Microwave Technologies*, vol. 3, pp. 1–10, 2020.

[3] A. Bhawadzier, A. Alhafiz, and N. B. Nugroho, "Security and Monitoring System of Locker Based on Internet of Things (IoT)," *Jurnal Sistem Komputer Triguna Dharma*, vol. 3, pp. 97–108, 2024.

[4] N. P. K. Rahil and R. Irawati, "Locker Security System Using Fingerprint, Touch, and Ultrasonic Sensors Integrated via Blynk Application," in *Proceedings of the National Student Seminar of the Faculty of Information Technology (SENAFTI)*, vol. 3, pp. 1045–1054, 2024.

[5] I. Nusufi, Design of Learning Media Application for Reading and Writing in Public Elementary School 1 Deah Rungkom Based on Android, *Banda Aceh: Faculty of Tarbiyah and Teacher Training*, 2023.

[6] D. Ramdani, "Design and Development of IoT-Based Temperature Automation and pH Monitoring System for Aquascape Using NodeMCU ESP8266 and Telegram Application," *Journal of Informatics Information System Software Engineering and Applications (INISTA)*, vol. 3, no. 1, pp. 59–68, 2020.

[7] R. I. Ramadhan and M. S. Ladjamuddin, "Web Filtering System Design Using DNS Forwarding Method on Computer Networks Based on Mikrotik RouterOS," *Jurnal Informatika dan Teknologi Komputer (JITEK)*, vol. 2, no. 2, pp. 146–157, 2022.

[8] M. Ilhamsyah and M. Fathurrahman, "Design of Android Application MySmartFishFeeding Using FlutterFlow and Firebase Firestore," *Jurnal Informatika dan Teknik Elektro Terapan (JITET)*, 2024.

[9] I. F. Maulana, "Implementation of Firebase Realtime Database on E-Ticketing Smartphone Application Based on Android Mobile," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 2020.

[10] A. A. Setyaningrum, A. P. Sasmito, and H. Z. Zahro, "Implementation of Finite State Machine Method in Noir Adventure Game," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, pp. 1298–130, 2024.

[11] A. A. Setyaningrum, A. P. Sasmito, and H. Z. Zahro, "Implementation of Finite State Machine Method in Adventure Game," *Jurnal Mahasiswa Teknik Informatika*, vol. 8, pp. 23–31, 2024.

[12] A. D. Kemalasari, H. Agustina, and I. Zulkarnin, "Effectiveness of Fingerprint Attendance Implementation on Civil Servants' Work Discipline at the Environmental Agency of Cirebon City," *Jurnal Ilmiah Publika*, vol. 9, 2019.

[13] M. Algarni, "An Extra Security Measurement for Android Mobile Applications," *Journal of Information Security and Cybercrimes Research*, vol. 6, no. 2, pp. 139–149, 2023.
[14] S. Gunawan, A. H. Anshor, and A. Amali, "IoT-Based Smart Park Monitoring and Control System," *Bulletin of Computer Science Research*, vol. 4, pp. 283–288, 2023.
[15] A. Nazarwati, A. Yuan, and D. M. Jannah, "Home Security and Door Unlocking System Using Arduino-Based Keypad," *Journal of Information System Research (JOSH)*, vol. 6, pp. 341–348, 2024.