



Sistem AC-Box (*Access Control Box*) dengan wi-fi manager pada ESP32 dan autentikasi *fingerprint* berbasis aplikasi android

Salsadila Puspitasari¹, Dwi Marisa Midyanti², Uray Ristian³

Email: ¹salsadilapuspitasari@gmail.com, ²dwi.marisa@siskom.untan.ac.id, ³eristian@siskom.untan.ac.id

Rekayasa Sistem Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Tanjungpura

Diterima: 12 Juni 2026 | Direvisi: 14 Agustus 2026 | Disetujui: 25 Agustus 2026

©2020 Program Studi Teknik Informatika Fakultas Ilmu Komputer,
Universitas Muhammadiyah Riau, Indonesia

Abstrak

Kotak penyimpanan merupakan wadah penting untuk menyimpan barang berharga yang fleksibel ditempatkan di berbagai lokasi. Penelitian ini bertujuan untuk mengimplementasikan Sistem AC-Box (*Access Control Box*) berbasis *Internet of Things* (IoT) dengan fokus pada kemudahan konfigurasi jaringan dan keamanan akses. Sistem ini menggunakan ESP32 yang diintegrasikan dengan fitur Wi-Fi Manager sehingga perangkat dapat terhubung ke jaringan baru secara dinamis tanpa perlu melakukan pemrograman ulang (*hardcoding*). Untuk mekanisme keamanan digunakan autentikasi *fingerprint* melalui sensor biometrik bawaan smartphone yang terhubung dengan aplikasi Android sebagai antarmuka pengguna. Berdasarkan hasil pengujian pada dua jenis jaringan yaitu jaringan Wi-Fi *access point* ZTE dan *hotspot* smartphone, sistem berjalan dengan cukup baik dalam merespons setiap perintah buka kunci yang diberikan. Rata-rata *delay* pada jaringan Wi-Fi *access point* ZTE adalah 384,2 ms masuk kategori sedang dan pada *hotspot* smartphone sebesar 566,7 ms masuk kategori jelek menurut standar TIPHON. Waktu koneksi awal melalui Wi-Fi Manager berkisar antara 806,1 ms hingga 3.151,4 ms. Pengujian keamanan autentikasi *fingerprint* berhasil mengenali 5 sidik jari yang terdaftar dan 25 sidik jari yang tidak terdaftar. Hasil ini membuktikan bahwa sistem mampu memvalidasi akses dengan akurat dan mencegah masuknya pengguna yang tidak sah. Dapat disimpulkan bahwa AC-Box mampu menjadi solusi sistem akses yang praktis, aman, dan efisien untuk kebutuhan penyimpanan pribadi.

Kata kunci: *Wi-Fi Manager, Autentikasi Sidik Jari, Aplikasi Android*

Access control box system with wi-fi manager on ESP32 and fingerprint authentication based on android application

Abstract

A storage box is an essential container for securing valuable items and offers the flexibility to be placed in various locations. This research aims to implement the AC-Box (*Access Control Box*) system based on the *Internet of Things* (IoT), focusing on ease of network configuration and access security. The system utilizes an ESP32 integrated with a Wi-Fi Manager feature, allowing the device to connect to new networks dynamically without the need for reprogramming (*hardcoding*). For the security mechanism, fingerprint authentication is employed via the smartphone's built-in biometric sensor, which is connected to an Android application as the user interface. Based on testing results across two network types—a ZTE Wi-Fi access point and a smartphone hotspot—the system performs adequately in responding to every unlock command. The average delay on the ZTE Wi-Fi access point was 384.2 ms, categorized as "Medium," while the smartphone hotspot recorded 566.7 ms, categorized as "Poor" according to TIPHON standards. The initial connection time via Wi-Fi Manager ranged from 806.1 ms to 3,151.4 ms. Fingerprint authentication security testing successfully recognized 5 registered fingerprints and 25 unregistered fingerprints. These results prove that the system can accurately validate access and prevent unauthorized entry. In conclusion, the AC-Box serves as a practical, secure, and efficient access system solution for personal storage needs.

Keywords: *Wi-Fi Manager, Fingerprint Authentication, Android Application*

1. PENDAHULUAN

Kotak penyimpanan adalah wadah yang berfungsi untuk menyimpan barang-barang berharga milik individu atau kelompok. Kotak ini dapat ditempatkan di berbagai lokasi seperti di rumah, kantor, sekolah maupun tempat lainnya[1]. Dalam penelitian ini sistem tersebut diberi nama AC-Box (*Access Control Box*) merupakan kotak tempat penyimpanan yang memiliki sistem akses kontrol berbasis IoT. Dengan penerapan Wi-Fi Manager, AC-Box dapat beradaptasi di berbagai tempat tanpa perlu konfigurasi jaringan perangkat melalui kode program atau pemrograman ulang. Sistem pada AC-Box juga dirancang dengan mekanisme autentikasi *fingerprint* sehingga pemiliknya saja yang dapat mengakses dan mengontrol kotak melalui aplikasi Android. Salah satu tantangan dalam perangkat IoT adalah konfigurasi Wi-Fi yang biasanya dituliskan langsung di dalam kode program. Hal ini menyebabkan perangkat menjadi kurang fleksibel, karena tidak dapat dengan mudah berpindah ke jaringan Wi-Fi[2]. Untuk mengatasi ketergantungan pada pengaturan jaringan manual, penelitian ini mengadopsi fitur Wi-Fi Manager pada perangkat ESP32. Dengan fitur ini perangkat dapat membentuk *access point* (AP) sementara saat pertama kali dinyalakan sehingga pengguna dapat mengonfigurasi jaringan Wi-Fi melalui antarmuka web tanpa perlu memprogram ulang kode. Dengan langkah ini meningkatkan fleksibilitas penempatan perangkat di tempat dengan jaringan Wi-Fi yang berbeda-beda, seperti rumah, kantor, atau tempat lainnya.

Penelitian sebelumnya yang berjudul “Pengembangan Prototipe Alat Pengusir Tikus Berbasis *Internet of Things* dengan Integrasi Notifikasi API WhatsApp” [3]. Sistem ini dirancang untuk mengusir hama tikus di area persawahan secara otomatis dengan integrasi API WhatsApp yang mengirimkan notifikasi peringatan serta tautan pemantauan langsung ke ponsel pengguna saat gerakan terdeteksi. Dengan tautan itu pengguna dapat memantau lahan secara *real-time* menggunakan kamera ESP32-CAM. Selain fitur kendali penelitian [3] juga menerapkan Wi-Fi Manager untuk meningkatkan fleksibilitas konfigurasi jaringan perangkat. Fitur ini memungkinkan alat untuk berpindah-pindah jaringan Wi-Fi tanpa perlu memprogram ulang kode. Namun bedanya dengan penelitian ini terdapat mekanisme keamanan autentikasi *fingerprint* memanfaatkan sensor biometrik bawaan smartphone untuk mengakses kotak penyimpanan.

Penelitian selanjutnya yang berkaitan dengan autentikasi *fingerprint* berjudul “*IoT based wifi fingerprint door lock system with raspberry pi & webcam*” [4]. Sistem ini membangun kunci pintu yang terhubung jaringan Wi-Fi agar bisa diakses dari jarak jauh melalui perangkat mobile dan integrasi API WhatsApp. Mekanisme autentikasi menggunakan sensor biometrik yang terhubung ke Arduino dan Raspberry Pi untuk memverifikasi identitas pengguna dengan kapasitas penyimpanan hingga 21 data sidik jari. Jika sidik jari tidak dikenal mencoba masuk, sistem akan memicu alarm *buzzer* dan mengirimkan notifikasi peringatan melalui WhatsApp kepada pemilik rumah. Namun pada penelitian [4] konfigurasi jaringan Wi-Fi masih bersifat statis (*hardcode*), berbeda dengan sistem AC-Box yang menawarkan fleksibilitas konfigurasi jaringan secara dinamis melalui fitur Wi-Fi Manager.

Penelitian lain yang memiliki objek sama yaitu kotak penyimpanan berjudul “Implementasi Protokol WebSocket Pada Sistem Keamanan *Safebox* Menggunakan Kode OTP (*One-Time Password*)” [5]. Sistem ini dirancang untuk mengatasi risiko lupa *password* atau pencurian PIN pada kunci manual. Perangkat utama yang digunakan meliputi NodeMCU ESP32, *keypad*, layar LCD, dan *solenoid door lock*. Mekanisme autentikasi bekerja dengan cara pengguna meminta kode OTP melalui aplikasi Android. Sistem kemudian mengirimkan 6 digit angka acak ke email pengguna yang hanya berlaku selama 5 menit. Kode tersebut harus diinput melalui *keypad* untuk diverifikasi secara *real-time* oleh *cloud server* via protokol WebSocket. Hasil pengujian menunjukkan performa yang sangat stabil dengan rata-rata *delay* 71 ms dan *throughput* 258 bps. Namun penelitian [5] masih berfokus pada keamanan berbasis kode OTP dan input manual, sementara penelitian ini menawarkan pendekatan alternatif melalui autentikasi biometrik *fingerprint* pada smartphone untuk alur akses yang tidak banyak perantara, serta dilengkapi fitur Wi-Fi Manager untuk konfigurasi jaringan yang dinamis.

Berdasarkan latar belakang ini, penelitian ini memberi solusi yang menggabungkan kemudahan konfigurasi jaringan dan sistem keamanan akses. Maka diusulkan sebuah sistem berjudul “AC-Box (*Access Control Box*) dengan Konfigurasi Wi-Fi Manager pada ESP32 dan Autentikasi *Fingerprint* Berbasis Aplikasi Android”. Dengan mengintegrasikan fitur Wi-Fi Manager dan memanfaatkan sensor biometrik bawaan smartphone, sehingga sistem ini dapat memberikan fleksibilitas dalam penempatan perangkat serta menjamin keamanan akses yang aman dan praktis.

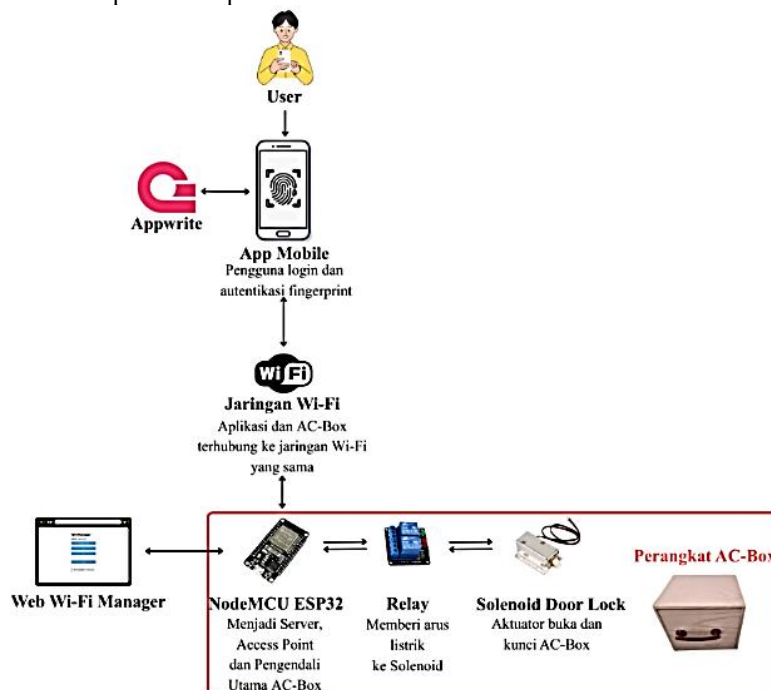
2. METODE PENELITIAN

Metodologi penelitian memuat langkah-langkah yang dilakukan dalam penelitian, diawali dengan studi literatur, analisis kebutuhan, perancangan sistem, implementasi sistem, pengujian sistem serta kesimpulan dan saran. Adapun uraian dari masing-masing langkah metode penelitian adalah sebagai berikut:

1. Studi Literatur: Pada tahap ini merupakan tahap mencari dan mengumpulkan referensi ilmiah yang relevan dengan topik penelitian termasuk jurnal, buku, dan dokumen lain mengenai sistem keamanan berbasis IoT, Wi-Fi Manager pada ESP32, autentikasi *fingerprint* melalui smartphone, verifikasi akun menggunakan Appwrite serta pembuatan aplikasi Android menggunakan Flutter.
2. Analisis Kebutuhan: Pada tahap ini dilakukan analisa kebutuhan untuk membangun sistem. Analisa kebutuhan pada penelitian terdapat kebutuhan perangkat keras dan kebutuhan perangkat lunak. Kebutuhan perangkat keras yang dibutuhkan dalam membangun sistem ini antara lain NodeMCU ESP32, relay, *solenoid door lock* dan adaptor 12V.

Kebutuhan perangkat lunak yang dibutuhkan dalam membangun arsitektur sistem ini antara lain Flutter, Visual Studio Code, Arduino IDE dan Appwrite.

3. Perancangan Sistem: Penelitian ini membangun sistem AC-Box merupakan sistem kontrol akses berbasis IoT untuk kotak penyimpanan yang menggabungkan Wi-Fi Manager pada ESP32 untuk mengatur konfigurasi jaringannya tanpa *hardcode* serta menggunakan autentikasi *fingerprint* untuk keamanan akses melalui aplikasi Android di smartphone. Saat pertama kali dinyalakan ESP32 membentuk *access point* bernama “AC-Box_Setup”. Pengguna menghubungkan smartphone ke *access point* yang dibuat oleh AC-Box agar aplikasi Android dan perangkat AC-Box berada dalam jaringan yang sama. Pengguna lalu diarahkan ke *captive portal* sebuah web tempat bagi pengguna memilih jaringan Wi-Fi dan memasukkan kredensial Wi-Fi. Setelah terkonfigurasi ESP32 menyimpan dan terhubung ke jaringan tersebut. Untuk membuka kotak pengguna login di aplikasi Android via Appwrite juga autentikasi *fingerprint* untuk mengakses perangkat AC-Box. Jika autentikasi berhasil aplikasi Android mengirim perintah “buka” ke IP lokal ESP32 dengan protokol HTTP. ESP32 aktifkan relay sehingga solenoid door lock terbuka selama lima detik. Gambaran deskripsi sistem AC-Box dapat dilihat pada Gambar 1.



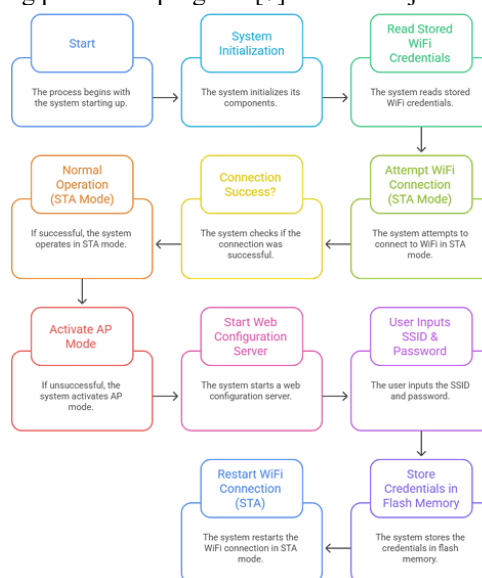
Gambar 1. Deskripsi Sistem AC-Box

4. Implementasi Sistem: Tahap implementasi sistem merupakan realisasi dari rancangan sistem yang telah dibuat. Pada tahap ini terdapat hasil perakitan fisik perangkat keras sesuai skematik koneksi antara NodeMCU ESP32, relay, dan *solenoid door lock*. *Firmware* ESP32 dikembangkan menggunakan Arduino IDE dengan *library* WiFiManager, WebServer dan beberapa *library* lainnya sesuai kebutuhan sistem. Aplikasi Android dikembangkan di Visual Studio Code menggunakan *framework* Flutter sehingga memiliki fitur seperti login, pencarian perangkat, akses dengan autentikasi *fingerprint* dan mengirim perintah buka ke AC-Box. Seluruh komunikasi antara aplikasi dan NodeMCU ESP32 berlangsung dalam jaringan yang sama dan lokal.
5. Pengujian Sistem: Tahap pengujian sistem dilakukan setelah implementasi sistem selesai dengan tujuan untuk memverifikasi bahwa seluruh fungsionalitas sistem berjalan sesuai rancangan. Pengujian ini mencakup beberapa aspek mulai dari kinerja Wi-Fi Manager pada ESP32 dapat terkoneksi dengan jaringan Wi-Fi yang ditentukan, pengukuran *delay* saat proses akses AC-Box agar mengetahui apakah sistem responsif dan berjalan dengan baik tanpa error dan akurasi autentikasi *fingerprint* pada aplikasi di smartphone.
6. Kesimpulan dan Saran: Tahap akhir dari penelitian tugas akhir ini yaitu menarik kesimpulan setelah menganalisis seluruh data dan mendapat hasil pengujian. Tujuannya adalah untuk menjawab permasalahan dan tujuan penelitian ini. Selanjutnya dibuat saran untuk pengembangan sistem AC-Box dimasa mendatang.

2.1. Wi-Fi Manager

Wi-Fi Manager merupakan modul yang terdapat pada mikrokontroler ESP32 berfungsi untuk mengelola koneksi ke jaringan Wi-Fi yang tersedia [6]. Proses kerja Wi-Fi Manager dimulai dengan inisialisasi mikrokontroler untuk membaca kredensial jaringan dari memori. ESP32 kemudian mencoba terhubung dalam mode *Station* (STA) ke Wi-Fi tujuan. Jika koneksi gagal karena data tidak valid atau jaringan tidak ditemukan, sistem secara otomatis beralih ke mode *Access Point* (AP) dan mengaktifkan *captive portal*. Melalui antarmuka web tersebut, pengguna dapat memasukkan SSID dan *password* baru yang kemudian disimpan ke dalam memori flash. Setelah data terkirim, sistem akan melakukan *restart* otomatis untuk memutuskan mode AP dan memulai

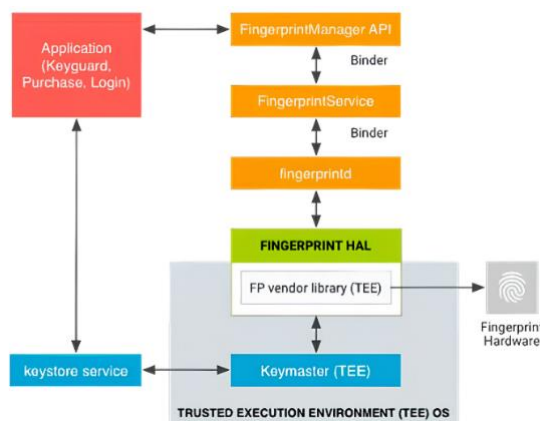
ulang koneksi dalam mode *Station*. Mekanisme ini memastikan perangkat tetap fleksibel dan dapat terhubung ke jaringan baru tanpa perlu melakukan pemrograman ulang pada kode program [7]. Proses kerja Wi-Fi Manager dapat dilihat pada Gambar 2.



Gambar 2. Kinerja Wi-Fi Manager

2.2. Autentikasi Fingerprint

Autentikasi *fingerprint* merupakan metode autentikasi identitas yang memanfaatkan karakteristik fisik unik setiap individu yaitu pola dan ciri khas sidik jari yang bersifat permanen, unik, dan tidak dapat dipindahtangankan [8]. Autentikasi *fingerprint* pada Android memanfaatkan sensor biometrik bawaan smartphone. Sensor sidik jari bekerja dengan merekam gambar digital dari pola sidik jari, dan rekaman tersebut akan dijadikan sebagai template biometrik yang digunakan untuk mencocokkan identitas pengguna [9]. Pada Gambar 5 merupakan arsitektur autentikasi *fingerprint* pada sistem operasi Android melibatkan integrasi berlapis antara perangkat keras dan perangkat lunak yang mencakup *Application*, *FingerprintManager API*, *FingerprintService*, *fingerprint daemon* (*fingerprintd*), *Hardware Abstraction Layer* (HAL) berfungsi sebagai penghubung utama antara perangkat keras dan sistem operasi dalam alur komunikasi data dari sensor menuju sistem keamanan Android, dan *Trusted Execution Environment* (TEE) sebuah sistem penyimpanan yang mengisolasi data biometrik sehingga tidak bisa diakses oleh aplikasi atau pihak ketiga. Proses ini diawali dengan permintaan autentikasi dari aplikasi yang dikelola oleh *FingerprintService* melalui mekanisme Binder IPC (*Inter-process communication*) menuju *fingerprintd* dan HAL sebagai penghubung utama ke perangkat keras. Data biometrik kemudian diproses dan diverifikasi secara aman di dalam TEE OS, di mana komponen seperti FP vendor library dan *Keystore* bekerja untuk memastikan bahwa operasi kriptografi serta penyimpanan kunci tetap terisolasi dari sistem operasi utama guna menjaga integritas keamanan [10]. Arsitektur autentikasi fingerprint dapat dilihat pada Gambar 3.

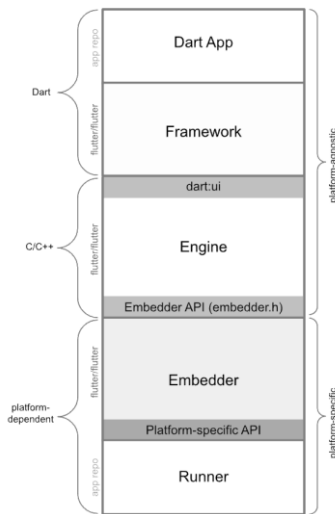


Gambar 3. Arsitektur Autentikasi Fingerprint

2.3. Flutter

Flutter adalah sebuah kerangka kerja yang dibuat oleh Google untuk membangun antarmuka pengguna (UI), baik untuk aplikasi mobile maupun web secara *native*. Dart adalah bahasa pemrograman yang dibuat oleh Google digunakan sebagai bahasa untuk mengembangkan aplikasi di berbagai platform seperti web, mobile, server, dan *Internet of Things* (IoT). Bahasa ini merupakan bahasa yang dominan dalam pengembangan Flutter, juga menjadi bahasa pemrograman yang umum digunakan dalam kerangka kerja tersebut [11]. Flutter adalah *toolkit* pengembangan UI *open-source*. Arsitektur ini terdiri dari beberapa lapisan. Lapisan teratas adalah "Dart App" yang berisi kode aplikasi yang ditulis oleh pengembang. Di bawahnya ada "Framework" yang menyediakan berbagai komponen UI dan utilitas yang ditulis dalam Dart. Lapisan "Engine" bertanggung jawab untuk rendering

grafis dan menangani fungsi inti seperti animasi dan input, yang ditulis dalam C++. Lapisan "Embedder" berfungsi sebagai jembatan antara engine dan platform spesifik, memungkinkan Flutter berjalan di berbagai sistem operasi. Di bagian bawah, "Runner" adalah aplikasi spesifik platform yang menjalankan kode Flutter di perangkat [12]. Lapisan arsitektur Flutter dapat dilihat pada Gambar 4.



Gambar 4. Flutter

2.4. Delay

Delay didefinisikan sebagai durasi waktu yang diperlukan oleh data atau informasi untuk berpindah dari sumber hingga mencapai titik tujuan pengiriman. Dalam ekosistem jaringan pemantauan terhadap besaran *delay* menjadi indikator krusial. Angka ini berfungsi sebagai landasan pengambilan keputusan saat administrator melakukan manajemen atau optimasi infrastruktur jaringan. Berdasarkan standar TIPHON (*Telecommunications and Internet Protocol Harmonization Over Networks*), kualitas jaringan dapat diukur melalui kategori *delay* tertentu. Semakin kecil nilainya semakin baik performa layanan (QoS) yang diberikan [13]. Untuk menghitung nilai rata-rata *delay* dalam suatu pengiriman data dapat dilihat pada Persamaan (1) [13].

$$\text{Rata-rata Delay} = \frac{\text{total delay}}{\text{total jumlah paket}} \quad (1)$$

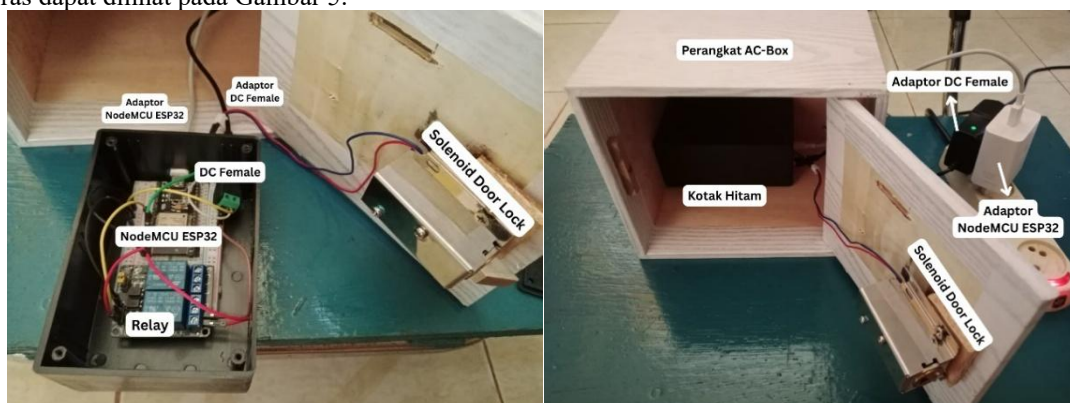
Untuk melihat kategori delay dapat dilihat pada Tabel 1 [13].

Besar Delay	Kategori Delay	Indeks
<150 ms	Sangat Bagus	4
150 – 300 ms	Bagus	3
300 – 450 ms	Sedang	2
>450 ms	Jelek	1

3. HASIL DAN PEMBAHASAN

3.1. Implementasi Perangkat Keras

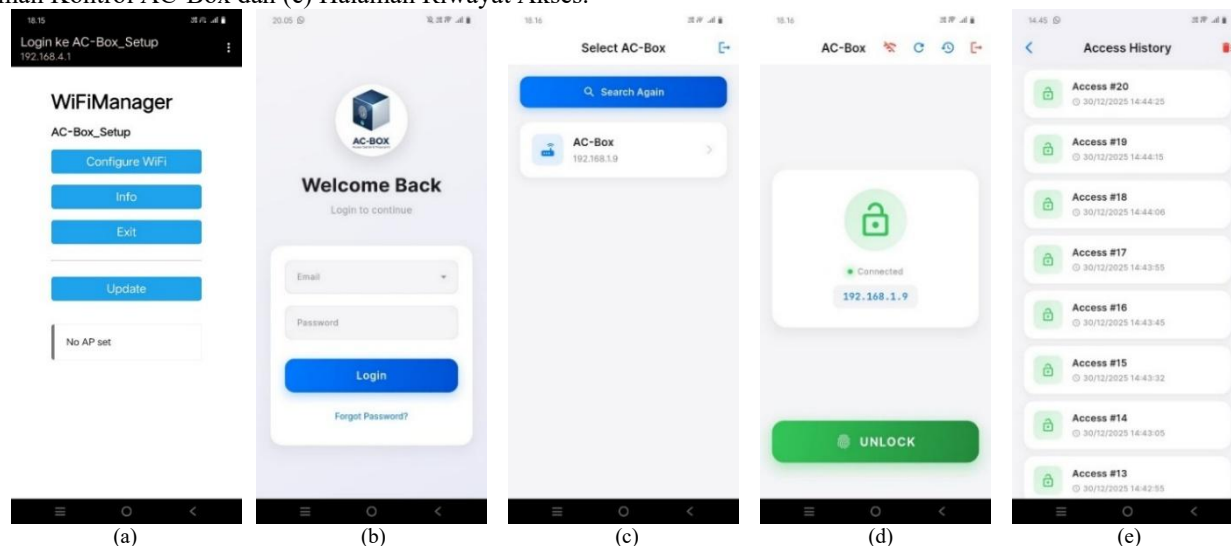
Implementasi perangkat keras dilakukan dengan merakit komponen fisik seperti NodeMCU ESP32, relay, *solenoid door lock*, dan adaptor sesuai skema. Sistem kendali berpusat pada NodeMCU ESP32 yang dikonfigurasi menggunakan *firmware* Arduino IDE agar dapat berkomunikasi dengan server dan aplikasi Android. Ketika perintah buka diterima dari aplikasi, ESP32 akan mendeskripsi perintah dan memberikan sinyal HIGH ke relay untuk mengaktifkan solenoid selama lima detik. Implementasi perangkat keras dapat dilihat pada Gambar 5.



Gambar 5. Implementasi Perangkat Keras

3.2. Implementasi Perangkat Lunak

Implementasi perangkat lunak yaitu realisasi aplikasi Android berbasis Flutter dan *firmware* ESP32 di Arduino IDE untuk sistem kontrol akses AC-Box. Adapun implementasi tampilan dari halaman aplikasi tersebut yaitu tampilan halaman Wi-Fi Manager, halaman *login*, halaman pencarian perangkat, halaman kontrol AC-Box, dan halaman riwayat akses. Implementasi tampilan aplikasi dapat dilihat pada Gambar 6. (a) Halaman Wi-Fi Manager (b) Halaman *Login*, (c) Halaman Pencarian Perangkat, (d) Halaman Kontrol AC-Box dan (e) Halaman Riwayat Akses.



Gambar 6. (a) Halaman Wi-Fi Manager (b) Halaman *Login*, (c) Halaman Pencarian Perangkat, (d) Halaman Kontrol AC-Box dan (e) Halaman Riwayat Akses

3.3. Pengujian Wi-Fi Manager Perpindahan Koneksi Antara Jaringan Wi-Fi Access Point ZTE Dan Hotspot Smartphone

Pengujian Wi-Fi Manager dilakukan pada dua jaringan Wi-Fi yaitu jaringan Wi-Fi menggunakan *access point* ZTE dan jaringan *hotspot* smartphone menggunakan smartphone Realme C75. Kedua jaringan bekerja pada frekuensi 2.4 GHz menyesuaikan pada modul ESP32. Pengujian perpindahan koneksi dilakukan untuk menguji performa dari Wi-Fi Manager dimana dengan fitur ini pengguna dapat mengkoneksikan jaringan Wi-Fi tanpa *hardcoding* serta secara dinamis dapat berpindah koneksi jaringan.

Tabel 2. Pengujian Wi-Fi Manager Perpindahan Koneksi Antara Jaringan Wi-Fi Access Point ZTE Dan Hotspot Smartphone

No	Jenis Pengujian	Waktu Jeda (ms)		Hasil
		Jaringan Wi-Fi AP ZTE	Jaringan Hotspot Smartphone	
1.	Percobaan fitur Wi-Fi Manager ke-1	3.008	708	Berhasil
2.	Percobaan fitur Wi-Fi Manager ke-2	3.003	804	Berhasil
3.	Percobaan fitur Wi-Fi Manager ke-3	3.410	910	Berhasil
4.	Percobaan fitur Wi-Fi Manager ke-4	3.101	710	Berhasil
5.	Percobaan fitur Wi-Fi Manager ke-5	3.208	710	Berhasil
6.	Percobaan fitur Wi-Fi Manager ke-6	3.109	710	Berhasil
7.	Percobaan fitur Wi-Fi Manager ke-7	3.010	910	Berhasil
8.	Percobaan fitur Wi-Fi Manager ke-8	3.208	710	Berhasil
9.	Percobaan fitur Wi-Fi Manager ke-9	3.108	810	Berhasil
10.	Percobaan fitur Wi-Fi Manager ke-10	3.108	711	Berhasil
11.	Percobaan fitur Wi-Fi Manager ke-11	3.108	909	Berhasil
12.	Percobaan fitur Wi-Fi Manager ke-12	3.110	911	Berhasil
13.	Percobaan fitur Wi-Fi Manager ke-13	3.109	706	Berhasil
14.	Percobaan fitur Wi-Fi Manager ke-14	5.008	909	Berhasil
15.	Percobaan fitur Wi-Fi Manager ke-15	3.008	811	Berhasil
16.	Percobaan fitur Wi-Fi Manager ke-16	3.108	910	Berhasil
17.	Percobaan fitur Wi-Fi Manager ke-17	3.209	910	Berhasil
18.	Percobaan fitur Wi-Fi Manager ke-18	3.109	909	Berhasil
19.	Percobaan fitur Wi-Fi Manager ke-19	3.008	809	Berhasil
20.	Percobaan fitur Wi-Fi Manager ke-20	3.008	710	Berhasil

21.	Percobaan fitur Wi-Fi Manager ke-21	3.008	910	Berhasil
22.	Percobaan fitur Wi-Fi Manager ke-22	3.009	911	Berhasil
23.	Percobaan fitur Wi-Fi Manager ke-23	3.108	907	Berhasil
24.	Percobaan fitur Wi-Fi Manager ke-24	3.109	811	Berhasil
25.	Percobaan fitur Wi-Fi Manager ke-25	3.008	710	Berhasil
26.	Percobaan fitur Wi-Fi Manager ke-26	3.108	709	Berhasil
27.	Percobaan fitur Wi-Fi Manager ke-27	3.008	808	Berhasil
28.	Percobaan fitur Wi-Fi Manager ke-28	3.009	810	Berhasil
29.	Percobaan fitur Wi-Fi Manager ke-29	3.108	710	Berhasil
30.	Percobaan fitur Wi-Fi Manager ke-30	3.008	711	Berhasil
Rata-rata Waktu Jeda		3.151,4	806,1	

Berdasarkan data hasil pengujian pada Tabel 2 terlihat perbedaan kecepatan saat perangkat AC-Box mencoba berpindah jaringan internet. Pengujian pada jaringan Wi-Fi *access point* ZTE menunjukkan waktu koneksi yang lebih lambat dengan rata-rata 3.151,4 ms, sedangkan pada jaringan *hotspot* smartphone proses koneksi berlangsung jauh lebih responsif dengan rata-rata hanya 806,1 ms.

3.4. Pengujian Akses Perangkat AC-box Menggunakan Jaringan Wi-Fi Access Point ZTE

Pada Tabel 3 menjelaskan bagaimana hasil dari pengujian akses AC-Box menggunakan jaringan Wi-Fi *access point* ZTE. Jenis pengujian yang dilakukan adalah membuka kotak penyimpanan AC-Box melalui aplikasi lalu autentikasi *fingerprint*. Kondisi pintu apakah terbuka atau tidak saat autentikasi *fingerprint* berhasil dan berapa *delay* tiap membuka perangkat AC-Box.

Tabel 3. Pengujian Akses Perangkat AC-box Menggunakan Jaringan Wi-Fi Access Point ZTE

No	Jenis Pengujian	Hasil	Kondisi Pintu	Delay (ms)
1.	Akses AC-Box	Berhasil	Terbuka	480
2.	Akses AC-Box	Berhasil	Terbuka	326
3.	Akses AC-Box	Berhasil	Terbuka	575
4.	Akses AC-Box	Berhasil	Terbuka	262
5.	Akses AC-Box	Berhasil	Terbuka	430
6.	Akses AC-Box	Berhasil	Terbuka	280
7.	Akses AC-Box	Berhasil	Terbuka	329
8.	Akses AC-Box	Berhasil	Terbuka	412
9.	Akses AC-Box	Berhasil	Terbuka	546
10.	Akses AC-Box	Berhasil	Terbuka	640
11.	Akses AC-Box	Berhasil	Terbuka	279
12.	Akses AC-Box	Berhasil	Terbuka	314
13.	Akses AC-Box	Berhasil	Terbuka	316
14.	Akses AC-Box	Berhasil	Terbuka	462
15.	Akses AC-Box	Berhasil	Terbuka	419
16.	Akses AC-Box	Berhasil	Terbuka	401
17.	Akses AC-Box	Berhasil	Terbuka	217
18.	Akses AC-Box	Berhasil	Terbuka	383
19.	Akses AC-Box	Berhasil	Terbuka	475
20.	Akses AC-Box	Berhasil	Terbuka	375
21.	Akses AC-Box	Berhasil	Terbuka	330
22.	Akses AC-Box	Berhasil	Terbuka	486
23.	Akses AC-Box	Berhasil	Terbuka	241
24.	Akses AC-Box	Berhasil	Terbuka	406
25.	Akses AC-Box	Berhasil	Terbuka	414
26.	Akses AC-Box	Berhasil	Terbuka	300
27.	Akses AC-Box	Berhasil	Terbuka	367
28.	Akses AC-Box	Berhasil	Terbuka	312

29.	Akses AC-Box	Berhasil	Terbuka	382
30.	Akses AC-Box	Berhasil	Terbuka	367
Rata-rata				384,2

Berdasarkan data pada Tabel 3, pengujian akses perangkat AC-Box yang dilakukan sebanyak 30 kali percobaan menghasilkan nilai rata-rata *delay* sebesar 384,2 ms. Jika merujuk pada standar kualitas layanan TIPHON pada Tabel 1, performa akses menggunakan jaringan Wi-Fi *access point* ZTE ini dikategorikan ke dalam kelompok Sedang karena berada pada rentang nilai 300 ms hingga 450 ms.

3.5. Pengujian Akses Perangkat AC-Box Menggunakan *Hotspot* Smartphone

Pada Tabel 4 menjelaskan bagaimana hasil dari pengujian akses AC-Box menggunakan jaringan *hotspot* smartphone.

Tabel 4. Pengujian Akses Perangkat AC-Box Menggunakan *Hotspot* Smartphone

No	Jenis Pengujian	Hasil	Kondisi Pintu	Delay (ms)
1.	Akses AC-Box	Berhasil	Terbuka	685
2.	Akses AC-Box	Berhasil	Terbuka	475
3.	Akses AC-Box	Berhasil	Terbuka	618
4.	Akses AC-Box	Berhasil	Terbuka	727
5.	Akses AC-Box	Berhasil	Terbuka	880
6.	Akses AC-Box	Berhasil	Terbuka	613
7.	Akses AC-Box	Berhasil	Terbuka	339
8.	Akses AC-Box	Berhasil	Terbuka	548
9.	Akses AC-Box	Berhasil	Terbuka	577
10.	Akses AC-Box	Berhasil	Terbuka	834
11.	Akses AC-Box	Berhasil	Terbuka	567
12.	Akses AC-Box	Berhasil	Terbuka	562
13.	Akses AC-Box	Berhasil	Terbuka	638
14.	Akses AC-Box	Berhasil	Terbuka	372
15.	Akses AC-Box	Berhasil	Terbuka	446
16.	Akses AC-Box	Berhasil	Terbuka	454
17.	Akses AC-Box	Berhasil	Terbuka	712
18.	Akses AC-Box	Berhasil	Terbuka	483
19.	Akses AC-Box	Berhasil	Terbuka	609
20.	Akses AC-Box	Berhasil	Terbuka	385
21.	Akses AC-Box	Berhasil	Terbuka	324
22.	Akses AC-Box	Berhasil	Terbuka	601
23.	Akses AC-Box	Berhasil	Terbuka	592
24.	Akses AC-Box	Berhasil	Terbuka	526
25.	Akses AC-Box	Berhasil	Terbuka	527
26.	Akses AC-Box	Berhasil	Terbuka	786
27.	Akses AC-Box	Berhasil	Terbuka	577
28.	Akses AC-Box	Berhasil	Terbuka	452
29.	Akses AC-Box	Berhasil	Terbuka	468
30.	Akses AC-Box	Berhasil	Terbuka	623
Rata-rata				566,7

Berdasarkan data pada Tabel 4, pengujian akses perangkat AC-Box menggunakan jaringan *hotspot* smartphone yang dilakukan sebanyak 30 kali percobaan menghasilkan nilai rata-rata *delay* sebesar 566,7 ms. Jika merujuk pada standar kualitas layanan TIPHON pada Tabel 1, performa akses menggunakan jaringan ini dikategorikan ke dalam kelompok Jelek karena nilainya berada di atas ambang batas 450 ms.

3.6. Pengujian Keamanan Mekanisme Autentikasi *Fingerprint*

Pengujian keamanan pada mekanisme autentikasi *fingerprint* dengan menggunakan 3 sampel orang yang didefinisikan Orang 1, Orang 2 dan Orang 3. Data sidik jari yang terdaftar ada 5 sidik jari yaitu jari manis kiri dan telunjuk kanan Orang 1, jari jempol kiri dan manis kanan Orang 2 lalu jari tengah kanan Orang 3. Data sidik jari yang tidak terdaftar adalah 25 sidik jari selain sidik jari yang didaftarkan. Hasil pengujian keamanan pada mekanisme autentikasi *fingerprint* dapat dilihat pada Tabel 5.

Tabel 5. Pengujian Keamanan Mekanisme Autentikasi *Fingerprint*

No	Sampel	Jenis Jari	Hak Akses	Kondisi Kotak	Keterangan
1.	Orang 1	Kelingking Kanan	Tidak	Tertutup	Berhasil
2.	Orang 1	Manis Kanan	Tidak	Tertutup	Berhasil
3.	Orang 1	Tengah Kanan	Tidak	Tertutup	Berhasil
4.	Orang 1	Telunjuk Kanan	Ya	Terbuka	Berhasil
5.	Orang 1	Jempol Kanan	Tidak	Tertutup	Berhasil
6.	Orang 1	Jempol Kiri	Tidak	Tertutup	Berhasil
7.	Orang 1	Telunjuk Kiri	Tidak	Tertutup	Berhasil
8.	Orang 1	Tengah Kiri	Tidak	Tertutup	Berhasil
9.	Orang 1	Manis Kiri	Ya	Terbuka	Berhasil
10.	Orang 1	Kelingking Kiri	Tidak	Tertutup	Berhasil
11.	Orang 2	Kelingking Kanan	Tidak	Tertutup	Berhasil
12.	Orang 2	Manis Kanan	Ya	Terbuka	Berhasil
13.	Orang 2	Tengah Kanan	Tidak	Tertutup	Berhasil
14.	Orang 2	Telunjuk Kanan	Tidak	Tertutup	Berhasil
15.	Orang 2	Jempol Kanan	Tidak	Tertutup	Berhasil
16.	Orang 2	Jempol Kiri	Ya	Terbuka	Berhasil
17.	Orang 2	Telunjuk Kiri	Tidak	Tertutup	Berhasil
18.	Orang 2	Tengah Kiri	Tidak	Tertutup	Berhasil
19.	Orang 2	Manis Kiri	Tidak	Tertutup	Berhasil
20.	Orang 2	Kelingking Kiri	Tidak	Tertutup	Berhasil
21.	Orang 3	Kelingking Kanan	Tidak	Tertutup	Berhasil
22.	Orang 3	Manis Kanan	Tidak	Tertutup	Berhasil
23.	Orang 3	Tengah Kanan	Ya	Terbuka	Berhasil
24.	Orang 3	Telunjuk Kanan	Tidak	Tertutup	Berhasil
25.	Orang 3	Jempol Kanan	Tidak	Tertutup	Berhasil
26.	Orang 3	Jempol Kiri	Tidak	Tertutup	Berhasil
27.	Orang 3	Telunjuk Kiri	Tidak	Tertutup	Berhasil
28.	Orang 3	Tengah Kiri	Tidak	Tertutup	Berhasil
29.	Orang 3	Manis Kiri	Tidak	Tertutup	Berhasil
30.	Orang 3	Kelingking Kiri	Tidak	Terbuka	Berhasil

Berdasarkan data hasil pengujian pada Tabel 5, sistem keamanan *fingerprint* pada AC-Box terbukti bekerja dengan akurat. Pengujian ini dilakukan dengan mencoba 30 jenis sidik jari berbeda dari sampel tiga orang untuk memastikan hanya jari yang sudah terdaftar saja yang bisa membuka kotak. Hasilnya sistem sukses mengenali 5 jari yang punya hak akses dan langsung membuka kunci solenoid. Sementara itu 25 jari lainnya yang tidak terdaftar langsung ditolak oleh sistem, sehingga aplikasi tidak mengirim perintah apa pun dan kotak tetap terkunci. Hal ini menunjukkan bahwa keamanan AC-Box sudah berjalan dengan baik karena hanya bisa dibuka oleh orang yang mempunyai akses.

4. KESIMPULAN

Penelitian ini berhasil mengimplementasikan sistem AC-Box dengan fitur Wi-Fi Manager yang memungkinkan konfigurasi jaringan dilakukan secara dinamis tanpa *hardcoding*. ESP32 terbukti mampu beralih secara stabil antara *Access Point* dan *Station Mode* dengan rata-rata waktu koneksi tercepat mencapai 806,1 ms pada jaringan *hotspot* smartphone dan 3.151,4 ms pada jaringan Wi-Fi *access point* ZTE. Pengujian mekanisme autentikasi *fingerprint* melalui aplikasi Android berjalan dengan baik di mana sistem berhasil memvalidasi sidik jari terdaftar dan menolak seluruh akses tidak sah. Meskipun performa *delay* bervariasi sesuai kualitas jaringan dengan kategori "Sedang" pada Wi-Fi *access point* ZTE (384,2 ms) dan "Jelek" pada jaringan *hotspot* smartphone (566,7 ms) berdasarkan standar TIPHON, seluruh proses kendali perangkat tetap berjalan konsisten tanpa kegagalan

sistem. Untuk pengembangan selanjutnya, disarankan melakukan optimasi performa melalui penggunaan protokol transmisi data lainnya seperti WebSocket atau MQTT. Sistem ini juga dapat dikembangkan untuk *multi-device* agar satu aplikasi dapat mengontrol beberapa unit AC-Box sekaligus. Fokus pada minimasi *delay* dan penggunaan protokol selain HTTP sangat penting untuk meningkatkan pengalaman pengguna (*user experience*) serta keandalan sistem dalam berbagai kondisi infrastruktur jaringan yang kompleks.

DAFTAR PUSTAKA

- [1] T. Wisjnuadji, N. Arsanto, and P. Hanang, "KOTAK PENYIMPANAN DENGAN SISTEM KEAMANAN BERBASIS ARDUINO," *Seminar Nasional Riset dan Inovasi Teknologi*, Jan. 2022.
- [2] D. D. Laksana and U. Zaky, "SISTEM MONITORING DAN PEMBERIAN PAKAN OTOMATIS PADA PETERNAKAN BURUNG DENGAN KONFIGURASI WI-FI DINAMIS ESP32," *Jurnal Informatika Teknologi dan Sains*, 2024.
- [3] F. E. Febriansyah, M. S. Raisyah, M. R. Kurniawan, D. Sakethi, and B. Hermanto, "Pengembangan Prototipe Alat Pengusir Tikus Berbasis Internet of Things dengan Integrasi Notifikasi API WhatsApp," *Jurnal Pepadun*, vol. 5, no. 3, pp. 203–212, 2024.
- [4] P. Shirisha, G. Satish Kumarr, K. Shivanjan, K. Shiva Rama Krishna, and K. Vineela, "IoT based wifi fingerprint door lock system with raspberry pi & webcam," *MATEC Web of Conferences*, vol. 392, p. 01066, 2024, doi: 10.1051/mateconf/202439201066.
- [5] I. T. Wenewasti, I. Ruslianto, and H. Muhandi, "Coding : Jurnal Komputer dan Aplikasi Implementasi Protokol WebSocket Pada Sistem Keamanan Safebox Menggunakan Kode OTP (One-Time Password)," *Jurnal Komputer dan Aplikasi*, 2024.
- [6] E. Marevac, E. Kadišić, N. Živić, N. Buzadija, and S. Lemeš, "Framework Design for the Dynamic Reconfiguration of IoT-Enabled Embedded Systems and 'On-the-Fly' Code Execution," *Future Internet*, vol. 17, no. 1, Jan. 2025, doi: 10.3390/fi17010023.
- [7] R. Nisa, E. D. Suryanto, E. Sipahutar, A. Budiansyah, and R. A. Candra, "Design and Implementation of a WiFi Manager System on the ESP8266 Module for IoT Applications," *Global Advances in Science, Engineering & Technology (GASET)*, vol. 1, no. 2, pp. 51–60, Feb. 2026, doi: 10.62671/gaset.v1i2.248.
- [8] V. Kumar, A. Saini, R. Nayak, and P. Kaur, "Biometric Fingerprint Authentication," *International Conference on Computing Sciences*, May 2023, [Online]. Available: <https://ssrn.com/abstract=4488021>
- [9] S. R. T. Bryan, Dasril, and H. Abduh, "Rancang Bangun Sistem Keamanan Sepeda Motor Menggunakan Fingerprint Berbasis Arduino," *Jurnal CoSciTech (Computer Science and Information Technology)*, vol. 5, no. 1, pp. 119–125, May 2024, doi: 10.37859/coscitech.v5i1.6294.
- [10] A. L. Tanzila and A. M. Reski, "Penerapan Hubungan Hardware-Software pada Sistem Operasi Android dalam Pengelolaan Sensor Fingerprint," *Polygon : Jurnal Ilmu Komputer dan Ilmu Pengetahuan Alam*, vol. 3, no. 6, pp. 50–59, Oct. 2025, doi: 10.62383/polygon.v3i6.817.
- [11] Muh. A. K. Lutfi, M. Mukramin, and D. Dasril, "Aplikasi Mobile Assistant Masa Kehamilan Berbasis Android," *Jurnal CoSciTech (Computer Science and Information Technology)*, vol. 5, no. 1, pp. 91–100, May 2024, doi: 10.37859/coscitech.v5i1.6812.
- [12] Flutter, "Flutter architectural overview," <https://docs.flutter.dev/resources/architectural-overview>.
- [13] R. F. Bari, A. Solehudin, and N. Heryana, "Analisis Quality of Service (QoS) Jaringan Internet Berbasis Wireless Local Area Network pada Layanan Indihome," *Jurnal Ilmiah Wahana Pendidikan*, Jul. 2022.