

## Rancang Bangun Smart Door Lock IoT Berbasis ESP32 Menggunakan Autentikasi NFC Perangkat Bergerak

Irfan Dwi Rangga Premana<sup>1\*</sup>, Aditiya Rasendra Putra<sup>2</sup>, Rahadyan Bintang Pamungkas<sup>3</sup>, Pramono<sup>4</sup>

<sup>1-4</sup>Fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Duta Bangsa, Surakarta, Indonesia

Email: <sup>1</sup>[irfandwiranggap@gmail.com](mailto:irfandwiranggap@gmail.com)\*, <sup>2</sup>[oppoadityaputra@gmail.com](mailto:oppoadityaputra@gmail.com), <sup>3</sup>[rahadyanbintang18@email.com](mailto:rahadyanbintang18@email.com),

<sup>4</sup>[pramono@udb.ac.id](mailto:pramono@udb.ac.id)

(\* : coresponding author)

**Abstrak**—Keamanan akses fisik ruangan memerlukan sistem yang andal dan terdesentralisasi. Penelitian ini merancang purwarupa sistem kendali akses pintu cerdas bernama "Aegis Door" yang memanfaatkan mikrokontroler ESP32 dan teknologi *Near Field Communication* (NFC) pada perangkat Android, menghilangkan ketergantungan pada modul RFID konvensional yang rentan manipulasi fisik. Sistem mikrokontroler menggunakan arsitektur *multithreading* FreeRTOS untuk memisahkan beban kerja komunikasi jaringan (*Task Cloud*) dan kontrol perangkat keras (*Loop Hardware*) yang dijumpai oleh antrian data *QueueHandle\_t*. Mekanisme mekanis mengintegrasikan aktuator motor servo 140 derajat, *solenoid door lock*, dan sensor PIR ganda yang dilengkapi fitur *fail-safe auto-reverse* anti-jepit. Pada sisi klien, aplikasi Android berbasis Kotlin beroperasi menggunakan metode otentikasi *zero-click* dengan memanfaatkan *intent filter* NDEF\_DISCOVERED, mengirimkan *Device ID* (UUID) dan *Secret Key* secara *Machine-to-Machine* (M2M) via API Blynk. Pengujian menunjukkan sistem sangat stabil tanpa *Watchdog Reset*, responsif, tangguh terhadap serangan *brute-force* melalui fitur pembekuan 30 detik, serta sangat hemat memori (RAM) pada *smartphone* berkat eksekusi *finishAndRemoveTask()*.

**Kata Kunci:** ESP32; FreeRTOS; Kotlin; Kunci Pintu Cerdas; NFC Android

**Abstract**—Physical room access security requires a reliable and decentralized system. This research designs a prototype of a smart door access control system called "Aegis Door" which utilizes the ESP32 microcontroller and Near Field Communication (NFC) technology on Android devices, eliminating reliance on conventional RFID modules prone to physical manipulation. The microcontroller system employs a FreeRTOS multithreading architecture to separate the workload of network communication (Task Cloud) and hardware control (Loop Hardware) bridged by a QueueHandle\_t data queue. The mechanical mechanism integrates a 140-degree servo motor actuator, solenoid door lock, and dual PIR sensors equipped with a fail-safe auto-reverse feature. On the client side, the Kotlin-based Android application operates using a zero-click authentication method utilizing the NDEF\_DISCOVERED intent filter, transmitting the Device ID (UUID) and Secret Key machine-to-machine (M2M) via the Blynk API. Tests show the system is highly stable without Watchdog Resets, responsive, resilient to brute-force attacks through a 30-second freeze feature, and highly memory efficient on smartphones due to the finishAndRemoveTask() execution.

**Keywords:** ESP32; FreeRTOS; Kotlin; Smart Door Lock; Android NFC

### 1. PENDAHULUAN

Perkembangan *Internet of Things* (IoT) telah mendorong transformasi yang masif pada infrastruktur sistem keamanan akses fisik. Transisi dari sistem kunci mekanis konvensional menuju kunci pintar (*smart door lock*) telah menjadi standar baru (Haris Asysyauqi et al., 2025). Namun, sebagian besar sistem akses pintu yang beredar saat ini masih menempatkan modul perangkat keras pembaca RFID di luar ruangan. Pendekatan desentralisasi ini sangat rentan terhadap insiden manipulasi perangkat keras (*vandalisme*), penyadapan data kelistrikan, hingga teknik penyalinan kartu atau *RFID cloning* (Aziz et al., 2025). Sebelumnya, pengembangan sistem pengaman pintu telah banyak dilakukan, misalnya menggunakan mikrokontroler Arduino dan modul *remote RF* konvensional (Yusuf et al., 2020). Namun, arsitektur IoT modern menuntut integrasi sistem keamanan rumah cerdas (*smart home security*) yang lebih terukur dan terhubung ke internet (Hoque & Davidson, 2019).

Sebagai solusi teknis, sistem "Aegis Door" diusulkan untuk mengatasi kerentanan fisik tersebut dengan memindahkan medium otentikasi ke dalam sistem operasi *smartphone* pengguna memanfaatkan protokol pasif *Near Field Communication* (NFC) (Hadi Subowo & Yulianti, 2021). Penggunaan Tag NFC pada area pintu direduksi murni hanya sebagai pemicu pasif mekanis

(*Physical Trigger*), sedangkan otorisasi tingkat lanjut ditangani oleh aplikasi klien berbasis Android secara *Serverless* (Daharmi et al., 2022).

Selain tantangan pada lapisan otentikasi, kendala stabilitas sering terjadi pada sisi mikrokontroler. ESP32 yang difungsikan untuk menangani banyak proses secara *real-time*—seperti komunikasi *Firebase*, *polling* API Telegram, pembacaan sensor ganda, dan kontrol aktuator—kerap mengalami tabrakan memori yang berujung pada kerusakan interupsi (*Watchdog Reset*) (Nguyen et al., 2023). Oleh karena itu, penelitian ini mengimplementasikan sistem operasi waktu-nyata (FreeRTOS) untuk menerapkan *multithreading* asinkron (Ramadhanil et al., 2025). Artikel ini menyajikan arsitektur Aegis Door secara komprehensif, mulai dari rekayasa *hardware* anti-jepit berlapis, stabilitas distribusi memori komputasi pada ESP32, hingga optimasi *zero-click execution* pada aplikasi Android berbasis Kotlin.

## 2. METODE PENELITIAN

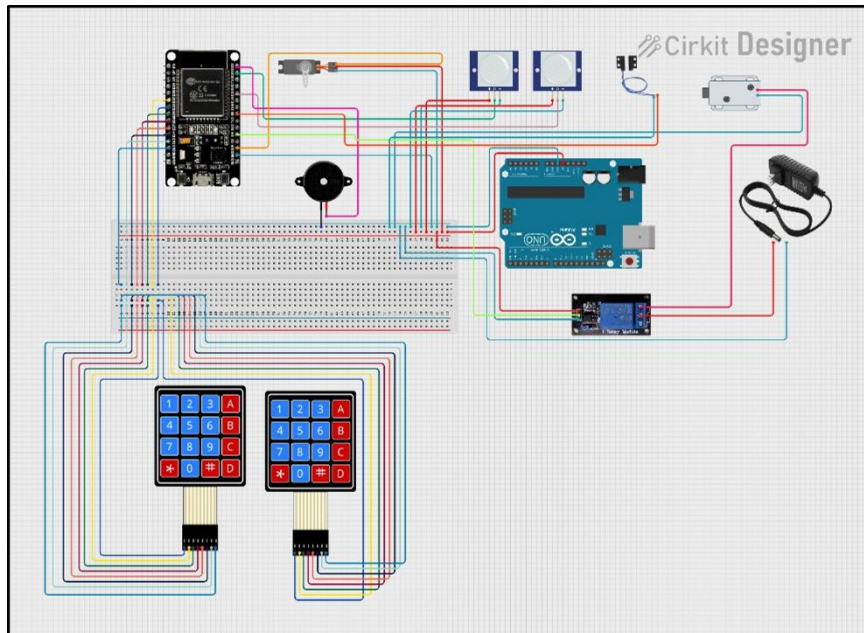
Metode penelitian difokuskan pada perancangan arsitektur berjenjang yang dibagi menjadi tiga sektor utama: perangkat keras mekanis, perangkat lunak mikrokontroler (FreeRTOS), dan spesifikasi klien Android.

### 2.1 Perancangan Perangkat Keras dan Logika Mekanis

Pusat komputasi dikendalikan oleh ESP32 Dev Module. Spesifikasi perangkat keras menggunakan Motor Servo sebagai aktuator pintu dengan rentang operasional 0° (buka penuh) hingga 140° (tutup rapat). *Solenoid Door Lock* beroperasi independen melalui modul *Relay*. Rincian komponen dapat dilihat pada Tabel 1.

**Tabel 1.** Spesifikasi Komponen Perangkat Keras

| Komponen Utama     | Fungsi Dalam Sistem                            | Spesifikasi dan Keteerangan             |
|--------------------|------------------------------------------------|-----------------------------------------|
| ESP32 Dev Module   | Pengontrol utama & Pemroses multithreading     | Menjalankan FreeRTOS (Core 0 & Core 1)  |
| Motor Servo        | Aktuator penggerak mekanis pintu               | Rentang kalibrasi sudut 0° hingga 140°  |
| Solenoid Door Lock | Pengunci pintu mekanis independen              | Diaktifkan terpisah via modul Relay     |
| Sensor PIR (Ganda) | Pendeteksi objek di area ayunan pintu          | Memicu fitur Fail-Safe AutoReverse      |
| Magnetic Switch    | Pendeteksi status rapatnya pintu               | Mengontrol batas waktu Grace Period     |
| Matriks Keypad 4x4 | Modul input PIN sekunder                       | Otentikasi offline berlatensi rendah    |
| Buzzer             | Indikator audio dan alarm keamanan             | Aktif saat Lockdown dan pintu terganjal |
| Tag NFC (Kartu)    | Pemicu fisik ( <i>Physical Trigger</i> ) pasif | Diformat NDEF memuat App Record (AAR)   |
| Smartphone NFC     | Klien otentikasi Zero-Click                    | Pengirim Device ID (UUID) via API Blynk |



Gambar 1. Skema Rangkaian Perangkat Keras Sistem

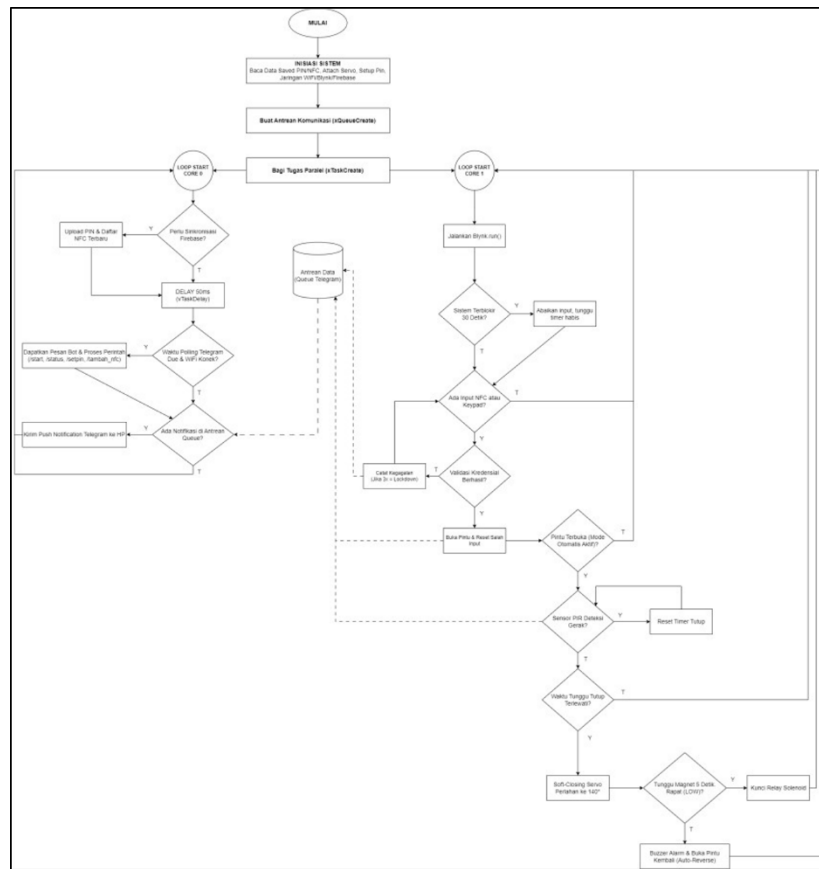
Keamanan mekanis tidak hanya dititikberatkan pada penguncian, tetapi juga pada keselamatan pengguna:

- 1) *Fail-Safe Auto-Reverse* : Menggunakan Sensor PIR Ganda (luar dan dalam). Saat pintu melakukan *soft-closing*, deteksi objek oleh PIR langsung memicu pintu berbalik ke posisi terbuka (Fauzi et al., 2024).
- 2) *Grace Period* : Solenoid tidak akan mengunci sebelum *Magnetic Switch* mendeteksi status pintu sepenuhnya rapat (LOW) untuk mencegah aktuatur *binding* Sistem mengizinkan toleransi ayunan maksimal 5 detik(Maulana et al., n.d.).

## 2.2 Pengembangan Arsitektur Mikrokontroler (FreeRTOS)

Stabilitas sistem dicapai dengan mengeksekusi dua alur (*thread*) paralel secara independen pada *core* yang berbeda.

- a) **Core 0 (Task Cloud)** : Bertanggung jawab penuh pada rutinitas jaringan, yakni melakukan *polling Webhook* perintah Bot Telegram secara berkala dan sinkronisasi perubahan PIN atau daftar UUID ke *Firestore Realtime Database*.
- b) **Core 1 (Loop Hardware)** : Didedikasikan untuk mengeksekusi perangkat keras berlatensi sangat rendah, seperti *debounce* Keypad 150ms dan pemantauan sensor. Kedua inti ini berkomunikasi silang menggunakan memori antrean *QueueHandle\_t*(Brajamusti & Nurjanah, 2023). Sistem juga dibekali lapisan *Anti-Bruteforce*; di mana tiga kali kegagalan otentikasi akan memicu status *Lockdown* otomatis selama 30 detik (mengabaikan pembacaan NFC dan Keypad) dan mengirimkan notifikasi interupsi ke Telegram.

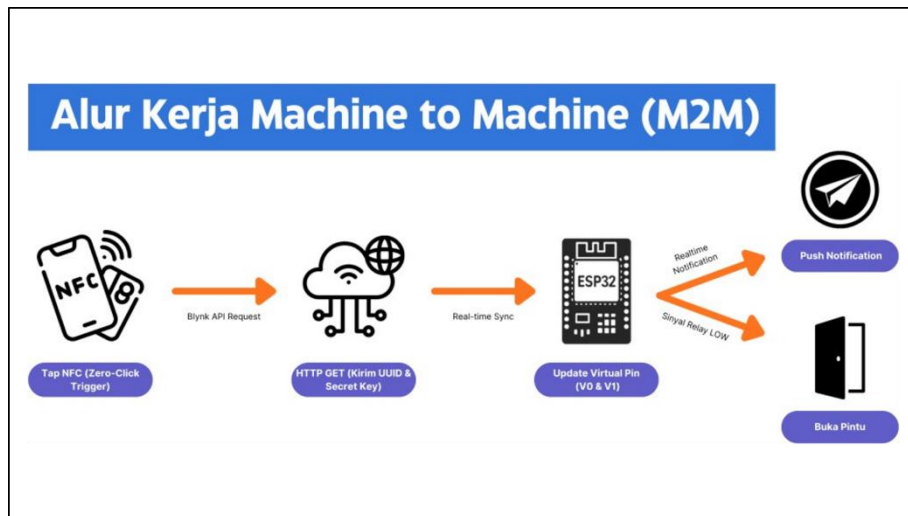


Gambar 2. Diagram Alir Multithreading pada ESP32

## 2.3 Implementasi Aplikasi Klien Android

Aplikasi dirancang menggunakan bahasa pemrograman Kotlin di Android Studio dengan pendekatan arsitektur *Serverless*.

- Zero-Click Execution** : Modul NFC pasif dikonfigurasi dalam format NDEF. *AndroidManifest* klien dilengkapi *intent filter* NDEF\_DISCOVERED dengan *contentType="text/plain"* (Purba et al., 2023). Hal ini memicu fungsi *bypass Intent Chooser*, di mana OS Android otomatis menjalankan *onCreate()* saat ponsel didekatkan ke pintu
- Otorisasi Device-Based** : Kredensial tidak disimpan di mikrokontroler luar, melainkan pada *SharedPreferences smartphone* pengguna dalam bentuk UUID statis. Saat NFC memicu *Intent*, aplikasi mendekripsi *Secret Key* menggunakan Base64 dan menembakkan HTTP GET secara *Machine-to-Machine (M2M)* menuju API *Cloud Blynk*.
- Optimasi RAM** : Permintaan HTTP berjalan pada *Background Thread*. Begitu *response header* HTTP 200 OK divalidasi, sistem memanggil fungsi *finishAndRemoveTask()*, secara instan menghapus instansi aplikasi dari memori RAM.

**Gambar 3.** Alur M2M**Gambar 4.** Antarmuka Klien Android

### 3. ANALISA DAN PEMBAHASAN

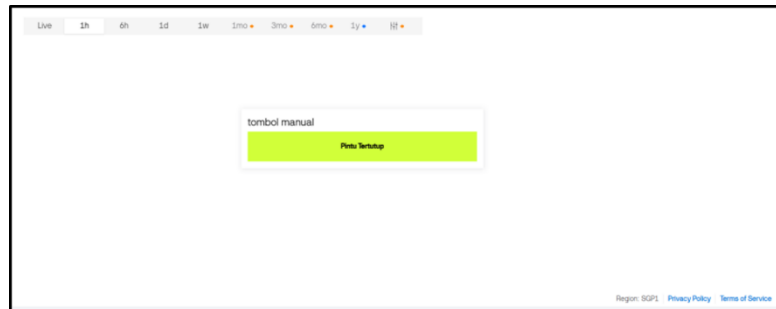
#### 3.1 Pengujian Mekanis dan Manajemen Memori (FreeRTOS)

Penerapan arsitektur asinkron berbasis QueueHandle\_t berhasil mengatasi isu beban komputasi. Saat sistem menerima pesan notifikasi yang harus dikirim ke Telegram, *Core 1* tidak melakukan pengiriman internet secara langsung, melainkan sekadar menitipkannya ke *Queue*. *Core 0* mengambil data dari antrian tersebut di sela-sela waktu istirahat jaringan. Metode ini terbukti menghilangkan 100% kasus *Watchdog Reset* saat sistem disimulasikan menerima beban eksekusi *hardware* dan notifikasi IoT secara simultan.

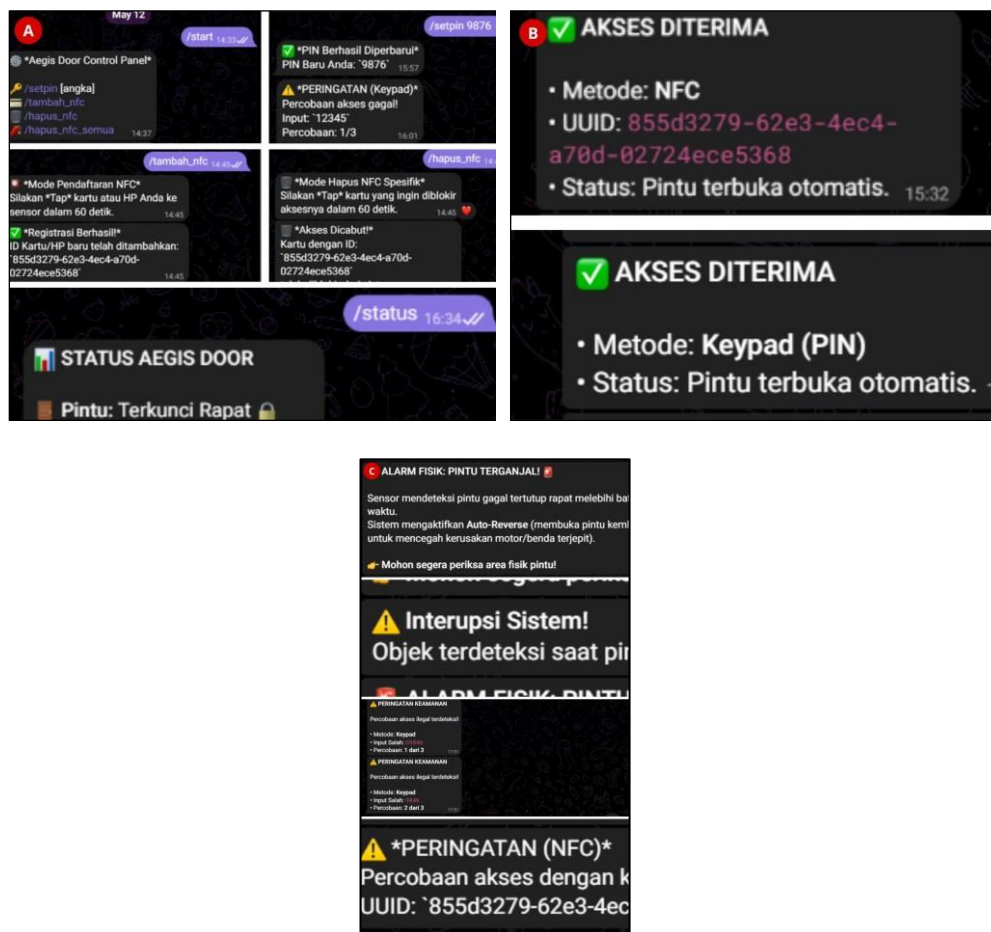
Tabel 2. Hasil Pengujian Sistem

| Skenario Pengujian         | Parameter Keberhasilan                                                                             | Waktu / Tingkat Keberhasilan                    |
|----------------------------|----------------------------------------------------------------------------------------------------|-------------------------------------------------|
| Eksekusi NFC (Zero-Click)  | Pintu terbuka setelah API Blynk ter-Panggil                                                        | < 2 detik (Berhasil 100%)                       |
| Eksekusi Android           | Aplikasi otomatis menutup dan terhapus dari RAM                                                    | < 1 detik setelah HTTP 200                      |
| Fail-Safe Auto-Reverse     | Pintu berbalik saat sensor PIR terhalang                                                           | Berhasil tanpa insiden                          |
| Grace Period (Anti-Ganjil) | Sistem gagal mendapat sinyal Magnetic Switch > 5 detik memicu alarm Buzzer dan membuka ulang pintu | Berhasil (Mencegah kerusakan aktuator/stalling) |
| Serangan Bruteforce        | Sistem menolak input setelah 3 kali gagal                                                          | Lockdown 30 detik (Berhasil)                    |
| Kondisi Tanpa Internet     | Akses offline via Keypad 4x4 tetap responsif                                                       | Responsif (Berhasil 100%)                       |
| Perintah Telegram Bot      | Sistem membalas perintah /status, /setpin, /tambah_nfc, /hapus_nfc, /hapus_nfc_semua               | Berhasil dieksekusi oleh Core 0                 |
| Push Notification Telegram | Laporan riwayat akses (diterima/ditolak), peringatan Lockdown, dan alarm terganjal masuk ke HP     | Terkirim instan via antrian Queue               |
| Kontrol Manual Blynk       | Perintah "Tahan Buka" menimpa mode otomatis                                                        | Aktuator merespons (Berhasil)                   |
| Sinkronisasi Firebase      | Perubahan PIN dan UUID NFC tersimpan di Cloud                                                      | Konsistensi data 100% terjaga                   |

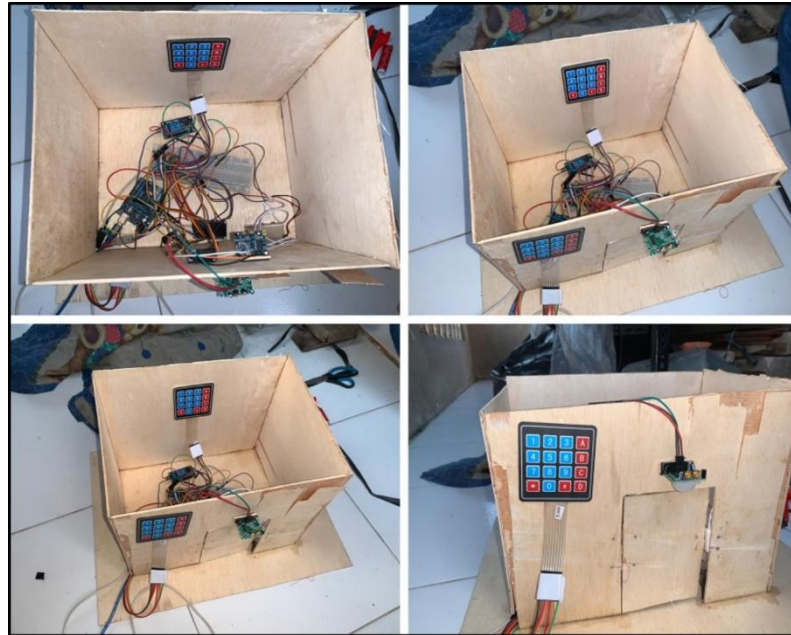
Keberhasilan pengujian komunikasi jaringan jarak jauh dibuktikan melalui responsivitas antarmuka pengendali pada aplikasi Blynk dan riwayat notifikasi komprehensif pada bot Telegram. Pengguna dapat melakukan intervensi mekanis secara *real-time* via *dashboard* Blynk. Sementara itu, sistem secara aktif dan mandiri melaporkan berbagai skenario keamanan melalui *push notification*, mulai dari pencatatan riwayat akses (diterima/ditolak), alarm pintu terganjal, hingga aktivasi protokol *Lockdown* saat terdeteksi indikasi serangan *bruteforce*.



**Gambar 6.** Antarmuka Pengendalian Jarak Jauh via Blynk Cloud



**Gambar 7.** Variasi Notifikasi Bot Telegram: (a) Perintah Interaktif, (b) Akses Diterima, (c) Akses Gagal



**Gambar 8.** Purwarupa Fisik Sistem Aegis Door Terpasang pada Pintu

### 3.2 Pengujian Otentikasi NFC dan Optimasi Android

Otentikasi Zero-Click berjalan mulus berkat implementasi NDEF dan AAR. Penggunaan fungsi `finishAndRemoveTask()` terbukti mengembalikan ruang memori (RAM) pada smartphone pengguna dalam waktu kurang dari 1 detik setelah sinyal HTTP 200 OK dari server Blynk diterima.

### 3.3 Hasil Pengujian Responsibilitas, Keamanan, dan Jaringan

Berdasarkan skenario pengujian yang telah dijalankan, performa keseluruhan sistem dianalisis melalui tiga aspek utama :

- Responsibilitas Sistem** : Waktu respons eksekusi otentikasi NFC (*Zero-Click*) tercatat sangat cepat, yakni kurang dari 2 detik sejak *smartphone* dipindai hingga aktuator membuka pintu. Selain itu, aplikasi Android berhasil melakukan optimasi RAM dengan mengeksekusi fungsi `finishAndRemoveTask()` dalam waktu kurang dari 1 detik pasca menerima respons HTTP 200 OK.
- Ketahanan Keamanan** : Mekanisme pertahanan sistem terbukti tangguh mengatasi anomali. Simulasi serangan tebakan PIN secara acak (*bruteforce*) memicu penguncian sistem (*lockdown*) secara presisi selama 30 detik setelah 3 kali kegagalan beruntun. Pada sisi perangkat keras, pengujian *Fail-Safe Auto-Reverse* dan *Grace Period* berhasil mencegah insiden terjepit maupun kerusakan pada dinamo aktuator.

- c) **Stabilitas Jaringan** : Integrasi komunikasi *Task Cloud* dengan Telegram Bot berjalan tanpa hambatan *Watchdog Reset*. Sistem secara konsisten membalas perintah interaktif (/status, /setpin, /tambah\_nfc) dan berhasil mendorong *push notification* (seperti laporan akses dan peringatan *lockdown*) ke perangkat pengguna secara instan melalui jembatan antrean *QueueHandle\_t*.

#### 4. KESIMPULAN

Sistem kendali akses pintar "Aegis Door" berhasil direkayasa dan diimplementasikan sebagai solusi desentralisasi titik otentikasi keamanan fisik ruangan. Penerapan arsitektur *multithreading* FreeRTOS pada ESP32 terbukti andal dalam menghilangkan risiko *Watchdog Reset* melalui pembagian beban kerja komputasi yang optimal antara *Task Cloud* pada Core 0 dan *Loop Hardware* pada Core 1, dengan sinkronisasi lintas inti yang dijembatani secara aman oleh antrean memori *QueueHandle\_t*. Pada sisi klien, aplikasi Android berbasis Kotlin berhasil mewujudkan otentikasi *zero-click* via filter intent NDEF\_DISCOVERED, sekaligus mengoptimalkan efisiensi memori RAM *smartphone* melalui eksekusi fungsi *finishAndRemoveTask()* pasca transmisi data M2M ke API Blynk. Selain itu, kombinasi fitur pertahanan *anti-bruteforce* (pembekuan akses 30 detik) serta rekayasa mekanis berlapis berupa *Fail-Safe Auto-Reverse* dan *Grace Period* menghasilkan sistem purwarupa yang tidak hanya cerdas, tetapi juga aman bagi keselamatan fisik pengguna serta memiliki ketahanan operasional luring (*offline*) yang responsif.

Sebagai rancangan pekerjaan masa depan untuk pengembangan penelitian ini, disarankan untuk mengeksplorasi penggunaan enkripsi kriptografi asimetris (*asymmetric cryptography*) tambahan pada pertukaran paket data API Blynk untuk memperkuat keamanan transmisi data. Selain itu, integrasi sistem cadangan daya (*Uninterruptible Power Supply*) sangat direkomendasikan guna menjaga kesinambungan dan ketersediaan fitur otentikasi lokal menggunakan matriks keypad 4x4 ketika terjadi pemadaman sumber daya listrik utama.

#### REFERENCES

- Aziz, M. A., Sutikno, T., Yuliansyah, H., Dewi, A. I., Rohmadi, Y. E., & Setiawati, D. (2025). Vulnerability Analysis of Smart Lock Using NIST SP 800-115 Method. *Jurnal Penelitian Pendidikan IPA*, 11(8), 264–272. <https://doi.org/10.29303/jppipa.v11i8.12219>
- Brajamusti, J. K., & Nurjanah, C. K. (2023). RANCANG BANGUN SISTEM SATELIT BUATAN BERBASIS ESP32 DENGAN FITUR KOMUNIKASI MENGGUNAKAN MODUL GSM SIM800L. *Jurnal Elektro Dan Telekomunikasi Terapan*, 10(1), 17. <https://doi.org/10.25124/jett.v10i1.6123>
- Daharmi, R. R., Bhawiyuga, A., & Basuki, A. (2022). Pengembangan IoT Cloud Platform berbasis pada Layanan Serverless Computing (Vol. 6, Number 12). <http://j-ptiik.ub.ac.id>
- Fauzi, A., Sembiring, H., & Kaputama, S. (2024). *Journal of Artificial Intelligence and Engineering Applications Internet-Based Smart Door Design of Things (IOT) with Visitor Access Controller Indoor* (Vol. 4, Number 1). <https://ioinformatic.org/>
- Hadi Subowo, A., & Yulianti, dan. (2021). *Jurnal Teknologi Sistem Informasi dan Aplikasi Perancangan dan Implementasi Sistem Keamanan Kendaraan dengan Teknologi Near Field Communication (NFC)*. 4(4), 260–266. <https://doi.org/10.32493/jtsi.v4i4.14870>
- Haris Asysyauqi, Moh. Ferdi Andriansyah, Lya Nurul Ulla, & Adi Sucipto. (2025). Sistem Keamanan Pintu Otomatis Berbasis IoT dengan Teknologi RFID dan Aplikasi Mobile Menggunakan Metode Fuzzy Mamdani. *Modem : Jurnal Informatika Dan Sains Teknologi*, 3(2), 42–50. <https://doi.org/10.62951/modem.v3i2.405>
- Hoque, M. A., & Davidson, C. (2019). Design and implementation of an IoT-based smart home security system. *International Journal of Networked and Distributed Computing*, 7(2), 85–92. <https://doi.org/10.2991/ijndc.k.190326.004>
- Maulana, F., Praja, D., Mardiansyah, A. Z., & Jatmika, A. H. (n.d.). RANCANG BANGUN SISTEM KEAMANAN PINTU RUMAH MENGGUNAKAN SENSOR MAGNETIK BERBASIS IOT.
- Nguyen, V. K., Tran, V. K., Pham, H., Nguyen, V. M., Nguyen, H. D., & Nguyen, C. N. (2023). A multi-microcontroller-based hardware for deploying Tiny machine learning model. *International Journal of Electrical and Computer Engineering*, 13(5), 5727–5736.

<https://doi.org/10.11591/ijece.v13i5.pp5727-5736>

Purba, P. M., Cipta Amandha, A., Putra, M. A., Harahap, N. M., Siregar, R. A., & Meilina, I. (2023). *Perancangan Sistem Absensi Mahasiswa Menggunakan Teknologi Near Field Communication Berbasis Android* (Vol. 1, Number 3). Pria Mitra Purba.

Ramadhanil, Selamat Meliala, Muhammad, & Yasir Amani. (2025). *MONITORING SUHU OVERHEAT PADA TRANSFORMATOR 1 FASA BERBASIS ESP32 MENGGUNAKAN GOOGLE SPREADSHEET*.

Yusuf, A. I., Samsugi, S., & Trisnawati, F. (2020). SISTEM PENGAMAN PINTU OTOMATIS DENGAN MIKROKONTROLER ARDUINO DAN MODULE RF REMOTE. In *Jurnal Ilmiah Mahasiswa Kendali dan Listrik* (Vol. 1, Number 1). <http://jim.teknokrat.ac.id/index.php/teknikelektro/index>