



**REAL TIME SYSTEM AND INTERNET OF THINGS FINAL PROJECT REPORT
DEPARTMENT OF ELECTRICAL ENGINEERING
UNIVERSITAS INDONESIA**

ATLAS

(Anti-cheat Tracking & Location Attendance System)

GROUP 18

Aliya Rizqiningrum Salamun	2306161813
Bonifasius Raditya Pandu H.	2306242350
Calvin Wirathama Katoroy	2306242395
Zhafira Zahra Alfarisy	2306250636



IoT dan Sistem Waktu Nyata

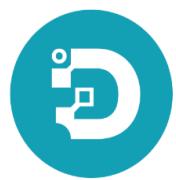
PREFACE

Proyek akhir mata kuliah *Internet of Things* (IoT) atau Sistem Waktu Nyata ini kami merancang **ATLAS (Anti-cheat Tracking & Location Attendance System)**, sebuah solusi yang dirancang untuk mengatasi masalah *proxy attendance* (titip absen) pada sistem presensi manual di lingkungan akademik. ATLAS merupakan solusi presensi IoT **berbasis protokol komunikasi MQTT** dan melalui mekanisme verifikasi ganda, akan mengeliminasi praktik titip absen. Inovasi utama terletak pada penerapan mekanisme **dual verification** mengimplementasikan teknologi fisik seperti **RFID** dan digital seperti **BLE Beacon Signal**, didukung oleh arsitektur *Publish-Subscribe* yang ringan dan *real-time* untuk menjamin kecepatan sinkronisasi data presensi antara ruang kelas dan *dashboard* pengawas.

Laporan ini menguraikan secara komprehensif latar belakang masalah, perancangan solusi, detail implementasi perangkat keras dan perangkat lunak, hingga rencana pengujian dan evaluasi proyek. Proyek ini diharapkan dapat meningkatkan integritas data kehadiran, memberikan *real-time monitoring* bagi dosen, dan menciptakan lingkungan akademik yang lebih fair.

Depok, December 4, 2025

Group 18



IoT dan Sistem Waktu Nyata

TABLE OF CONTENTS

CHAPTER 1.....	5
INTRODUCTION.....	5
1.1 PROBLEM STATEMENT.....	5
1.3 ACCEPTANCE CRITERIA.....	7
1.4 ROLES AND RESPONSIBILITIES.....	7
1.5 TIMELINE AND MILESTONES.....	8
CHAPTER 2.....	9
IMPLEMENTATION.....	9
2.1 HARDWARE DESIGN.....	9
2.2 SOFTWARE DEVELOPMENT.....	11
2.3 HARDWARE AND SOFTWARE INTEGRATION.....	13
CHAPTER 3.....	14
TESTING AND EVALUATION.....	14
3.1 TESTING.....	14
3.2 RESULT.....	16
3.3 EVALUATION.....	18
CHAPTER 4.....	19
CONCLUSION.....	19



IoT dan Sistem Waktu Nyata

CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT

Sistem presensi manual yang mengandalkan tanda tangan rentan terhadap praktik *titip absen*. Fenomena ini terjadi ketika seorang mahasiswa menandatangani daftar hadir untuk temannya yang tidak hadir. Hal ini menimbulkan beberapa dampak negatif signifikan, yaitu:

1. Ketidakakuratan data kehadiran, mengakibatkan Dosen tidak dapat memperoleh data kehadiran yang akurat, yang merupakan dasar penting untuk evaluasi dan penilaian performa mahasiswa.
2. *Lack of Fairness*, merugikan mahasiswa yang hadir dan rajin, menciptakan lingkungan akademik yang tidak *fair* dan merusak motivasi.
3. Penurunan integritas akademik, titip absen secara langsung merusak integritas proses belajar-mengajar dan kredibilitas sistem pendidikan.

Penggunaan sistem presensi elektronik sederhana, seperti hanya menggunakan RFID, seringkali masih memiliki celah, seperti KTM yang bisa dipinjamkan. Oleh karena itu, diperlukan sebuah solusi yang menggabungkan beberapa lapisan verifikasi untuk menjamin kehadiran fisik mahasiswa di dalam ruang kelas.

1.2 PROPOSED SOLUTION

Untuk mengatasi masalah proxy attendance, diusulkan implementasi **ATLAS (Anti-cheat Tracking & Location Attendance System)** berbasis IoT. Sistem ini dirancang dengan **client-server architecture** yang terdiri dari dua komponen utama: **Client Node** sebagai terminal presensi di ruang kelas dan **Admin Node** sebagai dashboard monitoring terpusat untuk dosen.



IoT dan Sistem Waktu Nyata

ATLAS menggunakan **dual-layer verification mechanism** yang menggabungkan autentikasi fisik dan deteksi kedekatan digital untuk memastikan kehadiran yang sah:

1. Physical Authentication Layer

Menggunakan teknologi RFID untuk verifikasi identitas mahasiswa melalui penempelan KTM secara contactless (tanpa sentuh).

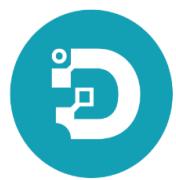
2. Digital Proximity Detection Layer

Mengimplementasikan **BLE (Bluetooth Low Energy) beacon scanning** untuk memastikan kehadiran fisik dari mobile device yang terdaftar milik mahasiswa pada saat yang bersamaan dengan verifikasi RFID.

Kedua node berkomunikasi melalui **MQTT (Message Queuing Telemetry Transport) protocol** over jaringan WiFi untuk transmisi data. Data presensi yang telah divalidasi secara real-time ditransmisikan ke platform monitoring berbasis cloud (Blynk) untuk dapat diakses oleh dosen.

Presensi dianggap valid **hanya jika kedua lapisan verifikasi berhasil secara bersamaan**. Jika kartu RFID terdeteksi namun sinyal BLE beacon yang sesuai tidak ditemukan dalam jarak proximity, sistem akan menolak upaya presensi dan memicu fraud alert, sehingga secara efektif mencegah praktik peminjaman KTM atau proxy attendance.

Sistem ini juga mendukung **remote user registration** dari admin dashboard, memungkinkan dosen untuk mengelola database mahasiswa tanpa perlu akses fisik ke terminal presensi. Untuk proof of concept, data mahasiswa (mapping RFID-to-BLE) disimpan secara lokal pada client node dengan opsi untuk pengembangan di masa depan menggunakan centralized cloud database.



IoT dan Sistem Waktu Nyata

1.3 ACCEPTANCE CRITERIA

The acceptance criteria of this project are as follows:

1. Berhasil memverifikasi presensi (Status OK) hanya ketika *RFID card* dan *BLE UUID* yang terdaftar terdeteksi secara simultan dalam *range* 2 meter.
2. Menolak presensi (Status GAGAL) dan mengaktifkan fraud alert jika hanya kartu *RFID* yang di-*tap* tanpa adanya *BLE UUID* yang terdaftar di sekitarnya.
3. Memastikan komunikasi MQTT antara Client Node dan Admin Node berjalan stabil untuk transmisi data presensi dan perintah registrasi remote.
4. Mengirimkan data presensi yang divalidasi ke Blynk Dashboard (Admin Interface) secara real-time untuk monitoring oleh dosen.
5. Memberikan *feedback* visual (LCD, LED RGB) yang jelas kepada pengguna mengenai status presensi (Berhasil/Gagal/Terlambat/Fraud Alert)

1.4 ROLES AND RESPONSIBILITIES

The roles and responsibilities assigned to the group members are as follows:

Roles	Responsibilities	Person
Concept Developer, Report Paper, Powerpoint Maker	Merancang ide dan konsep awal, menulis laporan proyek, dan membuat PPT.	Aliya Rizqiningrum Salamun
Hardware Designer, Code Writer, dan Circuit Maker	Mengembangkan ide awal dari segi teknis, merancang rangkaian <i>hardware</i> dan membuat <i>code</i> sampai final.	Bonifasius Raditya Pandu Herdianto
Team Leader, Advanced Conceptor, Hardware	Mengembangkan ide awal dari segi teknis, merancang rangkaian	Calvin Wirathama Katoroy



IoT dan Sistem Waktu Nyata

Designer, Code Writer, dan Circuit Maker	<i>hardware</i> dan membuat <i>code</i> sampai final.	
Concept Developer, Report Paper, Powerpoint Maker	Merancang ide dan konsep awal, menulis laporan proyek, dan membuat PPT.	Zhafira Zahra Alfarisy

Table 1. Roles and Responsibilities

1.5 TIMELINE AND MILESTONES

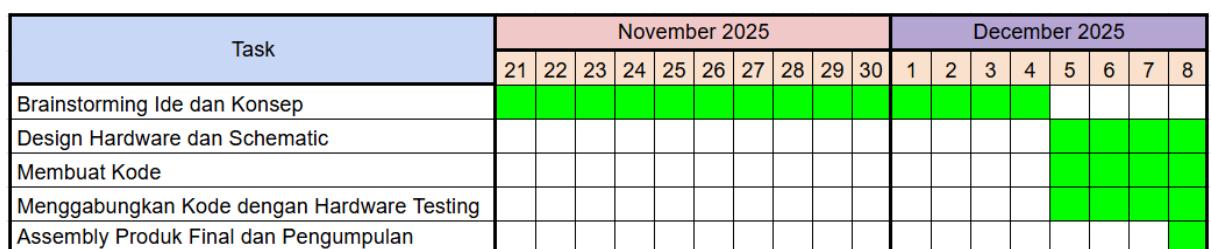


Fig 1. Gantt Chart



IoT dan Sistem Waktu Nyata

CHAPTER 2

IMPLEMENTATION

2.1 HARDWARE DESIGN

Perancangan *hardware* pada proyek ATLAS menggunakan pendekatan arsitektur terpusat (*Centralized Architecture*) dengan topologi *Star*. Proyek ini memanfaatkan protokol **MQTT (Message Queuing Telemetry Transport)** via koneksi WiFi yang memungkinkan komunikasi data *real-time* dengan latensi rendah. Sistem terdiri dari dua rangkaian fisik utama yaitu ***Client Node*** yang ditempatkan di pintu kelas sebagai bagian *scanning* presensi, dan ***Admin Node (Gateway)*** yang berada di sisi dosen sebagai pusat *control* dan perantara ke *Cloud Interface Blynk*. Pemilihan komponen didasarkan pada kebutuhan *Two-Factor Authentication* yaitu RFID dan BLE serta efisiensi sumber daya mikrokontroler.

2.1.1 Component List

- **Client Node (Slave)**

Berfungsi sebagai titik interaksi utama bagi pengguna. Dengan bagian *logic* utamanya adalah **ESP32 DevKit V1** yang dipilih karena memiliki modul WiFi dan *Bluetooth Low Energy* (BLE) terintegrasi dalam satu *chip*, memungkinkan pemindaian *Beacon* dan komunikasi MQTT berjalan bersamaan. *Input* fisik dilakukan dengan **RFID Reader RC522** yang berkomunikasi melalui antarmuka SPI untuk membaca UID KTM. Komponen *display* proyek ini dan seluruh validasi diproses secara *backend*.

- **Admin Node (Master)**

Berfungsi sebagai koordinator proyek. Dengan menggunakan **ESP32 DevKit V1**, *device* ini berperan sebagai *MQTT Subscriber* yang menerima data validasi dari *Client* dan meneruskannya ke **Blynk IoT Platform** untuk visualisasi. Bagian ini berfungsi sebagai *gateway* tanpa memerlukan sensor eksternal.

2.1.2 Schematic and Wiring



IoT dan Sistem Waktu Nyata

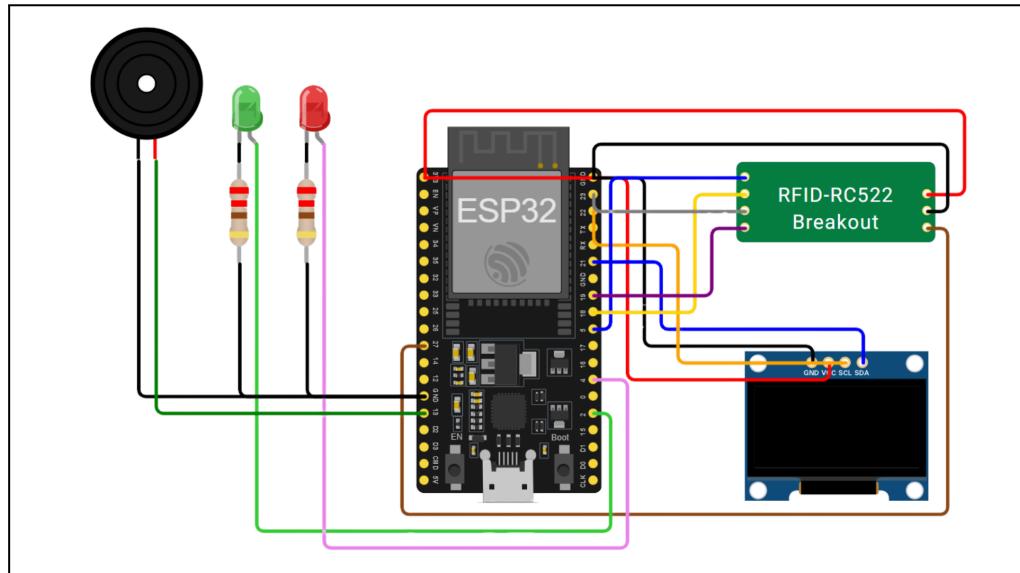


Fig 2. Skema Rangkaian Client Node

Implementasi rangkaian *hardware* pada proyek berfokus pada koneksi antarmuka SPI antara ESP32 dan modul RFID. Dimana konfigurasi pin dimaksimalkan untuk stabilitas komunikasi sensor. Modul RFID RC522 dihubungkan ke ESP32 menggunakan jalur SPI standar yaitu SCK (GPIO 18), MOSI (GPIO 23), dan MISO (GPIO 19). Pin *control SS* (*Chip Select*) dihubungkan ke **GPIO 21** dan pin *RST (Reset)* dihubungkan ke **GPIO 22**.

Strategi *wiring* ini memastikan jalur komunikasi data sensor terhubung langsung ke mikrokontroler tanpa interferensi periferal lain. Sumber daya listrik untuk ESP32 dan RFID diambil dari tegangan 3.3V untuk memastikan level *logic* yang sesuai. Dan indikasi keberhasilan proses sepenuhnya bergantung pada data yang terkirim ke *dashboard* Admin atau melalui pemantauan *Serial Monitor* saat tahap *debugging*. Dan catatan implementasi *buzzer* dan LED hanya dilakukan pada rangkaian *virtual* melalui wokwi sebagai tahapan konsep *hardware design*.



IoT dan Sistem Waktu Nyata

2.2 SOFTWARE DEVELOPMENT

Pengembangan *software* ATLAS dilakukan menggunakan bahasa C++ dengan *framework* Arduino di lingkungan PlatformIO. Struktur *software* dirancang menggunakan **Real-Time Operating System (FreeRTOS)** yang memungkinkan mikrokontroler menangani *multitasking* dan tugas berat seperti memindai sinyal BLE dan menjaga koneksi MQTT secara paralel tanpa saling memblokir (*non-blocking*).

2.2.1 Software Architecture and Libraries

Struktur *software* proyek kami bergantung pada *library* utama sebagai berikut:

- 1) **PubSubClient** digunakan untuk menangani protokol MQTT (Publish/Subscribe).
- 2) **NimBLE-Arduino** digunakan untuk implementasi *Bluetooth Low Energy* (BLE) yang efisien memori untuk memindai *Beacon* pengguna.
- 3) **MFRC522** digunakan sebagai *driver* untuk komunikasi SPI dengan modul RFID Reader.
- 4) **BlynkSimpleEsp32** digunakan untuk mengelola koneksi dan pertukaran data dengan aplikasi *mobile* Blynk IoT.
- 5) **Preferences** digunakan untuk menyediakan antarmuka penyimpanan *Non-Volatile Storage* (NVS) dan menyimpan *database* pengguna (UID, NPM, UUID) secara lokal pada *flash memory* ESP32 agar data *persistent* saat *restart*.

2.2.2 Software Program

Logic program dirancang dengan implementasi pada modul-modul praktikum yang telah dipelajari. Inisialisasi periferal dasar seperti konfigurasi SPI merupakan penerapan **Modul 1 (Introduction & I/O)**. Struktur inti program mengimplementasikan **Modul 2 (Task Management)** dengan membagi proses menjadi tiga *task* FreeRTOS utama yaitu rfidScanTask sebagai pembacaan KTM, validationProcessTask sebagai *logic* verifikasi, dan mqttTask sebagai komunikasi jaringan.



IoT dan Sistem Waktu Nyata

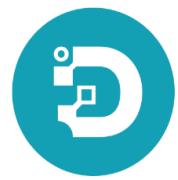
Dan untuk menjamin keamanan data antar *task*, kode program menerapkan **Modul 3 (Memory Management & Queue)** dengan menggunakan *Queue* untuk mengirim data UID KTM dari *task* sensor ke *task* validasi. Kemudian, **Modul 4 (Synchronization)** diterapkan menggunakan *Mutex Semaphore* untuk mencegah *Race Condition* saat mengakses *Preferences* atau mengirim paket MQTT. Dan fitur anti-cheating menggunakan *logic pemindaian sinyal* via pada **Modul 6 (Bluetooth & BLE)**, dan seluruh pertukaran data antara Client dan Admin dilakukan melalui protokol standar sesuai **Modul 7 (MQTT & WiFi)**. Terakhir, visualisasi data pada *dashboard Admin* merupakan implementasi **Modul 9 (IoT Platform)** via Blynk.

ID	Name	Type	Color	Data Type	Unit	Is Raw	Min	Max	Decimals	Default Value	Automation Type	Condition	Action	Actions
1	Register Mode	V0	Red	Integer		false	0	1	-	0	Switch	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Edit
2	NPM Input	V1	Cyan	String		false	-	-	-	-	Color	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Edit
3	UID Display	V2	Light Blue	String		false	-	-	-	-	Color	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Edit
4	Attendance Log	V3	Blue	String		false	-	-	-	-	Color	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Edit
5	Total Scans Today	V4	Yellow	Integer		false	0	999	-	0	Switch	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Edit
6	Lan Status	V5	Grey	String		false	-	-	-	-	Color	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Edit
7	Students In Class	V6	Green	Integer		false	0	100	-	0	Switch	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Edit
8	Invalid Attempts	V7	Red	String		false	-	-	-	-	Color	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Edit
9	Student List	V8	Cyan	String		false	-	-	-	-	Color	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Edit
10	Clear Counter	V9	Orange	Integer		false	0	1	-	0	Switch	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Edit
11	Clear Database	V10	Brown	Integer		false	0	1	-	0	Switch	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Edit
12	Average Duration	V11	Purple	Float		false	-	-	-	-	Color	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Edit

Fig 3. Konfigurasi Datastreams pada Blynk

REGISTER MODE	NPM	TOTAL SCANS	IN CLASS	Avg. DURATION
<input checked="" type="radio"/> OFF	296645200	0	0	-
GENERATED UID 00000000-0000-0000-002306242350				
STUDENTS IN CLASS		INVALID ATTEMPTS		
ATTENDANCE LOG		CLEAR COUNTER		
CLEAR DATABASE		Image Button Copy		

Fig 4. User Interface Dashboard Admin pada Blynk



IoT dan Sistem Waktu Nyata

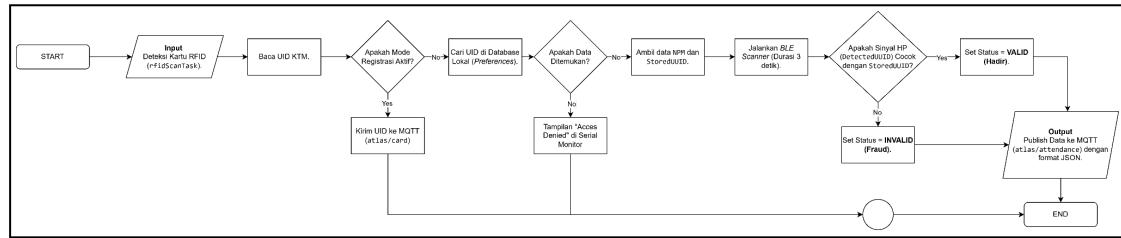


Fig 5. Flowchart Logika Dual-Factor pada Client Node

2.3 HARDWARE AND SOFTWARE INTEGRATION

Tahap integrasi menggabungkan *firmware* yang telah di *compile* ke dalam setiap bagian *hardware* masing-masing. Fokus utama integrasi adalah sinkronisasi *logic* antara *Client Node* dan *Admin Node*. Pengujian dilakukan untuk memastikan bahwa pembacaan sensor RFID pada *Client* mampu memicu rangkaian proses validasi di mulai dari pengecekan *database* lokal, pemindaian BLE selama 3 detik, hingga pengiriman status presensi ke *Admin Node* via topik MQTT (*atlas/attendance*).

Tantangan dalam integrasi proyek ini adalah fitur "Remote Registration" tanpa *interface* visual di sisi *Client*. Proses ini mengharuskan sinkronisasi presisi di mana perintah dari aplikasi Blynk pada *Admin Node* harus dikirim via MQTT, diterima oleh *Client Node* untuk mengaktifkan mode baca KTM, menangkap UID, mengirimkannya kembali ke *Admin* untuk digabungkan dengan data NPM, dan akhirnya dikembalikan ke *Client* untuk disimpan di *Flash Memory*. Hasil pengujian membuktikan bahwa penggunaan *Queue* dan *Mutex* FreeRTOS berhasil menjaga stabilitas proyek selama proses pertukaran data intensif ini, memastikan data presensi terekam akurat di *dashboard* Blynk meskipun tanpa indikator fisik di perangkat *scanning*.



IoT dan Sistem Waktu Nyata

CHAPTER 3

TESTING AND EVALUATION

3.1 TESTING

Tahap pengujian dilakukan menjadi tiga tahapan utama yang merepresentasikan alur dan cara kerja rangkaian asli sebagai berikut:

1) Tahap Pertama - Konektivitas & Inisialisasi Sistem

Memastikan bahwa *Client Node* dan *Admin Node* berhasil terhubung ke jaringan WiFi dan *Broker HiveMQ* saat pertama kali dinyalakan. Indikator keberhasilan dengan munculnya log "Connected" pada *Serial Monitor* dan status *online* pada aplikasi Blynk.

2) Tahap Kedua- Remote Registration

Sinkronisasi dua arah antara *dashboard Admin* dan perangkat *Client*. Dengan langkah pengujian meliputi:

- Admin mengaktifkan "Register Mode" melalui *switch V0* di aplikasi Blynk.
- Admin memasukkan NPM mahasiswa pada kolom input V1.
- Pengguna menempelkan KTM baru pada *Client Node*.
- Sistem diharapkan mengirimkan UID KTM ke *Admin*, menggabungkannya dengan NPM dan UUID *default*, lalu mengirim balik ke *Client* untuk disimpan di memori *Preferences*.

3) Tahap Ketiga - Validasi Presensi

- Kondisi Valid: Pengguna menempelkan KTM yang terdaftar sambil membawa *smartphone* dengan aplikasi *Beacon Simulator* aktif (UUID sesuai) dalam radius < 1 meter.



IoT dan Sistem Waktu Nyata

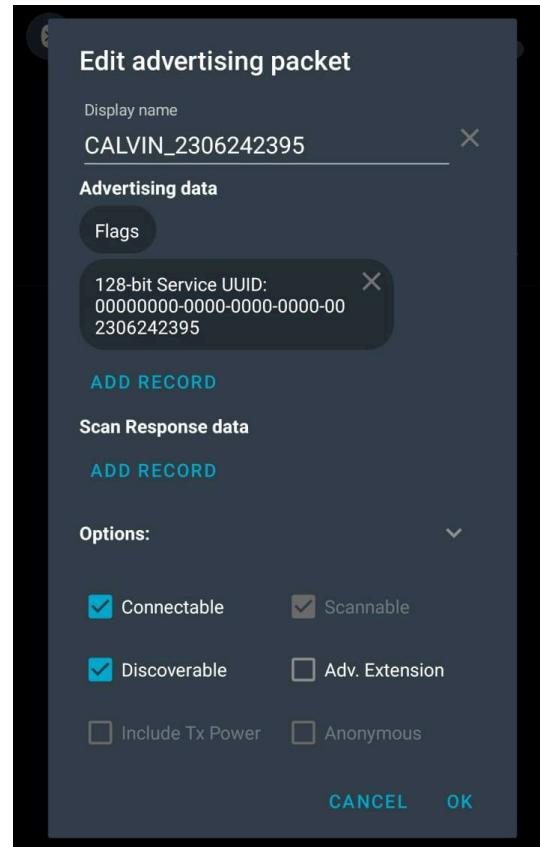


Fig 6. Konfigurasi UUID

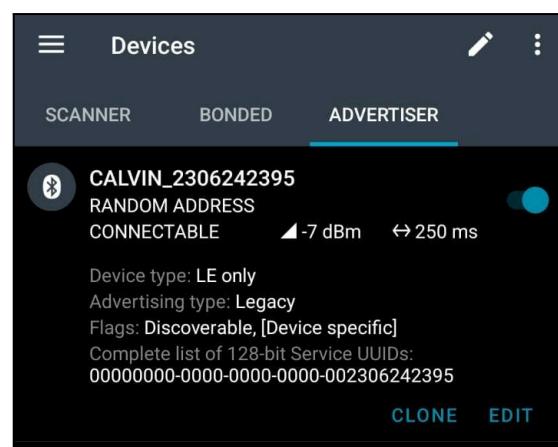
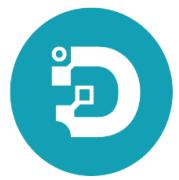


Fig 7. Beacon Simulator - Mode Advertising Aktif

```
[CARD] 04 6C 0D 9A 10 18 90
[BLE] Scanning 3s...
VALID #5 [MASUK] - 2025-12-08 23:02:14 | NPM: 2306242350 | RSSI: -58 dBm
```



IoT dan Sistem Waktu Nyata

Fig 8. Pengujian Kondisi Valid (Masuk)

```
[CARD] 04 6C 0D 9A 10 18 90
[BLE] Scanning 3s...
VALID #6 [KELUAR] - 2025-12-08 23:02:25 | NPM: 2306242350 | RSSI: -62 dBm
```

Fig 9. Pengujian Kondisi Valid (Keluar)

- Kondisi Fraud (Titip Absen): Pengguna menempelkan KTM yang terdaftar, namun *smartphone* pemilik KTM dimatikan atau berada di luar jangkauan (BLE Mismatch).

```
[CARD] 04 6C 0D 9A 10 18 90
[BLE] Scanning 3s...
INVALID - 2025-12-08 23:02:20 | BLE mismatch
```

Fig 8. Pengujian Kondisi Fraud

3.2 RESULT

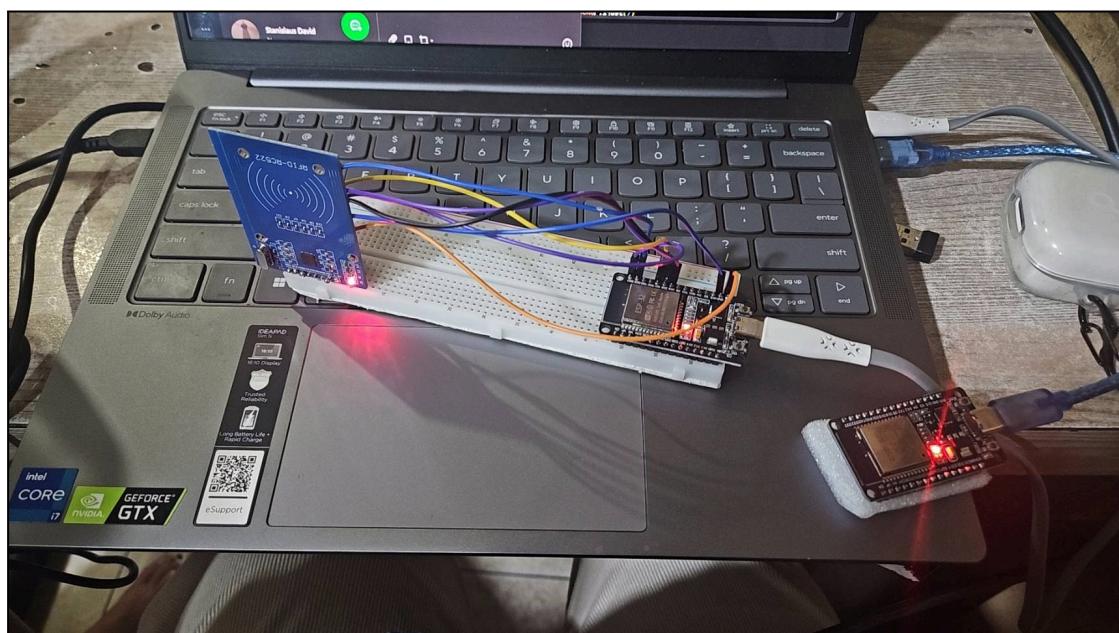


Fig 9. Foto Rangkaian Asli



IoT dan Sistem Waktu Nyata

Hasil pengujian keseluruhan menunjukkan bahwa proyek ATLAS mampu berjalan sesuai dengan spesifikasi perancangan. Pada *Client Node* dan *Admin Node* secara otomatis menjalankan prosedur inisialisasi *hardware* dan berhasil menjalin koneksi ke *access point* serta *Broker MQTT* dalam waktu singkat. Indikator keberhasilan awal ini terlihat dari kesiapan sistem menerima perintah *control* dari *dashboard admin* tanpa adanya latensi. Ketika mode registrasi diaktifkan oleh *admin* melalui aplikasi Blynk, sistem merespons secara *real-time* dengan mengubah status operasional *Client Node* untuk siap menerima pendaftaran KTM (Kartu Tanda Mahasiswa) baru. Alur pertukaran data dua arah di mulai dari pembacaan UID KTM, penggabungan data identitas mahasiswa, hingga penyimpanan permanen ke memori *flash* berjalan dengan baik, memastikan integritas data pengguna terjaga meskipun perangkat mengalami *restart*.

Pada tahap validasi kehadiran, algoritma *Dual-Factor Authentication* yang diterapkan terbukti efektif dalam menegakkan aturan keamanan. Sistem mampu membedakan secara presisi antara kehadiran yang sah dan percobaan manipulasi. Dalam kondisi normal di mana pengguna menempelkan KTM disertai sinyal identitas digital dari *smartphone* (via aplikasi *Beacon Simulator*), sistem langsung memverifikasi validitas tersebut dan memperbarui *log* kehadiran di *dashboard* dosen secara instan. Sedangkan, ketika sistem pada skenario percobaan *cheating* seperti KTM yang digunakan tanpa kehadiran pemiliknya yang tidak mendeteksi sinyal BLE, maka logika pertahanan sistem secara otomatis menolak akses tersebut dan mengirimkan notifikasi *fraud* ke server. Responsivitas dan akurasi ini memverifikasi integrasi antara manajemen *task* FreeRTOS, protokol komunikasi MQTT, dan antarmuka *cloud blynk* telah berhasil menciptakan presensi cerdas fungsional.



IoT dan Sistem Waktu Nyata

```

KARTU TERDETEKSI
UID: A1 B2 C3 D4

[VALIDATION] Checking database...
NPM: 2306242350
Stored UUID: 00000000-0000-0000-0000-002306242350

[BLE] Scanning...
[BLE] Found: 00000000-0000-0000-0000-002306242350, RSSI: -65

HASIL VALIDASI
Status: ✓✓✓ VALID ✓✓✓
RSSI : -65 dBm
✓ Kartu: OK
✓ BLE : OK
ABSENSI DITERIMA

```

Fig 10. Hasil Validasi Absensi Berhasil

3.3 EVALUATION

Implementasi proyek ATLAS telah berhasil memenuhi sebagian besar Acceptance Criteria yang ditetapkan dan efektif mengatasi Problem Statement awal, dimana mekanisme Dual Verification (RFID dan BLE Beacon) terbukti fungsional karena dalam testing sistem mampu membedakan presensi yang valid dengan upaya fraud sehingga secara langsung memitigasi risiko titip absen yang merupakan tujuan utama proyek, sementara keberhasilan implementasi Task Management FreeRTOS membuat proses BLE Scanning dapat berjalan paralel dengan MQTT connection tanpa mengganggu kestabilan sistem/blocking. Komunikasi WiFi/MQTT juga dapat mentransmisikan data presensi dan fraud alert ke Blynk Dashboard secara real-time sehingga dosen dapat melakukan real-time monitoring secara akurat, serta fitur penting Remote Registration Control dari Admin Node berhasil diimplementasikan yang memvalidasi arsitektur terpusat dan meningkatkan user experience administrasi.

Namun, proyek ini masih memiliki beberapa limitasi yang perlu diperhatikan untuk pengembangan lebih lanjut, yaitu ketiadaan feedback visual langsung pada Client Node (LCD/LED) yang membuat user experience kurang optimal karena pengguna harus mengandalkan monitoring melalui Serial Monitor atau menunggu



IoT dan Sistem Waktu Nyata

konfirmasi dari admin, penyimpanan database secara lokal menggunakan Preferences.h yang membatasi skalabilitas sistem untuk deployment multi-classroom yang membutuhkan centralized database management, serta proximity detection range yang terbatas pada radius 2 meter yang berpotensi menghasilkan false negative jika terdapat interferensi sinyal BLE di lingkungan dengan banyak perangkat wireless.



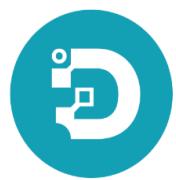
IoT dan Sistem Waktu Nyata

CHAPTER 4

CONCLUSION

Proyek ATLAS (Anti-cheat Tracking & Location Attendance System) telah berhasil dirancang dan diimplementasikan sebagai solusi berbasis IoT untuk mengatasi permasalahan proxy attendance pada sistem presensi akademik melalui penerapan dual-layer verification mechanism yang menggabungkan RFID physical authentication dan BLE digital proximity detection, sehingga sistem ini mampu menjamin kehadiran fisik mahasiswa secara valid serta secara efektif mencegah praktik card sharing atau titip absen.

Implementasi client-server architecture dengan protokol MQTT terbukti mampu menyediakan komunikasi yang stabil dan real-time antara terminal presensi di ruang kelas dengan dashboard monitoring dosen, serta didukung oleh penggunaan FreeRTOS untuk task management yang memastikan sistem dapat menjalankan multiple concurrent processes secara andal tanpa mengalami penurunan performa. Hasil pengujian menunjukkan bahwa proyek ini telah memenuhi seluruh acceptance criteria yang ditetapkan, di mana sistem mampu membedakan secara akurat antara legitimate attendance dan fraud attempt, mengirimkan data presensi secara real-time, serta secara otomatis mengaktifkan fraud alert ketika terdeteksi adanya upaya kecurangan. Untuk pengembangan kedepannya, lebih baik ditambah visual feedback interface pada Client Node, implementasi centralized cloud database untuk meningkatkan scalability, dan peningkatan security terhadap kemungkinan spoofing UUID agar ATLAS dapat di-deploy dan sebagai standardized attendance system di lingkungan akademik.



IoT dan Sistem Waktu Nyata

REFERENCES

[1] Digilab DTE, “Module 1 - Introduction to SMP with RTOS,” *Internet of Things*.

Available:

<https://learn.digilabdte.com/books/internet-of-things/chapter/module-1-introduction-to-smp-with-rtos>.

[2] Digilab DTE, “Module 2 - Task Management,” *Internet of Things*. Available:

<https://learn.digilabdte.com/books/internet-of-things/chapter/module-2-task-management>.

[3] Digilab DTE, “Module 3 - Memory Management & Queue,” *Internet of Things*.

Available:

<https://learn.digilabdte.com/books/internet-of-things/chapter/module-3-memory-management-queue>.

[4] Digilab DTE, “Module 4 - Deadlock & Synchronization,” *Internet of Things*.

Available:

<https://learn.digilabdte.com/books/internet-of-things/chapter/module-4-deadlock-synchronization>.

[5] Digilab DTE, “Module 6 - Bluetooth & BLE,” *Internet of Things*. Available:

<https://learn.digilabdte.com/books/internet-of-things/chapter/module-6-bluetooth-ble>.

[6] Digilab DTE, “Module 7 - MQTT, HTTP & Wi-Fi,” *Internet of Things*. Available:

<https://learn.digilabdte.com/books/internet-of-things/chapter/module-7-mqtt-http-wifi>.

[7] Digilab DTE, “Module 9 - IoT Platforms: Blynk and Node-RED,” *Internet of Things*.

Available:

<https://learn.digilabdte.com/books/internet-of-things/chapter/module-9-iot-platforms-blynk-and-red-node>.



IoT dan Sistem Waktu Nyata

APPENDICES

Appendix A: Library Dependencies

- **platformio.ini**

```
[env:slave]
platform = espressif32
board = esp32doit-devkit-v1
framework = arduino
lib_deps =
    miguelbalboa/MFRC522
    h2zero/NimBLE-Arduino
    bblanchon/ArduinoJson
monitor_speed = 115200
build_src_filter = +<slave_main.cpp>

[env:master]
platform = espressif32
board = esp32doit-devkit-v1
framework = arduino
lib_deps =
    blynkbb/Blynk
    bblanchon/ArduinoJson
monitor_speed = 115200
build_src_filter = +<master_main.cpp>
```