



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

CLASSROOM MANAGER

GROUP 23

Muhammad Hilmi Al Muttaqi	2306267082
R. Aisha Syauqi Ramadhani	2306250554
Wesley Frederick Oh	2306202763
Muhammad Riyanto Satrio Wibowo	2306229323

PREFACE

Puji dan syukur kami panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, kami dapat menyelesaikan laporan proyek akhir berjudul "Classroom Manager". Penulisan laporan ini disusun untuk memenuhi salah satu syarat kelulusan mata kuliah Sistem Waktu Nyata dan IoT.

Pada kesempatan ini, kami ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada Bapak Fransiskus Astha Ekadiyanto, S.T., M.Sc. selaku dosen pengampu, serta para asisten laboratorium yang telah memberikan bimbingan teknis maupun teori selama pelaksanaan proyek ini. Apresiasi juga kami berikan kepada rekan-rekan satu tim dan teman-teman seperjuangan yang telah saling mendukung dan berkontribusi sehingga berbagai kendala teknis dalam pengembangan alat ini dapat teratasi dengan baik.

Kami menyadari bahwa laporan dan purwarupa sistem ini masih memiliki kekurangan karena keterbatasan pengetahuan dan pengalaman kami. Oleh karena itu, kritik dan saran yang membangun sangat kami harapkan guna penyempurnaan di masa mendatang. Akhir kata, kami berharap laporan ini dapat memberikan manfaat bagi pembaca dan menambah wawasan mengenai penerapan teknologi IoT untuk manajemen ruang kelas yang efisien.

Depok, December 06, 2025

Group 23

TABLE OF CONTENTS

CHAPTER 1.....	4
INTRODUCTION.....	4
1.1 PROBLEM STATEMENT.....	4
1.3 ACCEPTANCE CRITERIA.....	5
1.4 ROLES AND RESPONSIBILITIES.....	6
1.5 TIMELINE AND MILESTONES.....	7
CHAPTER 2.....	8
IMPLEMENTATION.....	8
2.1 HARDWARE DESIGN AND SCHEMATIC.....	8
2.2 SOFTWARE DEVELOPMENT.....	9
2.3 HARDWARE AND SOFTWARE INTEGRATION.....	12
CHAPTER 3.....	14
TESTING AND EVALUATION.....	14
3.1 TESTING.....	14
3.2 RESULT.....	16
3.3 EVALUATION.....	20
CHAPTER 4.....	22
CONCLUSION.....	22
REFERENCES.....	23
APPENDICES.....	25

CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT

Pengelolaan fasilitas ruang kelas yang efisien dan aman menjadi aspek krusial dalam operasional institusi pendidikan. Saat ini, sistem manajemen ruang kelas masih sangat bergantung pada metode konvensional yang melibatkan peran petugas (*janitor*) secara fisik. Ketergantungan ini menimbulkan masalah efisiensi operasional, di mana petugas harus berkeliling secara manual untuk membuka atau mengunci pintu setiap kali jadwal perkuliahan dimulai atau berakhir. Selain memakan waktu, metode ini rentan terhadap kesalahan manusia (*human error*), seperti keterlambatan pembukaan kelas atau kelalaian dalam mengunci ruangan yang tidak terpakai.

Selain aspek operasional, penggunaan kunci fisik tradisional memiliki kelemahan signifikan dari sisi keamanan dan logistik. Kunci fisik mudah hilang, tertukar, atau bahkan diduplikasi oleh pihak yang tidak bertanggung jawab tanpa terdeteksi. Tidak adanya sistem pencatatan (*log*) otomatis membuat institusi kesulitan melacak siapa yang mengakses ruangan dan kapan akses tersebut dilakukan, sehingga mempersulit investigasi jika terjadi kehilangan aset inventaris di dalam kelas.

Masalah lain yang tidak kalah mendesak adalah inefisiensi konsumsi energi. Seringkali, ruang kelas ditinggalkan dalam keadaan kosong dengan pendingin ruangan (AC) dan lampu yang masih menyala karena pengguna lupa mematikannya. Tanpa adanya sistem otomatisasi yang mampu mendeteksi keberadaan manusia di dalam ruangan, pemborosan energi ini terus terjadi dan meningkatkan beban biaya operasional kampus secara sia-sia. Oleh karena itu, diperlukan sebuah inovasi sistem manajemen kelas berbasis *Internet of Things* (IoT) yang mampu mengintegrasikan kontrol akses yang aman dengan manajemen energi yang cerdas secara otomatis.

1.2 PROPOSED SOLUTION

Sebagai solusi dari permasalahan manajemen ruang kelas yang telah dipaparkan, proyek ini menawarkan Classroom Manager, sebuah sistem kontrol akses dan manajemen energi otomatis berbasis *Internet of Things* (IoT) dan *Embedded Systems*. Sistem ini dirancang untuk menggantikan metode pengelolaan konvensional yang bergantung sepenuhnya pada kunci fisik dan tenaga manual petugas (*janitor*).

Berbeda dengan solusi kunci pintar standar yang hanya mendigitalkan kunci, Classroom Manager mengimplementasikan mekanisme keamanan Challenge-Response menggunakan *Bluetooth Low Energy* (BLE). Metode ini memastikan bahwa hanya pengguna yang memiliki kredensial terverifikasi secara kriptografi (HMAC-SHA256) yang dapat mengakses ruangan, sehingga menghilangkan risiko duplikasi kunci fisik atau *replay attack*. Selain itu, sistem menerapkan Role-Based Access Control (RBAC) yang membedakan hak akses antara Administrator (akses ke seluruh ruangan) dan Dosen (akses terbatas pada ruang kelas tertentu), memberikan kontrol keamanan yang lebih granular. Sistem juga dilengkapi dengan arsitektur *hybrid* yang memungkinkan autentikasi tetap berjalan menggunakan basis data lokal (*offline mode*) apabila koneksi jaringan kampus terputus.

Dari sisi efisiensi energi, sistem ini mengintegrasikan sensor gerak (PIR) dan logika otomatisasi berbasis FreeRTOS. Sistem secara cerdas mampu mendeteksi keberadaan manusia di dalam ruangan untuk mengontrol penguncian pintu dan aliran listrik ke perangkat elektronik (seperti AC dan lampu) secara otomatis. Dengan pendekatan ini, Classroom Manager tidak hanya menawarkan keamanan yang lebih tinggi, tetapi juga solusi konkret untuk penghematan energi dan efisiensi operasional di lingkungan kampus.

1.3 ACCEPTANCE CRITERIA

Sistem Classroom Manager dinyatakan berhasil apabila memenuhi kriteria berikut:

1. Sistem mampu mengendalikan solenoid *door lock* secara otomatis melalui perintah dari ESP32 menggunakan modul *relay*.
2. Mekanisme autentikasi berbasis BLE berfungsi menggunakan metode *Challenge-Response* (HMAC-SHA256) untuk mencegah duplikasi akses (*replay attack*).

3. Sistem menerapkan *Role-Based Access Control* (RBAC) yang membedakan hak akses antara Admin (akses penuh) dan Dosen (akses terbatas ruang tertentu).
4. Sistem mampu melakukan sinkronisasi basis data pengguna dari server (*Backend*) melalui protokol HTTP saat dinyalakan (*startup*) dan menyimpannya ke memori lokal (*flash memory*).
5. Fitur *Offline Mode* berfungsi dengan baik, memungkinkan autentikasi tetap berjalan menggunakan data lokal saat koneksi WiFi terputus.
6. Sensor PIR mampu mendeteksi keberadaan pengguna di dalam ruang untuk mereset *timer* penguncian otomatis (*auto-lock*).
7. Sistem mengirimkan log akses (Siapa, Di mana, Kapan) secara *real-time* ke server database dan dashboard monitoring melalui protokol MQTT.
8. Sistem dapat menerima perintah jarak jauh untuk masuk ke mode hemat daya (*Deep Sleep*) melalui MQTT.
9. Display LED Matrix MAX7219 dapat memberikan pesan yang sesuai dengan keadaan atau *state* kunci pintu.

1.4 ROLES AND RESPONSIBILITIES

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

Roles	Responsibilities	Person
Role 1	Membeli barang, membuat laporan dan PPT, merangkai fisik, dan testing	Muhammad Hilmi Al Muttaqi
Role 2	Membuat laporan, PPT, dan readme	R. Aisha Syauqi Ramadhani
Role 3	Mengembangkan <i>Mobile App</i> untuk fungsi autentikasi BLE, membeli dan mengumpulkan komponen <i>hardware</i> , membuat kode ESP32, dan <i>testing</i> .	Wesley Frederick Oh
Role 4	Mengembangkan dan men-deploy backend database MongoDB, membuat kode ESP32, dan <i>testing</i> .	Muhammad Riyan Satrio Wibowo

Table 1. Roles and Responsibilities

1.5 TIMELINE AND MILESTONES

No	Kegiatan	November				Desember							
		27	28	29	30	1	2	3	4	5	6	7	8
1	Perumusan ide												
2	Finalisasi ide												
3	Hardware design												
4	Pengembangan software												
5	Menyusun rangkaian												
6	Integrasi dan uji coba hardware dan software												
7	Perakitan dan pengujian akhir												

Fig 1. Gantt Chart

CHAPTER 2

IMPLEMENTATION

2.1 HARDWARE DESIGN AND SCHEMATIC

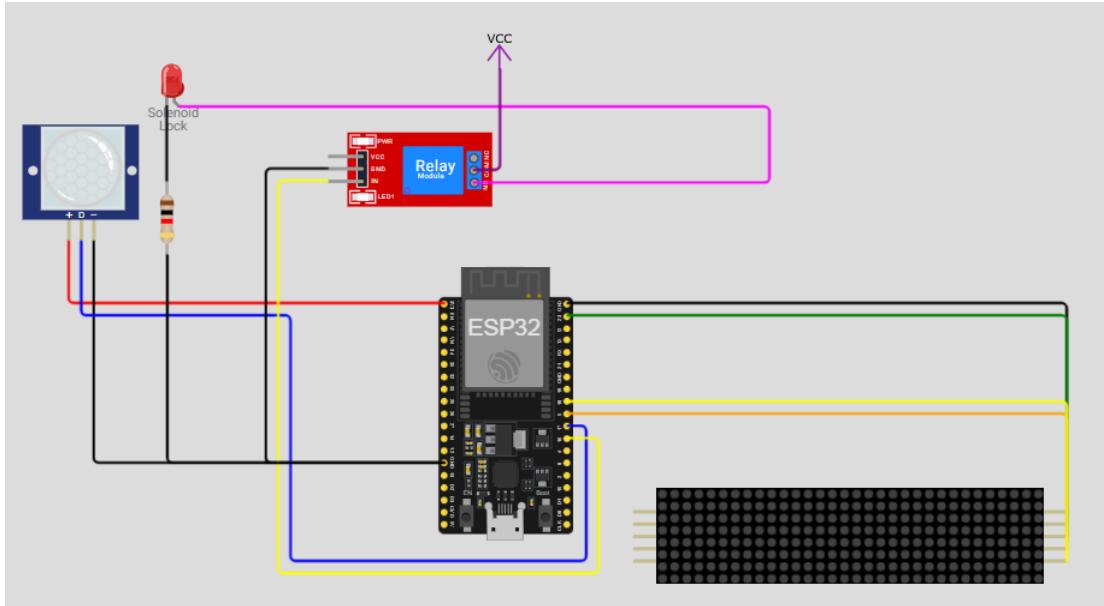


Fig 2. Desain Perangkat Keras dan Skematik Desain

Desain perangkat keras sistem *Classroom Manager* disusun dengan pendekatan modular untuk memastikan stabilitas daya dan keandalan operasi. Catu daya utama bersumber dari adaptor 12V DC yang bertugas menyuplai beban induktif solenoid *door lock*. Tegangan 12V ini didistribusikan melalui dua jalur terpisah: jalur daya langsung menuju terminal *Common* (COM) pada modul *relay*, dan jalur regulasi menuju *Buck Converter* yang menurunkan tegangan menjadi 5V DC yang stabil untuk menyuplai daya ke mikrokontroler ESP32. Pemisahan jalur ini diterapkan untuk mencegah *voltage sag* (penurunan tegangan) pada mikrokontroler saat solenoid diaktifkan.

Mikrokontroler ESP32 digunakan sebagai unit pemrosesan pusat yang menangani seluruh logika kendali, pemrosesan sinyal sensor, dan komunikasi nirkabel (WiFi/BLE). Untuk pemantauan okupansi ruangan, sensor gerak PIR (HC-SR501) dihubungkan ke GPIO 17 sebagai input digital. Interface visual bagi pengguna difasilitasi oleh modul OLED SSD1306 yang terhubung melalui protokol komunikasi I2C pada pin GPIO 21 (SDA) dan GPIO 22 (SCL).

Pengendalian akses fisik dilakukan menggunakan solenoid *door lock* 12V yang dioperasikan melalui modul *relay* elektromagnetik. Relay diaktifkan oleh sinyal digital dari

GPIO 16 pada ESP32, berfungsi sebagai saklar isolator yang memisahkan sirkuit logika tegangan rendah (3.3V) dengan sirkuit daya tegangan tinggi (12V). Seluruh subsistem dihubungkan pada referensi *Common Ground* yang sama untuk menjamin integritas sinyal. Desain ini menghasilkan arsitektur perangkat keras yang tangguh dan aman untuk operasional jangka panjang.

2.2 SOFTWARE DEVELOPMENT

Pengembangan perangkat lunak pada sistem *Classroom Manager* dilakukan secara terintegrasi yang mencakup dua sisi utama: *firmware* pada perangkat tertanam (Embedded System) dan layanan *backend* pada sisi server. Pendekatan ini memastikan sistem dapat berjalan secara mandiri (*standalone*) namun tetap dikelola secara terpusat.

A. Firmware ESP32 (Embedded Logic)

Perangkat lunak pada ESP32 dikembangkan menggunakan platform Arduino dengan basis sistem operasi waktu nyata (FreeRTOS). Arsitektur perangkat lunak dibagi menjadi beberapa *Task* independen yang berjalan secara konkuren untuk menjamin stabilitas dan responsivitas sistem:

1. Manajemen Task FreeRTOS:

- *doorTask* (High Priority): Bertanggung jawab atas logika kritis pengendalian akses. Task ini membaca antrian perintah (*Queue*), mengaktifkan *relay* solenoid pada GPIO 16, dan memantau sensor PIR pada GPIO 17 secara *real-time*. Logika sensor PIR berfungsi untuk mereset *software timer* secara otomatis (*auto-lock extension*) apabila terdeteksi pergerakan, mencegah pintu terkunci saat masih digunakan.
- Komunikasi & Sinkronisasi: ask ini menangani koneksi WiFi dan protokol MQTT di latar belakang agar proses komunikasi data tidak mengganggu responsivitas kunci pintu.

2. Autentikasi & Keamanan (BLE):

Sistem menggunakan protokol komunikasi *Bluetooth Low Energy* (BLE) dengan skema *Challenge-Response*. Saat pengguna terhubung, ESP32 mengirimkan *Nonce* (bilangan acak). Aplikasi pengguna membalas dengan tanda tangan digital yang dienkripsi menggunakan algoritma HMAC-SHA256. Sistem kemudian memverifikasi tanda tangan tersebut dan memeriksa hak akses pengguna (*Role-Based Access Control*).

Control), apakah sebagai ADMIN (akses penuh) atau LECTURER (akses terbatas pada ruang tertentu).

3. Sinkronisasi Data & Mode Offline: Salah satu fitur vital sistem ini adalah mekanisme operasi *hybrid*. Saat perangkat dinyalakan (*booting*), fungsi syncDatabase() melakukan permintaan HTTP GET ke *Backend Server* untuk mengunduh data pengguna terbaru. Data ini kemudian disimpan ke dalam memori non-volatile (Preferences/Flash Memory) di ESP32. Hal ini memungkinkan sistem tetap dapat memverifikasi pengguna dan membuka pintu menggunakan data lokal meskipun koneksi internet/WiFi kampus terputus (*Offline Mode*).

B. Backend Service & Database

Sisi server (*Backend*) dibangun menggunakan lingkungan eksekusi Node.js dengan kerangka kerja Express.js dan basis data MongoDB. Layanan ini berfungsi sebagai pusat manajemen data dengan fitur:

1. Manajemen Pengguna (User Repository): Menyediakan API RESTful untuk pendaftaran dan pengelolaan data dosen. Sistem secara otomatis membuat *Secret Key* unik dan mengenkripsi kata sandi pengguna menggunakan *library* bcrypt sebelum disimpan ke basis data.
2. Pencatatan Log: Menerima data riwayat akses dari ESP32 (melalui MQTT atau HTTP) yang berisi informasi ID pengguna, ruangan, dan waktu akses, lalu menyimpannya dalam koleksi log MongoDB untuk keperluan audit.

C. Alur Logika dan Diagram Sistem

Integrasi antara logika autentikasi, komunikasi data, dan kontrol perangkat keras digambarkan melalui diagram-diagram berikut:

1. Alur Autentikasi

Proses validasi pengguna dari koneksi BLE hingga pembukaan kunci ditunjukkan pada Fig 3.

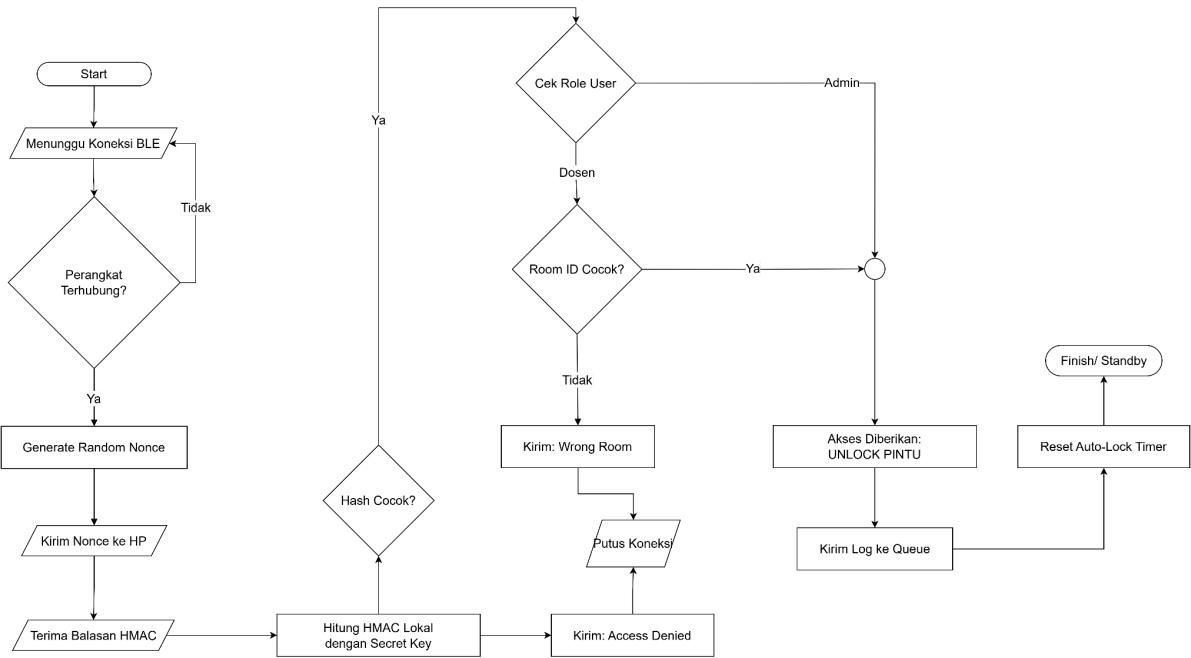


Fig 3. Flowchart proses autentikasi BLE dan kontrol solenoid

2. Alur Komunikasi Data

Untuk keperluan pencatatan data status ruangan, sistem menggunakan protokol MQTT. Data peristiwa dikirim ke *dashboard* hanya ketika terdapat perubahan status valid, dengan memanfaatkan *message queue* pada ESP32 untuk menghindari kehilangan data saat koneksi tidak stabil.

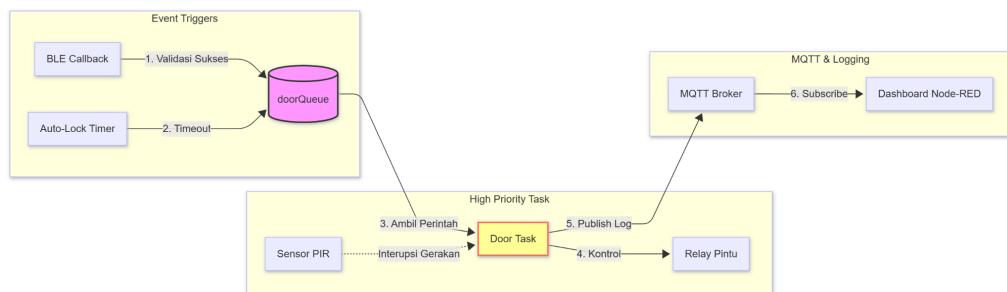


Fig 4. Flowchart pengiriman data melalui MQTT dan queue buffering

3. State Machine Operasional

Sistem beroperasi dalam beberapa kondisi (*state*) utama seperti *Idle*, *Verifying*, *Unlocked*, dan *Deep Sleep*. Sensor PIR berperan dalam memperpanjang *state* 'Unlocked' selama ada aktivitas, sebelum akhirnya kembali terkunci otomatis berdasarkan *timer*.

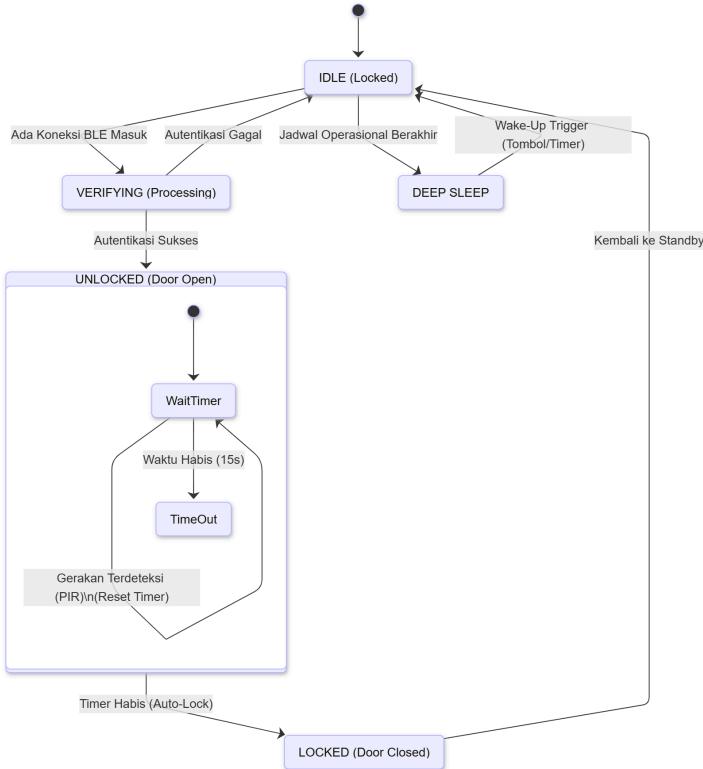


Fig 5. State Machine Operasi Sistem Classroom Manager

2.3 HARDWARE AND SOFTWARE INTEGRATION

Integrasi sistem *Classroom Manager* berpusat pada mikrokontroler ESP32 yang berfungsi memetakan sinyal fisik dari perangkat keras ke dalam logika perangkat lunak berbasis *FreeRTOS*. Pada sisi pengendalian aktuator, pin GPIO 16 dihubungkan ke modul *relay* sebagai jembatan antara logika program dan mekanisme fisik. Ketika kredensial pengguna divalidasi oleh sistem, instruksi buka kunci dikirimkan melalui *message queue* ke *Door Task*, yang kemudian mengirimkan sinyal logika *HIGH* untuk memicu *relay* dan mengaktifkan solenoid 12V. Pendekatan ini tidak hanya menjamin responsivitas kunci yang instan, tetapi juga memastikan isolasi yang aman antara sirkuit logika tegangan rendah dan beban daya tinggi.

Untuk mendukung fitur otomatisasi cerdas, sistem mengintegrasikan sensor gerak PIR pada GPIO 17 yang berfungsi memantau aktivitas ruangan secara *real-time*. Sinyal deteksi dari sensor ini terhubung langsung dengan logika *Software Timer* untuk mekanisme penguncian otomatis (*auto-lock*). Saat pintu terbuka, timer akan memulai hitungan mundur selama 15 detik, namun jika sensor mendeteksi adanya pergerakan, hitungan ini akan di-reset ulang secara otomatis. Mekanisme ini memastikan pintu hanya akan terkunci kembali jika

tidak ada aktivitas gerakan yang terdeteksi hingga durasi waktu habis, mencegah penguncian yang tidak diinginkan saat ruangan masih digunakan.

Sebagai antarmuka visual, modul MAX7219 LED Matrix dihubungkan melalui jalur komunikasi SPI pada pin GPIO 23 (DIN), GPIO 18 (CLK), dan GPIO 5 (CS) untuk menampilkan status "LOCKED" atau "OPEN" secara bergulir (*running text*). Di saat yang bersamaan, seluruh perubahan status operasional dan riwayat akses ditransmisikan secara paralel melalui protokol MQTT. Data ini ditampilkan secara langsung pada Dashboard Node-RED untuk pemantauan *real-time* dan disimpan secara permanen ke dalam Database melalui layanan Backend, menciptakan ekosistem pengawasan dan audit sistem yang terintegrasi penuh.

CHAPTER 3

TESTING AND EVALUATION

3.1 TESTING

Tahap pengujian dilaksanakan untuk memverifikasi bahwa seluruh modul perangkat keras dan perangkat lunak pada sistem *Classroom Manager* beroperasi sesuai dengan spesifikasi rancangan dan kriteria penerimaan yang telah ditetapkan. Pendekatan pengujian yang digunakan adalah *Black-Box Testing*, di mana sistem dievaluasi berdasarkan fungsionalitas input-output tanpa membedah struktur kode internal secara mendalam saat pengujian berlangsung.

Rangkaian pengujian mencakup beberapa aspek kunci yang menjadi parameter keberhasilan implementasi sistem, meliputi:

1. Memastikan mekanisme *Challenge-Response* berjalan dengan benar dan mampu menolak akses dari perangkat yang tidak memiliki kunci rahasia yang valid atau mengirimkan respon yang salah (kadaluarsa).
2. Memvalidasi logika pembedaan hak akses antara pengguna dengan peran 'ADMIN' dan 'LECTURER' terhadap ruangan tertentu sesuai data di *backend*.
3. Menguji kemampuan sistem untuk tetap beroperasi menggunakan basis data lokal (*Preferences*) saat koneksi WiFi ke server terputus.
4. Memastikan sensor PIR berfungsi memperpanjang durasi kunci terbuka (*timer extension*) saat mendeteksi aktivitas, serta memastikan fitur *auto-lock* bekerja saat ruangan kosong.
5. Memverifikasi bahwa setiap aktivitas akses terkirim dan tercatat dengan benar di *Backend Server* dan *Dashboard Node-Red*.
6. Menguji respons sistem terhadap perintah "SLEEP" yang dikirimkan melalui protokol MQTT untuk memasuki mode hemat daya (*Deep Sleep*).

Untuk memastikan objektivitas penilaian, setiap skenario pengujian disusun dalam format test case terstandarisasi yang memuat kondisi awal (initial state), prosedur uji (test procedure), hasil yang diharapkan (expected result), hasil aktual (actual result), serta status kelulusan.

ID Uji	Fitur	Skenario Pengujian	Hasil yang Diharapkan
TC-01	BLE Auth	Pengguna mengirimkan respon HMAC yang benar terhadap <i>Nonce</i> yang diberikan.	Autentikasi berhasil, akses diterima, dan log percobaan akses tercatat di Serial Monitor.
TC-02	Login/Register	Pengguna melakukan <i>login</i> dan <i>register</i> pada aplikasi di <i>smartphone</i> mereka untuk mendaftarkan kredensial mereka ke basis data.	Pengguna berhasil melakukan <i>login</i> maupun <i>register</i> dan hasilnya terlihat pada basis data yang telah dibuat.
TC-03	RBAC (Dosen dan Admin)	Pengguna dengan peran 'LECTURER' (izin Ruang 303) mencoba membuka Ruang 404. Pengguna dengan peran 'ADMIN' bisa membuka pintu ruangan manapun.	Akses ditolak, sistem mengirim pesan balik "ACCESS DENIED" ke aplikasi pengguna dan display MAX7219, dan tidak membuka kunci.
TC-04	Offline Mode	Memutuskan koneksi WiFi (simulasi gangguan), lalu melakukan akses dengan pengguna valid.	Sistem mendeteksi <i>timeout WiFi</i> , memuat data dari <i>Flash Memory</i> (Offline Data Found), dan pintu tetap dapat dibuka.
TC-05	Auto-Lock & PIR	Pintu terbuka, lalu ada gerakan terus-menerus di depan sensor PIR sebelum 15 detik habis.	<i>Timer</i> penguncian di-reset ulang terus-menerus, pintu tetap terbuka selama masih ada aktivitas gerakan.
TC-06	Logging	Akses berhasil dilakukan oleh pengguna.	Data log (UserID, Room) muncul secara <i>real-time</i> pada topik MQTT esp32/lock/log dan tersimpan di database MongoDB.
TC-07	Remote Sleep	Mengirim pesan "SLEEP" ke topik esp32/lock/sleep melalui dashboard.	Sistem membalas status "SLEEPING_NOW" dan memasuki mode <i>Deep Sleep</i> untuk menghemat daya.

TC-08	Spam Protection	Mengirim perintah "OPEN" berkali-kali dalam waktu singkat (< 2 detik).	Sistem mengabaikan perintah berulang (<i>debounce</i>) untuk mencegah <i>spamming</i> pada mekanisme mekanik kunci.
TC-09	LED Matrix Display	Mengirimkan indikator sesuai ruangan yang tertera, menunjukkan status pintu, dan pesan selamat datang pada orang yang berhasil membuka kunci.	Sistem berhasil menginformasikan ke display ruangan mana yang diasosiasikan dengan sistem, serta pesan yang sesuai saat kunci dibuka.

Table 1. Test Case Documentation Classroom Manager

3.2 RESULT

Berdasarkan rangkaian skenario pengujian yang telah dijalankan, berikut adalah pemaparan hasil teknis yang diperoleh dari *Serial Monitor*, *Dashboard Node-RED*, dan observasi fisik perangkat:

1. Validasi Autentikasi Kriptografi (TC-01: BLE Authentication)

Mekanisme *Challenge-Response* berhasil memverifikasi pengguna yang sah. Log sistem menunjukkan alur lengkap mulai dari koneksi, pertukaran *Nonce*, hingga verifikasi *HMAC*. Sistem secara tepat memberikan status [ACCESS] Granted hanya ketika tanda tangan digital valid, dan menolak akses jika terjadi ketidakcocokan kunci.

```
17:17:48.844 -> [BLE] Device Connected.
17:17:51.771 -> [BLE] ID Rx: LEC_E
17:17:51.771 -> [AUTH] User Found: LEC_E
17:17:51.841 -> [BLE] Nonce Rx: 434497
17:17:51.841 -> [BLE] Sent Hash: 2798289f036d38147b767bdc40e829f8ad1f111697e57eca7acf4f6ecf62e33d
17:17:51.962 -> [BLE] ID Rx: OPEN
17:17:51.962 -> [ACCESS] Granted (Lecturer)
17:17:52.004 -> [DOOR] Unlocking...
```

Fig 6. Log Validasi Kriptografi HMAC-SHA256

2. Verifikasi Sinkronisasi Kredensial (TC-02: Login/Register)

Pada pengujian komunikasi antar sistem, konsistensi *data propagation* dari server menuju perangkat berhasil divalidasi. Identitas pengguna yang diregistrasi melalui antarmuka aplikasi seluler terdeteksi secara akurat oleh perangkat embedded, ditandai dengan keluaran [AUTH] User Found: LEC_E. Proses ini mencerminkan

keberhasilan integrasi sistem dalam hal pengelolaan data pengguna dan pemetaan kredensial. Keberhasilan sinkronisasi ini berimplikasi langsung terhadap reliabilitas fitur kontrol akses berbasis otentikasi server.

```
[AUTH] User Found: LEC_E
[BLE]Nonce Rx: 434497
[BLE]Sent Hash: 2798289f036d38147b767bdc40e829f8ad1f111697e57eca7acf4f6ecf62e33d
[BLE]ID Rx: OPEN
[ACCESS] Granted (Lecturer)
[DOOR] Unlocking...
```

Fig 7. Log Sistem Menunjukkan Identifikasi Pengguna Terdaftar

3. Evaluasi Kontrol Akses Berbasis Peran (TC-03: Role-Based Access Control / RBAC)

Sistem berhasil menerapkan pembatasan akses berdasarkan peran (*Role*). Ketika pengguna dengan peran 'LECTURER' (yang hanya diizinkan di Ruang 303) mencoba membuka perangkat ini (Ruang 404), sistem secara otomatis menolak instruksi tersebut dan tidak mengaktifkan *relay*.

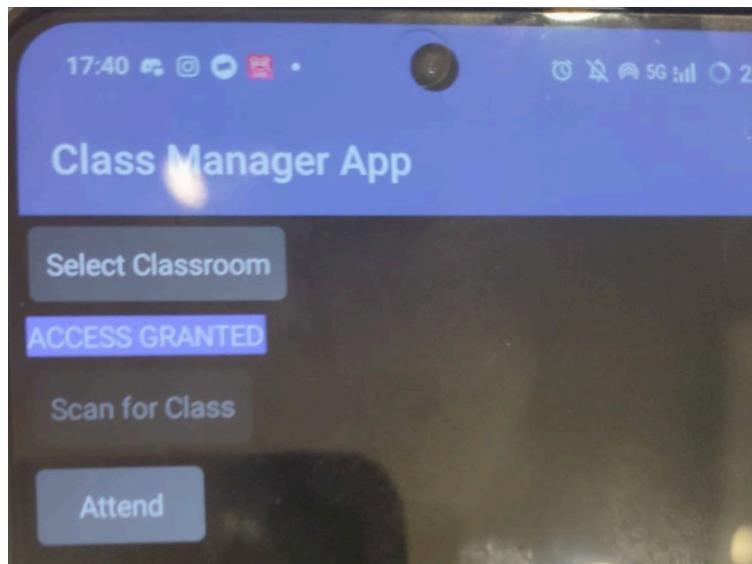


Fig 8. Log Implementasi Otorisasi Berbasis Peran

4. Uji Resiliensi dan Toleransi Kesalahan (TC-04: Offline Mode / Fail-Safe)

Mekanisme *fail-safe* berjalan efektif. Saat koneksi WiFi diputus secara paksa, sistem mendeteksi *timeout* dan beralih ke mode *offline*. Autentikasi tetap berjalan normal menggunakan data yang tersimpan di *Flash Memory*, menjamin ketersediaan layanan saat gangguan jaringan.

```

Connecting to WiFi.....  

WiFi Timeout. Switching to Offline Mode.  

Attempting to load Offline Database...  

Offline Data Found.  

--- RAW PAYLOAD START ---  

{"success":true,"message":"Found all users","data":[]}  

--- RAW PAYLOAD END ---

```

Fig 9. Log Transisi Sistem Offline & Pemuatan Basis Data Lokal

5. Validasi Otomatisasi Berbasis Sensor Okupansi (TC 05 Auto Lock dan PIR)

Integrasi sensor PIR dengan *Software Timer* FreeRTOS berfungsi responsif. Saat pintu dalam keadaan *UNLOCKED* dan sensor mendeteksi gerakan, sistem memperpanjang durasi buka kunci, mencegah pintu terkunci saat masih ada aktivitas.

```

[DOOR] Unlocking...
[PIR] Motion Detected - Timer Extended.

```

Fig 10. Log Respons Timer dan Deteksi Gerakan PIR

6. Integritas Telemetri Data dan Pemantauan Real-Time (TC 06 Logging)

Seluruh aktivitas akses berhasil direkam secara *real-time*. Data yang dikirimkan ESP32 via MQTT berhasil diterima oleh *broker*, divisualisasikan pada Dashboard Node-RED, dan tersimpan permanen di database MongoDB melalui layanan *backend*.

Live Status		Access Logs		
Door Status:	LOCKED	Time	User	Room
		07/12/2025, 17.17.52	LEC_E	ROOM_404
		06/12/2025, 18.09.01	LEC_C	ROOM_303
		06/12/2025, 18.39.08	LEC_B	ROOM_404

Fig 11. Dashboard Node-RED Menampilkan Akses Pengguna Secara Real-Time

7. Pengujian Manajemen Daya Jarak Jauh (TC 07 Remote Sleep)

Sistem merespons perintah manajemen daya yang dikirimkan melalui dashboard. Perintah "SLEEP" via MQTT berhasil memicu fungsi `esp_deep_sleep_start()`, menonaktifkan fitur non-esensial untuk penghematan energi.

<pre>[TIMER] Auto-lock triggered. [DOOR] Locking...</pre>	<pre>[MQTT] Connected MQTT Message arrived [esp32/lock/sleep] SLEEP [SYSTEM] Sleep command received. [SYSTEM] Entering Deep Sleep for 12 Hours...</pre>
---------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------

Fig 12. Log Transisi Sistem Menuju Mode Deep Sleep

8. Uji Stabilitas Terhadap Flooding Attack (TC 08 Spam Protection)

Fitur *debounce* berhasil mencegah eksekusi berulang dari perintah "OPEN" yang dikirimkan dalam interval sangat singkat (< 2 detik), melindungi solenoid dari kerusakan mekanis.

```
[BLE] ID Rx: OPEN  
[ACCESS] Granted (Lecturer)  
[DOOR] Unlocking...  
[PIR] Motion Detected - Timer Extended.  
[BLE] OPEN command ignored (spam/cool-down protection)
```

Fig 13. Log Perlindungan Sistem terhadap Flooding Command

9. Verifikasi Sinkronisasi Umpaman Balik Visual (TC 09 LED Matrix Display)

Modul MAX7219 berhasil menampilkan status sistem secara sinkron. Saat terkunci, layar menampilkan teks berjalan "LOCKED", dan saat terbuka menampilkan pesan "Welcome [UserID]".

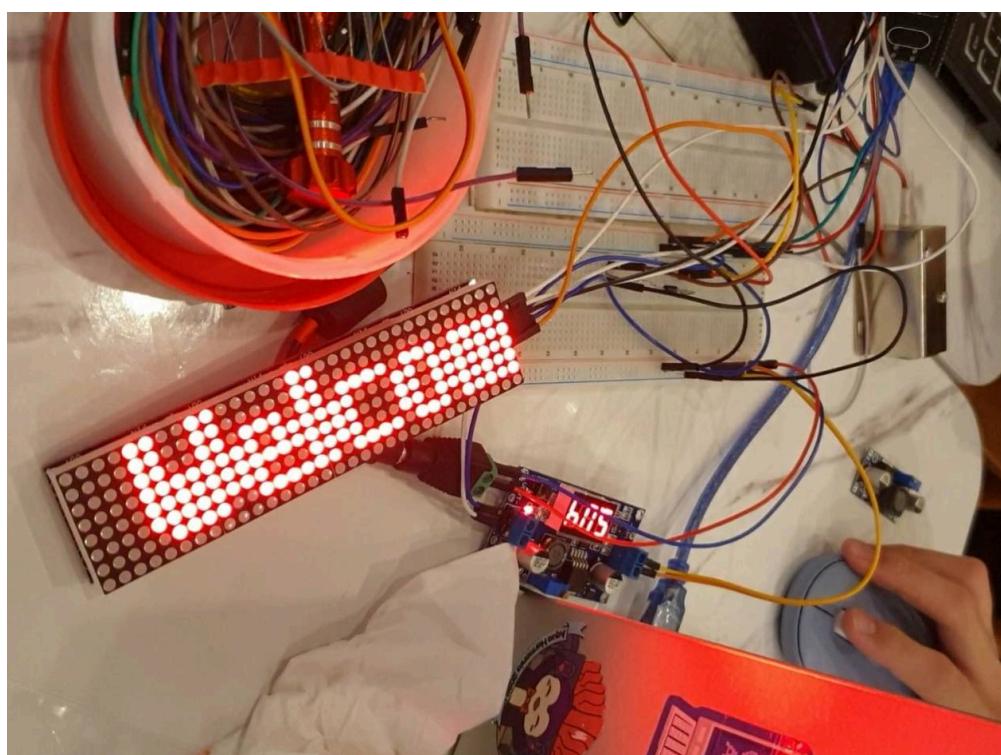


Fig 14. Log Pemicu Display dari Status Unlocking

3.3 EVALUATION

Evaluasi kinerja sistem *Classroom Manager* dilakukan dengan menganalisis hasil pengujian empiris terhadap standar kriteria penerimaan yang telah ditetapkan. Berdasarkan data pengujian yang diperoleh, berikut adalah analisis mendalam mengenai performa sistem:

1. Efektivitas Keamanan dan Kontrol Akses (Security Evaluation)

Implementasi keamanan berbasis HMAC-SHA256 terbukti efektif dalam mencegah akses tidak sah. Berbeda dengan kunci pintu digital standar yang rentan terhadap *replay attack* (perekaman sinyal), mekanisme *Challenge-Response* yang diterapkan memastikan bahwa setiap sesi pembukaan kunci menggunakan token (*nonce*) yang unik dan sekali pakai. Selain itu, fitur *Role-Based Access Control* (RBAC) berhasil membatasi hak akses pengguna secara granular; sistem secara konsisten menolak akses Dosen pada ruangan yang tidak sesuai dengan izinnya di *backend*, memenuhi kriteria keamanan tingkat lanjut untuk lingkungan kampus.

2. Ketahanan Sistem dan Konektivitas (System Reliability)

Sistem menunjukkan tingkat resiliensi (ketahanan) yang tinggi melalui fitur Offline Mode. Pengujian pemutusan koneksi WiFi membuktikan bahwa sistem tidak mengalami kegagalan fungsi (*system failure*); sebaliknya, sistem secara cerdas beralih menggunakan basis data lokal (*Preferences*) untuk melakukan autentikasi. Hal ini mengatasi kelemahan umum perangkat IoT yang biasanya lumpuh tanpa koneksi internet, memastikan operasional kelas tetap berjalan lancar meskipun infrastruktur jaringan kampus mengalami gangguan.

3. Kinerja Otomatisasi dan Efisiensi Energi

Integrasi antara sensor PIR dan *Software Timer* pada FreeRTOS terbukti responsif dalam menangani logika "Janitor Mode". Sistem berhasil menghilangkan risiko pintu terkunci saat ruangan masih berpenghuni, sekaligus menjamin pintu terkunci otomatis saat ruangan kosong untuk keamanan aset. Efisiensi energi lebih lanjut dicapai melalui fitur Remote Deep Sleep, yang memungkinkan administrator mematikan sistem secara terpusat di luar jam operasional, memperpanjang umur komponen dan menghemat daya secara signifikan.

4. Integrasi Ekosistem (End-to-End Integration)

Secara keseluruhan, integrasi antara perangkat keras (ESP32, Relay, MAX7219) dan perangkat lunak (*Backend* Node.js, MQTT) berjalan tanpa kendala (*seamless*). Data log yang tercatat di *Dashboard* dan *Database* memiliki akurasi waktu dan identitas yang tinggi, memenuhi kebutuhan audit keamanan. Transisi status visual pada LED Matrix juga sinkron dengan status logika internal, memberikan pengalaman pengguna (*User Experience*) yang intuitif dan informatif.

CHAPTER 4

CONCLUSION

Proyek pengembangan sistem *Classroom Manager* telah berhasil direalisasikan sebagai solusi komprehensif untuk modernisasi manajemen akses ruang kelas berbasis *Internet of Things* (IoT). Sistem ini mengintegrasikan mikrokontroler ESP32 dengan arsitektur perangkat lunak *multitasking* berbasis *FreeRTOS* untuk mengendalikan mekanisme penguncian elektromagnetik secara cerdas. Berdasarkan hasil pengujian, integrasi antara perangkat keras (Sensor PIR, Solenoid, LED Matrix MAX7219) dan ekosistem perangkat lunak (*Backend Node.js*, Protokol MQTT) terbukti berjalan stabil dan responsif dalam menangani permintaan akses, pemantauan okupansi ruangan, serta pelaporan data telemetri secara *real-time* ke dashboard pusat.

Keunggulan utama dari sistem ini terletak pada penerapan standar keamanan hibrida yang menggabungkan proteksi fisik dan siber. Implementasi mekanisme autentikasi *Challenge-Response* berbasis algoritma HMAC-SHA256 pada komunikasi BLE terbukti efektif dalam mencegah akses ilegal dan serangan duplikasi (*replay attack*). Keandalan sistem semakin diperkuat dengan fitur *Offline Mode* yang memungkinkan perangkat beroperasi secara mandiri menggunakan basis data lokal (*Preferences*) saat terjadi gangguan jaringan, memastikan bahwa kegiatan operasional akademik tidak terhambat oleh masalah konektivitas internet yang sering terjadi pada infrastruktur IoT konvensional.

Selain aspek keamanan dan keandalan, sistem ini juga menawarkan efisiensi operasional melalui fitur otomatisasi lingkungan. Integrasi sensor PIR dengan logika *timer* adaptif berhasil menghilangkan risiko kelalaian manusia (*human error*) dalam mengunci pintu, sekaligus mencegah penguncian yang tidak diinginkan saat ruangan masih digunakan. Ditambah dengan kemampuan manajemen daya jarak jauh (*Remote Deep Sleep*), sistem *Classroom Manager* tidak hanya memberikan kemudahan akses bagi dosen dan administrator, tetapi juga berkontribusi signifikan terhadap penghematan energi dan perpanjangan masa pakai perangkat untuk penggunaan jangka panjang di lingkungan kampus.

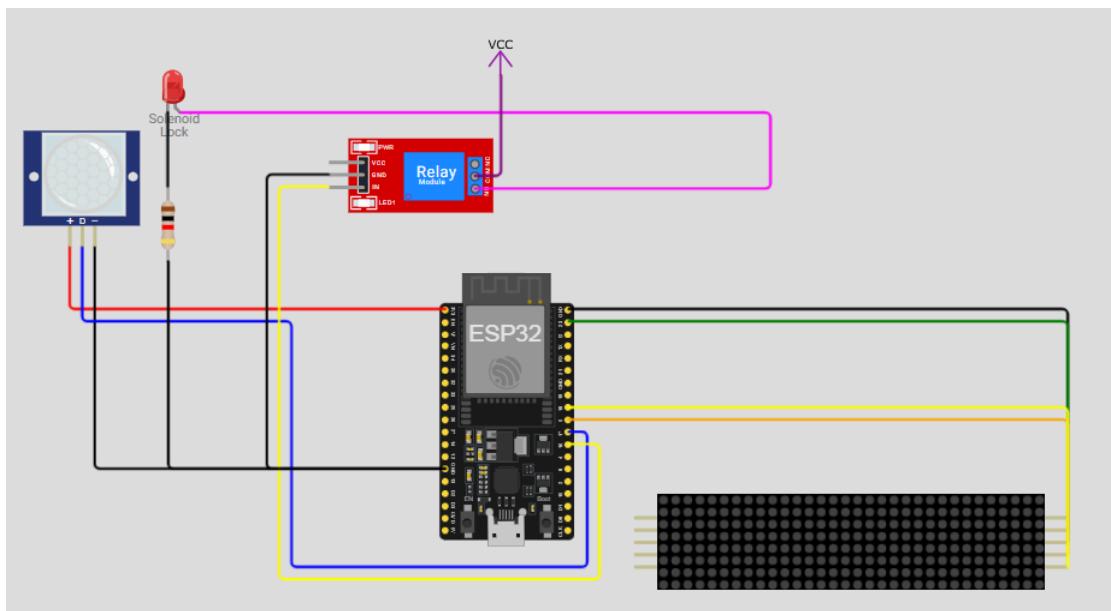
REFERENCES

- [1] EMQX Platform Docs, “Connect with ESP32”, 2025. [Online]. Available: https://docs.emqx.com/en/cloud/latest/connect_to_deployments/esp32.html [Accessed: Dec. 08, 2025].
- [2] EMQX Blog, “MQTT on ESP32: A Beginner's Guide”, Aug. 05, 2024. [Online]. Available: <https://www.emqx.com/en/blog/esp32-connects-to-the-free-public-mqtt-broker> [Accessed: Dec. 08, 2025]. www.emqx.com
- [3] Random Nerd Tutorials, “ESP32 MQTT – Publish DS18B20 Temperature Readings ...”, 2024. [Online]. Available: <https://randomnerdtutorials.com/esp32-mqtt-publish-ds18b20-temperature-arduino/> [Accessed: Dec. 08, 2025].
- [4] Random Nerd Tutorials, “ESP32 MQTT Publish/Subscribe with Arduino IDE”, 2025. [Online]. Available: <https://randomnerdtutorials.com/esp32-mqtt-publish-subscribe-arduino-ide/> [Accessed: Dec. 08, 2025].
- [5] Random Nerd Tutorials, “ESP32 with PIR Motion Sensor using Interrupts and Timers”, Oct. 27, 2025. [Online]. Available: <https://randomnerdtutorials.com/esp32-pir-motion-sensor-interrupts-timers/> [Accessed: Dec. 08, 2025].
- [6] SunFounder Documentation, “Lesson 12: PIR Motion Module (HC-SR501)”, 2025. [Online]. Available: https://docs.sunfounder.com/projects/umsk/en/latest/03_esp32/esp32_lesson12_pir_motion.html [Accessed: Dec. 08, 2025].
- [7] Instructables, “How to Use PIRs With Arduino & Raspberry Pi”, 2023. [Online]. Available: <https://www.instructables.com/PIR-Motion-Sensor-How-to-Use-PIRs-With-Arduino-Ras/> [Accessed: Dec. 08, 2025].
- [8] ArXiv, J. Dizdarevic, M. Michalke, A. Jukan, “Engineering and Experimentally Benchmarking Open Source MQTT Broker Implementations”, May 2023. [Online]. Available: <https://arxiv.org/abs/2305.13893> [Accessed: Dec. 08, 2025].

- [9] ArXiv, M. Ahmed, M. M. Akhtar, “Smart Home: Application using HTTP and MQTT as Communication Protocols”, Dec. 2021. [Online]. Available: <https://arxiv.org/abs/2112.10339> [Accessed: Dec. 08, 2025].

APPENDICES

Appendix A: Project Schematic



Appendix B: Documentation

