



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



**RANCANG BANGUN SISTEM LOKER PROYEK DOSEN
TELEKOMUNIKASI MENGGUNAKAN MODUL 4G BERBASIS AI DAN
ANDROID**

**“RANCANG BANGUN SISTEM LOKER DOSEN TELEKOMUNIKASI
MENGGUNAKAN TRANSMISI MODUL 4G”**

TUGAS AKHIR

**POLITEKNIK
NEGERI
JAKARTA**

Diajukan sebagai salah satu syarat untuk memperoleh gelar Diploma Tiga

Muhammad Zatiero Adrafa

2203332043

**PROGRAM STUDI TELEKOMUNIKASI
JURUSAN TEKNIK ELEKTRO
POLITEKNIK NEGERI JAKARTA**

2025



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

HALAMAN PERNYATAAN ORISINALITAS

Tugas akhir ini adalah hasil karya saya sendiri dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Nama : Muhammad Zatihero Adrafa

NIM : 2203332043

Tanda Tangan :

Tanggal : 4 Juli 2025





© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

LEMBAR PENGESAHAN
TUGAS AKHIR

Tugas Akhir diajukan oleh :
Nama : Muhammad Zatiero Adrafa
NIM : 2203332043
Program Studi : Teknik Telekomunikasi
Judul Tugas Akhir : Rancang Bangun Sistem Loker Dosen
Telekomunikasi Menggunakan Modul 4G

Telah diuji oleh tim penguji dalam Sidang Tugas Akhir pada Jumat, 4 Juli 2025 dan dinyatakan **LULUS**.

Pembimbing

Benny Nixon, S.T., M.T.,

196811072000031001

Depok, 24 Juli 2025

(.....)

Disahkan oleh

Ketua Jurusan Teknik Elektro



Dr. Murie Dwiyaniti, S.T., M.T.

NIP. 197803312003122002



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

KATA PENGANTAR

Puji dan syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas rahmat dan karunia-Nya, penulis dapat menyelesaikan tugas akhir. Penulisan tugas akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Diploma Tiga Politeknik.

Tugas akhir ini berjudul “Rancang Bangun Aplikasi Pada Sistem Loker Proyek Dosen Telekomunikasi”. Penulis menyadari bahwa terselesaikannya tugas akhir ini tanpa bimbingan dan bantuan dari beberapa pihak tidak akan selesai pada waktunya. Oleh karena itu, penulis mengucapkan terimakasih kepada :

1. Benny Nixon, S.T., M.T. selaku dosen pembimbing yang telah bersedia meluangkan waktu serta menyediakan tenaga dan pikiran untuk mengarahkan penulis dalam melaksanakan tugas akhir.
2. Orang tua dan keluarga penulis yang memberikan doa serta kasih sayang dan dukungan moral maupun material.
3. Rahul Asrialdi selaku rekan sekaligus sahabat dalam mengerjakan tugas akhir.
4. Revalina Serlidya selaku rekan sekaligus sahabat dalam mengerjakan tugas akhir dan teman-teman dari Program Studi Telekomunikasi 2022 terkhusus kelas B yang telah mendukung serta bekerja sama untuk menyelesaikan tugas akhir.
5. Seluruh Kelas Telekomunikasi B 2022 yang selalu mendukung dan menghibur penulis saat mengerjakan tugas akhir.

Akhir kata, penulis berharap Tuhan Yang Maha Esa berkenan membalas kebaikan semua pihak yang telah membantu. Semoga tugas akhir ini membawa manfaat bagi pengembangan ilmu.

Depok, 15 Juli 2025
Penulis



ABSTRAK

Perkembangan teknologi Internet of Things (IoT) dan Artificial Intelligence (AI) memungkinkan terciptanya sistem keamanan yang kompleks dan handal. Dosen sering menghadapi kendala dalam menyimpan barang pribadi penting seperti dokumen pembelajaran yang rentan hilang jika ditinggalkan di kampus atau merepotkan jika harus dibawa pulang setiap hari. Penelitian ini bertujuan merancang, merealisasikan, dan menguji sistem loker pintar khusus dosen Telekomunikasi yang mengintegrasikan IoT dan AI dengan transmisi modul 4G. Sistem menggunakan Raspberry Pi 5 sebagai unit pemrosesan utama, Arduino Nano untuk sistem RFID, modul 4G SIM7600G-H untuk transmisi data, Firebase sebagai database real-time, face recognition berbasis AI, dan aplikasi Android. Komponen pendukung meliputi solenoid door lock dan RFID sebagai kunci cadangan. Hasil pengujian menunjukkan sistem berhasil diimplementasikan dengan baik. Sistem RFID mencapai keberhasilan 100% membaca kartu UID, komunikasi I2C antara Arduino Nano dan Raspberry Pi 5 berjalan stabil, relay 12 channel beroperasi optimal dengan tegangan 4.9V (HIGH) dan 0.78-0.79V (LOW). Pengujian modul 4G menghasilkan rata-rata kecepatan download 18 Mbps, upload 8,4 Mbps, dan latency 40ms yang memadai untuk transmisi data ke Firebase. Sistem terbukti efektif meningkatkan keamanan dan efisiensi pengelolaan loker di lingkungan kampus dengan fleksibilitas operasi di lokasi dengan akses WiFi terbatas.

Kata Kunci: Internet of Things, Artificial Intelligence, Face Recognition, RFID, Firebase, Modul 4G, Raspberry Pi 5, Arduino Nano

POLITEKNIK
NEGERI
JAKARTA

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Design and Construction of a Telecommunication Lecturer Locker System Using a 4G Module

ABSTRACT

The development of Internet of Things (IoT) and Artificial Intelligence (AI) technologies enables the creation of complex and reliable security systems. Lecturers often face challenges in storing important personal belongings such as learning documents that are vulnerable to loss if left on campus or troublesome if carried home daily. This research aims to design, implement, and test a smart locker system specifically for Telecommunications lecturers that integrates IoT and AI with 4G module transmission. The system uses Raspberry Pi 5 as the main processing unit, Arduino Nano for RFID system, 4G SIM7600G-H module for data transmission, Firebase as real-time database, AI-based face recognition, and Android application. Supporting components include solenoid door lock and RFID as backup key. Test results show the system was successfully implemented. The RFID system achieved 100% success in reading UID cards, I2C communication between Arduino Nano and Raspberry Pi 5 ran stably, 12-channel relay operated optimally with voltages of 4.9V (HIGH) and 0.78-0.79V (LOW). 4G module testing resulted in average download speed of 18 Mbps, upload speed of 8.4 Mbps, and 40ms latency adequate for data transmission to Firebase. The system proved effective in improving security and efficiency of locker management on campus with operational flexibility in locations with limited WiFi access.

Keywords: *Internet of Things, Artificial Intelligence, Face Recognition, RFID, Firebase, 4G Module, Raspberry Pi 5, Arduino Nano*

POLITEKNIK
NEGERI
JAKARTA

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS.....	ii
LEMBAR PENGESAHAN.....	iii
KATA PENGANTAR.....	iv
ABSTRAK	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL	ix
DAFTAR GAMBAR.....	x
DAFTAR RUMUS	xi
DAFTAR LAMPIRAN.....	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	2
1.3 Tujuan.....	2
1.4 Luaran.....	2
BAB II TINJAUAN PUSTAKA.....	3
2.1 Raspberry Pi	3
2.2 Internet.....	3
2.3 Arduino Nano	4
2.4 Modul LTE	4
2.5 RFID	5
2.6 Relay	5
2.7 Buzzer	6
2.8 Webcam	6
2.8 OLED LCD.....	7
2.9 Solenoid Door Lock.....	7
2.10 Expander I/O MCP.....	8
2.11 Quality of Service (QoS).....	9
BAB III RANCANGAN DAN REALISASI.....	11
3.1 Perancangan Sistem Alat.....	11
3.1.1 Deskripsi Alat.....	11
3.1.2 Cara Kerja Alat.....	13
3.1.3 Flowchart Cara Kerja Alat	13
3.1.4 Diagram Blok	15
3.1.5 Spesifikasi Alat.....	16
3.2 Perencanaan Alat.....	18
3.2.1 Perencanaan Sistem Arduino Nano	18
3.2.2 Perencanaan Sistem Raspberry Pi 5	20
3.2.3 Perencanaan Penyambungan Buzzer	21
3.2.4 Perencanaan Penyambungan OLED.....	22
3.2.5 Perencanaan Penyambungan MCP23017 (GPIO Expander).....	24
3.2.6 Perencanaan Penyambungan Relay	25
3.2.7 Perencanaan Penyambungan Solenoid Doorlock	27
3.3 Realisasi Alat.....	28
3.3.1. Realisasi Perangkat Sistem.....	28
3.3.2. Realisasi LCD OLED I2C	29
3.3.3. Realisasi Sistem Relay	29
3.3.4. Realisasi Solenoid Door Lock	30
3.3.5. Realisasi Webcam AI untuk Akses Utama	31
3.3.6. Realisasi RFID Card untuk Akses cadangan	32

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

3.3.7. Realisasi Modul 4G	33
3.3.8. Realisasi Pemrograman Mikrokontroller Arduino Nano.....	33
3.3.9. Realisasi Pemrograman Komponen.....	36
3.3.10. Realisasi Pemrograman Raspberry Pi dan Firebase	39
BAB IV PEMBAHASAN.....	43
4.1 Pengujian Sistem RFID	43
4.1.1 Deskripsi Pengujian Sistem RFID	43
4.1.2 Prosedur Pengujian Sistem RFID	44
4.1.3 Hasil Pengujian <i>Scanner</i> RFID.....	45
4.2 Pengujian Koneksi I2C	46
4.2.1 Deskripsi Pengujian Koneksi I2C.....	46
4.2.2 Prosedur Pengujian Koneksi I2C.....	47
4.2.3 Hasil Pengujian Koneksi I2C	47
4.3 Pengujian Sistem <i>Relay</i>	49
4.3.1 Deskripsi Pengujian Sistem <i>Relay</i>	49
4.3.2 Prosedur Pengujian Sistem <i>Relay</i>	50
4.3.3 Hasil Pengujian Sistem <i>Relay</i>	50
4.4 Pengujian Koneksi Firebase	51
4.4.1 Deskripsi Pengujian Koneksi Firebase.....	51
4.4.2 Prosedur Pengujian Koneksi Firebase	52
4.4.3 Hasil Pengujian Koneksi Firebase.....	52
4.5 Pengujian <i>Display</i> OLED	53
4.5.1 Deskripsi Pengujian <i>Display</i> OLED.....	53
4.5.2 Prosedur Pengujian <i>Display</i> OLED	54
4.5.3 Hasil Pengujian <i>Display</i> OLED.....	54
4.6 Pengujian Kecepatan Modul 4G.....	55
4.6.1 Deskripsi Pengujian Kecepatan Modul 4G	55
4.6.2 Prosedur Pengujian Kecepatan Modul 4G.....	55
4.6.3 Hasil Pengujian Kecepatan Modul 4G	56
4.7 Analisa Sistem	56
BAB V KESIMPULAN.....	58
DAFTAR RIWAYAT HIDUP	59
DAFTAR PUSTAKA	60
LAMPIRAN.....	61



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR TABEL

Tabel 2. 1 Kategori <i>Throughput</i>	9
Tabel 2.2 Kategori <i>Packet Loss</i>	10
Tabel 2.3 Kategori <i>Delay</i>	10
Tabel 3.1 Spesifikasi Komponen.....	16
Tabel 3.2 Pin-Pin Pada Arduino Nano.....	19
Tabel 3.3 <i>Pin-Pin</i> Pada Raspberry Pi 5	21
Tabel 3.4 <i>Pin-Pin</i> Pada Penyambungan <i>Buzzer</i>	22
Tabel 3.5 <i>Pin-Pin</i> Pada Penyambungan OLED.....	23
Tabel 3.6 Pin-Pin Pada Penyambungan MCP23017	24
Tabel 3.7 Pin-Pin Pada Penyambungan <i>Relay</i>	26
Tabel 3.8 Pin-Pin Pada Penyambungan Solenoid <i>Doorlock</i>	27
Table 4.1 Hasil Pengujian <i>Scanner</i> RFID.....	45
Table 4.2 Hasil Pengujian Koneksi I2C	48
Tabel 4.3 Hasil Pengujian Data Relay.....	51
Tabel 4.4 Hasil Pengujian <i>Speedtest</i> Modul 4G	56

**POLITEKNIK
NEGERI
JAKARTA**



DAFTAR GAMBAR

Gambar 2.1 Raspberry Pi Model 5	3
Gambar 2.2 Arduino Nano	4
Gambar 2.3 Modul LTE	5
Gambar 2.4 RFID	5
Gambar 2.5 <i>Relay</i>	6
Gambar 2.6 <i>Buzzer</i>	6
Gambar 2.7 Webcam	7
Gambar 2.8 OLED LCD	7
Gambar 2.9 Solenoid <i>Door Lock</i>	8
Gambar 2.9 MCP23017.....	9
Gambar 3.1 Ilustrasi Alat Sistem Loker Dosen Telekomunikasi.....	12
Gambar 3.2 Ilustrasi Penempatan Sistem Pada Loker.....	12
Gambar 3.3 <i>Flowchart</i> sistem loker dosen telekomunikasi	14
Gambar 3.4 Diagram Blok Sistem Loker Dosen Telekomunikasi	15
Gambar 3.5 Rangkaian Skematik Arduino Nano	19
Gambar 3.6 Rangkaian Skematik Raspberry Pi 5	20
Gambar 3.7 Rangkaian Skematik Penyambungan <i>Buzzer</i>	22
Gambar 3.8 Rangkaian Skematik Penyambungan OLED.....	23
Gambar 3.9 Rangkaian Skematik MCP23017 (<i>GPIO Expander</i>)	24
Gambar 3.10 Rangkaian Skematik Penyambungan <i>Relay</i>	25
Gambar 3.11 Rangkaian Skematik Penyambungan Solenoid <i>Doorlock</i>	27
Gambar 3.12 Realisasi Perangkat Sistem <i>Smart Loker</i>	28
Gambar 3.13 Realisasi LCD OLED I2C 128x64	29
Gambar 3.14 Realisasi Sistem Relay.....	29
Gambar 3.15 Realisasi Solenoid <i>Door Lock</i>	30
Gambar 3.16 Webcam AI	31
Gambar 3.17 RFID <i>Card</i>	32
Gambar 3.18 Realisasi Modul 4G	33
Gambar 4.1 <i>Set-up</i> Rangkaian Sistem RFID.....	44
Gambar 4.2 <i>Serial Monitor</i> Arduino Nano.....	45
Gambar 4.3 <i>Set-up</i> Rangkaian Pengujian Koneksi I2C.....	46
Gambar 4.4 Serial Monitor <i>board</i> Arduino Nano.....	47
Gambar 4.5 <i>Terminal</i> Pi 5 Menerima data UID	48
Gambar 4.6 <i>Set-Up</i> Rangkaian <i>Relay</i> Sistem	49
Gambar 4.7 Ilustrasi Pengujian Koneksi Firebase	52
Gambar 4.8 Tampilan <i>Real-Time</i> Firebase	53
Gambar 4.9 Terminal Raspberry Pi 5	53
Gambar 4.10 Display OLED Menampilkan Data RFID	54
Gambar 4.11 Display OLED Menampilkan Firebase.....	55

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR RUMUS

(2.1) <i>Throughput</i>	12
(2.2) <i>Packet Loss</i>	13
(2.3) <i>Delay</i>	13





DAFTAR LAMPIRAN

L - 1 Luaran Sistem Utama Alat.....	54
L - 2 <i>Casing</i> Sistem Utama Alat <i>Smart Loker</i>	55
L - 3 Raspberry Pi 5 Sistem <i>Smart Loker</i>	56
L - 4 Source Code Arduino Nano	57
L - 5 Source Code Raspberry Pi 5	61



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

BAB I PENDAHULUAN

1.1. Latar Belakang

Pada era modern, teknologi seperti *Internet of Things* (IoT) hadir untuk mempermudah pekerjaan manusia dengan sistemnya yang handal dan termutakhir, ditambah dengan adanya teknologi *Artificial Intelligence* (AI) menjadi kunci dalam menciptakan sistem keamanan yang kompleks dan sulit dibobol. Salah satu penerapannya adalah sistem loker berbasis *Face Recognition* dan Android, yang memungkinkan autentikasi cepat dan akurat untuk melindungi serta menyimpan barang dengan aman.

Sistem ini didukung oleh Raspberry Pi 5 sebagai *Single Board Computer* (SBC), modul *Long-Term Evolution* (LTE/4G) dengan model SIM7600 M.2 untuk transmisi data ke Firebase sebagai database, solenoid *door lock* sebagai pengunci otomatis, dan *Radio Frequency Identification* (RFID) sebagai kunci cadangan. AI disini mampu bekerja untuk mempelajari dan membuat keputusan sendiri, sehingga IoT mampu menjadi lebih cerdas dan responsif terhadap lingkungannya. Dengan menggunakan aplikasi android mempermudah penggunaan dalam sistem. Kombinasi perangkat ini memastikan keamanan, kemudahan, dan keandalan sistem loker.

Dosen program studi Telekomunikasi sering menghadapi kendala dalam menyimpan barang pribadi yang penting, seperti dokumen untuk proses pembelajaran dan penilaian mahasiswa. Barang-barang ini rentan hilang jika ditinggalkan di kampus atau merepotkan jika harus dibawa pulang setiap hari. Kendala ini dapat mengganggu efisiensi proses mengajar.

Berdasarkan latar belakang tersebut, dibuatlah sistem *smart locker* untuk dosen Telekomunikasi dengan judul "Rancang Bangun Sistem Loker Dosen Telekomunikasi Menggunakan Transmisi Modul 4G." Sistem ini tidak hanya mengintegrasikan teknologi AI dan IoT, tetapi juga mengusung konsep keamanan digital dan kenyamanan akses melalui perangkat Android, sehingga dapat membantu dosen menyimpan barang penting dengan aman dan efisien.



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

1.2. Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan diatas, terdapat beberapa permasalahan yang akan dibahas dalam tugas akhir ini ialah sebagai berikut:

1. Bagaimana cara merancang Sistem Loker Dosen Telekomunikasi Menggunakan Transmisi Modul 4G?
2. Bagaimana cara Sistem Loker Dosen Menggunakan Transmisi Modul 4G?
3. Bagaimana kinerja Sistem Loker Dosen Telekomunikasi Menggunakan Transmisi Modul 4G?

1.3. Tujuan

Tujuan yang dapat tercapai dari tugas akhir ini ialah sebagai berikut:

1. Mampu merancang Sistem Loker Dosen Telekomunikasi Menggunakan Transmisi Modul 4G.
2. Mampu merealisasikan Sistem Loker Dosen Telekomunikasi Menggunakan Transmisi Modul 4G.
3. Mampu melakukan pengujian Sistem Loker Dosen Telekomunikasi Menggunakan Transmisi Modul 4G serta pengujian kecepatan jaringan internet.

1.4. Luaran

Adapun bentuk luaran dari tugas akhir ini adalah:

1. *Prototype* Alat.
2. Laporan Tugas Akhir
3. Artikel ilmiah.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

BAB V KESIMPULAN

Berdasarkan hasil perancangan, realisasi, dan pengujian sistem loker dosen telekomunikasi yang telah dilakukan, maka dapat ditarik kesimpulan sebagai berikut:

1. Perancangan sistem loker dosen telekomunikasi menggunakan transmisi modul 4G telah berhasil direalisasikan dengan menggunakan beberapa komponen utama yang terdiri dari Arduino Nano sebagai pengendali sistem RFID, Raspberry Pi 5 sebagai unit pemrosesan utama, dan koneksi Firebase sebagai *database real-time* untuk komunikasi dengan aplikasi dan sistem AI *Face Recognition*.
2. Realisasi sistem loker dosen telekomunikasi telah berhasil diimplementasikan dengan baik. Hasil pengujian menunjukkan, Sistem RFID berhasil membaca 100% kartu dengan UID unik oleh *scanner* RFID, menggunakan komunikasi I2C yang stabil antara Arduino Nano dan Raspberry Pi 5 untuk transfer data UID kartu RFID, dan mampu Mengendalikan 12 *channel relay* dengan tegangan operasi 4.9V (*HIGH*) dan 0.78-0.79V (*LOW*).
3. Pengujian kecepatan modul 4G SIM7600G-H menggunakan provider IM3 menunjukkan rata-rata kecepatan *Download* 18 Mbps, rata-rata *upload* 8,4 Mbps, dan 40ms *latency* yang konsisten, Kecepatan jaringan tersebut terbukti mencukupi untuk melakukan transmisi data ke Firebase.

Sistem ini dapat diimplementasikan untuk meningkatkan keamanan dan efisiensi pengelolaan loker di lingkungan kampus. Penggunaan transmisi modul 4G memastikan sistem dapat beroperasi di lokasi dengan akses WiFi terbatas sambil tetap terhubung ke sistem. Adapun saran untuk pengembangan lanjutan yaitu: membuat sistem *auto start* saat di inisialisasi awal, dan optimasi dalam konsumsi daya sistem.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR RIWAYAT HIDUP



Muhammad Zatiero Adrafa Lahir di Tangerang, 28 Maret 2005. Lulus dari SD Karawaci 8 Kota Tangerang Tahun 2016, SMP Islamic Centre Kota Tangerang Tahun 2019, dan SMA Islamic Centre Kota Tangerang Tahun 2022. Gelar Diploma Tiga (D3) diperoleh tahun 2025 dari Program Studi Telekomunikasi, Jurusan Teknik Elektro, Politeknik Negeri Jakarta.





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR PUSTAKA

- Arsyad, O. R., & Kartika, K. P. (2021). Rancang Bangun Alat Pengaman Brankas Menggunakan Sensor Sidik Jari Berbasis Arduino. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 5(1), 1-6.
- Berliani, D., Saragih, Y., & Latifa, U. (2021). Pemanfaatan 4G LTE dengan mikrokomputer Raspberry pi. *JET (Journal of Electrical Technology)*, 6(1), 1-6
- Clinton, R. M., & Sengkey, R. (2019). Purwarupa Sistem Daftar Pelanggaran Lalulintas Berbasis Mini-Komputer Raspberry Pi. *Jurnal Teknik Elektro dan Komputer*, 8(3), 181-192.
- Jakaria, D. A., & Fauzi, M. R. (2020). Aplikasi Smartphone Dengan Perintah Suara Untuk Mengendalikan Saklar Listrik Menggunakan Arduino. *Jurnal Teknik Informatika (JUTEKIN)*, 8(1).
- Luthfiah, N. I. (2023). Optimasi Pelayanan Perpustakaan Menggunakan Teknologi Rfid Di Upt Perpustakaan Itb. *Jurnal Multidisipliner Kapalamada*, 2(04), 240-252.
- Nugroho, H., Kurniawan, M., & Saidatin, N. (2019, September). Deteksi Wajah dan Mata dengan Menggunakan Metode Fitur Haar-Like pada Kamera WebCam. In *Prosiding Seminar Nasional Sains dan Teknologi Terapan (Vol. 1, No. 1, pp. 261-266)*.
- Oktodinata, W., & Purnomo, Y. (2024). Perangkat Jam Portabel dengan Fungsi Pembaca Suhu dan Pelacakan Suara Melalui Buzzer Menggunakan Modul Nrf Berbasis Arduino. *Jurnal Teknik Indonesia*, 3(3), 115-128.
- Rory, D. I., Putro, I. H., & Khoswanto, H. (2024). Perancangan Dan Pemantauan Locker Menggunakan Ktm Berbasis Esp8266. *Jurnal Teknik Elektro*, 17(1), 36-43.



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Triawan, Y., Sardi, J., & Hamka Air Tawar, J. (2020). Perancangan Sistem Otomatisasi Pada Aquascape Berbasis Mikrokontroller Arduino Nano. JTEIN: Jurnal Teknik Elektro Indonesia, 1(2), 76-83.

Triatmojo, F. (2023). SISTEM MONITORING KEPADATAN PENUMPANG BUSWAY BERBASIS UBIDOTS (Doctoral dissertation, Universitas Teknologi Digital Indonesia).

Zakiansyah, M., & Sutabri, T. (2025). Integrasi Internet of Things (IoT) dan Kecerdasan Buatan (AI) untuk Smart City di Indonesia. JOURNAL SAINS STUDENT RESEARCH, 3(2), 346-354.





© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

LAMPIRAN



L-1

Luaran Sistem Utama Alat



**PROGRAM STUDI
TELEKOMUNIKASI
JURUSAN TEKNIK
ELEKTRO
POLITEKNIK NEGERI
JAKARTA**

Digambar :

Muhammad Zatihero
Adrafa

Diperiksa :

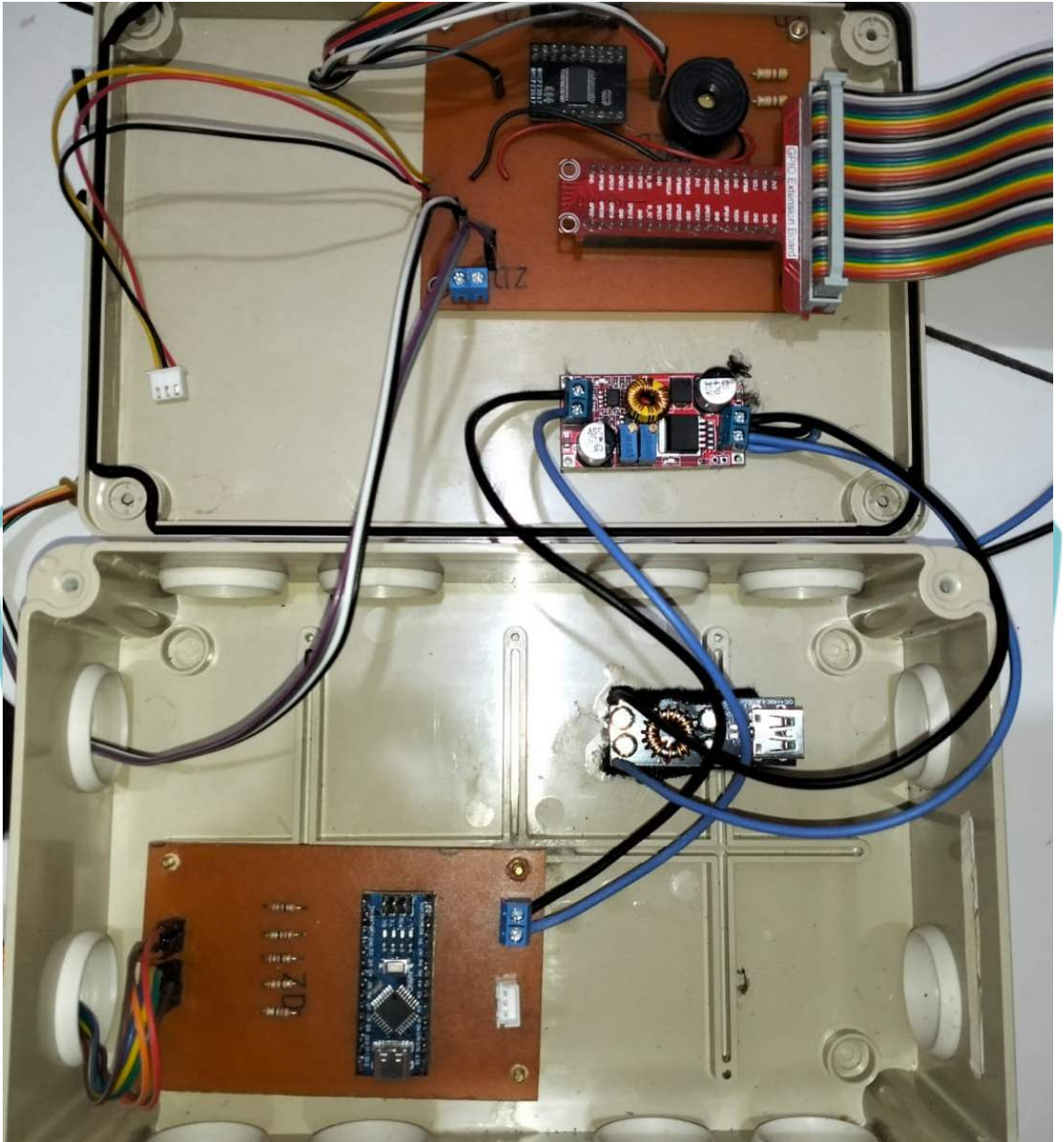
Benny Nixon, S.T.,
M.T.

Tanggal:

15 Juli 2025

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



L-2

Casing Sistem Utama Alat Smart Loker



**PROGRAM STUDI
TELEKOMUNIKASI
JURUSAN TEKNIK
ELEKTRO
POLITEKNIK NEGERI
JAKARTA**

<i>Digambar :</i>	Muhammad Zatihero Adrafa
<i>Diperiksa :</i>	Benny Nixon, S.T., M.T.
<i>Tanggal:</i>	17 Juli 2025



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



L-3

Raspberry Pi 5 Sistem *Smart Loker*



**PROGRAM STUDI
TELEKOMUNIKASI
JURUSAN TEKNIK
ELEKTRO
POLITEKNIK NEGERI
JAKARTA**

<i>Digambar :</i>	Muhammad Zatihero Adrafa
<i>Diperiksa :</i>	Benny Nixon, S.T., M.T.
<i>Tanggal:</i>	17 Juli 2025



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
#include <SPI.h>
#include <MFRC522.h>
#include <Wire.h>

#define RST_PIN 9
#define SS_PIN_1 10
#define SS_PIN_2 8
#define SS_PIN_3 7
#define I2C_ADDRESS 0x08

// Create MFRC522 instances
MFRC522 rfid1(SS_PIN_1, RST_PIN);
MFRC522 rfid2(SS_PIN_2, RST_PIN);
MFRC522 rfid3(SS_PIN_3, RST_PIN);

// Data buffer structure
struct RFIDData {
    uint8_t readerID;
    uint8_t uid[10];
    uint8_t uidSize;
    uint8_t data[16];
    uint8_t dataSize;
    bool hasData;
};

RFIDData dataBuffer;
bool dataReady = false;
bool writeMode = false;
uint8_t writeData[16];
uint8_t writeDataSize = 0;
uint8_t targetReader = 0;

void setup() {
    Serial.begin(9600);
    SPI.begin();

    // Initialize I2C as slave
    Wire.begin(I2C_ADDRESS);
    Wire.onReceive(receiveEvent);
    Wire.onRequest(requestEvent);

    // Initialize RFID readers
    rfid1.PCD_Init();
    rfid2.PCD_Init();
    rfid3.PCD_Init();

    // Clear buffer
    clearBuffer();

    Serial.println("Arduino Nano RFID Controller Ready");
    Serial.println("Commands:");
    Serial.println("w<reader><data> - Write data to tag (e.g.,
        w1Hello World)");
    Serial.println("r - Read mode");
}
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
void loop() {
    // Check serial commands
    if (Serial.available()) {
        handleSerialCommand();
    }

    // Check RFID readers if not in write mode or if data already
    sent
    if (!writeMode && !dataReady) {
        checkRFIDReader(1, rfid1);
        checkRFIDReader(2, rfid2);
        checkRFIDReader(3, rfid3);
    }

    // Handle write mode
    if (writeMode && writeDataSize > 0) {
        performWrite();
    }

    delay(100);
}

void handleSerialCommand() {
    String command = Serial.readString();
    command.trim();

    if (command.startsWith("w")) {
        // Write command: w<reader><data>
        if (command.length() > 2) {
            targetReader = command.charAt(1) - '0';
            String data = command.substring(2);

            if (targetReader >= 1 && targetReader <= 3 && data.length()
                <= 16) {
                writeMode = true;
                writeDataSize = data.length();
                data.getBytes(writeData, writeDataSize + 1);
                Serial.println("Write mode activated. Present tag to reader
" + String(targetReader));
            } else {
                Serial.println("Invalid write command format");
            }
        }
    } else if (command == "r") {
        writeMode = false;
        Serial.println("Read mode activated");
    }
}

void checkRFIDReader(uint8_t readerID, MFRC522 &rfid) {
    if (!rfid.PICC_IsNewCardPresent()
        || !rfid.PICC_ReadCardSerial()) {
        return;
    }

    Serial.println("Card detected on reader " + String(readerID));

    // Store UID
    dataBuffer.readerID = readerID;
}
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
dataBuffer.uidSize = rfid.uid.size;
for (uint8_t i = 0; i < rfid.uid.size; i++) {
    dataBuffer.uid[i] = rfid.uid.uidByte[i];
}

// Read data from tag
readDataFromTag(rfid);

dataBuffer.hasData = true;
dataReady = true;

// Print UID
Serial.print("UID: ");
for (uint8_t i = 0; i < dataBuffer.uidSize; i++) {
    Serial.print(dataBuffer.uid[i], HEX);
    Serial.print(" ");
}
Serial.println();

// Print data
Serial.print("Data: ");
for (uint8_t i = 0; i < dataBuffer.dataSize; i++) {
    Serial.print((char)dataBuffer.data[i]);
}
Serial.println();

rfid.PICC_HaltA();
rfid.PCD_StopCrypto1();
}

void readDataFromTag(MFRC522 &rfid) {
    MFRC522::StatusCode status;
    uint8_t buffer[18];
    uint8_t size = sizeof(buffer);

    // Read from block 4 (first data block)
    status = rfid.MIFARE_Read(4, buffer, &size);

    if (status == MFRC522::STATUS_OK) {
        // Find actual data length (stop at null terminator or padding)
        dataBuffer.dataSize = 0;
        for (uint8_t i = 0; i < 16; i++) {
            if (buffer[i] != 0x00 && buffer[i] != 0xFF) {
                dataBuffer.data[dataBuffer.dataSize++] = buffer[i];
            } else {
                break;
            }
        }
    } else {
        dataBuffer.dataSize = 0;
        Serial.println("Failed to read data from tag");
    }
}

void performWrite() {
    MFRC522* currentRFID = nullptr;

    switch (targetReader) {
        case 1: currentRFID = &rfid1; break;
    }
}
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
case 2: currentRFID = &rfid2; break;
case 3: currentRFID = &rfid3; break;
default: return;
}

if (!currentRFID->PICC_IsNewCardPresent() || !currentRFID->PICC_ReadCardSerial()) {
    return;
}

Serial.println("Writing to tag...");

// Prepare data buffer (16 bytes)
uint8_t dataBlock[16];
memset(dataBlock, 0, 16);
memcpy(dataBlock, writeData, writeDataSize);

// Write to block 4
MFRC522::StatusCode status = currentRFID->MIFARE_Write(4,
    dataBlock, 16);

if (status == MFRC522::STATUS_OK) {
    Serial.println("Write successful!");
} else {
    Serial.println("Write failed!");
}

currentRFID->PICC_HaltA();
currentRFID->PCD_StopCrypto1();

// Reset write mode
writeMode = false;
writeDataSize = 0;
Serial.println("Write mode deactivated");
}

void receiveEvent(int bytes) {
    uint8_t command = Wire.read();

    if (command == 0x01) {
        // Request to send data - Pi is ready to receive
        // Data will be sent in requestEvent
    } else if (command == 0x02) {
        // Clear buffer command
        clearBuffer();
    }
}

void requestEvent() {
    if (dataReady && dataBuffer.hasData) {
        // Send data structure
        Wire.write(dataBuffer.readerID);
        Wire.write(dataBuffer.uidSize);
        Wire.write(dataBuffer.uid, dataBuffer.uidSize);
        Wire.write(dataBuffer.dataSize);
        Wire.write(dataBuffer.data, dataBuffer.dataSize);

        // Clear buffer after sending
        clearBuffer();
    }
}
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
Serial.println("Data sent to Pi, buffer cleared");
} else {
    // Send empty response
    Wire.write(0x00);
}
}

void clearBuffer() {
    memset(&dataBuffer, 0, sizeof(dataBuffer));
    dataReady = false;
}
```

L - 5 Source Code Raspberry Pi 5

```
#!/usr/bin/env python3
"""
Simplified Unified RFID Relay Controller
Runs in both RFID + Firebase monitoring mode
"""

import time
import threading
import json
import os
from typing import Dict, Any
from threading import Thread, Lock
import smbus2

# Hardware imports
import board
import busio
import digitalio
from adafruit_mcp230xx.mcp23017 import MCP23017
from gpiozero import Buzzer

# OLED imports
from luma.core.interface.serial import i2c
from luma.core.render import canvas
from luma.oled.device import sh1106
from PIL import ImageFont

# Firebase imports
import firebase_admin
from firebase_admin import credentials, db

class UnifiedRelayController:
    def __init__(self):
        """Initialize the unified relay controller"""
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
print("Initializing Unified RFID Relay Controller...")

# Hardcoded Firebase configuration
self.firebase_url = "https://your-project-default-
rtddb.firebaseio.com/"

self.service_account_path =
"/path/to/your/serviceAccountKey.json"

# Hardware components
self.mcp = None
self.relay_pins = []
self.buzzer = None
self.oled = None

# State variables
self.running = False
self.lock = Lock()

# Firebase-related variables
self.current_waktu = {}
self.relay_timers = {}
self.uid_to_relay = {str(1000 + i): i-1 for i in range(1,
13)} # 1001-1012 -> 0-11

# RFID-related variables
self.i2c_bus = None
self.arduino_address = 0x08
self.config_file = "rfid_config.json"
self.tag_relay_map = {}
self.current_tag = None
self.current_reader = None
self.current_data = None
self.last_tag_time = None

# Initialize components
self.setup_hardware()
self.setup_firebase()
self.setup_rfid()
self.tag_relay_map = self.load_config()

def setup_hardware(self):
    """Initialize MCP23017, buzzer, and OLED"""
    # Initialize I2C bus for MCP23017
    i2c_bus = busio.I2C(board.SCL, board.SDA)

    # Initialize MCP23017 (default address 0x20)
    self.mcp = MCP23017(i2c_bus)
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
# Setup relay pins (12 relays on pins 0-11)
self.relay_pins = []
for i in range(12):
    pin = self.mcp.get_pin(i)
    pin.direction = digitalio.Direction.OUTPUT
    pin.value = True # Relay modules are usually active
LOW, so True = OFF
    self.relay_pins.append(pin)

# Setup buzzer
self.buzzer = Buzzer(17) # GPIO 17

# Setup OLED display
serial = i2c(port=1, address=0x3C)
self.oled = sh1106(serial, width=128, height=64)

# Load fonts for OLED
self.font = ImageFont.truetype("/usr/share/fonts/truetype/dejavu/DejaVuSans.ttf", 10)
self.font_small = ImageFont.truetype("/usr/share/fonts/truetype/dejavu/DejaVuSans.ttf", 8)

# Initialize display
self.update_oled("Unified Relay System", "Mode: BOTH",
"Initializing...", "")

def setup_firebase(self):
    """Initialize Firebase connection"""
    # Initialize Firebase Admin SDK
    cred = credentials.Certificate(self.service_account_path)
    firebase_admin.initialize_app(cred, {
        'databaseURL': self.firebase_url
    })

def setup_rfid(self):
    """Initialize RFID components"""
    # Initialize I2C bus for Arduino communication
    self.i2c_bus = smbus2.SMBus(1)

def load_config(self):
    """Load tag-to-relay mapping from JSON file"""
    if os.path.exists(self.config_file):
        with open(self.config_file, 'r') as f:
            config = json.load(f)
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
        return config
    return {}

def save_config(self):
    """Save tag-to-relay mapping to JSON file"""
    with open(self.config_file, 'w') as f:
        json.dump(self.tag_relay_map, f, indent=2)

def update_oled(self, line1, line2="", line3="", line4=""):
    """Update OLED display with up to 4 lines"""
    with canvas(self.oled) as draw:
        draw.text((0, 0), str(line1), font=self.font,
            fill="white")
        if line2:
            draw.text((0, 12), str(line2), font=self.font,
            fill="white")
        if line3:
            draw.text((0, 24), str(line3), font=self.font_small,
            fill="white")
        if line4:
            draw.text((0, 36), str(line4), font=self.font_small,
            fill="white")

def buzz_indicator(self, pattern="single"):
    """Buzzer indicator patterns"""
    if pattern == "single":
        self.buzzer.on()
        time.sleep(0.1)
        self.buzzer.off()
    elif pattern == "double":
        for _ in range(2):
            self.buzzer.on()
            time.sleep(0.1)
            self.buzzer.off()
            time.sleep(0.1)
    elif pattern == "success":
        self.buzzer.on()
        time.sleep(0.3)
        self.buzzer.off()
    elif pattern == "error":
        for _ in range(3):
            self.buzzer.on()
            time.sleep(0.1)
            self.buzzer.off()
            time.sleep(0.1)
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
def set_relay(self, relay_number: int, state: bool, duration: float = 0):
    """Set relay state"""
    if 1 <= relay_number <= 12:
        pin_index = relay_number - 1
        # Invert state because relay modules are usually active
        LOW
        self.relay_pins[pin_index].value = not state

        # Auto-off timer if duration specified
        if state and duration > 0:
            timer = threading.Timer(duration, self.set_relay,
args=[relay_number, False])
            timer.start()

        return True
    else:
        return False

def turn_off_all_relays(self):
    """Turn off all relays"""
    for i in range(1, 13):
        self.set_relay(i, False)

# === FIREBASE MODE METHODS ===

def process_waktu_change(self, uid: str, waktu: str):
    """Process waktu (time) change for a specific UID (Firebase
mode)"""
    if uid in self.uid_to_relay:
        relay_number = self.uid_to_relay[uid] + 1 # Convert
back to 1-12

        # Check if waktu changed (indicating new detection)
        if self.current_waktu.get(uid) != waktu:
            old_waktu = self.current_waktu.get(uid, "None")
            self.current_waktu[uid] = waktu

            # Cancel existing timer for this relay if any
            if uid in self.relay_timers:
                self.relay_timers[uid].cancel()

            # Turn ON relay when waktu changes (detection
occurred)
            self.set_relay(relay_number, True)

        # Update OLED
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
self.update_oled(
    "Firebase Mode",
    f"UID: {uid}",
    f"Relay {relay_number} ON",
    f"Time: {waktu}"
)

# Buzzer indication
self.buzz_indicator("success")

# Set timer to turn OFF relay after 5 seconds
self.relay_timers[uid] = threading.Timer(
    5.0,
    self.auto_turn_off_relay,
    args=[uid, relay_number]
)
self.relay_timers[uid].start()

def auto_turn_off_relay(self, uid: str, relay_number: int):
    """Automatically turn off relay after timer expires
    (Firebase mode)"""
    self.set_relay(relay_number, False)

    # Remove timer from tracking
    if uid in self.relay_timers:
        del self.relay_timers[uid]

def firebase_listener(self, event):
    """Firebase database event listener"""
    if event.data is not None:
        # Get current data from status_pengenalan
        ref = db.reference('status_pengenalan')
        data = ref.get()

        if data:
            for uid in [str(1000 + i) for i in range(1, 13)]: #
1001-1012
                if uid in data and 'waktu' in data[uid]:
                    waktu = data[uid]['waktu']
                    self.process_waktu_change(uid, waktu)

def start_firebase_monitoring(self):
    """Start monitoring Firebase database"""
    # Set up Firebase listener
    ref = db.reference('status_pengenalan')
    ref.listen(self.firebase_listener)
```



Hak Cipta:

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
# Initial state check
data = ref.get()
if data:
    for uid in [str(1000 + i) for i in range(1, 13)]: #
1001-1012
        if uid in data and 'waktu' in data[uid]:
            waktu = data[uid]['waktu']
            self.current_waktu[uid] = waktu # Store initial
waktu without triggering relay

        self.update_oled("Firebase Mode", "Monitoring Active",
"Waiting for changes...", "")

# === RFID MODE METHODS ===

def uid_to_string(self, uid_bytes):
    """Convert UID bytes to hex string"""
    return ''.join(f'{b:02X}' for b in uid_bytes)

def read_rfid_data(self):
    """Read RFID data from Arduino Nano via I2C"""
    # Send request command to Arduino
    self.i2c_bus.write_byte(self.arduino_address, 0x01)
    time.sleep(0.05) # Small delay for Arduino to prepare data

    # Read response from Arduino
    response =
self.i2c_bus.read_i2c_block_data(self.arduino_address, 0, 32)

    if response[0] == 0x00: # No data available
        return None

    # Parse the response
    reader_id = response[0]
    uid_size = response[1]

    if uid_size > 10 or uid_size == 0: # Sanity check
        return None

    uid = response[2:2+uid_size]
    data_size = response[2+uid_size]

    if data_size > 16: # Sanity check
        data_size = 16

    if data_size > 0:
        tag_data = response[3+uid_size:3+uid_size+data_size]
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
        tag_data_str = bytes(tag_data).decode('utf-8',
errors='ignore').strip()
    else:
        tag_data_str = ""

    return {
        'reader_id': reader_id,
        'uid': uid,
        'data': tag_data_str
    }

def process_rfid_tag(self, tag_info):
    """Process detected RFID tag"""
    with self.lock:
        uid_str = self.uid_to_string(tag_info['uid'])

        # Update current tag info
        self.current_tag = uid_str
        self.current_reader = tag_info['reader_id']
        self.current_data = tag_info['data']
        self.last_tag_time = time.strftime("%H:%M:%S")

        # Update OLED display with tag data
        mapped_relay = self.tag_relay_map.get(uid_str, "None")
        self.update_oled(
            f"Reader: {self.current_reader}",
            f"UID: {uid_str[:16]}",
            f>Data: {self.current_data[:18]}",
            f"Relay: {mapped_relay} | {self.last_tag_time}"
        )

        # Buzzer indicator for tag detection
        self.buzz_indicator("single")

        # Check if tag is mapped to a relay
        if uid_str in self.tag_relay_map:
            relay_num = self.tag_relay_map[uid_str]
            if self.set_relay(relay_num, True, 5.0): # Auto-
off after 5 seconds
                self.buzz_indicator("success")
            else:
                self.buzz_indicator("error")
        else:
            self.buzz_indicator("double")

def monitor_rfid(self):
    """Continuously monitor RFID tags from Arduino"""
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
self.update_oled("RFID Mode", "Monitoring Active", "Waiting
for tags...", "")

while self.running:
    tag_info = self.read_rfid_data()
    if tag_info:
        self.process_rfid_tag(tag_info)
        time.sleep(0.3) # Check every 300ms

# === INTERACTIVE MENU METHODS ===

def show_mappings(self):
    """Display current tag-relay mappings"""
    if not self.tag_relax_map:
        print("\nNo tag mappings configured yet.")
        return

    print(f"\nCurrent Tag-Relay Mappings
({len(self.tag_relax_map)} total):")
    print("-" * 50)
    for i, (tag, relay) in
enumerate(self.tag_relax_map.items(), 1):
        print(f"{i:2d}. Tag: {tag} -> Relay: {relay}")

def add_edit_mapping(self):
    """Add or edit tag-relay mapping"""
    print("\n--- ADD/EDIT TAG MAPPING ---")
    print("Options:")
    print("1. Use current detected tag")
    print("2. Enter tag UID manually")

    option = input("Choose option (1/2): ").strip()

    if option == '1':
        if not self.current_tag:
            print("No tag currently detected. Present a tag
first.")
            return
        uid = self.current_tag
        print(f"Using current tag: {uid}")
    elif option == '2':
        uid = input("Enter tag UID (hex format, e.g., 1A2B3C4D):
").strip().upper()
        if not uid:
            print("Invalid UID entered.")
            return
    else:
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
print("Invalid option.")
return

    relay_num = int(input(f"Enter relay number (1-12) for tag
{uid}: "))
    if 1 <= relay_num <= 12:
        self.tag_relay_map[uid] = relay_num
        self.save_config()
        print(f"✓ Mapping saved: {uid} -> Relay {relay_num}")
    else:
        print("Invalid relay number. Must be 1-12.")

def test_relay(self):
    """Test individual relay"""
    relay_num = int(input("Enter relay number to test (1-12):
"))
    if 1 <= relay_num <= 12:
        print(f"Testing relay {relay_num}...")
        self.set_relay(relay_num, True)
        time.sleep(2)
        self.set_relay(relay_num, False)
        print("✓ Relay test complete")
    else:
        print("Invalid relay number. Must be 1-12.")

def interactive_menu(self):
    """Interactive menu system"""
    while self.running:
        print("\n" + "="*60)
        print("      UNIFIED RFID RELAY CONTROL SYSTEM")
        print("      Mode: BOTH")
        print("="*60)
        print("1. View tag-relay mappings")
        print("2. Add/Edit tag mapping")
        print("3. Remove tag mapping")
        print("4. Test relay")
        print("5. Test buzzer patterns")
        print("6. Manual relay control")
        print("7. Turn off all relays")
        print("8. Show current tag info")
        print("9. System status")
        print("0. Exit system")
        print("="*60)

        if self.current_tag:
            print(f>Last Tag: {self.current_tag} | Reader:
{self.current_reader} | {self.last_tag_time}")
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
print("-"*60)

choice = input("Enter your choice (0-9): ").strip()

if choice == '1':
    self.show_mappings()
elif choice == '2':
    self.add_edit_mapping()
elif choice == '3':
    self.remove_mapping()
elif choice == '4':
    self.test_relay()
elif choice == '5':
    self.test_buzzer()
elif choice == '6':
    self.manual_relay_control()
elif choice == '7':
    self.turn_off_all_relays()
    print("All relays turned OFF")
elif choice == '8':
    self.show_current_tag()
elif choice == '9':
    self.show_system_status()
elif choice == '0':
    print("Shutting down system...")
    self.stop()
    break
else:
    print("Invalid choice! Please enter 0-9.")

def remove_mapping(self):
    """Remove tag-relay mapping"""
    if not self.tag_relay_map:
        print("No mappings to remove.")
        return

print("\n--- REMOVE TAG MAPPING ---")
self.show_mappings()

uid = input("\nEnter tag UID to remove: ").strip().upper()

if uid in self.tag_relay_map:
    relay_num = self.tag_relay_map[uid]
    del self.tag_relay_map[uid]
    self.save_config()
    print(f"✓ Mapping removed: {uid} (was -> Relay {relay_num})")
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
else:
    print("Tag UID not found in mappings.")

def test_buzzer(self):
    """Test buzzer patterns"""
    print("\n--- BUZZER TEST ---")
    patterns = ["single", "double", "success", "error"]

    for pattern in patterns:
        print(f"Testing {pattern} pattern...")
        self.buzz_indicator(pattern)
        time.sleep(1)

    print("✓ Buzzer test complete")

def manual_relay_control(self):
    """Manual relay control"""
    relay_num = int(input("Enter relay number (1-12): "))
    if 1 <= relay_num <= 12:
        state = input("Turn ON or OFF? (on/off): ")
        state = state.strip().lower()
        if state in ['on', '1', 'true']:
            self.set_relay(relay_num, True)
            print(f"✓ Relay {relay_num} turned ON")
        elif state in ['off', '0', 'false']:
            self.set_relay(relay_num, False)
            print(f"✓ Relay {relay_num} turned OFF")
        else:
            print("Invalid state. Use 'on' or 'off'.")
    else:
        print("Invalid relay number. Must be 1-12.")

def show_current_tag(self):
    """Show current tag information"""
    with self.lock:
        if self.current_tag:
            print(f"\n--- CURRENT TAG INFO ---")
            print(f"UID: {self.current_tag}")
            print(f"Reader: {self.current_reader}")
            print(f>Data: '{self.current_data}'")
            print(f"Detection Time: {self.last_tag_time}")
            print(f"Mapped Relay: {self.tag_relay_map.get(self.current_tag, 'None')}")
        else:
            print("\nNo tag currently detected.")

def show_system_status(self):
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
"""Show system status"""
print(f"\n--- SYSTEM STATUS ---")
print(f"Operating Mode: BOTH")
print(f"Total Mappings: {len(self.tag_relay_map)}")
print(f"MCP23017 Relays: {len(self.relay_pins)} channels")
print(f"OLED Display: SH1106 128x64")
print(f"Buzzer GPIO: 17")
print(f"Config File: {self.config_file}")
    print(f>Last Tag: {self.current_tag if self.current_tag
else 'None'}")
    print(f"Firebase URL: {self.firebase_url}")
    print(f"Firebase UIDs: 1001-1012")
print(f"Arduino I2C Address: 0x{self.arduino_address:02X}")

def start(self):
    """Start the system"""
    self.running = True

    # Start Firebase monitoring thread
    firebase_thread = Thread(target=self.start_firebase_monitoring, daemon=True)
    firebase_thread.start()

    # Start RFID monitoring thread
    rfid_thread = Thread(target=self.monitor_rfid, daemon=True)
    rfid_thread.start()

    # Start interactive menu
    self.interactive_menu()

def stop(self):
    """Stop the system and cleanup"""
    self.running = False

    # Cancel all active timers
    for timer in self.relay_timers.values():
        timer.cancel()
    self.relay_timers.clear()

    # Turn off all relays
    self.turn_off_all_relays()

    # Cleanup hardware
    if self.oled:
        self.oled.clear()
    if self.buzzer:
        self.buzzer.close()
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
def main():  
    """Main function"""  
    print("=== UNIFIED RFID RELAY CONTROLLER ===")  
    print("Running in BOTH modes (RFID + Firebase)")  
  
    # Create and start controller  
    controller = UnifiedRelayController()  
    controller.start()  
  
if __name__ == "__main__":  
    main()
```

