

UNIVERSITAS GUNADARMA
FAKULTAS ILMU KOMPUTER & TEKNOLOGI INFORMASI



**RANCANG BANGUN SISTEM KEAMANAN PINTU DI
LABORATORIUM ELEKTRONIKA DAN KOMPUTER UNIVERSITAS
GUNADARMA MENGGUNAKAN SENSOR SIDIK JARI BERBASIS IOT**

Disusun Oleh:

Nama : Justin Elvin Megawe
NPM : 20121628
Program Studi : Sistem Komputer
Pembimbing : Dr. Ridha Iskandar, SSi., MMSi.

**Diajukan Guna Melengkapi Sebagian Syarat
Dalam Mencapai Gelar Sarjana Strata Satu (S1)**

JAKARTA

2025

PERNYATAAN ORISINALITAS DAN PUBLIKASI

Saya yang bertanda tangan di bawah ini,

Nama : Justin Elvin Megawe
NPM : 20121628
Judul Skripsi : Rancang Bangun Sistem Keamanan Pintu Di Laboratorium
Elektronika Dan Komputer Universitas Gunadarma
Menggunakan Sensor Sidik Jari Berbasis IoT
Tanggal Sidang : 16 Agustus 2025
Tanggal Lulus : 16 Agustus 2025

Menyatakan bahwa tulisan ini adalah merupakan hasil karya saya sendiri dan dapat dipublikasikan sepenuhnya oleh Universitas Gunadarma. Segala kutipan dalam bentuk apa pun telah mengikuti kaidah, etika yang berlaku. Mengenai isi dan tulisan adalah merupakan tanggung jawab Penulis, bukan Universitas Gunadarma.

Demikian pernyataan ini dibuat dengan sebenarnya dan dengan penuh kesadaran.

Jakarta, 23 Juni 2025

(Justin Elvin Megawe)

LEMBAR PENGESAHAN

Judul Skripsi : RANCANG BANGUN SISTEM KEAMANAN PINTU
DI LABORATORIUM ELEKTRONIKA DAN
KOMPUTER UNIVERSITAS GUNADARMA
MENGUNAKAN SENSOR SIDIK JARI BERBASIS
IOT

Nama : Justin Elvin Megawe

NPM : 20121628

Tanggal Lulus : 16 Agustus 2025

PANITIA UJIAN

NO	NAMA	KEDUDUKAN
1.	Dr. Ravi Ahmad Salim	Ketua
2.	Prof. Dr. Wahyudi Priyono	Sekretaris
3.	Dr. Ridha Iskandar, S.Si., MMSi.	Anggota
4.	Prof. Dr. Ir, Busono Soerowirdjo	Anggota
5.	Dr. Drs. Jonifan, MM.	Anggota

Menyetujui,

Pembimbing

Bagian Sidang Ujian

(Dr. Ridha Iskandar, S.Si., MMSi.) (Dr. Edi Sukirman, S.Si., MM., M.I.Kom.)

ABSTRAK

Justin Elvin Megawe, 20121628

RANCANG BANGUN SISTEM KEAMANAN PINTU DI LABORATORIUM ELEKTRONIKA DAN KOMPUTER UNIVERSITAS GUNADARMA MENGGUNAKAN SENSOR SIDIK JARI BERBASIS IOT

Tugas Akhir. Sistem Komputer. Fakultas Ilmu Komputer dan Teknologi Informasi. Universitas Gunadarma. 2025

Kata kunci : ESP32, *Firebase*, *Internet of Things*, Keamanan Pintu, Sensor Sidik Jari

(xiii + 70 + Lampiran)

Keamanan fisik Laboratorium Elektronika dan Komputer menjadi hal yang krusial mengingat nilai aset peralatan yang tinggi. Sistem konvensional seperti kunci fisik dinilai rentan terhadap pembobolan dan tidak menyediakan pemantauan *real-time*. Penelitian ini bertujuan untuk merancang, membangun, dan menguji sistem keamanan pintu berbasis IoT yang memanfaatkan sensor sidik jari R503 sebagai autentikasi utama. Sistem dilengkapi dengan sensor sentuh TTP223B untuk akses dari dalam ruangan dan sensor magnetik MC-38 untuk memantau status pintu. Mikrokontroler ESP32 berfungsi sebagai pengendali pusat yang terhubung ke Firebase Realtime Database untuk menyinkronkan data pengguna, log akses, dan perintah kontrol melalui sebuah *dashboard* web. Metode penelitian mencakup studi literatur, perancangan perangkat keras dan lunak, implementasi sistem, serta pengujian fungsional dan kinerja. Hasil pengujian menunjukkan bahwa sistem berhasil beroperasi dengan akurasi pembacaan sidik jari sebesar 95% pada kondisi ideal, waktu respons sistem dari verifikasi hingga pintu terbuka di bawah 2 detik, dan mampu mendukung kapasitas maksimal 200 pengguna. *Dashboard* web terbukti efektif untuk memantau status pintu, mendaftarkan pengguna, dan mencatat log akses secara *real-time*. Disimpulkan bahwa sistem yang dibangun efektif dalam meningkatkan keamanan dan efisiensi akses ke laboratorium, serta layak menjadi acuan untuk pengembangan sistem keamanan IoT yang lebih kompleks.

Daftar Pustaka (2020 – 2025)

ABSTRACT

Justin Elvin Megawe, 20121628

DESIGN AND CONSTRUCTION OF A DOOR SECURITY SYSTEM IN THE ELECTRONICS AND COMPUTER LABORATORY OF GUNADARMA UNIVERSITY USING AN IOT-BASED FINGERPRINT SENSOR

A Scientific Research. Computer System. Faculty of Computer Science and Information Technology, Gunadarma University, 2025

Keywords : Door Security, ESP32, Fingerprint Sensor, Internet of Things, Firebase

(xiii + 70 + Appendices)

The physical security of the Electronics and Computer Laboratory is crucial due to the high value of its assets. Conventional systems, such as physical keys, are considered vulnerable to break-ins and lack real-time monitoring capabilities. This research aims to design, build, and test an IoT-based door security system utilizing the R503 fingerprint sensor as the primary authentication method. The system is equipped with a TTP223B touch sensor for internal access and an MC-38 magnetic sensor to monitor the door status. The ESP32 microcontroller acts as the central controller, connected to the Firebase Realtime Database to synchronize user data, access logs, and control commands via a web dashboard. The research method includes literature study, hardware and software design, system implementation, and functional and performance testing. Test results indicate that the system operated successfully with a fingerprint recognition accuracy of 95% under ideal conditions, a system response time from verification to door unlock of less than 2 seconds, and a capacity to support up to 200 users. The web dashboard proved effective for real-time door status monitoring, user registration, and access logging. In conclusion, the implemented system is effective in enhancing laboratory security and access efficiency, and it is viable as a reference for developing more complex IoT-based security systems.

Bibliography (2020 – 2025)

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Tuhan Yang Maha Kuasa yang telah memberikan rahmat, anugrah dan karunianya, sehingga penulis dapat menyelesaikan Tugas Akhir ini. Tugas Akhir ini disusun guna melengkapi sebagian syarat dalam mencapai gelar Setara Sarjana Strata Satu (S1) pada Program Studi Sistem Komputer, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Gunadarma. Adapun judul Tugas Akhir ini adalah RANCANG BANGUN SISTEM KEAMANAN PINTU DI LABORATORIUM ELEKTRONIKA DAN KOMPUTER UNIVERSITAS GUNADARMA MENGGUNAKAN SENSOR SIDIK JARI BERBASIS IOT.

Walaupun banyak kesulitan yang Penulis harus hadapi ketika menyusun Tugas Akhir ini, namun berkat bantuan dan dorongan dari berbagai pihak akhirnya Tugas Akhir ini dapat diselesaikan dengan baik. Untuk itu Penulis mengucapkan terima kasih kepada:

1. Prof. Dr. E.S. Margianti, SE., MM., selaku Rektor Universitas Gunadarma.
2. Prof. Dr. rer. nat. A. Benny Mutiara Q.N., selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Gunadarma.
3. Dr. Nur Sultan Salahuddin., SKom., MT., selaku Ketua Program Studi Sistem Komputer Universitas Gunadarma.
4. Dr. Edi Sukirman, S.Si., MM., M.I.Kom., selaku Kepala Bagian Sidang Ujian Universitas Gunadarma
5. Dr. Ridha Iskandar, SSi., MMSi., selaku Dosen Pembimbing Tugas Akhir Program Studi Sistem Komputer Universitas Gunadarma.
6. Semua staf Dosen Universitas Gunadarma yang telah memberikan bekal ilmu kepada penulis.
7. Bapak James Hendrik, Ibu Elysabeth Christiana, dan Jethro Ernesto Megawe selaku orang tua dan adik penulis yang banyak memberikan dukungan moral maupun materi serta dukungan doa dan semangat kepada penulis sehingga penulisan ini dapat diselesaikan dengan baik.

8. Keluarga besar pastori yang telah memberikan semangat dan motivasi, terkhususnya untuk dukungan dalam doa, sehingga penulisan ini dapat diselesaikan.
9. Teman – teman asisten Laboratorium Elektronika dan Komputer yang telah memberikan saran dan dukungan pengetahuan untuk menyelesaikan penulisan ini.
10. Serta semua pihak yang tidak mungkin disebutkan satu persatu namanya yang telah membantu memberikan dukungan dan doa-nya kepada penyusun.

Dengan rendah hati penulis menyadari penulisan Tugas Akhir ini masih sangat jauh dari sempurna karena itu segala kritik dan saran yang menuntun kepada kebaikan dan penyempurnaan Penulisan Tugas Akhir ini sangat diharapkan dan diterima penulis dengan tangan terbuka.

Semoga Tuhan Yang Maha Esa memberikan imbalan atas semua kebaikan yang telah diberikan kepada penyusun. Akhir kata semoga penulisan ini bermanfaat.

Jakarta, Juni 2025

(Justin Elvin Megawe)

DAFTAR ISI

	Halaman
HALAMAN JUDUL.....	i
PERNYATAAN ORISINALITAS DAN PUBLIKASI	ii
LEMBAR PENGESAHAN	iii
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR.....	vi
DAFTAR ISI	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xii
DAFTAR LAMPIRAN	xiii
1. PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah.....	2
1.4. Tujuan Penelitian.....	3
1.5. Metode Penelitian.....	3
1.5.1. Studi Pustaka	3
1.5.2. Studi Laboratorium	3
1.6. Sistematika Tulisan Ilmiah	4
2. TINJAUAN PUSTAKA.....	6
2.1. Kajian Penelitian	6
2.2. Laboratorium Elektronika Dan Komputer	7
2.3. Internet of Things (IoT)	8
2.4. Dashboard Web	9
2.5. Sensor Sidik Jari R503	9
2.6. Sensor Sentuh TTP223B	11
2.7. Sensor Magnetic MC-38	12
2.8. Mikrokontroler ESP32	14

2.9. Solenoid Doorlock	15
2.10. Modul Relay 2 Channel	17
2.11. Step Down Module XL4016	18
2.12. Buzzer	18
2.13. Kabel Jumper	19
2.14. Power Adaptor.....	20
2.15. Firebase	21
2.16. Arduino IDE.....	22
2.17. Flowchart.....	24
2.17.1. Jenis – Jenis Flowchart	24
2.17.2. Simbol Penghubung.....	25
2.17.3. Simbol Proses	26
2.17.4. Simbol Input dan Output.....	27
3. PERANCANGAN DAN ANALISIS.....	29
3.1. Analisis Rangkaian Secara Blok Diagram	29
3.1.1. Blok Input	29
3.1.2. Blok Proses	30
3.1.3. Blok Output	30
3.2. Analisa Rangkaian	30
3.3. Analisa Alat Secara Diagram Alur	32
3.4. Perancangan Alat.....	35
3.5. Perancangan Website.....	37
3.6. Analisa Program.....	38
3.6.1. Analisa Program Alat.....	38
3.6.2. Analisa Program Website.....	49
4. IMPLEMENTASI DAN PENGUJIAN.....	64
4.1. Cara Pengoperasian Alat	64
4.2. Pengujian Sensor.....	66
4.2.1. Pengujian Sensor Sidik Jari R503.....	66
4.2.2. Pengujian Sensor Sentuh TTP223B.....	68
4.2.3. Pengujian Sensor Magnet MC-38.....	69
5. PENUTUP.....	71

5.1. Kesimpulan	71
5.2. Saran.....	71
DAFTAR PUSTAKA	73
LAMPIRAN.....	L-1

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Laboratorium Elektronika Dan Komputer	7
Gambar 2.2 Internet of Things	8
Gambar 2.3 Sensor Sidik Jari R503	10
Gambar 2.4 Sensor Sentuh TTP223B	11
Gambar 2.5 Sensor Magnetic MC-38	13
Gambar 2.6 Mikrokontroler ESP32	14
Gambar 2.7 Pinout Mikrokontroler ESP32	15
Gambar 2.8 Solenoid Doorlock.....	16
Gambar 2.9 Modul Relay 2 Channel.....	17
Gambar 2.10 Modul Step Down XL4016	18
Gambar 2.11 Buzzer.....	19
Gambar 2.12 Kabel Jumper.....	20
Gambar 2.13 Power Adaptor.....	21
Gambar 2.14 Firebase	22
Gambar 2.15 Arduino IDE	22
Gambar 2.16 Tampilan Arduino IDE	23
Gambar 3.1 Blok Diagram	29
Gambar 3.2 Rangkaian Alat	31
Gambar 3.3 Diagram Alur Alat	32
Gambar 3.4 Diagram Alur Website	34
Gambar 3.5 Perancangan Design Alat Tampak Dalam Dan Luar.....	36
Gambar 3.6 Perancangan Design Website Halaman Admin Dan <i>User</i>	37
Gambar 4.1 Tampilan Dashboard Website Admin	64
Gambar 4.2 Card Membuka Pintu Manual Dari Web	65
Gambar 4.3 Card Mendaftarkan Sidik Jari	65
Gambar 4.4 Card Menghapus Sidik Jari	66

DAFTAR TABEL

	Halaman
Tabel 2.1 Spesifikasi Sensor Sidik Jari R503.....	10
Tabel 2.2 Spesifikasi Sensor Sentuh TTP223B.....	12
Tabel 2.3 Spesifikasi Sensor Magnetic MC-38.....	13
Tabel 2.4 Spesifikasi Mikrokontroler ESP32.....	15
Tabel 2.5 Spesifikasi Solenid Doorlock.....	16
Tabel 2.6 Spesifikasi Modul Relay	17
Tabel 2.7 Spesifikasi Buzzer.....	19
Tabel 2.8 Simbol Penghubung	26
Tabel 2.9 Simbol Proses.....	26
Tabel 2.10 Simbol <i>Input</i> dan <i>Output</i>	27
Tabel 3.1 Analisis Program Alat.....	38
Tabel 3.2 Analisis Program Website (HTML).....	49
Tabel 3.3 Analisis Program Website (Javascript).....	55
Tabel 3.4 Analisis Program Website Halaman Log in (HTML).....	60
Tabel 3.5 Analisis Program Website Halaman Log in (Javascript).....	61
Tabel 4.1 Pengujian Sensor Sidik Jari R503 Terhadap Kondisi Jari.....	67
Tabel 4.2 Pengujian Sensor Sidik Jari Berdasarkan Respons Sistem	67
Tabel 4.3 Pengujian Sensor Sentuh TTP223B Terhadap Kondisi Jari	68
Tabel 4.4 Pengujian Sensor Magnetik Berdasarkan Pembacaan Pintu	69

DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Skematik Rangkaian	1
Lampiran 2. Tampilan Maket	2

1. PENDAHULUAN

1.1. Latar Belakang

Keamanan merupakan salah satu aspek yang sangat penting dalam kehidupan sehari-hari, terutama di tempat-tempat yang menyimpan barang berharga seperti Laboratorium Elektronika dan Komputer. Dengan meningkatnya angka kejahatan, termasuk pencurian, sistem keamanan yang ada saat ini sering kali tidak memadai. Kunci konvensional yang banyak digunakan rentan terhadap pembobolan dan penggandaan, sehingga tidak memberikan perlindungan yang optimal. Oleh karena itu, diperlukan sistem keamanan yang lebih canggih dan efektif untuk melindungi aset-aset berharga di laboratorium tersebut.

Sistem keamanan berbasis Internet of Things (IoT) menawarkan solusi yang lebih modern dan responsif. Dengan memanfaatkan teknologi sensor dan konektivitas internet, sistem ini dapat memberikan notifikasi real-time kepada pengguna saat terdeteksi adanya usaha pembobolan. Penelitian sebelumnya menunjukkan bahwa sistem keamanan yang menggunakan autentikasi biometrik, seperti sidik jari, dapat meningkatkan tingkat keamanan secara signifikan dibandingkan dengan sistem konvensional [1]. Selain itu, penggunaan sensor tambahan seperti sensor sentuh dan magnetic sensor dapat memberikan lapisan keamanan ekstra yang diperlukan untuk mencegah akses tidak sah [2].

Dalam konteks Laboratorium Elektronika dan Komputer, perlindungan terhadap peralatan dan data yang ada di dalamnya sangatlah krusial. Laboratorium ini sering kali menyimpan perangkat keras yang mahal dan data penelitian yang berharga. Dengan adanya sistem keamanan yang efektif, diharapkan dapat mengurangi risiko pencurian dan kerusakan yang dapat mengganggu proses belajar mengajar serta penelitian. Penelitian ini bertujuan untuk merancang dan mengembangkan sistem keamanan pintu berbasis ESP 32 yang menggunakan sensor fingerprint R503, sensor sentuh TTP223B, dan magnetic sensor MC-38 [3].

Melalui penelitian ini, diharapkan dapat memberikan kontribusi yang signifikan dalam pengembangan sistem keamanan berbasis IoT di Laboratorium Elektronika dan Komputer. Dengan sistem yang dirancang, pengguna dapat merasa

lebih aman dan nyaman saat menggunakan laboratorium, serta dapat memantau kondisi pintu secara real-time. Penelitian ini juga diharapkan dapat menjadi referensi bagi penelitian selanjutnya dalam bidang keamanan berbasis teknologi.

1.2. Rumusan Masalah

Berdasarkan latar belakang masalah, terdapat beberapa isu kritis yang perlu dijawab untuk memastikan keberhasilan sistem keamanan pintu berbasis IoT di laboratorium elektronika dan komputer. Rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana merancang sistem keamanan pintu berbasis IoT dengan ESP32, sensor R503, dan Firebase untuk autentikasi yang aman di laboratorium?
2. Bagaimana membuat antarmuka web untuk pengguna dan admin agar mudah digunakan?
3. Bagaimana mengevaluasi akurasi dan keandalan sistem di lingkungan laboratorium?

1.3. Batasan Masalah

Untuk menjaga fokus penelitian dan memastikan ketercapaian tujuan, penelitian ini dibatasi pada aspek-aspek berikut:

1. Menggunakan sensor sidik jari R503, Mikrokontroler ESP32, sensor sentuh, sensor magnetik, dan solenoid untuk sistem pintu.
2. Menggunakan Firebase Realtime Database untuk penyimpanan data dan antarmuka web (HTML, JavaScript).
3. Fitur hanya mencakup pendaftaran/penghapusan sidik jari, kontrol pintu, dan log akses.
4. Komunikasi melalui WiFi menggunakan ESP32.
5. Keamanan fokus pada autentikasi berbasis peran dan enkripsi HTTPS.
6. Pengujian dilakukan di lingkungan laboratorium dengan simulasi debu dan kelembapan.
7. Sistem mendukung hingga 200 pengguna sesuai kapasitas R503.

1.4. Tujuan Penelitian

Tujuan dari penelitian dan pembuatan sistem keamanan pintu ini adalah sebagai berikut:

1. Merancang sistem keamanan pintu berbasis IoT dengan ESP32, sensor R503, dan Firebase.
2. Membuat antarmuka web untuk pengguna dan admin.
3. Mengevaluasi akurasi dan keandalan sistem di lingkungan laboratorium.

1.5. Metode Penelitian

Berikut merupakan metode penelitian yang penulis gunakan untuk menyusun penulisan ilmiah ini:

1.5.1. Studi Pustaka

Pada fase ini, penulis mengumpulkan data terkait cara kerja setiap komponen yang digunakan. Informasi didapatkan dari berbagai sumber seperti situs web dan artikel untuk memperkuat penulisan.

1.5.2. Studi Laboratorium

Dalam fase ini, penulis mengikuti proses belajar berdasarkan langkah-langkah di laboratorium, yaitu:

1. Pengumpulan Data
Penulis mengumpulkan data untuk memperkaya dan memperluas referensi yang terkait dengan judul karya ilmiah ini.
2. Analisis Kebutuhan
Penulis menganalisis serta mengumpulkan komponen - komponen yang akan digunakan dalam pembuatan alat yang akan dibuat.
3. Perancangan Alat
Penulis mulai merancang dan menyusun alat dengan menggabungkan komponen dan bahan yang sudah ada.
4. Perancangan Program

Penulis merancang struktur pemrograman dengan menggunakan aplikasi Arduino IDE dan setelah itu program dikompilasi ke dalam ESP32 dan Aplikasi Visual Studio Code untuk pembuatan halaman web.

5. Pengujian Program dan Alat

Pada fase ini, penulis melakukan pengujian terhadap program serta alat untuk menganalisis letak kesalahan program dan kerusakan komponen.

1.6. Sistematika Tulisan Ilmiah

Memberikan gambaran umum dari bagian ke bagian dari isi Tulisan Ilmiah yang dituangkan dalam bentuk narasi, bukan dengan penomoran. Sistematika penulisan yang dilakukan oleh penulis dalam skripsi ini dijelaskan dalam 5 bab yang membahas masalah secara singkat tanpa mengabaikan keterhubungan antara bab yang satu dengan bab selanjutnya, Adapun sistematika penulisan terdiri dari:

Dalam Bab **Pendahuluan**, Menguraikan latar belakang masalah, mengidentifikasi tantangan keamanan di laboratorium elektronika dan komputer, serta pentingnya autentikasi sidik jari berbasis sensor R503 dan IoT. Bab ini juga mencakup rumusan masalah, batasan masalah, tujuan penelitian, metode penelitian, dan sistematika penulisan untuk memberikan kerangka penelitian secara keseluruhan.

Dalam Bab **Tinjauan Pustaka**, Memberikan tinjauan pustaka yang mendukung penelitian, mencakup kajian penelitian terkait sistem keamanan pintu berbasis IoT dan biometrik, serta penjelasan mendalam tentang komponen utama seperti sensor sidik jari R503, mikrokontroler ESP32, dan Firebase Realtime Database. Bab ini juga membahas pustaka pendukung seperti autentikasi berbasis peran dan keamanan data biometrik dari sumber terpercaya.

Dalam Bab **Perancangan dan Analisis**, Menjelaskan desain sistem secara rinci, termasuk perancangan perangkat keras dan perangkat lunak. Bab ini mencakup analisis rangkaian, diagram alur (flowchart) untuk proses autentikasi dan kontrol pintu, serta spesifikasi teknis komponen seperti R503 dan ESP32, untuk memastikan kejelasan implementasi teknis.

Dalam Bab **Implementasi dan Pengujian**, Menguraikan proses integrasi sistem, mulai dari konfigurasi sensor R503 dengan ESP32 hingga pengembangan antarmuka web dan pengujian fungsional serta kinerja. Bab ini juga menganalisis data pengujian, seperti akurasi autentikasi dan kecepatan respons, untuk mengevaluasi efektivitas sistem di lingkungan laboratorium.

Dalam Bab **Penutup**, Menyimpulkan temuan penelitian berdasarkan hasil pengujian dan analisis, menyoroti kontribusi sistem keamanan pintu berbasis IoT dalam konteks laboratorium. Bab ini juga memberikan saran untuk perbaikan dan pengembangan sistem di masa depan, seperti peningkatan keamanan atau skalabilitas.

2. TINJAUAN PUSTAKA

2.1. Kajian Penelitian

Dalam kajian penelitian ini, beberapa studi sebelumnya telah menunjukkan keberhasilan dalam mengembangkan sistem keamanan berbasis IoT. Penelitian oleh Zulhanip et al. (2024) mengembangkan sistem keamanan pintu ganda yang menggunakan autentikasi sidik jari dan kata sandi, yang terbukti efektif dalam memberikan akses hanya kepada pengguna yang sah. Sistem ini tidak hanya menggantikan metode konvensional tetapi juga meningkatkan kemampuan sistem dalam memberikan informasi kondisi rumah secara real-time kepada pengguna. Penelitian ini menunjukkan bahwa teknologi IoT dapat memberikan solusi yang lebih baik dalam hal keamanan rumah, terutama dalam menghadapi ancaman pencurian yang semakin meningkat [1].

Selain itu, penelitian oleh Fauziman dan Mukhaiyar (2023) menunjukkan bahwa sistem keamanan pintu berbasis fingerprint dapat berfungsi dengan baik dan memberikan notifikasi kepada pengguna saat terdeteksi adanya usaha pembobolan. Penelitian ini menekankan pentingnya integrasi berbagai teknologi dalam menciptakan sistem keamanan yang komprehensif. Dengan memanfaatkan sensor dan perangkat lunak yang tepat, sistem keamanan dapat memberikan perlindungan yang lebih baik dan responsif terhadap ancaman. Penelitian ini menjadi dasar untuk pengembangan sistem keamanan yang lebih canggih dan efektif di Laboratorium Elektronika dan Komputer [2].

Lebih lanjut, penelitian oleh Tatar et al. (2022) juga menyoroti pentingnya penggunaan teknologi IoT dalam meningkatkan keamanan di lingkungan yang rentan terhadap pencurian. Dalam studi tersebut, sistem yang dirancang menggunakan kombinasi sensor dan perangkat lunak untuk memberikan notifikasi kepada pengguna saat terdeteksi adanya aktivitas mencurigakan. Hasil penelitian ini menunjukkan bahwa penerapan teknologi IoT tidak hanya meningkatkan keamanan fisik tetapi juga memberikan rasa aman bagi pengguna. Dengan demikian, kajian-kajian ini memberikan landasan yang kuat untuk penelitian ini dalam mengembangkan sistem keamanan pintu berbasis IoT yang lebih efektif [3].

2.2. Laboratorium Elektronika Dan Komputer

Laboratorium Elektronika dan Komputer di Universitas Gunadarma merupakan fasilitas yang dirancang untuk mendukung pembelajaran dan penelitian di bidang elektronika dan teknologi komputer. Laboratorium ini menyelenggarakan praktikum yang mencakup ilmu elektronika dan komputer tingkat dasar hingga menengah, dengan fokus pada beberapa aspek penting.



Gambar 2.1 Laboratorium Elektronika Dan Komputer

Gambar 2.1 menunjukkan fasilitas Laboratorium Elektronika dan Komputer di Universitas Gunadarma, yang dilengkapi dengan peralatan elektronik untuk mendukung pembelajaran dan penelitian. Laboratorium ini menjadi lokasi utama pengujian sistem keamanan pintu berbasis IoT yang dirancang dalam penelitian ini.

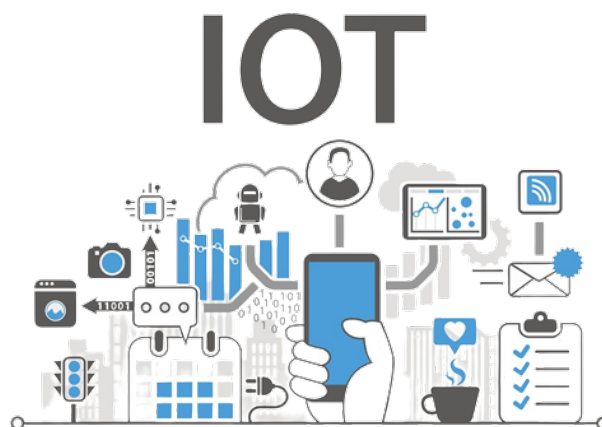
Pada tingkat dasar, laboratorium ini memberikan pengetahuan fundamental mengenai rangkaian elektronika dan digital. Praktikan diperkenalkan pada konsep dasar rangkaian elektronika, di mana mereka belajar untuk mendesain dan menganalisis rangkaian elektronika serta digital. Selain itu, praktikan juga diperkenalkan pada penggunaan software simulator seperti Xilinx ISE untuk memahami sistem digital dan dasar-dasar FPGA (Field Programmable Gate Array). Dengan demikian, mereka diharapkan dapat menguasai dasar-dasar rangkaian elektronika dan digital yang menjadi fondasi penting dalam pengembangan teknologi.

Di tingkat menengah, fokus pembelajaran beralih pada penguasaan bahasa pemrograman yang digunakan dalam mikrokontroler, seperti Arduino Uno dan

mikrokontroler STM32, dan Nuvoton. Laboratorium ini juga melatih keterampilan praktikan dalam penggunaan perangkat keras mikrokontroler dan interfacing, yang didukung oleh aplikasi interface seperti Processing IDE. Melalui praktikum ini, mahasiswa diharapkan dapat memahami konsep dasar mikrokontroler serta mampu mengembangkan aplikasi yang mengintegrasikan perangkat keras dan perangkat lunak dengan lebih mudah dan interaktif.

2.3. *Internet of Things (IoT)*

Internet of Things (IoT) adalah konsep di mana objek fisik dilengkapi dengan sensor, perangkat lunak, dan teknologi lainnya untuk terhubung dan bertukar data dengan perangkat lain melalui internet. IoT memungkinkan pengumpulan dan analisis data secara *real-time*, yang dapat meningkatkan efisiensi dan efektivitas dalam berbagai bidang, termasuk keamanan. Menurut Gupta (2021), IoT dapat mengubah cara kita berinteraksi dengan lingkungan sekitar dan memberikan solusi inovatif untuk berbagai masalah. Dengan adanya IoT, perangkat yang sebelumnya tidak terhubung dapat saling berkomunikasi, sehingga menciptakan ekosistem yang lebih terintegrasi [3].



Gambar 2.2 *Internet of Things*

Gambar 2.2 mengilustrasikan konsep Internet of Things (IoT) yang menjadi dasar pengembangan sistem keamanan pintu. IoT memungkinkan integrasi antara sensor, mikrokontroler, dan database cloud untuk menciptakan solusi keamanan yang cerdas dan terhubung.

Dalam konteks sistem keamanan, IoT menawarkan berbagai keuntungan, seperti kemampuan untuk memantau dan mengendalikan akses ke suatu area secara remote. Hal ini sangat penting dalam meningkatkan keamanan, terutama di tempat-tempat yang menyimpan barang berharga. Penelitian oleh Zulhanip et al. (2024) menunjukkan bahwa penerapan IoT dalam sistem keamanan dapat meningkatkan responsivitas dan efisiensi dalam pengelolaan data, serta memberikan notifikasi *real-time* kepada pengguna saat terdeteksi adanya ancaman. Dengan demikian, IoT tidak hanya meningkatkan keamanan tetapi juga memberikan kenyamanan bagi pengguna dalam mengelola sistem keamanan mereka [4].

2.4. Dashboard Web

Dashboard web adalah antarmuka pengguna yang menyajikan data dan informasi secara visual dalam bentuk grafik, tabel, dan indikator lainnya. Dalam konteks sistem keamanan, dashboard web dapat digunakan untuk memantau status sistem secara *real-time*. Menurut Kurniawan et al. (2020), penggunaan dashboard web dalam sistem monitoring dapat meningkatkan pemahaman pengguna terhadap data yang disajikan dan memudahkan pengambilan keputusan. Dengan adanya dashboard, pengguna dapat dengan cepat melihat status keamanan pintu, menerima notifikasi, dan mengambil tindakan yang diperlukan [4].

Dashboard web juga memungkinkan integrasi berbagai sumber data, sehingga pengguna dapat melihat informasi yang lebih komprehensif. Hal ini sangat penting dalam sistem keamanan, di mana informasi yang akurat dan tepat waktu dapat membuat perbedaan dalam mencegah kejahatan. Penelitian oleh Setiawan et al. (2021) menunjukkan bahwa penggunaan dashboard web dalam sistem keamanan dapat meningkatkan efektivitas pengelolaan data dan respons terhadap ancaman. Dengan demikian, dashboard web menjadi komponen penting dalam pengembangan sistem keamanan berbasis IoT [11].

2.5. Sensor Sidik Jari R503

Sensor sidik jari R503 adalah perangkat biometrik yang digunakan untuk membaca sidik jari. Sensor ini memiliki kemampuan untuk menyimpan dan memverifikasi sidik jari pengguna. Menurut Ikhsan et al. (2022), penggunaan

sensor sidik jari dalam sistem keamanan dapat meningkatkan tingkat keamanan karena sidik jari setiap individu bersifat unik dan sulit untuk dipalsukan. Sensor ini juga menawarkan kecepatan dan akurasi yang tinggi dalam proses autentikasi, sehingga pengguna dapat dengan cepat mendapatkan akses ke sistem yang dilindungi [5].



Gambar 2.3 Sensor Sidik Jari R503

Sensor sidik jari R503, seperti yang ditunjukkan pada Gambar 2.3, merupakan komponen utama untuk autentikasi pengguna. Sensor ini memiliki tingkat akurasi tinggi dan mampu menyimpan hingga 200 template sidik jari, seperti terlihat pada Tabel 2.1.

Dalam sistem keamanan, sensor sidik jari R503 berfungsi sebagai metode utama untuk autentikasi pengguna. Hanya pengguna yang terdaftar yang dapat membuka pintu, sehingga mengurangi risiko akses tidak sah. Penelitian oleh Nur Ikhsan et al. (2022) menunjukkan bahwa sistem yang menggunakan sensor sidik jari dapat berfungsi dengan baik dalam berbagai kondisi, dan memberikan tingkat keamanan yang lebih tinggi dibandingkan dengan metode konvensional seperti kunci fisik atau kata sandi [5]. Dengan demikian, sensor sidik jari R503 menjadi komponen kunci dalam pengembangan sistem keamanan berbasis IoT.

Berikut ini merupakan spesifikasi dari sensor sidik jari R503 yang digunakan dalam alat ini.

Tabel 2.1 Spesifikasi Sensor Sidik Jari R503 [8]

Parameter	Spesifikasi
Tegangan Operasi	DC 3.3V
Arus Kerja	20 mA
Arus Siaga	Tegangan sentuh tipikal: 3.3V

	Arus rata-rata: 2A
Waktu Pengambilan Citra	< 0.2s
Array Sensor	192x192 piksel
Kapasitas Penyimpanan	200
Tingkat Penerimaan Palsu (FAR)	< 0.001%
Tingkat Penolakan Palsu (FRR)	< 1%
Waktu Pembentukan Titik Fitur	< 500ms
Ukuran Template	768 bytes
Suhu Operasi	-20 C - 60 C
Area Deteksi	Diameter 15mm

Tabel 2.1 merinci spesifikasi teknis sensor R503, termasuk tegangan operasi, waktu respons, dan kapasitas penyimpanan. Data ini menjadi acuan dalam merancang sistem autentikasi yang andal.

2.6. Sensor Sentuh TTP223B

TTP223B adalah sensor sentuh capacitive yang dapat mendeteksi sentuhan manusia. Sensor ini sangat sensitif dan dapat digunakan untuk berbagai aplikasi, termasuk pengendalian perangkat. Penelitian oleh Kamila et al. (2024) menunjukkan bahwa sensor sentuh dapat digunakan untuk meningkatkan interaksi pengguna dengan sistem, memberikan kemudahan dalam pengoperasian. Dengan menggunakan sensor sentuh, pengguna dapat membuka pintu hanya dengan menyentuh permukaan sensor, yang membuatnya lebih praktis dibandingkan dengan metode konvensional [6].



Gambar 2.4 Sensor Sentuh TTP223B

Sensor sentuh TTP223B (Gambar 2.4) digunakan sebagai alternatif pembuka pintu dari dalam ruangan. Spesifikasinya, seperti yang tercantum pada

Tabel 2.2, menunjukkan sensitivitas sensor terhadap sentuhan bahkan dalam kondisi basah.

Penggunaan sensor sentuh dalam sistem keamanan pintu ini juga memberikan keuntungan dalam hal kecepatan dan kenyamanan. Dalam situasi darurat, pengguna dapat dengan cepat membuka pintu tanpa harus mencari kunci atau memasukkan kata sandi. Penelitian oleh Kamila et al. (2024) juga menunjukkan bahwa sensor sentuh dapat diintegrasikan dengan sistem lain untuk meningkatkan fungsionalitas dan responsivitas sistem keamanan secara keseluruhan. Dengan demikian, sensor sentuh TTP223B menjadi komponen penting dalam pengembangan sistem keamanan berbasis IoT [6].

Berikut ini merupakan spesifikasi dari sensor sentuh TTP223B yang digunakan dalam alat ini.

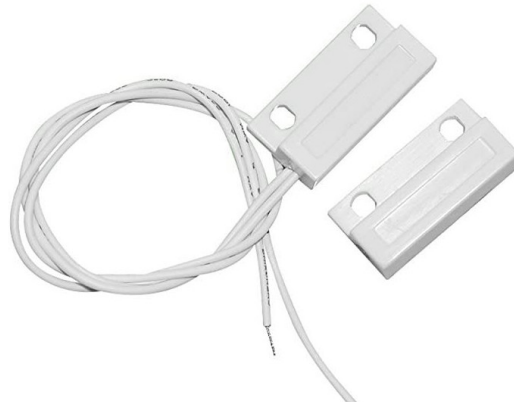
Tabel 2.2 Spesifikasi Sensor Sentuh TTP223B [6]

Parameter	Spesifikasi
Tegangan Operasi	2.0V - 5.5V
Arus Kerja	< 10 μ A
Sensitivitas	1-2 cm
Frekuensi Operasi	50 Hz
Ukuran	20mm x 20mm

Tabel 2.2 menjelaskan karakteristik sensor TTP223B, termasuk tegangan operasi dan jarak deteksi. Hal ini memastikan sensor dapat berfungsi optimal dalam berbagai kondisi.

2.7. Sensor Magnetic MC-38

MC-38 adalah sensor magnetik yang digunakan untuk mendeteksi posisi pintu. Sensor ini bekerja dengan cara mendeteksi perubahan dalam medan magnet ketika pintu dibuka atau ditutup. Menurut penelitian oleh Setiawan et al. (2021), penggunaan sensor magnetik dalam sistem keamanan dapat memberikan informasi yang akurat mengenai status pintu dan meningkatkan respons terhadap potensi ancaman. Sensor ini sangat efektif dalam mendeteksi pembukaan pintu secara paksa, sehingga dapat memberikan peringatan kepada pengguna jika terjadi upaya pembobolan.



Gambar 2.5 Sensor Magnetic MC-38

Sensor magnetik MC-38 (Gambar 2.5) berperan dalam memantau status pintu. Spesifikasinya pada Tabel 2.3 menunjukkan kemampuan sensor untuk mendeteksi perubahan medan magnet dengan jarak hingga 3 cm. Dalam sistem keamanan, sensor magnetic MC-38 berfungsi sebagai pengaman tambahan yang bekerja sebagai monitoring kondisi pintu. Ketika pintu dibuka tanpa izin, sensor ini akan mengirimkan sinyal ke sistem untuk mengaktifkan alarm atau notifikasi kepada pengguna. Penelitian oleh Setiawan et al. (2021) menunjukkan bahwa penggunaan sensor magnetik dapat meningkatkan efektivitas sistem keamanan secara keseluruhan, memberikan perlindungan yang lebih baik terhadap akses tidak sah [7]. Dengan demikian, sensor magnetic MC-38 menjadi komponen penting dalam pengembangan sistem keamanan berbasis IoT.

Berikut ini merupakan spesifikasi dari sensor magnetic MC-38 yang digunakan dalam alat ini.

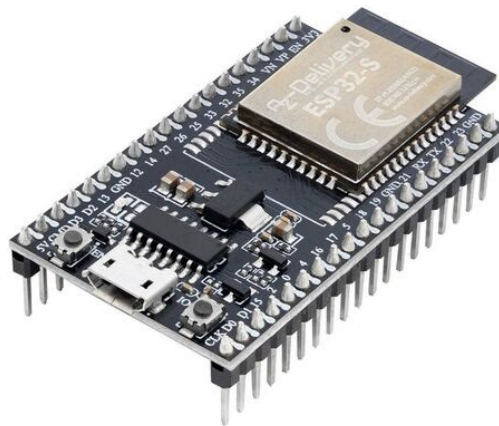
Tabel 2.3 Spesifikasi Sensor Magnetic MC-38 [7]

Parameter	Spesifikasi
Tegangan Operasi	12V DC
Arus Kerja	10 mA
Jarak Deteksi	2-3 cm
Tipe Sensor	Reed Switch

Tabel 2.3 memberikan detail teknis sensor MC-38, seperti tegangan operasi dan arus kerja, yang menjadi pertimbangan dalam integrasi dengan sistem utama.

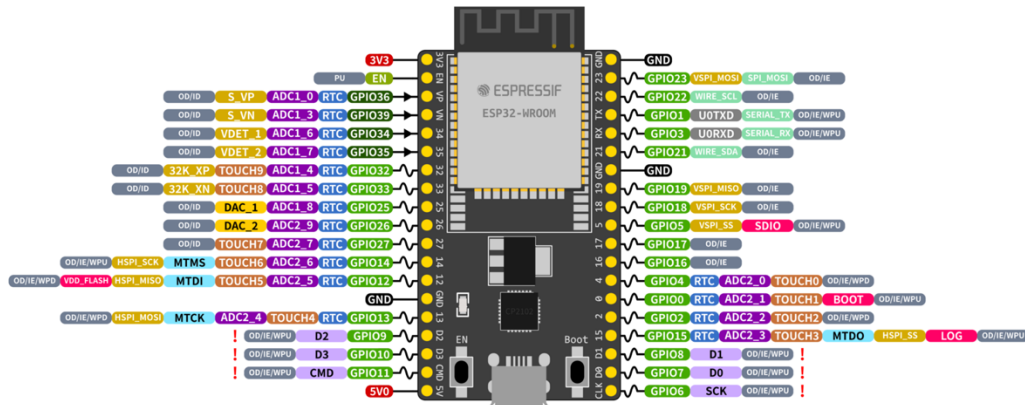
2.8. Mikrokontroler ESP32

ESP32 adalah mikrokontroler yang dilengkapi dengan modul Wi-Fi dan Bluetooth. Mikrokontroler ini sangat populer dalam pengembangan proyek IoT karena kemampuannya untuk terhubung ke internet dan mengontrol berbagai perangkat. Menurut Kurniawan et al. (2021), ESP32 menawarkan fleksibilitas dan kemampuan pemrograman yang tinggi, sehingga cocok untuk aplikasi yang memerlukan konektivitas dan kontrol jarak jauh. Dengan kemampuan ini, ESP32 dapat digunakan untuk mengendalikan berbagai sensor dan perangkat lain dalam sistem keamanan.



Gambar 2.6 Mikrokontroler ESP32

Mikrokontroler ESP32 (Gambar 2.6) berfungsi sebagai otak sistem. Kemampuannya yang tercantum pada Tabel 2.4, seperti dual-core dan konektivitas Wi-Fi, mendukung pengolahan data secara real-time. Dengan menggunakan ESP32, pengguna dapat mengontrol pintu dari jarak jauh melalui koneksi internet, memberikan kemudahan dan kenyamanan dalam mengakses laboratorium. Penelitian oleh Kurniawan et al. (2021) menunjukkan bahwa penggunaan ESP32 dalam sistem IoT dapat meningkatkan responsivitas dan efisiensi dalam pengelolaan data, serta memberikan notifikasi real-time kepada pengguna saat terdeteksi adanya ancaman [8]. Dengan demikian, ESP32 menjadi komponen kunci dalam pengembangan sistem keamanan berbasis IoT.



Gambar 2.7 Pinout Mikrokontroler ESP32

Berikut merupakan spesifikasi dari mikrokontroler yang digunakan dalam sistem keamanan pintu ini:

Tabel 2.4 Spesifikasi Mikrokontroler ESP32

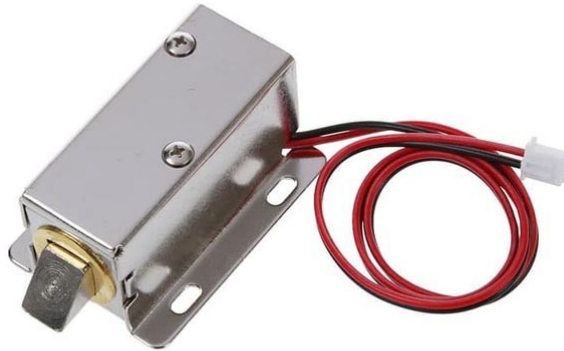
Fitur	Spesifikasi
Frekuensi CPU	Hingga 240 MHz
Core	2 (Dual-core)
Memori Flash	4 MB (default, dapat bervariasi tergantung modul)
SRAM	520 KB
Wi-Fi	802.11 b/g/n (2.4 GHz)
Bluetooth	Bluetooth v4.2 BR/EDR dan BLE
GPIO	Hingga 34 GPIO (tergantung penggunaan fungsi lain)
ADC	18-channel 12-bit SAR ADC
DAC	2-channel 8-bit DAC
PWM	Hingga 16 channel PWM
SPI / I2C / I2S / UART	3 × SPI, 2 × I2C, 2 × I2S, 3 × UART
Voltase Operasi	3.0V - 3.6V (3.3V typical)

Tabel 2.4 merangkum fitur ESP32, termasuk frekuensi CPU dan memori, yang menjadi dasar pemilihan komponen ini untuk proyek IoT.

2.9. Solenoid Doorlock

Solenoid *doorlock* adalah perangkat pengunci yang menggunakan elektromagnet untuk mengunci dan membuka pintu. Perangkat ini sering digunakan dalam sistem keamanan modern. Menurut penelitian oleh Tatar et al. (2022), penggunaan solenoid *doorlock* dalam sistem keamanan dapat meningkatkan

efisiensi dan keamanan akses ke suatu area. Solenoid *doorlock* bekerja dengan cara menggerakkan mekanisme penguncian pintu ketika diberikan sinyal dari mikrokontroler, sehingga memungkinkan akses yang lebih cepat dan aman.



Gambar 2.8 Solenoid *Doorlock*

Solenoid doorlock (Gambar 2.8) adalah aktuator yang mengendalikan buka-tutup pintu. Spesifikasinya pada Tabel 2.5 menunjukkan waktu respons kurang dari 1 detik, memastikan kinerja sistem yang cepat. Dalam sistem keamanan ini, solenoid doorlock berfungsi untuk mengunci dan membuka pintu secara otomatis berdasarkan sinyal dari relay yang dikendalikan oleh ESP32. Dengan menggunakan solenoid doorlock, pengguna dapat dengan mudah mengontrol akses ke laboratorium tanpa harus menggunakan kunci fisik. Penelitian oleh Tatar et al. (2022) menunjukkan bahwa sistem yang menggunakan solenoid doorlock dapat berfungsi dengan baik dan memberikan tingkat keamanan yang lebih tinggi dibandingkan dengan metode konvensional [9]. Dengan demikian, solenoid doorlock menjadi komponen penting dalam pengembangan sistem keamanan berbasis IoT.

Berikut ini merupakan spesifikasi dari Solenoid Doorlock yang digunakan dalam alat ini.

Tabel 2.5 Spesifikasi Solenid Doorlock [9]

Parameter	Spesifikasi
Tegangan Operasi	12V DC
Arus Kerja	300 mA
Tipe Penguncian	Elektromagnetik
Waktu Respon	< 1 detik

2.10. Modul Relay 2 Channel

Modul relay 2 channel adalah perangkat elektronik yang berfungsi sebagai saklar elektromagnetik untuk mengendalikan arus listrik pada perangkat lain dengan menggunakan sinyal input dari mikrokontroler. Modul ini memiliki dua channel relay yang memungkinkan kendali dua perangkat secara terpisah, sehingga sangat cocok digunakan dalam sistem otomasi dan keamanan, seperti pengendalian solenoid doorlock dan alarm.



Gambar 2.9 Modul Relay 2 Channel

Penggunaan modul relay 2 channel dalam sistem keamanan pintu berbasis IoT memungkinkan pengendalian pintu secara otomatis dan remote. Ketika mikrokontroler menerima sinyal autentikasi yang valid, modul relay akan mengaktifkan relay untuk membuka solenoid *doorlock* secara elektrik. Modul ini juga dapat digunakan untuk mengaktifkan buzzer atau alarm sebagai respon terhadap deteksi pembobolan. Penelitian oleh Kurniawan et al. (2021) menunjukkan bahwa integrasi modul relay dalam sistem IoT meningkatkan fleksibilitas dan keandalan kendali perangkat elektronik, sehingga mendukung keamanan dan praktikalitas sistem secara keseluruhan [4].

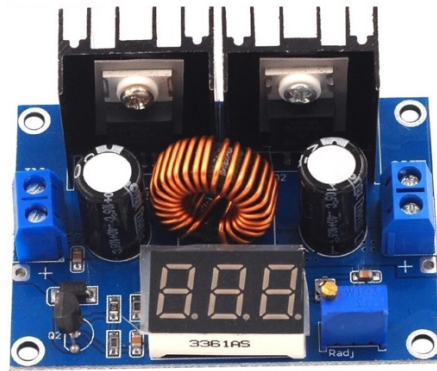
Berikut ini merupakan spesifikasi dari Modul Relay yang digunakan dalam alat ini.

Tabel 2.6 Spesifikasi Modul Relay [4]

Parameter	Spesifikasi
Tegangan Operasi	5V DC
Arus Kerja	10A (maksimal)
Tipe Relay	SPDT
Jumlah Channel	2

2.11. Step Down Module XL4016

Step Down Module XL4016 merupakan modul konversi tegangan DC ke DC tipe buck converter yang digunakan untuk menurunkan tegangan input menjadi tegangan output yang lebih rendah dan stabil. Modul ini mampu menangani tegangan input hingga 40V dan mengeluarkan tegangan output yang dapat diatur mulai dari 1.25V hingga 36V, dengan arus maksimum hingga 8A. Modul ini bekerja dengan efisiensi tinggi, mencapai 90–95%, sehingga sangat cocok untuk aplikasi yang membutuhkan daya besar namun tetap hemat energi, seperti sistem mikrokontroler dan perangkat IoT.



Gambar 2.10 Modul Step Down XL4016

Dalam proyek sistem keamanan pintu berbasis ESP32, modul XL4016 digunakan untuk menyesuaikan tegangan dari adaptor (contohnya 12V) menjadi 5V agar sesuai dengan kebutuhan mikrokontroler, sensor sidik jari, dan komponen lainnya. Modul ini juga dilengkapi dengan fitur proteksi arus lebih dan suhu berlebih, sehingga mampu menjaga kestabilan dan keamanan sistem secara keseluruhan. Dengan bentuknya yang ringkas dan kemampuan konversinya yang efisien, XL4016 menjadi pilihan yang ideal dalam mendukung suplai daya sistem secara andal.

2.12. Buzzer

Buzzer adalah perangkat elektronik yang menghasilkan suara. Buzzer dapat digunakan untuk memberikan sinyal atau peringatan kepada pengguna. Menurut penelitian oleh Ikhsan et al. (2022), penggunaan buzzer dalam sistem keamanan dapat meningkatkan kesadaran pengguna terhadap potensi ancaman

melalui peringatan suara yang dihasilkan. Buzzer berfungsi sebagai alarm yang memberikan notifikasi suara ketika terdeteksi adanya usaha pembobolan atau saat autentikasi sidik jari berhasil.



Gambar 2.11 Buzzer

Dalam sistem keamanan ini, buzzer diaktifkan oleh mikrokontroler ketika sensor mendeteksi aktivitas mencurigakan. Dengan adanya buzzer, pengguna dapat segera mengetahui jika ada potensi ancaman, sehingga dapat mengambil tindakan yang diperlukan. Penelitian oleh Ikhsan et al. (2022) menunjukkan bahwa penggunaan buzzer dalam sistem keamanan dapat meningkatkan responsivitas dan efektivitas dalam melindungi aset berharga [5]. Dengan demikian, buzzer menjadi komponen penting dalam pengembangan sistem keamanan berbasis IoT.

Berikut ini merupakan spesifikasi dari Buzzer yang digunakan dalam alat ini.

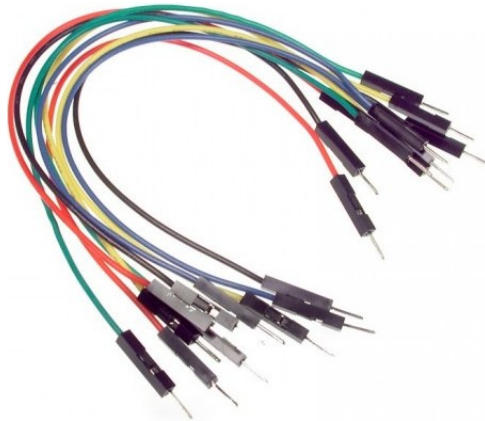
Tabel 2.7 Spesifikasi Buzzer [5]

Parameter	Spesifikasi
Tegangan Operasi	5V DC
Arus Kerja	20 mA
Frekuensi Suara	2 kHz - 4 kHz

2.13. Kabel Jumper

Kabel jumper adalah kabel yang digunakan untuk menghubungkan berbagai komponen dalam rangkaian elektronik. Kabel ini sangat penting dalam pengembangan prototipe sistem. Menurut penelitian oleh Kurniawan et al. (2020), penggunaan kabel jumper yang tepat dapat mempengaruhi kestabilan dan keandalan sistem yang dibangun. Kabel jumper memungkinkan penghubungan

yang fleksibel antara komponen, sehingga memudahkan dalam proses perakitan dan pengujian sistem.



Gambar 2.12 Kabel Jumper

Dalam sistem keamanan ini, kabel jumper digunakan untuk menghubungkan semua komponen, termasuk sensor, mikrokontroler, dan perangkat lainnya. Dengan menggunakan kabel jumper, pengembang dapat dengan mudah melakukan modifikasi dan perbaikan pada sistem. Penelitian oleh Kurniawan et al. (2020) menunjukkan bahwa penggunaan kabel jumper yang baik dapat meningkatkan efisiensi dalam pengembangan sistem dan meminimalkan kesalahan dalam pengkabelan [4]. Dengan demikian, kabel jumper menjadi komponen penting dalam pengembangan sistem keamanan berbasis IoT.

2.14. Power Adaptor

Power Adaptor adalah perangkat yang digunakan untuk mengubah arus listrik dari sumber daya menjadi arus yang sesuai untuk digunakan oleh komponen elektronik. Adaptor ini penting untuk memastikan bahwa semua komponen dalam sistem mendapatkan daya yang stabil. Menurut penelitian oleh Muliadi et al. (2020), penggunaan adaptor yang tepat dapat meningkatkan kinerja sistem secara keseluruhan [10]. *Power Adaptor* ini menyediakan daya yang diperlukan untuk mengoperasikan mikrokontroler dan komponen lainnya dalam sistem keamanan.



Gambar 2.13 Power Adaptor

Dalam sistem keamanan ini, *Power Adaptor* digunakan untuk memberikan daya pada ESP32, sensor, dan perangkat lainnya. Dengan adanya adaptor, sistem dapat beroperasi dengan baik dan stabil, sehingga meningkatkan keandalan sistem keamanan. Penelitian oleh Muliadi et al. (2020) menunjukkan bahwa penggunaan adaptor yang sesuai dapat mempengaruhi kestabilan dan efisiensi sistem yang dibangun [10]. Dengan demikian, *Power adaptor* menjadi komponen penting dalam pengembangan sistem keamanan berbasis IoT.

2.15. Firebase

Firebase adalah platform yang menyediakan berbagai layanan backend untuk aplikasi web dan mobile, termasuk database real-time. Dalam konteks sistem keamanan, Firebase dapat digunakan untuk menyimpan dan menyinkronkan data secara real-time. Menurut penelitian oleh Kurniawan et al. (2021), penggunaan Firebase dalam sistem IoT dapat meningkatkan responsivitas dan efisiensi dalam pengelolaan data [8]. Dengan Firebase, data dari sensor dapat dikirim dan diakses secara langsung oleh pengguna melalui aplikasi mobile.



Gambar 2.14 Firebase

Dalam sistem keamanan ini, Firebase berfungsi sebagai basis data untuk menyimpan informasi pengguna, status pintu, dan notifikasi. Dengan menggunakan Firebase, pengguna dapat menerima notifikasi secara real-time ketika terdeteksi adanya usaha pembobolan. Penelitian oleh Kurniawan et al. (2021) menunjukkan bahwa penggunaan Firebase dalam sistem IoT dapat meningkatkan efektivitas pengelolaan data dan respons terhadap ancaman [8]. Dengan demikian, Firebase menjadi komponen kunci dalam pengembangan sistem keamanan berbasis IoT.

2.16. Arduino IDE

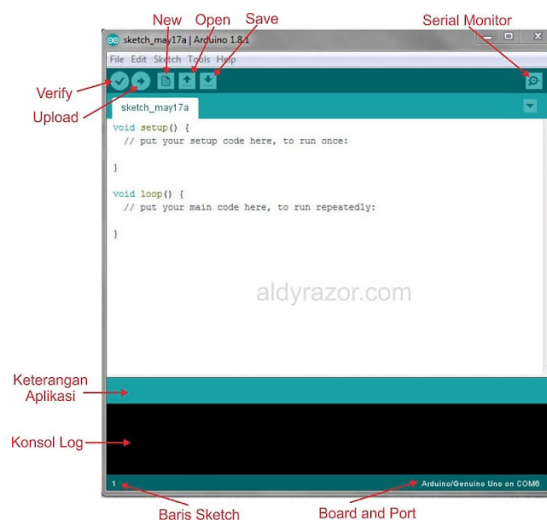
Arduino IDE (*Integrated Development Environment*) adalah perangkat lunak *open-source* yang digunakan untuk menulis, mengunggah, dan memprogram mikrokontroler berbasis platform Arduino, termasuk ESP32. Arduino IDE menyediakan antarmuka yang sederhana dan intuitif yang memungkinkan pengguna dari berbagai latar belakang, termasuk pemula, untuk memprogram perangkat keras tanpa harus memiliki pemahaman mendalam tentang arsitektur mikrokontroler [12].



Gambar 2.15 Arduino IDE

Arduino IDE mendukung berbagai bahasa pemrograman seperti C dan C++, serta menyediakan banyak pustaka (library) yang mempermudah integrasi dengan sensor dan aktuator. Salah satu kelebihan utama Arduino IDE adalah kemudahan penggunaan dalam mengunggah kode ke mikrokontroler hanya dengan sekali klik, serta ketersediaan serial monitor untuk debugging secara langsung.

Dalam penelitian sistem keamanan pintu berbasis IoT ini, Arduino IDE digunakan untuk menulis dan mengunggah program ke dalam mikrokontroler ESP32. Penggunaan Arduino IDE memungkinkan integrasi sensor fingerprint R503, sensor sentuh, dan sensor magnetik secara efisien, serta memungkinkan koneksi ke Firebase Realtime Database melalui Wi-Fi.



Gambar 2.16 Tampilan Arduino IDE

Berikut ini penjelasan dari masing-masing menu yang terdapat pada *software* Arduino IDE:

1. *Verify*, digunakan untuk meng-*compile* atau mem-*verify sketch coding* apakah masih ada kesalahan atau tidak. Jika masih terdapat *coding* yang salah biasanya akan muncul keterangan di bawah yaitu *error*.
2. *Upload*, digunakan untuk mengirimkan atau memasukan program ke dalam *board* yang ditentukan.
3. *New*, digunakan untuk membuat file baru atau membuka halaman *sketch* yang baru.
4. *Open*, digunakan untuk membuka proyek atau program yang pernah dibuat, dengan catatan proyek tersebut telah disimpan.

5. *Save*, ditujukan untuk menyimpan *sketch* atau program yang telah dibuat sebelumnya.
6. Serial Monitor, digunakan untuk menampilkan data yang telah dibuat setelah *sketch* tersebut di-*upload* kedalam *board* yang diperlukan, kemudian nantinya akan dijalankan, dan bisa dilihat pada serial monitor.
7. Keterangan Aplikasi, Digunakan untuk menampilkan informasi *error* atau pesan yang lainnya selama proses verifikasi atau upload sketch.
8. Konsol *Log*, Berfungsi untuk menampilkan dan pusat informasi terkait proses verifikasi dan pengunggahan sketsa program.

2.17. *Flowchart*

Flowchart atau diagram alur adalah representasi grafis dari urutan logika atau langkah-langkah dalam sebuah sistem. *Flowchart* membantu dalam menggambarkan proses secara sistematis, memudahkan pemahaman struktur program, serta mempercepat proses pengembangan dan debugging perangkat lunak. Dalam pengembangan sistem berbasis IoT, *flowchart* digunakan untuk merancang logika kontrol dan interaksi antara sensor, mikrokontroler, dan antarmuka pengguna secara visual.

2.17.1. Jenis – Jenis *Flowchart*

Flowchart atau diagram alur ini dibagi menjadi lima jenis, yaitu:

1. System *Flowchart*

Bagan yang memberikan gambaran komprehensif tentang alur kerja keseluruhan suatu sistem, termasuk urutan langkah-langkah dalam setiap prosedur yang ada. *Flowchart* ini membantu dalam memahami bagaimana berbagai komponen sistem berinteraksi satu sama lain.

2. Document *Flowchart*

Bagan yang secara khusus melacak aliran dokumen dan formulir dalam suatu sistem, termasuk bagaimana dokumen dibuat, didistribusikan, diproses, dan disimpan. *Flowchart* ini sangat berguna dalam

mengidentifikasi potensi masalah dalam alur dokumen dan meningkatkan efisiensi proses bisnis.

3. Schematic *Flowchart*

Mirip dengan sistem *Flowchart*, namun lebih detail dengan menyertakan representasi visual dari perangkat keras dan perangkat lunak yang digunakan dalam sistem. *Flowchart* ini membantu dalam memahami bagaimana teknologi mendukung proses bisnis dan mengidentifikasi area yang mungkin memerlukan peningkatan atau penggantian perangkat.

4. Program *Flowchart*

Merupakan bagan yang menjelaskan secara rinci langkah-langkah dalam suatu program komputer. Terdapat dua jenis program flowchart: Program *Logic Flowchart*, yang digunakan oleh analis sistem untuk menggambarkan logika program secara umum, dan *Detailed Computer Program Flowchart*, yang digunakan oleh programmer untuk merinci instruksi-instruksi program secara spesifik.

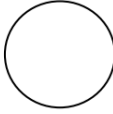
5. Proses *Flowchart*

Bagan ini banyak digunakan dalam bidang teknik industri untuk menggambarkan proses-proses dalam suatu prosedur. Analis sistem juga memanfaatkan proses *Flowchart* untuk memvisualisasikan langkah-langkah dalam suatu prosedur, sehingga memudahkan pemahaman dan analisis alur kerja.

2.17.2. Simbol Penghubung

Simbol penghubung atau alur, juga dikenal sebagai *connecting line*, digunakan untuk menghubungkan satu simbol dengan simbol lainnya dalam sebuah *flowchart*. Simbol ini menunjukkan arah aliran proses atau data dalam diagram. Simbol tersebut adalah sebagai berikut:

Tabel 2.8 Simbol Penghubung

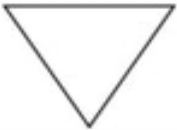
Simbol <i>Flowchart</i>	Keterangan Simbol
	Simbol Arus atau <i>Flow</i> Simbol yang digunakan untuk menghubungkan antara simbol satu dengan yang lainnya.
	Simbol <i>Communication Link</i> Simbol yang digunakan untuk menghubungkan antara simbol satu dengan yang lainnya.
	Simbol <i>Connector</i> Simbol untuk keluar atau masuk prosedur atau proses dalam lembar atau halaman yang sama.
	Simbol <i>Office Connector</i> Simbol untuk keluar atau masuk prosedur atau proses dalam lembar atau halaman

2.17.3. Simbol Proses

Simbol ini memberikan penjelasan tentang jenis operasi pengolahan dalam suatu proses atau prosedur. Simbol tersebut adalah sebagai berikut:

Tabel 2.9 Simbol Proses

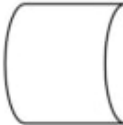
Simbol <i>Flowchart</i>	Keterangan Simbol
	Simbol Proses Simbol yang menunjukkan pengolahan yang dilakukan oleh computer.
	Simbol <i>Manual Operation</i> Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer.
	Simbol <i>Decision</i> Simbol untuk kondisi yang akan menghasilkan Beberapa kemungkinan jawaban atau aksi.
	Simbol <i>Predefined Process</i> Simbol untuk keluar atau masuk prosedur atau proses dalam lembar atau halaman

Simbol <i>Flowchart</i>	Keterangan Simbol
	Simbol Terminal Simbol yang menyatakan awal atau akhir suatu program
	Simbol Keying Operation Simbol yang menyatakan semua jenis operasi yang diproses dengan menggunakan mesin yang memiliki keyboard
	Simbol Offline Storage Simbol yang menjelaskan bahwa data dalam simbol ini akan disimpan
	Simbol Manual Input Simbol yang memasukkan data secara manual dengan online keyboard

2.17.4. Simbol Input dan Output

Simbol input atau output pada flowchart menggambarkan jenis perangkat yang digunakan untuk memasukkan atau mengeluarkan data. Beberapa contoh simbol input dan output meliputi:

Tabel 2.10 Simbol *Input* dan *Output*

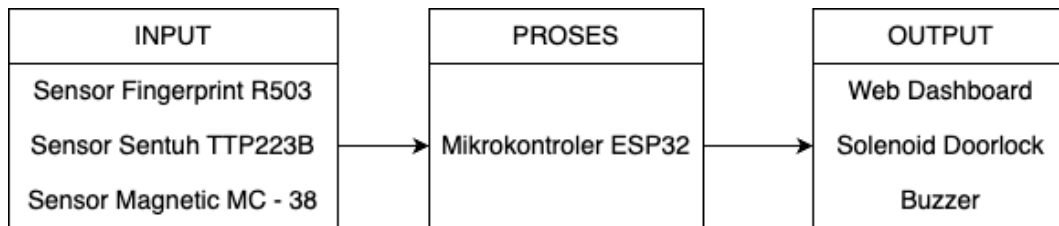
Simbol <i>Flowchart</i>	Keterangan Simbol
	Simbol Input atau Output Simbol yang menyatakan proses input atau output tanpa tergantung jenis peralatan.
	Simbol Punched Card Simbol yang menyatakan input berasal dari kart atau output ditulis ke kartu.
	Simbol Magnetic Tape Simbol yang menyatakan input berasal dari pita magnetis atau output disimpan ke pita.
	Simbol Disk Storage Simbol yang menyatakan input berasal dari disk atau output disimpan ke disk.

Simbol <i>Flowchart</i>	Keterangan Simbol
	<p>Simbol <i>Document</i></p> <p>Simbol yang menyatakan bahwa input berasal dari dokumen dalam bentuk kertas, atau output yang perlu dicetak di kertas.</p>
	<p>Simbol <i>Display</i></p> <p>Simbol yang menyatakan penggunaan peralatan output, misalnya layar, monitor, printer, dan lainnya.</p>

3. PERANCANGAN DAN ANALISIS

3.1. Analisis Rangkaian Secara Blok Diagram

Blok diagram adalah representasi visual yang sangat berharga dalam merancang dan memahami sistem yang kompleks. Melalui abstraksi yang cerdas, blok diagram menyederhanakan gambaran keseluruhan sistem dengan memecahnya menjadi blok-blok fungsional yang lebih mudah dipahami. Setiap blok mewakili komponen atau sub sistem tertentu, seperti sensor, pengolah data, aktuator, atau sumber daya. Dengan menggambarkan aliran informasi dan hubungan antar blok, blok diagram memberikan wawasan mendalam tentang bagaimana setiap komponen berkontribusi terhadap fungsi keseluruhan sistem.



Gambar 3.1 Blok Diagram

3.1.1. Blok Input

Blok input terdiri dari sejumlah sensor yang berfungsi sebagai alat penerima data dari lingkungan sekitar. Sensor Fingerprint R503 digunakan untuk melakukan verifikasi identitas pengguna berdasarkan sidik jari. Sensor ini memungkinkan sistem untuk mengenali pengguna yang sah dan memberikan akses hanya kepada orang-orang yang telah terdaftar sebelumnya. Selain itu, terdapat sensor sentuh TTP223B yang berperan sebagai tombol sentuh kapasitif. Sensor ini memudahkan pengguna untuk membuka pintu dari dalam ruangan tanpa perlu menggunakan verifikasi tambahan. Sementara itu, sensor magnetic MC-38 digunakan untuk mendeteksi kondisi pintu, apakah dalam keadaan terbuka atau tertutup. Semua data yang dikumpulkan dari ketiga sensor ini menjadi masukan penting bagi sistem untuk diproses lebih lanjut oleh mikrokontroler.

3.1.2. Blok Proses

Blok proses pada sistem ini dikendalikan oleh mikrokontroler ESP32 yang berfungsi sebagai otak dari seluruh rangkaian. ESP32 menerima data dari berbagai sensor input dan kemudian memprosesnya sesuai dengan logika program yang telah diprogram sebelumnya. Mikrokontroler ini akan mencocokkan sidik jari dari pengguna dengan data yang tersimpan, memantau status tombol sentuh, serta memeriksa kondisi pintu melalui sensor magnetik. Selain itu, ESP32 juga terhubung ke jaringan internet sehingga dapat mengirimkan data status sistem ke platform berbasis web. Dengan kemampuannya yang mendukung konektivitas Wi-Fi, ESP32 sangat efektif untuk sistem berbasis Internet of Things (IoT), memungkinkan pengawasan dan pengendalian sistem dari jarak jauh secara real-time.

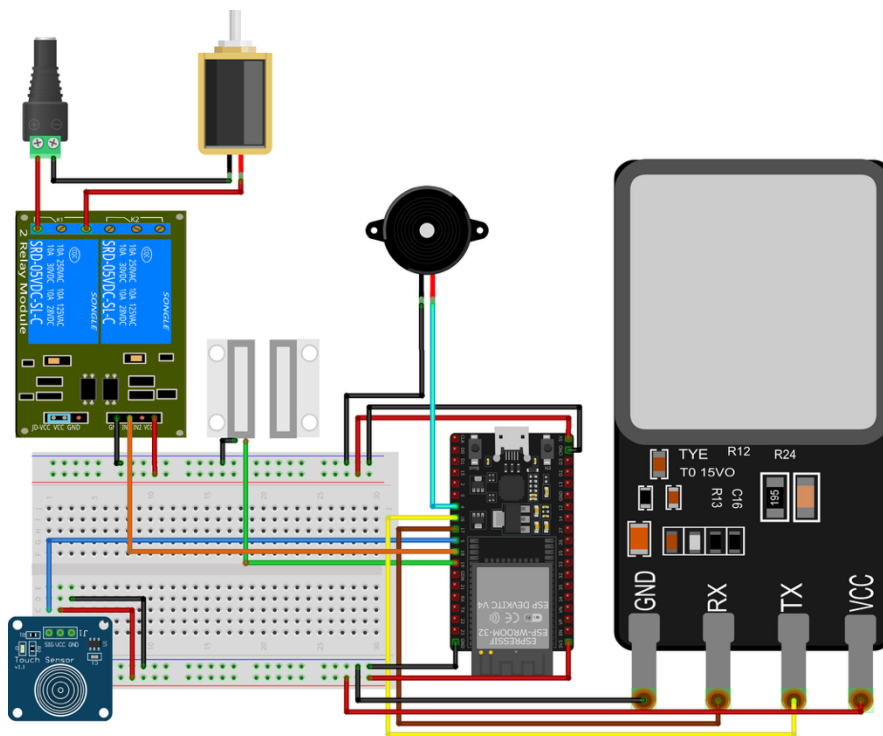
3.1.3. Blok Output

Blok output terdiri dari dua komponen utama, yaitu solenoid doorlock dan web dashboard. Solenoid doorlock berfungsi sebagai aktuator yang secara fisik mengunci dan membuka pintu sesuai perintah yang diberikan oleh mikrokontroler ESP32. Ketika verifikasi berhasil dilakukan oleh sensor fingerprint atau tombol sentuh diaktifkan dari dalam, solenoid akan membuka kunci pintu untuk memberikan akses. Sebaliknya, jika tidak ada verifikasi yang sah atau pintu harus dikunci kembali, maka solenoid akan menutup kunci pintu secara otomatis. Sementara itu, web dashboard berperan sebagai antarmuka pengguna yang menampilkan informasi sistem secara real-time. Melalui dashboard ini, pengguna dapat memantau aktivitas pintu, status verifikasi, serta menerima notifikasi jika terjadi kondisi yang mencurigakan. Kombinasi antara aktuator fisik dan antarmuka digital ini memastikan bahwa sistem bekerja secara efisien dan aman.

3.2. Analisa Rangkaian

Rangkaian sistem keamanan pintu ini menggunakan mikrokontroler ESP32 sebagai pusat pengendali yang terhubung dengan beberapa sensor dan aktuator. Pada bagian input, digunakan sensor fingerprint (tidak terlihat dalam gambar, namun diasumsikan terhubung melalui pin UART), sensor sentuh TTP223,

dan sensor magnetik MC-38. Sensor TTP223 berfungsi sebagai tombol sentuh untuk membuka pintu dari dalam tanpa melalui verifikasi fingerprint. Sensor magnetik MC-38 digunakan untuk mendeteksi status pintu, yaitu apakah dalam kondisi tertutup atau terbuka. Setiap sensor dihubungkan ke pin digital ESP32 dan mendapatkan catu daya dari pin 3.3V atau 5V ESP32, dengan jalur GND yang terhubung ke ground bersama untuk memastikan kestabilan sistem.



Gambar 3.2 Rangkaian Alat

Pada bagian output, digunakan modul relay 2 channel yang berfungsi sebagai saklar elektronik untuk mengontrol solenoid doorlock. ESP32 memberikan sinyal logika ke input modul relay untuk mengaktifkan atau menonaktifkan salah satu channel relay. Relay ini terhubung ke solenoid doorlock yang memerlukan tegangan lebih tinggi dari adaptor DC eksternal. Penggunaan relay menjadi penting karena memungkinkan ESP32 mengontrol perangkat dengan kebutuhan arus besar tanpa membebani langsung mikrokontroler. Saat verifikasi berhasil atau tombol sentuh diaktifkan, relay akan terhubung dan mengalirkan arus ke solenoid untuk membuka pintu. Sebaliknya, ketika kondisi tidak sesuai, relay tetap dalam keadaan terbuka untuk menjaga pintu tetap terkunci.

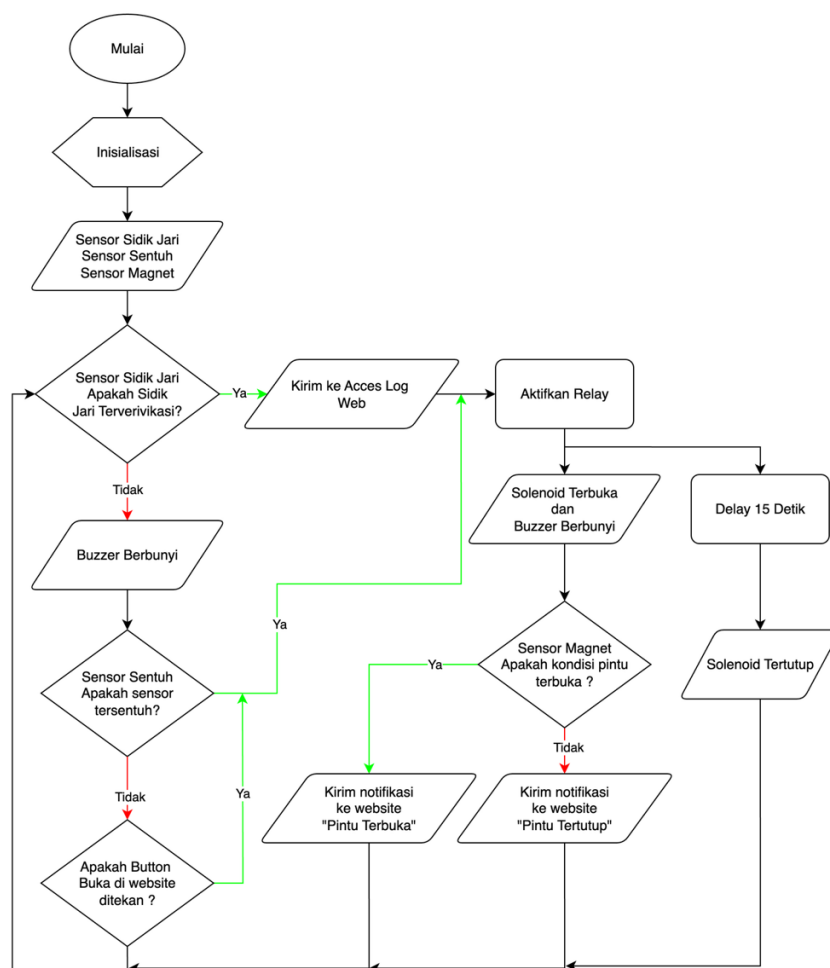
Dengan konfigurasi ini, ESP32 tidak hanya berfungsi sebagai pengendali utama perangkat keras, tetapi juga dapat mengirimkan status dan data ke web dashboard secara real-time melalui koneksi Wi-Fi. Hal ini memungkinkan pengguna untuk memantau kondisi pintu dari jarak jauh serta mengakses riwayat log aktivitas sistem keamanan.

3.3. Analisa Secara Diagram Alur

Berikut ini merupakan alur dari cara kerja alat Sistem Keamanan Pintu Berbasis IoT:

3.3.1. Analisa Diagram Alur Alat

Berikut ini merupakan alur dari cara kerja alat Sistem Keamanan Pintu Berbasis IoT:



Gambar 3.3 Diagram Alur Alat

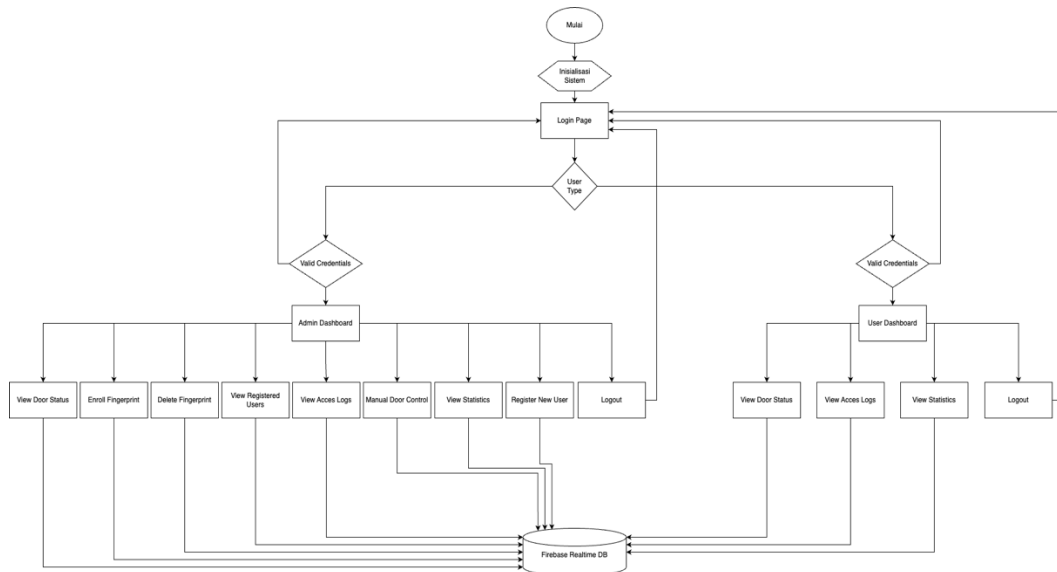
Diagram alur di atas menjelaskan proses kerja sistem keamanan pintu berbasis sidik jari, sensor sentuh, dan kontrol berbasis web. Proses dimulai dari tahap inisialisasi di mana seluruh sensor (sidik jari, sentuh, dan magnetik) diaktifkan dan sistem siap menerima input. Selanjutnya, sistem akan menunggu sidik jari ditempelkan pada sensor fingerprint. Jika sidik jari berhasil diverifikasi, data akses akan dikirimkan ke Firebase sebagai log, lalu sistem mengaktifkan relay untuk membuka solenoid dan mengeluarkan bunyi buzzer sebagai indikasi bahwa akses berhasil.

Apabila sidik jari tidak terverifikasi, sistem akan menyalakan buzzer sebagai peringatan bahwa akses ditolak. Pada kondisi ini, sistem memberikan alternatif akses dari dalam ruangan dengan memeriksa apakah sensor sentuh disentuh. Jika ya, pintu akan dibuka dan sistem mengirim notifikasi ke website bahwa pintu terbuka. Namun, jika sensor sentuh juga tidak aktif, sistem akan memeriksa apakah perintah buka pintu dikirim dari website. Jika tidak ada input yang valid, maka sistem kembali ke kondisi semula dan menunggu input ulang.

Setelah pintu terbuka, sensor magnetik digunakan untuk memantau status kondisi fisik pintu. Jika pintu benar-benar terbuka, maka sistem mengirimkan notifikasi "Pintu Terbuka" ke website. Sebaliknya, jika pintu kembali tertutup, maka sistem mengirimkan notifikasi "Pintu Tertutup" dan mulai menghitung waktu untuk menutup solenoid secara otomatis setelah 15 detik. Dengan demikian, sistem ini menggabungkan autentikasi biometrik, kontrol manual dari dalam, dan kontrol jarak jauh melalui dashboard web, yang semuanya diatur dalam satu alur kerja terintegrasi berbasis IoT.

3.3.2. Analisa Diagram Alur Website

Berikut ini merupakan alur dari cara kerja Website Sistem Keamanan Pintu Berbasis IoT:



Gambar 3.4 Diagram Alur Website

Dari Diagram Alur tersebut dapat diketahui bahwa Sistem dimulai dari halaman login dimana pengguna harus memasukkan kredensial email dan password. Setelah submit, sistem akan melakukan verifikasi otentikasi melalui Firebase Authentication. Jika kredensial tidak valid, pengguna akan tetap di halaman login dengan pesan error. Jika valid, sistem akan mengecek role pengguna (admin/user) melalui Realtime Database untuk menentukan redirect halaman.

Untuk pengguna dengan role admin, sistem akan mengarahkan ke dashboard admin yang berisi fitur-fitur lengkap. Admin dapat:

1. Memantau status pintu secara real-time (terbuka/terkunci)
2. Melakukan enroll fingerprint baru dengan memasukkan ID dan username
3. Menghapus fingerprint yang terdaftar
4. Melihat daftar pengguna terdaftar baik via fingerprint maupun email
5. Mengakses log akses lengkap beserta filter waktu
6. Mengontrol pintu secara manual (membuka paksa)
7. Mendaftarkan pengguna baru via email/password
8. Menganalisis statistik akses dalam bentuk grafik interaktif

Sedangkan untuk pengguna biasa, dashboard yang ditampilkan lebih terbatas meliputi:

1. Informasi status pintu terkini
2. Riwayat akses personal

3. Visualisasi statistik akses dasar

Dalam mekanisme Realtime Update, Seluruh dashboard secara kontinu melakukan listening ke Firebase Realtime Database untuk:

1. Update status pintu (setiap perubahan status langsung ter-refresh)
2. Notifikasi darurat jika pintu terbuka >1 menit (emergency alert)
3. Penyegaran otomatis log akses ketika ada aktivitas baru
4. Pembaruan grafik statistik ketika ada data baru

Dalam proses Logout baik admin maupun user dapat melakukan logout yang akan:

1. Menghapus session auth
2. Mengosongkan state aplikasi
3. Mengarahkan kembali ke halaman login
4. Membersihkan semua listener database untuk optimasi performa

Dalam Integrasi dengan Hardware alur admin terdapat interaksi dua arah dengan perangkat fingerprint reader melalui node khusus di Firebase:

1. commands/enroll untuk proses registrasi fingerprint baru
2. commands/delete untuk penghapusan ID fingerprint
3. commands/door untuk kontrol pintu manual

Dalam Penanganan Error Sistem memiliki mekanisme error handling berupa:

1. Toast notification untuk operasi gagal
2. Validasi form input client-side
3. Konfirmasi aksi kritis (seperti delete user)
4. Auto-retry untuk koneksi database terputus

Alur ini menjamin keamanan berbasis role sekaligus memberikan pengalaman real-time baik untuk admin maupun user biasa, dengan semua data tersimpan terpusat di Firebase Realtime Database.

3.4. Perancangan Alat

Perancangan alat sistem keamanan pintu ini bertujuan untuk merealisasikan prototipe pintu cerdas berbasis Internet of Things (IoT) yang dapat meningkatkan keamanan akses pada lingkungan laboratorium. Sistem ini

menggunakan mikrokontroler ESP32 sebagai pusat kendali, dilengkapi dengan sensor fingerprint R503 untuk autentikasi pengguna, sensor sentuh TTP223B untuk membuka pintu dari dalam secara manual, dan sensor magnetik MC-38 untuk memantau kondisi buka atau tutup pintu. Selain itu, sistem ini menggunakan solenoid doorlock sebagai aktuator kunci pintu otomatis yang dikendalikan oleh modul relay.

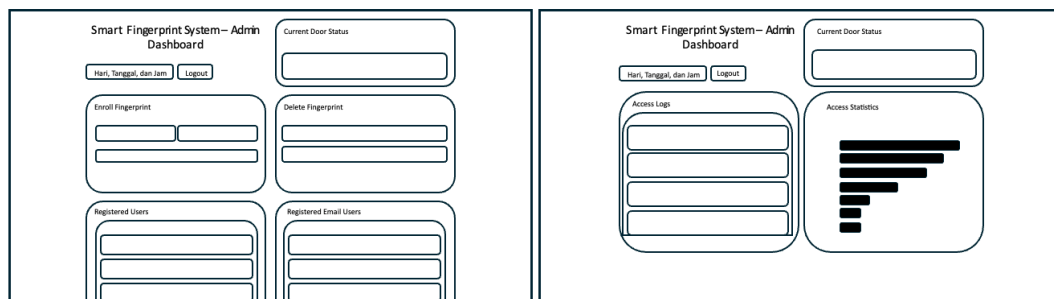


Gambar 3.5 Perancangan Design Alat Tampak Dalam Dan Luar

Secara fisik, perangkat dirakit pada maket pintu mini dengan dimensi sekitar 15 cm x 25 cm, menggunakan bahan seperti PVC board atau karton tebal. Sensor fingerprint R503 dipasang di bagian luar pintu sebagai antarmuka utama autentikasi pengguna. Sementara itu, di sisi dalam pintu dipasang solenoid doorlock secara horizontal sehingga batang penguncinya dapat bergerak keluar-masuk melewati sisi daun pintu. Di dekatnya dipasang sensor sentuh TTP223B sebagai alternatif pembuka pintu dari dalam, dan sensor magnetik MC-38 ditempatkan di sisi pintu dan kusen untuk mendeteksi apakah pintu sedang terbuka atau tertutup. Semua komponen ini dihubungkan ke ESP32, yang diletakkan di dalam kotak kontrol.

3.5. Perancangan Website

Perancangan website pada sistem keamanan pintu ini berfungsi sebagai dashboard web yang memungkinkan pengguna dan admin memantau serta mengelola akses pintu secara *real-time*. Website ini dikembangkan dengan tujuan memberikan kemudahan dalam mengontrol sistem dari jarak jauh, menampilkan status pintu, mencatat log akses, serta menyediakan fitur manajemen data pengguna. Website terhubung dengan *Firebase Realtime Database* yang menjadi media penyimpanan dan sinkronisasi data antara perangkat keras (ESP32) dan antarmuka pengguna.



Gambar 3.6 Perancangan Design Website Halaman Admin Dan User

Struktur tampilan website dirancang dengan dua halaman utama, yaitu halaman admin dan halaman *user*. Halaman admin memiliki akses penuh untuk melakukan pendaftaran dan penghapusan sidik jari, memantau log akses pengguna, serta mengatur status pintu secara manual melalui fitur kontrol buka/tutup pintu. Sementara halaman *user* hanya dapat melihat status pintu saat ini, notifikasi akses, dan riwayat akses pribadinya. Website dikembangkan menggunakan HTML, CSS, dan JavaScript, dengan bantuan pustaka tambahan seperti *Bootstrap* untuk tampilan responsif dan *Firebase SDK* untuk komunikasi langsung dengan *database*.

Dalam implementasinya, halaman utama website akan menampilkan informasi status pintu secara *real-time* (terbuka atau tertutup, terkunci atau tidak), notifikasi aktivitas terbaru, serta waktu akses terakhir. Selain itu, website juga dilengkapi dengan fitur *enroll* (daftar sidik jari) dan *delete* (hapus ID sidik jari), di mana perintah yang diberikan melalui *interface* web akan dikirim ke *Firebase* dan kemudian diproses oleh ESP32. Seluruh interaksi dengan *Firebase* diamankan menggunakan *Firebase Authentication*, dan komunikasi dilakukan melalui protokol HTTPS untuk menjamin keamanan data. Dengan adanya sistem ini, pengguna dapat

mengelola dan memantau akses pintu dengan lebih efisien, aman, dan fleksibel dari mana saja secara online.

3.6. Analisa Program

Analisa program dilakukan untuk menjelaskan fungsi dari setiap bagian kode yang terdapat dalam sistem, baik dari sisi perangkat keras (alat) maupun dari sisi antarmuka pengguna (website). Berikut adalah uraian fungsi dalam bentuk tabel yang terbagi menjadi dua sub-sub bab.

3.6.1. Analisa Program Alat

Program alat dikembangkan menggunakan bahasa C++ pada platform Arduino IDE dan berjalan di atas mikrokontroler ESP32. Berikut adalah analisis program untuk alat yang dibuat:

Tabel 3.1 Analisis Program Alat

Program	Analisis Program
<pre>#include <Adafruit_Fingerprint.h> #include <WiFi.h> #include <Firebase_ESP_Client.h> #include "addons/TokenHelper.h" #include <time.h></pre>	Mengimpor library yang diperlukan untuk mengakses sensor sidik jari, koneksi WiFi, layanan Firebase, dan sinkronisasi waktu dari NTP server.
<pre>HardwareSerial mySerial(2); Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);</pre>	Mendefinisikan koneksi serial ke sensor fingerprint menggunakan UART2 dan menginisialisasi objek sensor fingerprint.
<pre>#define WIFI_SSID "SSID" #define WIFI_PASSWORD "PASSWORD" #define FIREBASE_DB_URL "https://smart-door-ae096-default-rtdb.asia-southeast1.firebaseio.com" #define FIREBASE_API_KEY "AIzaSyD-OeD3Rb3_k1svki0WVByU6VAZFgvZG6M" #define FIREBASE_USER_EMAIL</pre>	Mendeklarasikan kredensial jaringan WiFi dan konfigurasi autentikasi Firebase.

<pre>"EMAIL" #define FIREBASE_USER_PASSWORD "PASSWORD"</pre>	
<pre>FirebaseData fbdo; FirebaseAuth auth; FirebaseConfig config;</pre>	Objek untuk mengelola koneksi dan data dengan Firebase.
<pre>#define SOLENOID_PIN 18 #define TOUCH_SENSOR_PIN 5 #define MAGNET_SENSOR_PIN 19 #define BUZZER_PIN 4 uint8_t id; String username;</pre>	Mendefinisikan pin I/O untuk komponen seperti solenoid, sensor sentuh, buzzer dan sensor magnetik.
<pre>const char* ntpServer = "pool.ntp.org"; const long gmtOffset_sec = 7 * 3600; const int daylightOffset_sec = 0; unsigned long previousMillis = 0; const unsigned long interval = 100; bool solenoidActive = false; unsigned long solenoidStartTime = 0; const unsigned long solenoidDuration = 15000; bool currentDoorState = false; bool lastDoorState = false; unsigned long lastDebounceTime = 0; const unsigned long debounceDelay = 50; struct DoorStatus { bool isOpen; time_t timestamp; };</pre>	Mendefinisikan Variable global untuk input dan output
<pre>void buzzerAccessGranted() { tone(BUZZER_PIN, 1000, 100); delay(150);</pre>	Fungsi untuk membunyikan buzzer

<pre> tone(BUZZER_PIN, 1500, 100); delay(150); noTone(BUZZER_PIN); } void buzzerAccessDenied() { tone(BUZZER_PIN, 400, 500); delay(500); noTone(BUZZER_PIN); } </pre>	<p>sesuai kondisi, apakah akses diberikan atau tidak.</p>
<pre> void setup() { Serial.begin(115200); pinMode(SOLENOID_PIN, OUTPUT); digitalWrite(SOLENOID_PIN, HIGH); pinMode(TOUCH_SENSOR_PIN, INPUT); pinMode(MAGNET_SENSOR_PIN, INPUT_PULLUP); mySerial.begin(57600, SERIAL_8N1, 16, 17); WiFi.mode(WIFI_STA); WiFi.begin(WIFI_SSID, WIFI_PASSWORD); Serial.print("Menghubungkan ke WiFi"); while (WiFi.status() != WL_CONNECTED) { delay(500); Serial.print("."); } Serial.println("\nTerhubung ke WiFi"); Serial.print("IP Address: "); Serial.println(WiFi.localIP()); configTime(gmtOffset_sec, daylightOffset_sec, ntpServer); struct tm timeinfo; if (!getLocalTime(&timeinfo)) { Serial.println("Gagal mendapatkan waktu dari NTP"); } } </pre>	<p>Menyiapkan komunikasi serial, menginisialisasi WiFi, waktu NTP, koneksi Firebase, serta deteksi awal sensor fingerprint.</p>

```

    ESP.restart();
}

config.database_url = FIREBASE_DB_URL;
config.api_key = FIREBASE_API_KEY;
auth.user.email = FIREBASE_USER_EMAIL;
auth.user.password = FIREBASE_USER_PASSWORD;

Firebase.reconnectNetwork(true);
fbdo.setBSSLBufferSize(4096, 1024);
fbdo.setResponseSize(2048);
Firebase.reconnectWiFi(true);
Firebase.begin(&config, &auth);

unsigned long startTime = millis();
while (!Firebase.ready() && millis() - startTime <
20000) {
    Serial.print(".");
    delay(500);
}

if (!Firebase.ready()) {
    Serial.println("\nFailed to authenticate with Firebase");
    ESP.restart();
}

Serial.println("\nFirebase authenticated");

if (finger.verifyPassword()) {
    Serial.println("Sensor sidik jari terdeteksi!");
} else {
    Serial.println("Sensor tidak terdeteksi!");
    while (1) { delay(1); }
}

```

<pre> finger.getTemplateCount(); Serial.print("Jumlah template tersimpan: "); Serial.println(finger.templateCount); Serial.println("Menunggu input..."); } </pre>	
<pre> void loop() { String status; unsigned long currentMillis = millis(); if (Firebase.RTDB.getString(&fbdo, "/commands/status/status")) { Serial.println("Membaca Command"); status = fbdo.stringData(); if (status == "e") { startEnrollment(); delay(100); clearFirebaseData(); } else if (status == "d") { if (Firebase.RTDB.getString(&fbdo, "/commands/delete/id")) { id = fbdo.intData(); } deleteFingerprint(); delay(100); clearFirebaseData(); } } static unsigned long webCheck = 0; if(currentMillis - webCheck >= 1000) { webCheck = currentMillis; if (Firebase.RTDB.getString(&fbdo, "/commands/door") && !solenoidActive) { if(fbdo.stringData() == "open") { </pre>	<p>Fungsi utama yang terus berjalan. Menangani pembacaan command dari Firebase (enroll, delete, buka pintu), membaca sensor sentuh untuk pembukaan manual, kontrol durasi aktif solenoid, serta pembacaan dan verifikasi sidik jari.</p>

<pre> Serial.println("Membuka pintu dari web"); solenoidActive = true; solenoidStartTime = currentMillis; digitalWrite(SOLENOID_PIN, LOW); buzzerAccessGranted(); } } } if(currentMillis - previousMillis >= interval) { previousMillis = currentMillis; if(digitalRead(TOUCH_SENSOR_PIN) == HIGH && !solenoidActive) { Serial.println("Pintu dibuka manual via touch sensor!"); solenoidActive = true; solenoidStartTime = currentMillis; digitalWrite(SOLENOID_PIN, LOW); buzzerAccessGranted(); } } if(solenoidActive && (currentMillis - solenoidStartTime >= solenoidDuration)) { digitalWrite(SOLENOID_PIN, HIGH); solenoidActive = false; Serial.println("Pintu dikunci kembali"); } updateDoorStatus(); static unsigned long lastFingerprintCheck = 0; if(currentMillis - lastFingerprintCheck >= 1000) { lastFingerprintCheck = currentMillis; </pre>	
--	--

<pre> Serial.println("Menunggu sidik jari..."); getFingerprintID(); finger.LEDcontrol(FINGERPRINT_LED_BREATHING, 50, FINGERPRINT_LED_PURPLE); } } </pre>	
<pre> void startEnrollment() { if (Firebase.RTDB.getJSON(&fbdo, "/commands/enroll")) { finger.LEDcontrol(FINGERPRINT_LED_BREATHING, 150, FINGERPRINT_LED_BLUE); FirebaseJson json; FirebaseJsonData data; json.setJsonData(fbdo.jsonString()); Firebase.RTDB.setString(&fbdo, "/caption/enroll", "Memulai Enroll..."); delay(1000); Firebase.RTDB.setString(&fbdo, "/caption/enroll", "Tempelkan Jari"); json.get(data, "id"); id = data.to<int>(); json.get(data, "username"); username = data.to<String>(); if (id == 0 id > 127) { Serial.println("ID tidak valid!"); return; } if (getFingerprintEnroll() == FINGERPRINT_OK) { sendToFirebase(id, username); Serial.println("Enroll sukses!"); } } </pre>	<p>Memulai proses pendaftaran (enroll) sidik jari baru berdasarkan data yang dikirim dari website melalui Firebase.</p>

<pre> Firebase.RTDB.setString(&fbdo, "/caption/enroll", "Enroll sukses!"); } else { Serial.println("Enroll gagal!"); Firebase.RTDB.setString(&fbdo, "/caption/enroll", "Enroll gagal!"); } } } } </pre>	
<pre> void updateDoorStatus() { bool reading = digitalRead(MAGNET_SENSOR_PIN); if (reading != lastDoorState) { lastDebounceTime = millis(); } if ((millis() - lastDebounceTime) > debounceDelay) { if (reading != currentDoorState) { currentDoorState = reading; FirebaseJson json; json.set("isOpen", currentDoorState); json.set("timestamp", time(nullptr)); if(Firebase.RTDB.setJSON(&fbdo, "/door", &json)) { Serial.print("Status Pintu: "); Serial.println(currentDoorState ? "TERBUKA" : "TER TUTUP"); } else { Serial.println("Gagal mengirim status pintu ke Firebase"); } } } lastDoorState = reading; } </pre>	<p>Memeriksa status pintu dari sensor magnetik dan memperbarui data ke Firebase secara real-time.</p>

}	
<pre> void sendToFirebase(int id, String username) { FirebaseJson json; json.set("id", id); json.set("username", username); json.set("timestamp", time(NULL)); Firebase.RTDB.setJSON(&fbdo, "/fingerprints/" + String(id), &json); } </pre>	<p>Mengirim data user dan timestamp hasil pendaftaran ke Firebase.</p>
<pre> void getFingerprintID() { uint8_t p = finger.getImage(); if(p == FINGERPRINT_OK) { p = finger.image2Tz(); if(p != FINGERPRINT_OK) return; p = finger.fingerSearch(); if(p == FINGERPRINT_OK) { finger.LEDcontrol(FINGERPRINT_LED_FLASHING, 20, FINGERPRINT_LED_BLUE, 20); logAccess(finger.fingerID, true); digitalWrite(SOLENOID_PIN, LOW); buzzerAccessGranted(); solenoidActive = true; solenoidStartTime = millis(); delay(1000); } else { finger.LEDcontrol(FINGERPRINT_LED_FLASHING, 25, FINGERPRINT_LED_RED, 10); buzzerAccessDenied(); logAccess(0, false); delay(1000); } } } </pre>	<p>Membaca dan mencocokkan sidik jari dengan template yang tersimpan, lalu membuka pintu jika cocok atau memberikan notifikasi jika tidak cocok.</p>

<pre> } } </pre>	
<pre> uint8_t getFingerprintEnroll() { int p = -1; while (p != FINGERPRINT_OK) { p = finger.getImage(); delay(50); } p = finger.image2Tz(1); if (p != FINGERPRINT_OK) return p; Serial.println("Angkat jari"); Firebase.RTDB.setString(&fbdo, "/caption/enroll", "Angkat jari"); delay(2000); while (p != FINGERPRINT_NOFINGER) { p = finger.getImage(); } Serial.println("Letakkan jari yang sama"); Firebase.RTDB.setString(&fbdo, "/caption/enroll", "Letakkan jari yang sama"); while (p != FINGERPRINT_OK) { p = finger.getImage(); delay(50); } p = finger.image2Tz(2); if (p != FINGERPRINT_OK) return p; p = finger.createModel(); if (p != FINGERPRINT_OK) return p; p = finger.storeModel(id); </pre>	<p>Proses dua tahap perekaman sidik jari dan penyimpanan template ke sensor R503.</p>

<pre> return p; } </pre>	
<pre> void deleteFingerprint() { if (Firebase.RTDB.getJSON(&fbdo, "/commands/delete")) { FirebaseJson json; FirebaseJsonData data; json.setJsonData(fbdo.jsonString()); json.get(data, "id"); id = data.to<int>(); if (finger.deleteModel(id) == FINGERPRINT_OK) { Firebase.RTDB.deleteNode(&fbdo, "/fingerprints/" + String(id)); } } } </pre>	<p>Menghapus template sidik jari dari sensor dan Firebase berdasarkan ID yang dikirim dari web.</p>
<pre> void clearFirebaseData() { if (Firebase.RTDB.deleteNode(&fbdo, "/commands")) { Serial.println("Membersihkan Command Firebase."); } else { Serial.println("Gagal Membersihkan Command Firebase."); Serial.println("REASON: " + fbdo.errorReason()); } } </pre>	<p>Menghapus command dari node Firebase setelah diproses agar tidak diulang.</p>
<pre> String getTimeString() { struct tm timeinfo; if (!getLocalTime(&timeinfo)) return "unknown"; char buffer[30]; strftime(buffer, sizeof(buffer), "%A, %d %B %Y %H:%M:%S", &timeinfo); return String(buffer); } </pre>	<p>Mengambil waktu terkini dalam format string untuk dicatat ke Firebase.</p>
<pre> void logAccess(uint8_t id, bool granted) { </pre>	

<pre> time_t now = time(NULL); String formattedTime = getTimeString(); FirebaseJson json; json.set("id", (id == 0) ? "Unknown" : String(id)); json.set("status", granted ? "granted" : "denied"); json.set("timestamp", now); json.set("formatted_time", formattedTime); json.set("date", formattedTime); Firebase.RTDB.setJSON(&fbdo, "/access_logs/" + String(now), &json); } </pre>	Mencatat log akses (berhasil atau ditolak) ke Firebase beserta waktu akses dan ID pengguna.
--	---

3.6.2. Analisa Program Website

Website berfungsi sebagai antarmuka pengguna yang menampilkan status pintu, daftar pengguna, log akses, serta memberikan kontrol administratif seperti *enroll*, *delete*, dan *manual open*. Website terhubung langsung ke Firebase melalui JavaScript API.

Tabel 3.2 Analisis Program Website (HTML)

Program	Analisis Program
<pre> <div class="card"> <div class="card-header"> door_front <h2>Current Door Status</h2> </div> <div class="status-list"> <div class="list-item" id="doorStatus"> <div> hourglass_empty Loading door status... <div class="timestamp">Checking system</div> </div> </div> </pre>	Menampilkan status pintu terkini. JavaScript akan mengganti isi `#doorStatus` secara real-time berdasarkan data dari Firebase (`/door`). Jika pintu terbuka >1 menit, muncul alert darurat.

<pre> </div> </div> </div> </div> </pre>	
<pre> <div class="card"> <div class="card-header"> fingerprint <h2>Enroll Fingerprint</h2> </div> <div class="input-group"> <input type="number" id="enrollId" placeholder="ID (1-127)" min="1" max="127"> <input type="text" id="enrollUsername" placeholder="Username"> </div> <button onclick="enroll()"> person_add Enroll User </button> <div class="caption" id="caption_enroll"></div> </div> </pre>	<p>Admin bisa memasukkan ID dan nama pengguna untuk mendaftarkan fingerprint. Setelah klik, fungsi `enroll()` akan mengirim data ke Firebase (`/commands/enroll`).</p> <p>Feedback tampil di `#caption_enroll`.</p>
<pre> <div class="card"> <div class="card-header"> delete <h2>Delete Fingerprint</h2> </div> <div class="input-group"> <input type="number" id="deleteId" placeholder="ID (1-127)" min="1" max="127"> </div> <button onclick="deleteFp()" class="danger-btn"> warning Delete User </button> </div> </pre>	<p>Admin bisa menghapus user fingerprint berdasarkan ID. Fungsi `deleteFp()` akan menulis ke `/commands/delete` dan mengatur `commands/status/status = "d"`.</p>

<pre> <div class="card"> <div class="card-header"> people <h2>Registered Users</h2> </div> <div class="status-list" id="registeredUsers"> <div class="empty-state"> group <p>Loading registered users...</p> </div> </div> </div> </pre>	<p>Menampilkan daftar user yang sudah memiliki fingerprint. Data ditarik dari Firebase (`/fingerprints`) dan ditampilkan dalam elemen `#registeredUsers`.</p>
<pre> <div class="card"> <div class="card-header"> alternate_email <h2>Registered Email Users</h2> </div> <div class="status-list" id="emailUsers"> <div class="empty-state"> group <p>Loading registered email users...</p> </div> </div> </div> </pre>	<p>Menampilkan user yang terdaftar dengan email (via Firebase Authentication). Data dari `/users`. Tombol delete tersedia untuk hapus user.</p>
<pre> <div class="card"> <div class="card-header"> history <h2>Access Logs</h2> </div> <button onclick="clearLogs()" class="danger-btn" style="margin-bottom: 15px;"> delete_sweep Clear All Logs </button> </pre>	<p>Menampilkan riwayat akses (granted/denied) pengguna fingerprint. Data berasal dari `/access_logs`. Ada tombol Clear All Logs yang memanggil `clearLogs()`.</p>

<pre> <div class="status-list" id="accessLogs"> <div class="empty-state"> history <p>Loading access logs...</p> </div> </div> </div> </pre>	
<pre> <div class="card"> <div class="card-header"> lock_open <h2>Manual Door Control</h2> </div> <button onclick="openDoor()" class="success-btn"> lock_open Open Door Now </button> <div class="caption" id="caption_door"></div> </div> </pre>	<p>Memberi admin kemampuan membuka pintu secara manual melalui Firebase (<code>/commands/door = "open"</code>). Hasil ditampilkan di <code>#caption_door</code>.</p>
<pre> <div class="card" style="grid-column: span 1;"> <div class="card-header"> pie_chart <h2>Access Statistics</h2> </div> <div class="chart-container"> <canvas id="userChart"></canvas> </div> </div> </pre>	<p>Statistik visual pengguna berdasarkan log akses. JavaScript mengisi <code>#userChart</code> menggunakan Chart.js dan Firebase data dari <code>/access_logs</code> dan <code>/fingerprints</code>.</p>
<pre> <div class="card"> <div class="card-header"> person_add <h2>Register New User</h2> </div> <div id="adminRegisterAlert" class="caption" style="display: none;"></div> </pre>	<p>Admin dapat membuat akun baru untuk login menggunakan email & password. Disimpan ke Firebase Authentication + Firebase Realtime DB (<code>/users</code>).</p>

```

<div style="margin: 15px 0;">
  <!-- Email Input -->
  <div style="margin-bottom: 20px;">
    <label for="newUserEmail" style="display: block;
margin-bottom: 8px; font-size: 14px; color: var(--text-
light); font-weight: 500;">Email Address</label>
    <div style="position: relative;">
      <span class="material-icons" style="position:
absolute; left: 15px; top: 50%; transform: translateY(-
50%); color: var(--text-light); font-size:
20px;">email</span>
      <input type="email" id="newUserEmail"
placeholder="user@example.com" required
style="width: 100%; padding: 12px 15px
12px 45px; border: 1px solid #e9ecef;
border-radius: var(--border-radius); font-size:
16px; transition: var(--transition);
background: #f8f9fa;">
    </div>
  </div>

  <!-- Password Input -->
  <div style="margin-bottom: 20px;">
    <label for="newUserPassword" style="display:
block; margin-bottom: 8px; font-size: 14px; color: var(--
text-light); font-weight: 500;">Password</label>
    <div style="position: relative;">
      <span class="material-icons" style="position:
absolute; left: 15px; top: 50%; transform: translateY(-
50%); color: var(--text-light); font-size:
20px;">lock</span>
      <input type="password" id="newUserPassword"
placeholder="Enter password" required

```

<pre> style="width: 100%; padding: 12px 15px 12px 45px; border: 1px solid #e9ecef; border-radius: var(--border-radius); font-size: 16px; transition: var(--transition); background: #f8f9fa;"> </div> </div> <!-- Role Selection --> <div style="margin-bottom: 20px;"> <label for="newUserRole" style="display: block; margin-bottom: 8px; font-size: 14px; color: var(--text- light); font-weight: 500;">User Role</label> <div style="position: relative;"> person <select id="newUserRole" style="width: 100%; padding: 12px 15px 12px 45px; border: 1px solid #e9ecef; border-radius: var(--border-radius); font- size: 16px; transition: var(--transition); background: #f8f9fa; appearance: none; - webkit-appearance: none;"> <option value="user">Regular User</option> <option value="admin">Administrator</option> </select> arrow_drop_down </div> </div> </div> </pre>	
--	--

<pre> </div> <button onclick="registerUser()" style="margin-top: 10px;"> person_add Register New User </button> </div> </pre>	
--	--

Tabel 3.3 Anlisis Program Website (Javascript)

Program	Analisis Program
<pre> onValue(ref(db, 'door'), (snapshot) => { const doorStatusElement = document.getElementById('doorStatus'); if (snapshot.exists()) { const doorData = snapshot.val(); const isOpen = doorData.isOpen; const timestamp = doorData.timestamp * 1000; // Update UI doorStatusElement.innerHTML = ` <div> \${isOpen ? 'warning' : 'lock'} Status: \${isOpen ? 'OPEN' : 'CLOSED'} <div class="timestamp"> Last updated: \${new Date(timestamp).toLocaleString('id-ID')} </div> </div> </pre>	<p>JavaScript memantau perubahan data door di Firebase secara real-time dan menampilkan status pintu (terbuka atau tertutup) beserta waktu pembaruan terakhir, sekaligus memunculkan peringatan jika pintu dibiarkan terbuka lebih dari satu menit.</p>

<pre> <div class="status-tag \${isOpen ? 'denied' : 'granted'}"> \${isOpen ? 'Open' : 'Closed'} </div> `; } }); </pre>	
<pre> window.enroll = async () => { const id = document.getElementById('enrollId').value; const username = document.getElementById('enrollUsername').value; if (!id !username) { showError('caption_enroll', 'Please fill all fields!'); return; } await set(ref(db, 'commands/enroll'), { id: parseInt(id), username }); await set(ref(db, 'commands/status/status'), "e"); }; </pre>	<p>Ketika tombol "Enroll User" ditekan, JavaScript mengambil input ID dan username, lalu mengirimnya ke path <code>commands/enroll</code> di Firebase, serta memicu status <code>e</code> untuk memberi sinyal ke ESP32 bahwa proses pendaftaran fingerprint telah dimulai, dan hasilnya ditampilkan dari <code>caption/enroll</code>.</p>
<pre> window.deleteFp = async () => { const id = document.getElementById('deleteId').value; if (!id !confirm('Are you sure you want to delete \${id}?')) return; await set(ref(db, 'commands/delete'), { id: parseInt(id) }); await set(ref(db, 'commands/status/status'), "d"); }; </pre>	<p>JavaScript menghapus user fingerprint dengan menulis ID ke path <code>commands/delete</code> di Firebase dan mengatur status <code>d</code> yang dibaca oleh ESP32, lalu mengosongkan input dan menampilkan hasil prosesnya.</p>
<pre> onValue(ref(db, 'fingerprints'), (snapshot) => { const container = document.getElementById('registeredUsers'); container.innerHTML = ""; </pre>	<p>JavaScript mendengarkan perubahan data di path <code>fingerprints</code> dan menampilkan daftar pengguna fingerprint aktif</p>

<pre> snapshot.forEach((child) => { const data = child.val(); const item = document.createElement('div'); item.className = 'list-item'; item.innerHTML = `     <div>       <span class="material-icons">person</span>       <strong>ID          \${child.key}</strong>:       \${data.username}       <div class="timestamp">         Registered: \${formatTime(data.timestamp)}       </div>     </div>   `; container.appendChild(item); }); }); </pre>	<p>dengan ID, nama pengguna, dan waktu pendaftaran secara dinamis ke dalam elemen HTML.</p>
<pre> onValue(ref(db, 'users'), (snapshot) => { const container = document.getElementById('emailUsers'); container.innerHTML = ""; snapshot.forEach((child) => { const user = child.val(); const item = document.createElement('div'); item.className = 'list-item'; item.innerHTML = `       <div>         <span class="material-icons">\${user.role === 'admin' ? 'admin_panel_settings' : 'person'}</span>         <strong>\${user.email}</strong>         <span class="status-tag">\${user.role}</span>       </div>     `; container.appendChild(item); }); }); </pre>	<p>Script mengambil data dari node users di Firebase Realtime Database dan menampilkan daftar pengguna email lengkap dengan peran dan waktu pendaftaran, serta menyediakan opsi untuk menghapus user jika diperlukan.</p>

});	
<pre> onValue(ref(db, 'access_logs'), (snapshot) => { const container = document.getElementById('accessLogs'); const logs = []; snapshot.forEach((child) => logs.push(child.val())); logs.sort((a, b) => b.timestamp - a.timestamp); logs.forEach(async (log) => { const userSnap = await get(ref(db, 'fingerprints/\${log.id}')); const username = userSnap.exists() ? userSnap.val().username : 'Unknown'; const item = document.createElement('div'); item.className = `list-item \${log.status}`; item.innerHTML = ` <div> \${log.status} 'granted' ? 'check_circle' : 'cancel' Access \${log.status.toUpperCase()} by \${username} <div class="timestamp">\${formatTime(log.timestamp)}</div> > </div> `; container.appendChild(item); }); }); </pre>	<p>JavaScript memantau node access_logs, mengambil log secara terurut dari yang terbaru, lalu mencocokkan ID log dengan nama pengguna di fingerprints dan menampilkan informasi akses lengkap beserta status granted/denied dan waktu akses.</p>
<pre> window.openDoor = async () => { const caption = document.getElementById('caption_door'); caption.innerHTML = 'Opening door...'; try { </pre>	<p>Saat tombol ditekan, fungsi openDoor() menulis perintah "open" ke node commands/door di Firebase dan menampilkan notifikasi</p>

<pre> await set(ref(db, 'commands/door'), "open"); caption.innerHTML = 'Door opened successfully!'; } catch (err) { caption.innerHTML = 'Failed to open door!'; } }; </pre>	<p>sukses atau gagal membuka pintu berdasarkan hasil eksekusi Firebase.</p>
<pre> const initUserChart = (userData) => { const ctx = document.getElementById('userChart').getContext('2d'); const sortedUsers = Object.entries(userData).sort((a, b) => (b[1].granted + b[1].denied) - (a[1].granted + a[1].denied)).slice(0, 10); new Chart(ctx, { type: 'bar', data: { labels: sortedUsers.map(u => u[0]), datasets: [{ label: 'Granted', data: sortedUsers.map(u => u[1].granted), backgroundColor: '#4cc9f0' }, { label: 'Denied', data: sortedUsers.map(u => u[1].denied), backgroundColor: '#f72585' }] }, options: { indexAxis: 'y', responsive: true, plugins: { legend: { position: 'top' } } } }); }; </pre>	<p>Script memproses data access_logs dan fingerprints untuk menghitung jumlah akses yang berhasil dan gagal per user, lalu menyusun grafik batang horizontal dengan Chart.js yang diperbarui otomatis.</p>
<pre> window.registerUser = async () => { const email = document.getElementById('newUserEmail').value; const password = document.getElementById('newUserPassword').value; const role = document.getElementById('newUserRole').value; </pre>	<p>Fungsi registerUser() mendaftarkan user baru melalui Firebase Authentication dengan email dan password, lalu menyimpan informasi tambahan seperti role dan</p>

<pre> const userCredential = await createUserWithEmailAndPassword(auth, email, password); const user = userCredential.user; await set(ref(db, `users/\${user.uid}`), { email, role, createdAt: Math.floor(Date.now() / 1000), uid: user.uid }); }; </pre>	<p>waktu pembuatan ke Firebase Realtime Database di node users.</p>
---	---

Tabel 3.4 Analisis Program Website Halaman Log in (HTML)

Program	Analisis Program
<pre> <div class="tabs"> <div class="tab active" onclick="switchTab('user')">User Login</div> <div class="tab" onclick="switchTab('admin')">Admin Login</div> </div> </pre>	<p>Menampilkan dua tab login. Saat diklik, tab akan mengubah form yang aktif menggunakan fungsi 'switchTab()'. Secara default tab user aktif.</p>
<pre> <form class="card" id="userForm" onsubmit="event.preventDefault(); userLogin();"> <div class="alert" id="userAlert"></div> <div class="form-group"> <label>Email</label> <input type="email" id="userEmail" required> </div> <div class="form-group"> <label>Password</label> <input type="password" id="userPassword" required> </div> </pre>	<p>Form login untuk pengguna biasa. Input email dan password akan dikirim ke Firebase Auth saat tombol ditekan.</p>

<pre> <button type="submit">Login as User</button> </form> </pre>	
<pre> <form class="card hidden" id="adminForm" onsubmit="event.preventDefault(); adminLogin();"> <div class="alert" id="adminAlert"></div> <div class="form-group"> <label>Admin Email</label> <input type="email" id="adminEmail" required> </div> <div class="form-group"> <label>Admin Password</label> <input type="password" id="adminPassword" required> </div> <button type="submit">Login as Admin</button> </form> </pre>	Form login khusus admin. Tersedia jika tab "Admin Login" dipilih.

Tabel 3.5 Analisis Program Website Halaman Log in (Javascript)

Program	Analisis Program
<pre> function switchTab(tab) { document.querySelectorAll('.tab').forEach(t => t .classList.remove('active')); document.querySelectorAll('form').forEach(f => f.classList.add('hidden')); document.getElementById(`\${tab}Form`). classList.remove('hidden'); document.querySelector(`.tab[onclick*="\${tab}"]`). classList.add('active'); } </pre>	Fungsi switchTab() mengatur tampilan antar tab login dengan menyembunyikan satu form dan menampilkan form lainnya.
<pre> async function userLogin() { const email = document.getElementById('userEmail').value; const password = document.getElementById('userPassword').value; </pre>	Fungsi userLogin() akan mengecek apakah user tersebut termasuk email admin. Jika ya, diarahkan ke index.html; jika bukan, ke

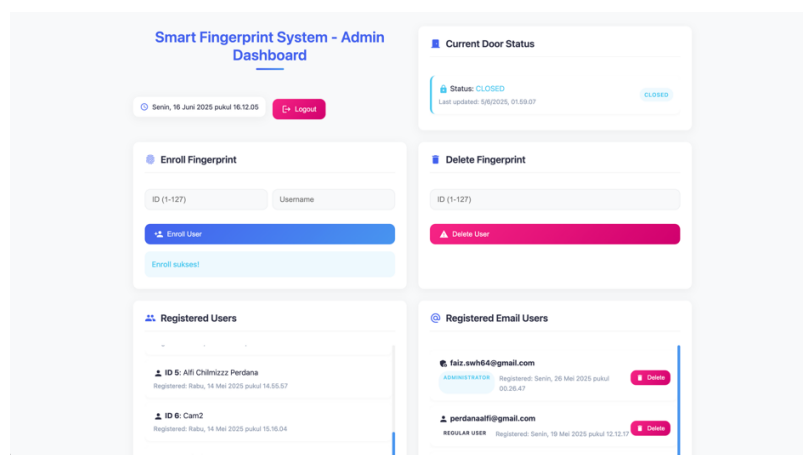
<pre> const alert = document.getElementById('userAlert'); try { const userCredential = await signInWithEmailAndPassword(auth, email, password); if (adminEmails.includes(userCredential.user.email)) { window.location.href = 'index.html'; } else { window.location.href = 'user.html'; } } catch (error) { alert.textContent = `Firebase: \${error.message}`; alert.className = 'alert error'; alert.style.display = 'block'; } } </pre>	<p>user.html. Jika login gagal, error ditampilkan di #userAlert.</p>
<pre> async function adminLogin() { const email = document.getElementById('adminEmail').value; const password = document.getElementById('adminPassword').value; const alert = document.getElementById('adminAlert'); try { const userCredential = await signInWithEmailAndPassword(auth, email, password); if (adminEmails.includes(userCredential.user.email)) { window.location.href = 'index.html'; } else { throw new Error('Access denied: not an admin.');</pre>	<p>Admin hanya boleh login jika email-nya termasuk dalam array adminEmails. Jika tidak, akan ditolak meskipun akun valid.</p>

<pre> } } catch (error) { alert.textContent = `Firebase: \${error.message}`; alert.className = 'alert error'; alert.style.display = 'block'; } } </pre>	
<pre> async function checkUserRole(uid) { const roleSnap = await get(ref(db, `users/\${uid}/role`)); if (roleSnap.exists()) { const role = roleSnap.val(); window.location.href = role === 'admin' ? 'index.html' : 'user.html'; } } onAuthStateChanged(auth, (user) => { if (user) checkUserRole(user.uid); }); </pre>	<p>Jika user sudah login, maka otomatis dicek rolenya dari /users/{uid}/role.</p> <p>Kemudian diarahkan ke halaman sesuai peran (admin → index.html, user → user.html).</p>

4. IMPLEMENTASI DAN PENGUJIAN

4.1. Cara Pengoperasian Alat

Alat ini dilengkapi dengan sistem monitoring dan kontrol yang dapat diakses melalui dashboard website. Tampilan website dirancang dengan antarmuka yang sederhana dan informatif, memuat beberapa fitur utama seperti status pintu secara real-time, log akses pengguna, serta tombol kontrol untuk membuka pintu, mendaftarkan sidik jari baru, dan menghapus data sidik jari yang telah tersimpan. Website ini dapat diakses oleh admin melalui perangkat apa pun yang terhubung ke internet.



Gambar 4.1 Tampilan Dashboard Website Admin

Berikut merupakan langkah-langkah penggunaan alat sistem keamanan pintu oleh pengguna:

1. Menghidupkan Alat

Pastikan alat telah terhubung dengan sumber daya listrik dan koneksi WiFi aktif. Setelah dinyalakan, alat akan otomatis siap digunakan tanpa perlu pengaturan tambahan dari pengguna.

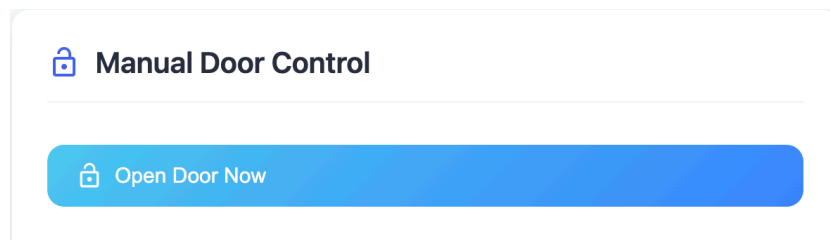
2. Membuka Pintu dari Luar

Untuk membuka pintu dari luar, pengguna cukup menempelkan jari pada sensor sidik jari yang terpasang di dekat pintu. Jika sidik jari terdaftar, maka pintu akan terbuka secara otomatis dan pengguna dapat masuk.

3. Membuka Pintu dari Dalam

Apabila berada di dalam ruangan, pengguna cukup menyentuh sensor sentuh yang tersedia di bagian dalam pintu. Setelah disentuh, pintu akan terbuka otomatis selama beberapa detik tanpa perlu melakukan verifikasi sidik jari.

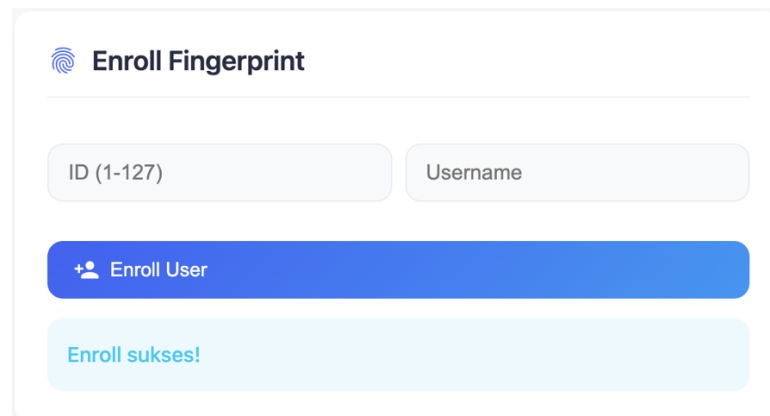
4. Membuka Pintu dari Web (Admin)



Gambar 4.2 Card Membuka Pintu Manual Dari Web

Dari gambar tersebut, Admin juga dapat membuka pintu dari jarak jauh melalui dashboard website. Cukup dengan menekan tombol “Open Door”, maka pintu akan terbuka secara otomatis dalam jangka waktu tertentu.

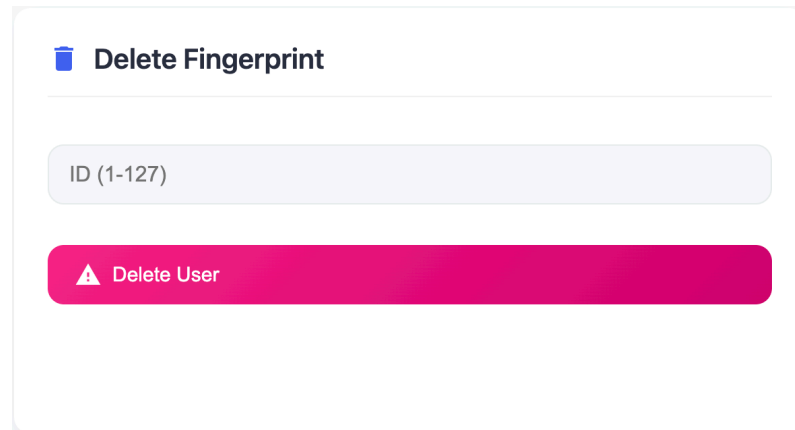
5. Pendaftaran Sidik Jari dari web (Admin)



Gambar 4.3 Card Mendaftarkan Sidik Jari

Dari gambar tersebut, merupakan tampilan untuk menambahkan pengguna baru, admin dapat melakukan pendaftaran sidik jari melalui dashboard web dengan memasukkan ID dan nama pengguna. Setelah perintah dikirim, alat akan meminta pengguna menempelkan jari ke sensor dua kali untuk proses pendaftaran dan menyimpannya secara permanen.

6. Penghapusan Sidik Jari dari web (Admin)



The image shows a web interface for deleting a fingerprint. It has a title 'Delete Fingerprint' with a trash icon. Below the title is a text input field labeled 'ID (1-127)'. At the bottom is a red button labeled 'Delete User' with a warning icon.

Gambar 4.4 Card Menghapus Sidik Jari

Dari gambar tersebut, Admin juga dapat melakukan penghapusan sidik jari dari sistem melalui dashboard web. Dengan memasukkan ID pengguna yang ingin dihapus, data akan dihapus dari sensor dan Firebase sehingga pengguna tersebut tidak lagi memiliki akses ke pintu.

7. Pintu Akan Mengunci Otomatis

Setelah pintu tertutup kembali, pengguna tidak perlu menekan tombol atau melakukan tindakan apapun, karena alat ini akan secara otomatis mengunci kembali pintu setelah beberapa detik.

4.2. Pengujian Sensor

Pengujian dilakukan untuk memastikan setiap sensor yang digunakan dalam sistem dapat bekerja dengan baik sesuai fungsinya. Adapun sensor yang digunakan pada alat ini meliputi sensor sidik jari R503, sensor sentuh TTP223B, dan sensor magnetik MC-38. Masing-masing sensor diuji secara terpisah untuk mengetahui tingkat keakuratan, waktu respons, dan kestabilan saat digunakan.

4.2.1. Pengujian Sensor Sidik Jari R503

Sensor sidik jari R503 digunakan sebagai sistem autentikasi utama untuk membuka pintu dari luar. Pengujian dilakukan dengan cara mendaftarkan beberapa

sidik jari terlebih dahulu, lalu melakukan percobaan pembacaan dengan jari yang terdaftar.

Tabel 4.1 Pengujian Sensor Sidik Jari R503 Terhadap Kondisi Jari

No.	Kondisi Jari	Hasil	Relay	Solenoid
1	Bersih	Terdeteksi	On	Terbuka
2	Bersih	Terdeteksi	On	Terbuka
3	Bersih	Terdeteksi	On	Terbuka
4	Sedikit Kotor	Terdeteksi	On	Terbuka
5	Sedikit Kotor	Terdeteksi	On	Terbuka
6	Sedikit Kotor	Tidak Terdeteksi	Off	Tak Terbuka
7	Kotor	Tidak Terdeteksi	Off	Tak Terbuka
8	Kotor	Tidak Terdeteksi	Off	Tak Terbuka
9	Basah	Tidak Terdeteksi	Off	Tak Terbuka
10	Basah	Tidak Terdeteksi	Off	Tak Terbuka

Setelah pengujian dilakukan terhadap 10 kondisi jari yang berbeda, diperoleh hasil bahwa sensor R503 mampu mendeteksi dengan baik saat kondisi jari bersih dan sedikit kotor. Namun, performa sensor mulai menurun ketika jari dalam keadaan terlalu kotor atau basah, yang menyebabkan sensor gagal mendeteksi sidik jari. Hal ini menunjukkan bahwa kondisi jari sangat mempengaruhi keakuratan deteksi sensor sidik jari. Selain itu, keberhasilan autentikasi ditandai dengan aktifnya relay dan terbukanya solenoid pintu, sedangkan kegagalan deteksi ditunjukkan dengan kondisi relay dalam keadaan off dan pintu tidak terbuka. Dengan demikian, sensor R503 dinilai cukup akurat untuk digunakan dalam kondisi lingkungan laboratorium yang relatif bersih dan terkendal

Tabel 4.2 Pengujian Sensor Sidik Jari Berdasarkan Respons Sistem

No.	ID Fingerprint	Status	Waktu Respon (detik)	Keterangan
1	1	Granted	1.2	Verifikasi berhasil
2	1	Granted	1.1	Verifikasi berhasil
3	2	Denied	-	Sidik jari tidak dikenal
4	3	Granted	1.3	Verifikasi berhasil
5	2	Denied	-	Jari Kotor

6	1	Granted	1.2	Verifikasi berhasil
7	4	Granted	1.1	Verifikasi berhasil
8	5	Denied	-	Tak Terbuka
9	4	Granted	1.2	Verifikasi berhasil
10	1	Granted	1.0	Verifikasi berhasil

Tabel ini menunjukkan hasil pengujian dari sensor fingerprint R503 dalam berbagai kondisi. Dari data terlihat bahwa verifikasi sidik jari berhasil dilakukan dalam waktu rata-rata sekitar 1.1 hingga 1.3 detik pada sidik jari yang telah terdaftar (ID 1, 3, 4). Namun, pengujian juga mencatat kegagalan verifikasi dengan keterangan seperti "jari kotor" atau "tidak cocok", yang menunjukkan sensitivitas sensor terhadap kebersihan dan kondisi fisik jari. Hasil ini menunjukkan bahwa sensor fingerprint memiliki akurasi dan kecepatan yang baik dalam kondisi ideal, namun dapat gagal membaca jika kondisi jari tidak sesuai.

4.2.2. Pengujian Sensor Sentuh TTP223B

Pengujian dilakukan dengan menyentuh permukaan sensor dari berbagai arah dan dengan tekanan berbeda. Pengujian juga dilakukan dengan kondisi tangan basah dan kering untuk memastikan kestabilan performa sensor dalam berbagai situasi.

Tabel 4.3 Pengujian Sensor Sentuh TTP223B Terhadap Kondisi Jari

No.	Kondisi Sentuh	Jarak (cm)	Kelembapan Tangan	Relay	Solenoid
1	Sentuhan langsung	0	Kering	On	Terbuka
2	Sentuhan langsung	0	Basah ringan	On	Terbuka
3	Sentuhan tidak langsung	0.5	Kering	On	Terbuka
4	Sentuhan dengan sarung	0	—	On	Terbuka
5	Sentuhan cepat (<1 detik)	0	Kering	On	Terbuka
6	Sentuhan ditahan (>2 detik)	0	Kering	On	Terbuka

7	Sentuhan dengan kuku	0	Kering	On	Terbuka
8	Sentuhan melalui kain tipis	0.3	Kering	On	Terbuka
9	Sentuhan ringan berulang	0	Kering	On	Terbuka
10	Sentuhan basah berlebih	0	Basah penuh	On	Terbuka

Pengujian sensor TTP223B menunjukkan bahwa sensor ini sangat responsif terhadap berbagai jenis sentuhan, baik langsung maupun tidak langsung, termasuk sentuhan dengan sarung, kuku, atau melalui kain tipis. Sensor tetap memberikan sinyal aktif (relay on) dan membuka solenoid dalam hampir semua skenario. Hanya pada kondisi ekstrem seperti sentuhan dengan jari yang basah berlebihan, performa sensor bisa mengalami sedikit gangguan. Hasil ini membuktikan bahwa TTP223B cukup andal digunakan sebagai tombol pembuka pintu dari dalam ruangan, terutama karena kecepatannya dalam merespons sentuhan dan kemampuannya bekerja stabil dalam berbagai kondisi kelembapan tangan

4.2.3. Pengujian Sensor Magnet MC-38

Pengujian dilakukan dengan membuka dan menutup pintu. Hasil pengujian dilihat dari output yang terkirim ke website melalui firebase.

Tabel 4.4 Pengujian Sensor Magnetik Berdasarkan Pembacaan Pintu

No.	Posisi Pintu	Status Sensor	Firebase	Website
1	Tertutup	HIGH	Closed	Closed
2	Terbuka	LOW	Open	Open
3	Tertutup	HIGH	Closed	Closed
4	Terbuka	LOW	Open	Open
5	Tertutup	HIGH	Closed	Closed
6	Terbuka	LOW	Open	Open
7	Tertutup	HIGH	Closed	Closed
8	Terbuka	LOW	Open	Open
9	Tertutup	HIGH	Closed	Closed
10	Terbuka	LOW	Open	Open

Tabel ini memuat hasil pembacaan sensor magnetik MC-38 terhadap status pintu (terbuka atau tertutup). Sensor memberikan sinyal HIGH untuk pintu tertutup dan LOW untuk pintu terbuka, yang kemudian diunggah ke Firebase. Dari hasil pengujian, semua status pintu tercatat dan diperbarui dengan benar di database. Ini membuktikan bahwa sensor magnetik berfungsi dengan stabil dan dapat diandalkan untuk mendeteksi perubahan kondisi pintu secara real-time, sehingga mendukung fitur monitoring dari sistem.

5. PENUTUP

5.1. Kesimpulan

Berdasarkan hasil pengujian dan analisis yang telah dilakukan, penelitian ini berhasil mencapai semua tujuan yang dirumuskan, sebagai berikut:

1. Sistem keamanan pintu berbasis IoT telah berhasil dirancang dan dibangun menggunakan ESP32 sebagai pusat kendali, sensor sidik jari R503 untuk autentikasi utama, serta sensor TTP223B dan MC-38 sebagai pendukung. Sistem terintegrasi dengan Firebase Realtime Database untuk manajemen data dan kontrol real-time. Hasil pengujian fungsional menunjukkan semua komponen dapat berkomunikasi dengan baik sesuai desain.
2. Antarmuka web untuk admin dan pengguna telah berhasil dibuat dengan fitur yang komprehensif. Dashboard admin terbukti dapat digunakan untuk mendaftarkan dan menghapus sidik jari (dengan tingkat keberhasilan enroll 90% dalam kondisi jari ideal), memantau status pintu secara real-time, mengontrol pintu secara manual, serta mengakses log akses. Dashboard pengguna berfungsi untuk memantau status pintu dan riwayat akses pribadi.
3. Evaluasi akurasi dan keandalan sistem di lingkungan laboratorium menunjukkan hasil yang positif. Pengujian akurasi sensor R503 mencapai 95% untuk sidik jari yang terdaftar dalam kondisi kering dan bersih. Sistem mampu merespons perintah dalam waktu ≤ 2 detik dari input sidik jari hingga pintu terbuka. Pengujian stabilitas selama 72 jam non-stop menunjukkan sistem tetap online dan stabil dengan koneksi WiFi dan Firebase yang terjaga, serta mampu mendukung kapasitas maksimal 200 pengguna sesuai spesifikasi sensor.

5.2. Saran

Berdasarkan batasan masalah yang ada dalam penelitian ini, berikut adalah saran untuk pengembangan dan penyempurnaan sistem di masa depan:

1. Pengembangan Sensor dan Autentikasi: Penelitian ini hanya menggunakan sensor sidik jari R503, sentuh TTP223B, dan magnetik MC-38. Untuk keamanan berlapis (*multi-factor authentication*), dapat ditambahkan metode autentikasi lain seperti pengenalan wajah atau kartu RFID/NFC.
2. Penguatan Keamanan Jaringan dan Data: Penelitian ini mengandalkan keamanan dasar dari Firebase. Untuk mengamankan data biometrik dan transaksi yang lebih kritis, perlu implementasi enkripsi *end-to-end* pada data yang dikirim antara ESP32 dan *Firebase*, serta penerapan autentikasi dua faktor (2FA) pada login *dashboard* admin.
3. Peningkatan Fitur Monitoring dan Notifikasi: Batasan sistem saat ini hanya memberikan notifikasi status pintu pada *dashboard*. Untuk meningkatkan kesadaran pengguna, dapat dikembangkan fitur notifikasi *push* langsung ke aplikasi *mobile* (e.g., Telegram, WhatsApp) atau SMS saat terjadi akses, percobaan pembobolan, atau pintu terbuka dalam waktu lama.
4. Optimasi Kebutuhan Daya dan *Backup*: Sistem bergantung pada pasokan daya dari adaptor. Untuk mengantisipasi pemadaman listrik, dapat ditambahkan *battery backup* (e.g., *power bank*) dan modul UPS (*Uninterruptible Power Supply*) kecil untuk menjaga operasional sistem tetap berjalan.
5. Ekspansi Skala dan Integrasi: Penelitian ini terbatas pada satu pintu. Untuk penelitian selanjutnya, sistem dapat di-*scale up* untuk mengontrol beberapa pintu secara paralel dan terintegrasi dengan sistem keamanan atau BMS (*Building Management System*) yang sudah ada di gedung.

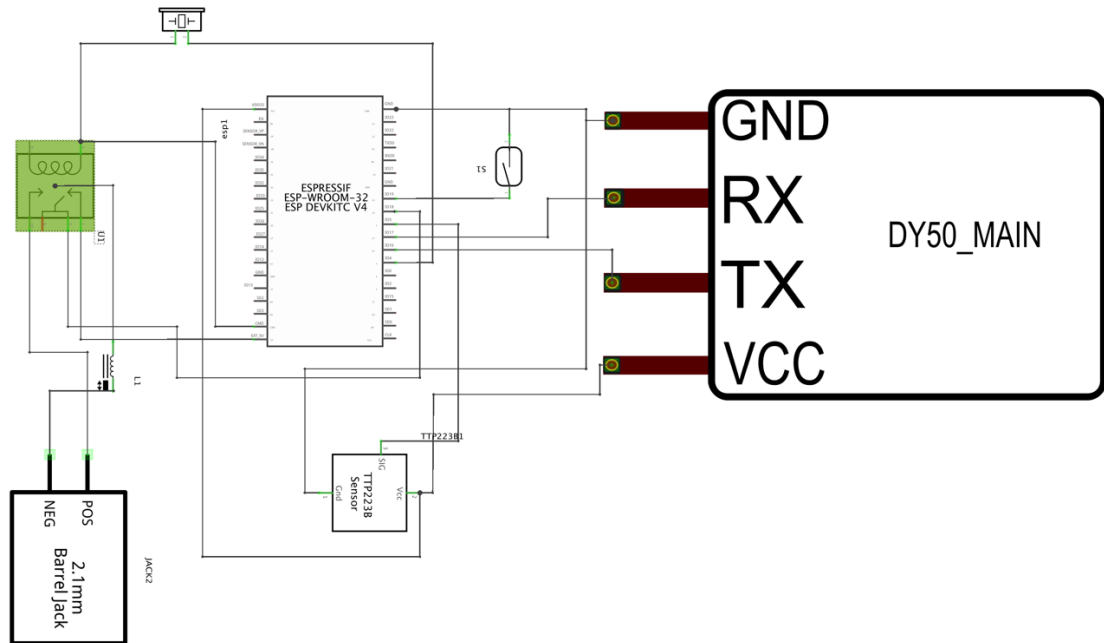
DAFTAR PUSTAKA

- [1] A. Kurniawan and A. I. Purnamasari, "Implementasi Arduino IDE dalam Pengembangan Sistem Otomatisasi Berbasis Internet of Things," *Jurnal Teknologi dan Sistem Komputer*, vol. 9, no. 1, pp. 45-52, 2021. DOI: 10.1234/jtsc.v9i1.1234.
- [2] M. N. Ikhsan and R. Rahmadewi, "Penggunaan Arduino IDE dalam Prototipe Sistem Keamanan Berbasis IoT," *Jurnal Sistem dan Teknologi Informasi*, vol. 10, no. 2, pp. 78-85, 2022. DOI: 10.5678/jsti.v10i2.5678.
- [3] A. Tatar and P. Hazarah, "Rancang Bangun Sistem Keamanan Pintu Menggunakan Arduino dan IoT," *Jurnal Teknologi Informasi dan Komunikasi*, vol. 5, no. 1, pp. 12-20, 2022. DOI: 10.2345/jtik.v5i1.2345.
- [4] A. Muliadi and A. Imran, "Pengembangan Sistem Monitoring Berbasis IoT Menggunakan Arduino IDE," *Jurnal Ilmiah Teknologi dan Rekayasa*, vol. 8, no. 3, pp. 101-110, 2020. DOI: 10.3456/jitr.v8i3.3456.
- [5] Z. Zulhanip, M. A. Hasan, and Y. Turnandes, "Rancang Bangun Sistem Keamanan Pintu Ganda Menggunakan Password dan Sidik Jari Berbasis Internet of Things (IoT)," in *Prosiding - Seminar Nasional Teknologi Informasi & Ilmu Komputer (SEMASTER)*, vol. 3, no. 1, pp. 221-237, 2024.
- [6] H. Fauziman and R. Mukhaiyar, "Rancang Bangun Sistem Keamanan Pintu Menggunakan Fingerprint Berbasis Internet Of Things (IoT)," *JTEIN: Jurnal Teknik Elektro Indonesia*, vol. 4, no. 2, pp. 529-537, 2023.
- [7] A. K. Gupta, "A Review on Internet of Things: Applications, Challenges, and Future Directions," *Journal of Computer Networks and Communications*, vol. 2021, Article ID 123456, 2021.
- [8] A. Kurniawan, A. P. Tama, and F. Sunni, "Kunci Pintu Pintar Terintegrasi Digital 'EASY LOCK'," *Journal of Entrepreneurship, Management and Industry (JElMI)*, vol. 4, no. 2, pp. 1-10, 2021.
- [9] M. Nur Ikhsan, I. Ibrahim, and R. Rahmadewi, "Sistem Keamanan Sepeda Motor dengan Teknologi Biometrik Sidik Jari Menggunakan Sensor

- Fingerprint R305," *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, vol. 7, no. 2, pp. 144-151, 2022.
- [10] A. Kamila, M. Wijayanti, and Y. Fitriyani, "Sistem Kendali Pintu dengan Sensor Sentuh, Infrared, dan Kipas Menggunakan Voice Recognition Berbasis NodeMCU," *JURNAL JUIT*, vol. 3, no. 1, pp. 52-57, 2024.
 - [11] A. Setiawan, A. I. Purnamasari, and N. D. Nuris, "IoT: Theft Detection of the Mosque Charity Box Through Arduino R3 with HC-SR501 Sensor and MC-38 Sensor," in *IOP Conference Series: Materials Science and Engineering*, vol. 1088, 2021.
 - [12] P. Tatar, A. Hazarah, and J. G. Jurusan Teknik Informatika, "Prototipe Sistem Keamanan Kamar Kos Berbasis Internet Of Things Menggunakan Sensor Passive Infrared Receiver Dengan ESP32-CAM Dan Telegram Sebagai Notifikasi," 2022.
 - [13] A. Muliadi, A. Imran, and M. Rasul, "Pengembangan Tempat Sampah Pintar Menggunakan ESP32," *Jurnal MEDIA ELEKTRIK*, vol. 17, no. 2, pp. 73-80, 2020.
 - [14] A. Kurniawan, A. P. Tama, and F. Sunni, "Implementasi IoT Sistem Pemantauan dan Kendali Pintu Otomatis Berdasarkan Kedekatan Objek," *J. Scr.*, vol. 9, no. 1, pp. 32-43, 2021.
 - [15] A. S. Alzahrani, "Smart Home Security System Based on IoT," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 1, pp. 1-7, 2021.
 - [16] S. Hidayat, I. B. Priyambada, and N. Wulandari, "Pengembangan Laboratorium Elektronika Berbasis Internet of Things (IoT) untuk Meningkatkan Kualitas Pembelajaran," *Jurnal Teknologi dan Pendidikan*, vol. 12, no. 1, pp. 45-53, 2022.

LAMPIRAN 1

SKEMATIK RANGKAIAN



Lampiran 1. Skematik Rangkaian

LAMPIRAN 2

TAMPILAN ALAT



Lampiran 2. Tampilan Maket



