

**SISTEM *MONITORING KEAMANAN RUMAH MENGGUNAKAN  
FINGERPRINT DAN FACE RECOGNITION BERBASIS  
MACHINE LEARNING***

**SKRIPSI**

Karya Tulis sebagai syarat memperoleh  
Gelar Sarjana Komputer dari Fakultas Teknologi Informasi  
Universitas Bale Bandung

Disusun Oleh:

SITI ALLANURIN

NPM : 301210002



PROGRAM STRATA 1  
PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS BALE BANDUNG  
BANDUNG

2025

**LEMBAR PERSETUJUAN PEMBIMBING**

**SISTEM MINITORING KEAMANAN RUMAH MENGGUNAKAN  
FINGERPRINT DAN FACE RECOGNITION BERBASIS  
MACHINE LEARNING**

Disusun Oleh:

**SITI ALLANURIN**

NPM : 301210002

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar  
**SARJANA KOMPUTER**

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS BALE BANDUNG**

Baleendah, Agustus 2025

Disetujui oleh:

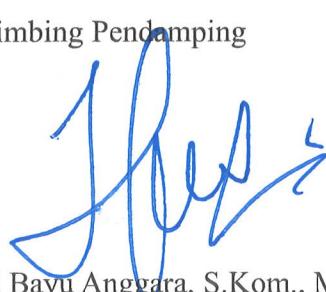
Pembimbing Utama



Yusuf Muharam, S.Kom., M.Kom.

NIK. 04104820003

Pembimbing Pendamping



Mohammad Bayu Anggara, S.Kom., M.Kom.

NIK. 04104823002

**LEMBAR PERSETUJUAN PENGUJI**

**SISTEM MINITORING KEAMANAN RUMAH MENGGUNAKAN  
FINGERPRINT DAN FACE RECOGNITION BERBASIS  
MACHINE LEARNING**

Disusun Oleh:

**SITI ALLANURIN**

NPM : 301210002

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar  
**SARJANA KOMPUTER**

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS BALE BANDUNG**

Baleendah, Agustus 2025

Disetujui oleh:

Pengaji 1



Yudi Herdiana, S.T., M.T.

NIK. 04104808008

Pengaji 2



Cecep Suwanda, S.Si., M.Kom

NIK. 3273110210820005

**LEMBAR PENGESAHAN PROGRAM STUDI**

**SISTEM MINITORING KEAMANAN RUMAH MENGGUNAKAN  
FINGERPRINT DAN FACE RECOGNITION BERBASIS  
MACHINE LEARNING**

Disusun Oleh:

**SITI ALLANURIN**

NPM : 301210002

SKRIPSI ini telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar

**SARJANA KOMPUTER**

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS BALE BANDUNG**

Baleendah, Agustus 2025

Mengetahui,

Dekan



Yudi Herdiana, S.T., M.T.

NIK. 04104808008

Mengetahui,

Ketua Program Studi

Yusuf Muhararni, S.Kom.,M. Kom.

NIK. 04104820003

## HALAMAN PERNYATAAN

Yang bertanda tangan dibawah ini, saya:

Nama : Siti Allanurin  
NIM : 301210002  
Jurusan : Teknik Informatika  
Fakultas : Fakultas Teknologi Informasi  
Judul : Sistem Monitoring Keamanan Rumah  
Menggunakan *Fingerprint* Dan *Face Recognition* Berbasis *Machine Learning*

Dengan ini, penulis menyatakan bahwa penulisan skripsi ini merupakan hasil dari penelitian, pemikiran, dan pemaparan asli penulis sendiri, baik dalam bentuk naskah laporan maupun program pemrograman yang menjadi bagian dari skripsi ini. Apabila *user* an terdapat karya orang lain, penulis telah mencantumkan sumbernya secara jelas dan sesuai dengan kaidah penulisan ilmiah yang berlaku.

Baleendah, Agustus 2025



Siti Allanurin

NPM. 301210002

## ABSTRAK

Keamanan rumah merupakan aspek penting dalam membatasi dan mengontrol akses masuk secara efektif. Banyak kasus pembobolan rumah yang terjadi akibat penggunaan sistem kunci konvensional yang belum mampu memberikan jaminan keamanan yang efektif. Penggunaan teknologi biometrik *Fingerprint* dan *Face Recognition* dengan *Machine Learning* punya keunggulan dalam hal kepraktisan dan akurasi tinggi.

Perancangan Sistem *Monitoring* keamanan rumah ini mencakup integrasi berbagai komponen utama seperti sensor *Fingerprint*, kamera CCTV, NodeMCU, dan *Solenoid Door Lock* sebagai kunci otomatis. Pada *Fingerprint* digunakan sensor AS806 yang menerapkan *algoritma minutiae based matching*, serta *Machine Learning* untuk *Face Recognition* yang dikembangkan dengan Python serta kontrol tampilan web dikembangkan menggunakan *Tailwind CSS* guna memberikan tampilan yang *modern, responsif*, dan ramah pengguna. Sistem ini dilengkapi dengan notifikasi melalui *WhatsApp* yang secara otomatis mengirimkan peringatan jika terdeteksi akses gagal atau aktivitas yang mencurigakan. Dengan fitur tersebut sistem ini diharapkan dapat meningkatkan keamanan rumah dengan akurasi tinggi, efisien, dan adaptif terhadap perkembangan teknologi. Proses implementasi sistem mengikuti pendekatan metode *Waterfall* yang meliputi tahapan analisis kebutuhan, perancangan sistem, implementasi perangkat keras dan perangkat lunak, serta pengujian sistem.

Hasil pengujian *black box* menunjukkan prototipe berfungsi sesuai spesifikasi. Sensor *fingerprint* akurat >90%, *face recognition* optimal pada jarak 30 cm dengan keberhasilan 90% dan menurun pada cahaya rendah dan penggunaan masker. *Relay, solenoid, buzzer, pushbutton*, serta dashboard web (*registrasi, monitoring CCTV, log aktivitas*) berjalan baik. Notifikasi WhatsApp terkirim 100% dengan *delay* ±1 detik. Secara keseluruhan, sistem berfungsi optimal sebagai solusi keamanan rumah *modern*.

**Kata Kunci:** *Face Recognition, Fingerprint, Internet of Things, Machine Learning, Sistem monitoring.*

## ***ABSTRACT***

*Home security is a crucial aspect in effectively limiting and controlling access. Many burglary cases occur due to the use of conventional locking systems that cannot provide an adequate level of protection. The use of biometric technologies such as fingerprint and face recognition with machine learning offers advantages in terms of practicality and high accuracy.*

*The design of this home security monitoring system integrates several main components, including a fingerprint sensor, CCTV camera, NodeMCU, and solenoid door lock as an automatic lock. The fingerprint module uses the AS608 sensor with a minutiae-based matching algorithm, while face recognition is developed using Python-based machine learning. The web dashboard is implemented with Tailwind CSS to provide a modern, responsive, and user-friendly interface. The system is also equipped with WhatsApp notifications that automatically send alerts when failed access attempts or suspicious activities are detected. With these features, the system is expected to improve home security with high accuracy, efficiency, and adaptability to technological developments. The implementation process follows the Waterfall methodology, consisting of requirements analysis, system design, hardware and software implementation, and system testing.*

*Black-box testing results show that the prototype functions according to specifications. The fingerprint sensor achieved accuracy above 90%, while face recognition performed optimally at a 30 cm distance with 90% success but decreased under low-light conditions and when users wore masks. Relay, solenoid, buzzer, push button, and the web dashboard (registration, CCTV monitoring, activity logs) all functioned properly. WhatsApp notifications were delivered with 100% success and an average delay of  $\pm 1$  second. Overall, the system operates optimally as a modern home security solution.*

***Keywords:*** Face Recognition, Fingerprint, Internet of Things, Machine Learning, Monitoring System

## KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kepada Allah SWT, yang telah memberikan rahmat serta karunia-Nya, sehingga penulis dapat menyelesaikan Skripsi ini tepat sesuai dengan waktunya. Tak lupa shalawat serta salam kita curahkan kepada junjungan Nabi Muhammad SAW, yang membawa kita dari zaman kegelapan menuju zaman penuh ilmu pengetahuan seperti sekarang ini. Atas izin Allah SWT, penulis dapat menyusun skripsi dengan judul "**SISTEM MINITORING KEAMANAN RUMAH MENGGUNAKAN FINGERPRINT DAN FACE RECOGNITION BERBASIS MACHINE LEARNING**". Skripsi ini disusun sebagai salah satu syarat untuk menyelesaikan pendidikan pada Program Studi Teknik Informatika. Dalam penyusunan laporan ini, penulis menyadari bahwa masih banyak kekurangan karena keterbatasan pengalaman dan pengetahuan. Oleh karena itu, saran dan kritik yang membangun dari semua pihak sangat penulis harapkan demi penyempurnaan skripsi ini di kemudian hari. Akhir kata, penulis mengucapkan terima kasih kepada semua pihak yang terlibat, karena penyusunan laporan ini tidak lepas dari bantuan dan dukungan dari banyak pihak, oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Allah SWT yang telah memberikan rahmat, kemudahan, serta kelancaran selama proses penelitian dan penulisan laporan ini.
2. Rasulullah SAW sebagai teladan hidup yang mulia, yang bimbingan sunnah dan ajarannya selalu menjadi pedoman penulis dalam beribadah, berakhhlak, serta menuntut ilmu hingga terselesaiannya laporan ini.
3. Kedua orang tua penulis yang selalu memberikan dukungan, doa, dan semangat yang tak ternilai harganya.
4. Bapak Yudi Herdiana, S.T., M.T., selaku Dekan Fakultas Teknologi Informasi Universitas Bale Bandung.
5. Bapak Yusuf Muharam, S. Kom., M.Kom, selaku Kepala Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Bale Bandung sekaligus pembimbing utama.

6. Bapak Mohammad Bayu Anggara, S.Kom., M.Kom., selaku pembimbing pendamping yang telah memberikan arahan dan masukan berharga.
7. Seluruh jajaran dosen dan staf Fakultas Teknologi Informasi Universitas Bale Bandung atas ilmu dan bimbingannya selama masa studi.
8. Sahabat saya, Riska Nurhayan, yang selalu mendampingi, membantu, dan menjadi tempat berbagi suka duka dari masa awal kuliah hingga tahap akhir penulisan skripsi ini.
9. Rekan-rekan Fakultas Teknologi Informasi angkatan 2021 yang senantiasa memberikan semangat, dukungan, dan bantuan selama proses penelitian dan penyusunan laporan skripsi ini.
10. Ucapan terima kasih juga penulis sampaikan kepada Anita Fitriani dan Yuniarti Wandari, sahabat terbaik di luar kampus. Terima kasih karena selalu mendukung dan bersedia mendengar keluh kesah penulis selama melewati masa sulit penyusunan skripsi ini.
11. Terakhir, untuk satu nama yang tidak bisa penulis sebutkan namanya, namun selalu hadir dalam ingatan. Terima kasih karena tanpa sadar menjadi semangat yang menyertai perjalanan penulis hingga titik ini. Tak terungkap namun tetap terkisah

Semoga skripsi ini dapat memberikan manfaat dan menjadi referensi yang berguna bagi pengembangan ilmu pengetahuan, khususnya di bidang *Internet of Things* dan keamanan sistem berbasis biometrik. Semoga upaya yang telah dilakukan dalam penyusunan Skripsi ini mendapatkan ridho dan keberkahan dari ALLAH SWT.

Baleendah, Agustus 2025

Penulis

## DAFTAR ISI

<b>ABSTRAK.....</b>	<b>vi</b>
<b>ABSTRACT .....</b>	<b>vii</b>
<b>KATA PENGANTAR.....</b>	<b>viii</b>
<b>DAFTAR ISI .....</b>	<b>x</b>
<b>DAFTAR GAMBAR.....</b>	<b>xiv</b>
<b>DAFTAR TABEL.....</b>	<b>xviii</b>
<b>DAFTAR LAMPIRAN .....</b>	<b>xx</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1    Latar Belakang .....	1
1.2    Rumusan Masalah .....	3
1.3    Batasan Masalah.....	3
1.4    Tujuan Penelitian.....	4
1.5    Metodologi Penelitian.....	4
1.6    Sistematika Penulisan .....	6
<b>BAB II TINJAUAN PUSTAKA.....</b>	<b>8</b>
2.1    Landasan Teori .....	8
2.2    Dasar Teori .....	10
2.2.1    Keamanan Rumah.....	11
2.2.2    Sistem <i>Monitoring</i> .....	11
2.2.3 <i>Fingerprint</i> .....	11
2.2.4 <i>Face Recognition</i> .....	12
2.2.5 <i>Machine Learning</i> .....	13
2.2.6    Model SDLC <i>Waterfall</i> .....	19

2.2.7	UML ( <i>Unified Modeling Language</i> ).....	21
2.2.8	<i>Internet of Things</i> (IoT).....	25
2.2.9	WhatsApp.....	25
2.2.10	ESP32.....	26
2.2.11	Sensor <i>Fingerprint AS608</i> .....	27
2.2.12	<i>Solenoid Door Lock</i> .....	28
2.2.13	<i>Push Button</i> (Tombol Tekan).....	29
2.2.14	Kamera CCTV 360 .....	30
2.2.15	<i>Relay Module</i> .....	31
2.2.16	<i>Power Supply 12V 2A</i> .....	31
2.2.17	Kabel <i>Micro USB</i> (untuk NodeMCU) .....	32
2.2.18	<i>Jumper wire</i> .....	33
2.2.19	<i>Base Plate ESP32</i> .....	34
2.2.20	<i>Website</i> .....	35
2.2.21	Arduino IDE .....	36
2.2.22	<i>Visual Studio Code</i> .....	37
2.2.23	<i>Python</i> .....	38
2.2.24	<i>Flask</i> .....	39
2.2.25	<i>MySQL</i> .....	40
2.2.26	<i>MQTT (Message Queuing Telemetry Transport)</i> .....	41
2.2.27	<i>Figma</i> .....	42
2.2.28	<i>Draw.io</i> .....	43
2.2.29	<i>Fritzing</i> .....	44
2.2.30	<i>HTML (Cascading Style Sheets)</i> .....	45
2.2.31	<i>CSS (Cascading Style Sheets)</i> .....	46
2.2.32	<i>Tailwind CSS</i> .....	47

2.2.33	<i>BlackBox Testing</i> .....	48
2.2.34	Aplikasi V380 Pr0 .....	49
<b>BAB III METODOLOGI PENELITIAN .....</b>		<b>50</b>
3.1	Kerangka Berfikir.....	50
3.2	Deskripsi .....	51
3.2.1	Identifikasi Masalah.....	51
3.2.2	Pengumpulan Data.....	51
3.2.3	Analisis Kebutuhan.....	51
3.2.4	Perancangan sistem.....	54
3.2.5	Implementasi .....	56
3.2.6	Pengujian.....	56
3.2.7	Pembuatan Laporan .....	56
<b>BAB IV ANALISIS DAN PERANCANGAN .....</b>		<b>57</b>
4.1	Analisis .....	57
4.1.1	Analisis Masalah.....	57
4.1.2	Analisis Kebutuhan.....	58
4.1.3	Analisis <i>User</i> .....	60
4.1.4	<i>Interface User</i> .....	60
4.1.5	Fitur-Fitur .....	61
4.1.6	Analisis Data .....	62
4.1.7	Analisis Biaya.....	62
4.2	Perancangan .....	63
4.2.1	Blok Diagram Rancangan Alat <i>Monitoring</i> Keamanan Rumah ...	63
4.2.2	<i>Flowmap</i> Sistem <i>Monitoring</i> Keamanan Rumah .....	65
4.2.3	Perancangan Rangkaian Sistem.....	66
4.2.4	Perancangan Sistem UML.....	72

4.2.5	Perancangan Antarmuka <i>Dashboard</i> Web.....	99
<b>BAB V IMPLEMENTASI DAN PENGUJIAN.....</b>		<b>111</b>
5.1	Implementasi .....	111
5.1.1	Listing Program .....	111
5.1.2	Implementasi Modul Elektronika .....	116
5.1.4	Instalasi Sistem.....	118
5.1.5	Menjalankan Sistem.....	120
5.2	Pengujian <i>Prototipe</i> .....	130
<b>BAB VI KESIMPULAN .....</b>		<b>138</b>
6.1	Kesimpulan .....	138
6.2	Saran .....	139
<b>DAFTAR PUSTAKA .....</b>		<b>140</b>
<b>LAMPIRAN .....</b>		<b>147</b>

## DAFTAR GAMBAR

Gambar 2. 1 Arsitektur <i>Fingerprint</i> .....	12
Gambar 2. 2 Arsitektur <i>Face Recognition</i> .....	13
Gambar 2. 3 Operasi LBP .....	17
Gambar 2. 4 Proses Ekstrasi Histogram.....	18
Gambar 2. 5 Model SDLC <i>Waterfall</i> .....	20
Gambar 2. 6 Aplikasi WhatsApp.....	25
Gambar 2. 7 ESP32.....	26
Gambar 2. 8 Sensor <i>Fingerprint</i> AS608 .....	27
Gambar 2. 9 <i>Selenoid Door Lock</i> .....	28
Gambar 2. 10 <i>Push Button</i> (Tombol Tekan).....	29
Gambar 2. 11 Kamera CCTV 360 .....	30
Gambar 2. 12 <i>Relay Module</i> .....	31
Gambar 2. 13 <i>Power Supply</i> 12V 2A.....	32
Gambar 2. 14 Kabel <i>Micro USB</i> (untuk NodeMCU).....	33
Gambar 2. 15 Jumper Wire .....	33
Gambar 2. 16 <i>Base Plate</i> ESP32 .....	34
Gambar 2. 17 Arsitektur Website.....	35
Gambar 2. 18 Aplikasi Arduino IDE .....	36
Gambar 2. 19 Aplikasi Visual Studio Code.....	37
Gambar 2. 20 Arsitektur Python.....	38
Gambar 2. 21 Arsitektur <i>Flask</i> .....	39
Gambar 2. 22 Arsitektur MySQL .....	40
Gambar 2. 23 Arsitektur MQTT .....	41
Gambar 2. 24 Aplikasi Figma.....	42
Gambar 2. 25 Aplikasi Draw.io .....	43
Gambar 2. 26 Aplikasi Fritzing .....	44
Gambar 2. 27 Arsitektur HTML .....	45
Gambar 2. 28 Arsitektur CSS .....	46
Gambar 2. 29 Arsitektur <i>Talwind</i> CSS.....	47
Gambar 2. 30 Arsitektur <i>BlackBox Testing</i> .....	48

Gambar 2. 31 Aplikasi V380 Pro.....	49
Gambar 3. 1 Kerangka Berfikir .....	50
Gambar 4. 1 Flowmap Akses Pintu Konvensional .....	58
Gambar 4. 2 Blok diagram perancangan alat monitoring keamanan rumah.....	63
Gambar 4. 3 Flowmap Sistem Monitoring Keamanan Rumah .....	65
Gambar 4. 4 Skema Rangkaian Koneksi Sensor .....	66
Gambar 4. 5 Implementasi Wiring Sensor Fingerprint AS608 ke ESP32 .....	67
Gambar 4. 6 Skema Rangkaian LCD dengan ESP32 .....	67
Gambar 4. 7 Implementasi Wiring LCD ke ESP32 .....	68
Gambar 4. 8 Skema Rangkaian PushButton terhubung dengan ESP32 .....	68
Gambar 4. 9 Implementasi <i>Wiring</i> LCD ke ESP32 .....	69
Gambar 4. 10 Skema Rangkaian <i>Buzzer</i> terhubung dengan ESP32.....	69
Gambar 4. 11 Implementasi <i>Wiring Buzzer</i> terhubung dengan ESP32 .....	70
Gambar 4. 12 Skema Rangkaian <i>Buzzer</i> terhubung dengan ESP32.....	70
Gambar 4. 13 Implementasi Wiring Selenoid , .....	71
Gambar 4. 14 <i>Skema</i> Rangkaian Alat Keseluruhan Sistem.....	71
Gambar 4. 15 Implementasi <i>Wiring</i> Seluruh Alat Sistem .....	72
Gambar 4. 16 <i>Use Case Diagram</i> .....	73
Gambar 4. 17 <i>Class Diagram</i> .....	80
Gambar 4. 18 <i>Activity Diagram Login</i> .....	84
Gambar 4. 19 <i>Activity Diagram</i> Menampilkan Halaman CCTV .....	85
Gambar 4. 20 <i>Activity Diagram</i> membuka menu kontrol CCTV.....	85
Gambar 4. 21 <i>Activity Diagram</i> membuka menu Full Screen .....	86
Gambar 4. 22 <i>Activity diagram</i> membuka menu Setting CCTV.....	87
Gambar 4. 23 <i>Activity Diagram</i> Mengedit Link CCTV .....	87
Gambar 4. 24 <i>Activity Diagram</i> Menampilkan Halaman User .....	88
Gambar 4. 25 <i>Activity Diagram</i> Menambahkan <i>User</i> .....	89
Gambar 4. 26 <i>Activity Diagram</i> Mengedit Data Diri <i>User</i> .....	90
Gambar 4. 27 <i>Activity Diagram</i> Untuk Menghapus <i>User</i> .....	91
Gambar 4. 28 <i>Activity Diagram</i> Menampilkan Halaman Akses Pintu .....	92
Gambar 4. 29 <i>Activity Diagram</i> Membuka Pintu dari sistem .....	92
Gambar 4. 30 <i>Activity Diagram</i> Menutup Pintu dari Sistem .....	93

Gambar 4. 31 <i>Activity Diagram</i> Menghapus Data.....	94
Gambar 4. 32 <i>Activity Diagram</i> Akses Pintu dengan Autentikasi.....	95
Gambar 4. 33 <i>Activity Diagram</i> Akses Pintu dengan Push Button .....	96
Gambar 4. 34 <i>Activity Diagram</i> Menampilkan WhatsApp.....	96
Gambar 4. 35 <i>Activity Diagram</i> Menambahkan Nomor WhatsApp.....	97
Gambar 4. 36 <i>Activity Diagram</i> Menghapus Nomor WhatsApp .....	98
Gambar 4. 37 <i>Activity Diagram</i> Notifikasi WhatsApp.....	98
Gambar 4. 38 <i>Activity Diagram</i> Logout.....	99
Gambar 4. 39 <i>Wireframe</i> Halaman Login .....	100
Gambar 4. 40 <i>Wireframe</i> Halaman Dashboard Utama .....	101
Gambar 4. 41 <i>Wireframe</i> Halaman Menu CCTV .....	102
Gambar 4. 42 <i>Wireframe</i> Halaman Pengguna .....	102
Gambar 4. 43 <i>Wireframe</i> Halaman Tambah pengguna .....	103
Gambar 4. 44 <i>Wireframe</i> Halaman Akses Pintu .....	104
Gambar 4. 45 <i>Wireframe</i> Halaman WahtsApp .....	104
Gambar 4. 46 Perancangan Antarmuka (UI) Halaman Login .....	105
Gambar 4. 47 Desain Antarmuka (UI) Dashboard .....	106
Gambar 4. 48 Perancangan Antarmuka (UI) Halaman Menu CCTV .....	107
Gambar 4. 49 Perancangan Antarmuka (UI) Halaman Pengguna .....	107
Gambar 4. 50 Perancangan Antarmuka (UI) Halaman Tambah Pengguna.....	108
Gambar 4. 51 Perancangan Antarmuka (UI) Halaman Akses Pintu .....	109
Gambar 4. 52 Perancangan Antarmuka (UI) Halaman Whatsapp .....	110
Gambar 4. 53 Tampilan Foto dengan Status <i>False</i> .....	129
Gambar 4. 54 Tampilan Notifikasi WhtasApp .....	130
Gambar 5. 1 Halaman Login .....	120
Gambar 5. 2 Halaman Monitoring CCTV.....	122
Gambar 5. 3 Halaman User .....	123
Gambar 5. 4 Halaman Tambah User .....	123
Gambar 5. 5 Proses Registrasi <i>Fingerprint</i> .....	124
Gambar 5. 6 Perintah Registrasi <i>Fingerprint</i> dari LCD .....	124
Gambar 5. 7 Halaman Akses Pintu .....	126
Gambar 5. 8 Proses Autentikasi.....	127

Gambar 5. 9 Halaman WhatsApp..... 129

## DAFTAR TABEL

Tabel 2. 1 Jurnal Acuan Penelitian .....	8
Tabel 2. 2 Simbol <i>Use Case Diagram</i> .....	22
Tabel 2. 3 Simbol <i>Activity Diagram</i> .....	23
Tabel 2. 4 Simbol <i>Class Diagram</i> .....	24
Tabel 3. 1 Kebutuhan Perangkat Keras.....	53
Tabel 3. 2 Kebutuhan Perangkat Lunak .....	53
Tabel 4. 1 Kebutuhan Hardware .....	59
Tabel 4. 2 Analisis Biaya.....	62
Tabel 4. 3 Deskripsi <i>Aktor</i> .....	74
Tabel 4. 4 Deskripsi <i>Use Case</i> Daftar Sidik Jari .....	74
Tabel 4. 5 Deskripsi <i>Use Case Scan</i> Sidik Jari.....	75
Tabel 4. 6 Deskripsi <i>Use Case Scan</i> Wajah.....	75
Tabel 4. 7 Deskripsi <i>Use Case</i> Membuka Pintu .....	76
Tabel 4. 8 Deskripsi <i>Use Case Login</i> .....	76
Tabel 4.9 Deskripsi <i>Use Case dashboard</i> .....	77
Tabel 4. 10 Deskripsi <i>Use Case</i> Kelola CCTV .....	77
Tabel 4. 11 Deskripsi <i>Use Case</i> Halaman Pengguna.....	78
Tabel 4. 12 Deskripsi <i>Use Case</i> Akses Pintu .....	79
Tabel 4. 13 Deskripsi <i>Use Case</i> Setting WA .....	79
Tabel 4. 14 Deskripsi <i>Use Case</i> Notifikasi WhatsApp .....	80
Tabel 4. 15 Perancangan tabel <i>user</i> .....	81
Tabel 4. 16 Perancangan tabel <i>door access</i> .....	82
Tabel 4. 17 Perancangan tabel <i>CCTV</i> .....	82
Tabel 4. 18 Perancangan tabel <i>Image</i> .....	83
Tabel 4. 19 Perancangan tabel <i>number phone</i> .....	83
Tabel 5. 1 Spesifikasi Perangkat Keras.....	117
Tabel 5. 2 Spesifikasi Perangkat Lunak .....	118
Tabel 5. 3 Pengujian Alat <i>Mikrokontroller</i> .....	131
Tabel 5. 4 Pengujian <i>Dashboard Monitoring</i> .....	132
Tabel 5. 5 Pengujian <i>Fingerprint</i> .....	134

Tabel 5. 6 Pengujian <i>Face Recognition</i> Berdasarkan Jarak .....	134
Tabel 5. 7 Pengujian <i>Face Recognition</i> Berdasarkan Kondisi Pengguna.....	134
Tabel 5. 8 Pengujian <i>Face Recognition</i> Berdasarkan Cahaya.....	135
Tabel 5. 9 Pengujian Notifikasi WhatsApp.....	135
Tabel 5. 10 Pengujian Penyimpanan Data ke <i>Dashboard Web</i> .....	135
Tabel 5. 11 Pengujian ESP32 (Komunikasi dan Kontrol).....	136
Tabel 5. 12 Pengujian <i>Relay + Selenoid Door Lock</i> .....	136
Tabel 5. 13 Pengujian <i>Buzzer</i> .....	136
Tabel 5. 14 Pengujian <i>Push Button</i> (buka/tutup manual).....	137
Tabel 5. 15 Pengujian ESP32 (Komunikasi dan Kontrol) .....	137

## **DAFTAR LAMPIRAN**

Lampiran 1: Hasil Wawancara .....	147
Lampiran 2 : Hasil Kuisioner Warga Desa Wargaluyu .....	153
Lampiran 3 : Dokumentasi Wawancara .....	157
Lampiran 4: TOR ( <i>Term of Reference</i> ) .....	160
Lampiran 5 : Listing Program .....	162

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Keamanan Rumah merupakan aspek krusial yang berperan penting dalam menjamin kenyamanan dan perlindungan bagi penghuni. Hal ini semakin krusial dengan meningkatnya kasus pencurian dan pembobolan dengan mayoritas rumah masih mengandalkan kunci konvensional yang rawan hilang maupun diduplikasi, sehingga tidak lagi mampu memberikan perlindungan optimal terhadap akses tidak sah. (Nasir & Al Gifari, 2024). Sistem *monitoring* keamanan rumah memungkinkan *user* untuk mengelola akses lebih aman dengan pemantauan aktivitas *real-time* dari jarak jauh melalui koneksi internet. Tren terkini menunjukkan peningkatan signifikan dalam pemanfaatan *Internet of Things* (IoT) yang terintegrasi dengan *Fingerprint* dan *Face Recognition*. Melalui integrasi teknologi *Fingerprint*, sistem dapat mengenali *user* dengan tingkat akurasi tinggi (Joevan Maulana Florian et al., 2024). Sistem keamanan *Face Recognition* memanfaatkan kamera CCTV 360 sebagai input data yang mampu memproses identifikasi wajah dengan cepat dan akurat (Andri Nugraha Ramdhon & Fadly Febriya, 2021). Teknologi ini dikembangkan dengan memanfaatkan metode *Machine Learning* guna meningkatkan akurasi dalam proses pendekripsi.

Desa Wargaluyu merupakan salah satu desa yang terletak di Kecamatan Arjasari Kabupaten Bandung Provinsi Jawa Barat, dengan luas wilayah 627 hektar dan 2232 rumah. Desa ini merupakan kawasan pedesaan yang mulai menunjukkan perkembangan dalam aktivitas sosial dan ekonomi. Meskipun demikian, infrastruktur teknologi dan sistem keamanannya masih terbatas. Sebagian besar rumah belum dilengkapi sistem keamanan modern dan masih mengandalkan metode konvensional. Lokasinya yang relatif jauh dari pusat kota turut memengaruhi rendahnya tingkat adopsi teknologi. Berdasarkan kondisi geografis dan sosial tersebut, Desa Wargaluyu dinilai tepat sebagai lokasi penelitian sistem *monitoring* keamanan rumah berbasis teknologi yang disesuaikan dengan karakteristik masyarakat pedesaan.

Permasalahan keamanan rumah masih menjadi isu penting, terutama di kawasan pedesaan yang belum terjangkau teknologi keamanan modern. Banyak masyarakat yang masih menggunakan sistem keamanan konvensional dan pastinya belum mampu menghadapi risiko kejahatan yang semakin kompleks. Di Desa Wargaluyu sendiri, mayoritas rumah masih menggunakan kunci biasa yang tentunya rentan terhadap pencurian, pembobolan, duplikasi, dan belum dilengkapi dengan sistem peringatan dini saat terjadi upaya akses ilegal. Kondisi ini menyulitkan pemilik rumah dalam mendeteksi dan merespon ancaman secara cepat dan akurat. Situasi tersebut menunjukkan perlunya solusi keamanan yang lebih canggih dan adaptif sesuai dengan kebutuhan masyarakat.

Beberapa penelitian sebelumnya telah mengembangkan sistem keamanan rumah berbasis IoT dengan berbagai pendekatan. (Lionar Putra & Indah Fenriana, 2024) merancang sistem smart home dengan integrasi autentikasi *Fingerprint* dan otomatisasi perangkat elektronik, sistem terbukti efektif namun tanpa *Face Recognition*, sehingga validasi identitas kurang maksimal. (Wijaya Setiady & Amanda Ginting, 2023) mengembangkan sistem kendali otomatis menggunakan NodeMCU dan sensor gerak untuk mendeteksi aktivitas fisik, sistem berjalan baik, namun tanpa autentikasi *user*, akses belum sepenuhnya aman. (Muhammad Nasir & Zainul Al Gifari, 2024) mengembangkan sistem keamanan pintu menggunakan *Solenoid Door Lock* dan *Magnetic Switch Sensor* dengan notifikasi Telegram. Sistem responsif dalam menampilkan status pintu secara *real-time*, tetapi masih menggunakan *keypad* manual. Berbeda dengan itu, penelitian ini menggabungkan autentikasi *Fingerprint* dan *Face Recognition* berbasis *Machine Learning*, serta notifikasi WhatsApp yang terhubung ke *dashboard* web untuk pemantauan *real-time*. Sistem dikembangkan dengan metode *Waterfall* dan *framework Flask* untuk integrasi komponen IoT dan basis data.

Berdasarkan identifikasi masalah dan studi literatur, dirancang sebuah sistem *monitoring* keamanan rumah berbasis IoT yang mengintegrasikan autentikasi biometrik *Fingerprint* dan *Face Recognition* berbasis *Machine Learning*. Sistem ini mampu mengirimkan notifikasi otomatis ke WhatsApp saat terjadi akses, serta menyediakan pemantauan jarak jauh secara *real-time* melalui *dashboard* berbasis web. *Fingerprint* diidentifikasi menggunakan sensor AS608

dengan metode *Minutiae-Based Matching*, dan *Face Recognition* dilatih dengan metode *Machine Learning* untuk meningkatkan akurasi. Sistem dikembangkan dengan metode *Waterfall*, melalui tahapan analisis kebutuhan, perancangan, implementasi, dan pengujian. Penelitian ini menghasilkan prototipe sistem keamanan rumah yang akan diuji pada satu rumah sebagai studi kasus. Sistem ini diharapkan mampu meningkatkan perlindungan terhadap akses tidak sah. Berdasarkan uraian tersebut, penelitian ini mengusung judul “**SISTEM MONITORING KEAMANAN RUMAH MENGGUNAKAN FINGERPRINT DAN FACE RECOGNITION BERBASIS MACHINE LEARNING**”.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka terdapat rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana mengatasi keterbatasan sistem keamanan rumah konvensional yang rentan terhadap akses tidak sah?
2. Bagaimana memberikan notifikasi *real-time* saat terjadi upaya akses gagal atau tidak terverifikasi?
3. Bagaimana membangun sistem *monitoring* keamanan rumah dengan pemantauan *real-time* dan manajemen riwayat akses?

## 1.3 Batasan Masalah

Adapun batasan masalah yang ada agar permasalahan sesuai yang dituju maka perlu ditetapkan batasan masalah sebagai berikut :

1. Sistem menggunakan autentikasi biometrik *Fingerprint* (Sensor AS806) dan *Face Recognition* dengan kamera CCTV 360° sebagai metode keamanan utama.
2. Notifikasi WhatsApp dikirim saat terjadi kegagalan autentikasi.
3. Sistem berbasis IoT dengan ESP32 *dashboard web* untuk pemantauan pintu oleh *user* terautentikasi.
4. Sistem ini masih terbatas pada tahap prototipe, sedangkan website hanya berjalan secara lokal tanpa proses *deployment* ke layanan hosting.

## 1.4 Tujuan Penelitian

Berdasarkan rumusan masalah sebelumnya, maka tujuan penelitian yang ingin dicapai oleh penulis, yaitu :

1. Merancang dan membangun sistem *monitoring* keamanan rumah menggunakan autentikasi biometrik *Fingerprint* dan *Face Recognition*.
2. Mengembangkan sistem notifikasi otomatis ke WhatsApp saat terdeteksi upaya akses gagal atau tidak terverifikasi.
3. Membangun *dashboard web real-time* terintegrasi notifikasi untuk memantau keamanan dan riwayat akses dari jarak jauh.

## 1.5 Metodologi Penelitian

Pada penelitian ini, penulis menerapkan metode *Waterfall* sebagai pendekatan dalam perancangan sistem, yang terdiri dari tahapan-tahapan berikut:

### 1. Identifikasi masalah

Tahap ini untuk mengidentifikasi permasalahan di lingkungan masyarakat Desa Wargaluyu terkait keamanan rumah. Proses dilakukan dengan mengamati fenomena yang terjadi, keterbatasan sistem keamanan konvensional, mengkaji kebutuhan sistem pengamanan yang *modern*.

### 2. Pengumpulan Data

Dalam proses pengumpulan data penulis menerapkan beberapa metode untuk mendapatkan data yang diperlukan dalam membangun sistem:

#### a. Observasi

Melaksanakan observasi terhadap sistem keamanan pintu rumah yang diterapkan oleh masyarakat Desa Wargaluyu sebagai studi lapangan.

#### b. Wawancara

Melakukan wawancara dengan warga dan pejabat desa setempat untuk menggali informasi mengenai kebutuhan sistem keamanan rumah.

#### c. Studi Pustaka

Penulis mengumpulkan dan mengkaji referensi dari jurnal ilmiah terkait dengan *Internet of Things*, biometrik, serta keamanan digital.

### 3. Perancangan Sistem

Proses perancangan sistem *monitoring* keamanan rumah dilakukan secara terstruktur dan menyeluruh, meliputi pengembangan perangkat keras dan perangkat lunak yang saling terintegrasi untuk mendukung autentikasi biometrik, notifikasi WhatsApp dan pemantauan *real-time* dengan *dashboard monitoring*. Adapun Tahapan perancangan tersebut meliputi:

a. Perancangan perangkat keras

Perancangan perangkat keras pada sistem *monitoring* keamanan rumah ini menggunakan ESP32 sebagai mikrokontroler utama yang bertugas mengelola komunikasi data dan kendali sistem, dilengkapi dengan kamera cctv sebagai input untuk proses *Face Recognition*, serta sensor *Fingerprint AS806* yang berfungsi untuk memverifikasi identitas *user* melalui sidik jari. Untuk mekanisme penguncian pintu, digunakan *relay* sebagai pengendali arus listrik dan *Solenoid Door Lock* sebagai aktuator utama yang membuka atau menutup kunci pintu secara otomatis berdasarkan hasil autentikasi. Semua perangkat keras ini saling terintegrasi untuk mendukung sistem *monitoring* keamanan rumah yang responsif.

b. Perancangan perangkat lunak

Kegiatan ini mencakup pemrograman ESP32 untuk mendukung proses autentikasi biometrik, integrasi komunikasi *Internet of Things* (IoT) menggunakan protokol MQTT, serta implementasi notifikasi otomatis melalui WhatsApp. Perancangan sistem juga dilakukan dengan menggunakan *Unified Modeling Language* (UML), yang meliputi *Use Case Diagram*, *Activity Diagram*, serta membuat *Entity Relationship Diagram* (ERD) untuk memodelkan alur sistem dan struktur basis data secara terstruktur dan sistematis.

### 4. Implementasi Sistem

Implementasi sistem dilakukan dengan merakit perangkat keras kemudian memprogram ESP32 menggunakan Arduino IDE untuk menangani autentikasi dan komunikasi data. Proses autentikasi wajah menggunakan kamera CCTV dilakukan dengan menerapkan algoritma *Machine Learning* berbasis *Local Binary Patterns Histogram* (LBPH) yang telah dilatih sebelumnya untuk mengenali wajah

pengguna berdasarkan kemiripan fitur wajah yang telah tersimpan di database. Sistem juga terhubung dengan jaringan menggunakan protokol MQTT sebagai media komunikasi antar perangkat IoT. Untuk antarmuka dibuat *Dashboard Monitoring* berbasis web menggunakan HTML, CSS, yang menampilkan status autentikasi serta log aktivitas. Notifikasi sistem dikirimkan ke WhatsApp, memberikan informasi hasil akses langsung kepada pengguna.

## 5. Pengujian dan Evaluasi

Pengujian dilakukan untuk mengevaluasi fungsi dan kinerja sistem secara menyeluruh dengan metode *Black Box Testing*. Fokus pengujian pada input dan output sistem tanpa memperhatikan struktur kode. Aspek yang diuji meliputi:

### a Pengujian Fungsional (Simulasi Fisik):

Pada pengujian fungsionalitas di lakukan dengan memberikan input sidik jari dan citra wajah yang sudah terdaftar untuk memastikan autentikasi berhasil membuka pintu, dan menampilkan status pada LCD, serta mengirimkan notifikasi dengan WhatsApp kemudian mencatat aktivitas, yaitu hasil dari akses pada *dashboard web* secara *real-time*.

### b Simulasi Kesalahan

Pengujian dilakukan dengan memberikan input yang tidak valid atau tidak dikenali oleh sistem untuk mengamati respons terhadap upaya akses ilegal. Tujuannya untuk menguji ketahanan sistem dalam menghadapi potensi penyusupan.

### c Pengujian Perangkat Keras

Pengujian perangkat keras dilakukan terhadap seluruh komponen fisik IoT untuk memastikan bahwa setiap perangkat yang telah dirangkai berfungsi secara stabil dan optimal.

## 1.6 Sistematika Penulisan

Sistematika penulisan skripsi ini disusun untuk menyajikan gambaran yang jelas, terstruktur mengenai alur pembahasan skripsi, sistematika penulisan dalam skripsi ini adalah sebagai berikut:

## **BAB I PENDAHULUAN**

Bab ini merupakan bagian yang memuat uraian mengenai latar belakang penelitian, rumusan masalah, batasan masalah, tujuan penelitian, metodologi penelitian, sistematika penulisan skripsi yang digunakan sebagai landasan dan pedoman dalam menyusun penelitian ini secara sistematis dan terstruktur.

## **BAB II TINJAUAN PUSTAKA**

Bab ini menjelaskan landasan teori yang bersumber dari mata kuliah yang telah dipelajari, tabel yang menyajikan acuan penelitian dari jurnal terdahulu. Pada bagian dasar teori, dijelaskan mengenai materi yang mendukung sistem Smart Door Lock, termasuk perangkat keras dan perangkat lunak yang digunakan.

## **BAB III METODOLOGI PENELITIAN**

Bab ini menjelaskan metodologi penelitian yang menjadi acuan dalam pelaksanaan penelitian, dimulai dari perencanaan, perancangan, implementasi pengkodean, pengujian. Setiap tahapan disusun sistematis untuk menggambarkan keseluruhan proses pengembangan sistem *monitoring* keamanan rumah.

## **BAB IV ANALISIS, PERANCANGAN DAN HASIL**

Bab ini akan membahas analisis kebutuhan sistem serta tahapan perancangan sistem *monitoring* keamanan rumah, yaitu perancangan perangkat keras dan perangkat lunak, serta integrasi dengan *server MQTT* dan notifikasi WhatsApp, Database, perancangan antarmuka.

## **BAB V IMPLEMENTASI DAN PENGUJIAN**

Bab ini membahas implementasi dan pengujian sistem *monitoring* keamanan rumah, mulai dari pemasangan perangkat keras, integrasi web *monitoring*, hingga pengaturan notifikasi. Pengujian menggunakan *BlackBox* untuk memastikan fungsi sistem sesuai dengan yang di rancang sebelumnya.

## **BAB VI PENUTUP**

Bab ini menyajikan kesimpulan dari hasil analisis pada bab-bab sebelumnya. Kemudian terdapat saran yang akan menjadi acuan dalam pengembangan sistem *monitoring* keamanan rumah di masa mendatang.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Landasan Teori

Sebagai landasan dalam penyusunan penelitian ini, penulis mengkaji sejumlah jurnal terkait untuk mengetahui pendekatan yang telah digunakan sebelumnya. Kajian ini dilakukan guna melihat persamaan dan perbedaan dengan penelitian yang akan dilakukan.

Tabel 2. 1 Jurnal Acuan Penelitian

No	Jurnal Penelitian	Metode	Hasil	Perbedaan
1.	<b>Judul :</b> Rancang bangun smart home system berbasis iot dengan integrasi sidik jari ( <i>Fingerprint</i> ) dan otomasi elektronik <b>Penulis :</b> Lionar Putra dan Indah Fenriana.	Metode autentikasi menggunakan <i>Fingerprint</i> .	Sistem Smart Home berbasis IoT dengan integrasi sidik jari dan otomasi elektronik terbukti efektif, dalam keamanan rumah.	Metode autentikasi <i>Fingerprint</i> dan <i>Face Recognition</i> .
2.	<b>Judul :</b> Perancangan Dan Implementasi Security Dan Sistem Kendali Otomatis <i>Smart Home</i> Menggunakan Nodemcu Design And Implementation Of Security And Smart Home Automatic Control Systems Using Nodemcu <b>Penulis :</b> Kevin Wijaya Setiady, Jusia Amanda Ginting, jginting.	Metode autentikasi Menggunakan sensor untuk mendeteksi pergerakan atau manipulasi fisik tanpa autentikasi <i>user</i> .	Mikrokontroler dapat di gunakan dengan baik dan berjalan sesuai dengan fungsi yang ditujukan.	Autentikasi dilakukan melalui <i>Fingerprint</i> dan <i>Face Recognition</i> untuk membatasi akses hanya bagi <i>user</i> yang berwenang.

3.	<p><b>Judul :</b> Rancang Bangun Sistem Keamanan Pintu Rumah Menggunakan Solenoid Door Lock Dan Magnetic Switch Sensor Dengan Notifikasi Dan Kontrol Melalui Telegram.</p> <p><b>Penulis :</b> Muhammad Nasir1 dan Zainul Al Gifari2.</p>	<p>User an</p> <p><i>Solenoid Door Lock, sensor Magnetic Switch</i></p> <p>dikendalikan</p> <p>NodeMCU</p> <p>ESP8266, serta integrasi</p> <p>Telegram untuk kontrol dan notifikasi.</p>	<p>Sistem berhasil memberikan kontrol pintu secara <i>real-time</i> melalui Telegram, serta mendeteksi status pintu secara akurat, dilengkapi alternatif <i>Keypad</i> jika koneksi internet bermasalah.</p>	<p>Sistem ini menggunakan autentikasi biometrik <i>Fingerprint</i> atau <i>Face Recognition</i>, dan notifikasi melalui WhatsApp terhubung dengan <i>dahsboard web</i>.</p>
----	---	--	--	---

Berdasarkan Tiga jurnal pada tabel diatas memiliki fokus utama pada pengembangan sistem keamanan rumah berbasis teknologi *Internet of Things* (IoT). Jurnal pertama oleh (Lionar Putra & Indah Fenriana, 2024) dengan judul Rancang Bangun *Smart Home System* Berbasis IoT Dengan Integrasi Sidik Jari (*Fingerprint*) dan otomasi elektronik. Sistem terbukti efektif menggunakan *Fingerprint* namun tanpa *Face Recognition* validasi identitas akan kurang maksimal. Jurnal kedua oleh (Wijaya Setiady & Amanda Ginting, 2023) dengan judul Perancangan Dan Implementasi Security Dan Sistem Kendali Otomatis *Smart Home* Menggunakan Nodemcu Design And Implementation Of Security And Smart Home Automatic Control Systems Using Nodemcu, yaitu sistem dirancang dengan NodeMCU dan sensor gerak untuk mendeteksi aktivitas fisik tanpa biometrik sehingga autentikasi menjadi kurang optimal. Jurnal ketiga oleh (Muhammad Nasir & Zainul Al Gifari, 2024) dengan judul Rancang Bangun Sistem Keamanan Pintu Rumah Menggunakan *Solenoid Door Lock* Dan *Magnetic Switch Sensor* Dengan Notifikasi Dan Kontrol Melalui Telegram, Sistem penguncian IoT ini terhubung ke Telegram, namun masih bergantung pada perintah manual, sehingga kurang responsif terhadap ancaman.

Ketiga jurnal mengusung prinsip IoT sebagai dasar sistem keamanan, namun dengan pendekatan dan fitur yang berbeda. Jurnal pertama sudah

menggunakan autentikasi biometrik yang kuat namun belum menyertakan notifikasi *real-time*. Jurnal kedua menitikberatkan pada deteksi gerakan, menjadikannya kurang aman karena tidak memverifikasi identitas. Jurnal ketiga sudah menggunakan notifikasi melalui Telegram sebagai media komunikasi, tetapi masih mengandalkan interaksi manual. Ketiganya belum menghadirkan solusi keamanan yang sepenuhnya otomatis, terutama dalam integrasi antara autentikasi biometrik ganda dengan notifikasi WhatsApp yang terhubung dengan dashboard web sebagai sistem *monitoring real-time*.

Penelitian ini menghadirkan solusi yang lebih menyeluruh melalui integrasi autentikasi ganda berbasis *Fingerprint* dan *Face Recognition* yang dibangun menggunakan metode *Deep Learning*, menjadikannya lebih akurat dan praktis. Sistem ini juga dilengkapi dengan kamera CCTV untuk identifikasi wajah, serta *Solenoid Door Lock* yang dikendalikan oleh NodeMCU sebagai pusat kendali sistem. Web *monitoring* dikembangkan menggunakan *Tailwind CSS* untuk pengalaman antarmuka *user* yang modern dan responsif, memberikan notifikasi otomatis melalui WhatsApp ketika terjadi upaya akses ilegal, serta pencatatan aktivitas secara *real-time* pada *dashboard web*.

Berdasarkan analisis terhadap ketiga jurnal, terdapat beberapa kekurangan yang menjadi celah penelitian, seperti tidak adanya integrasi autentikasi biometrik ganda, belum adanya sistem notifikasi otomatis, dan keterbatasan dalam pencatatan aktivitas secara *real-time*. Sistem pada penelitian ini mengatasi *gap* tersebut dengan menggabungkan teknologi biometrik *Fingerprint* dan *Face Recognition*, serta menyediakan sistem *monitoring* dan notifikasi otomatis berbasis web dan WhatsApp. Dengan demikian, sistem ini tidak hanya mampu meningkatkan tingkat keamanan rumah secara signifikan, tetapi juga menghadirkan pengalaman *user* yang lebih praktis, cerdas, dan selaras dengan kemajuan teknologi digital masa kini.

## 2.2 Dasar Teori

Dalam membangun sistem *monitoring* keamanan rumah, penulis menggunakan berbagai alat dan bahan dengan merujuk pada teori-teori relevan sebagai landasan konseptual dan teknis dari sistem yang dibangun.

### **2.2.1 Keamanan Rumah**

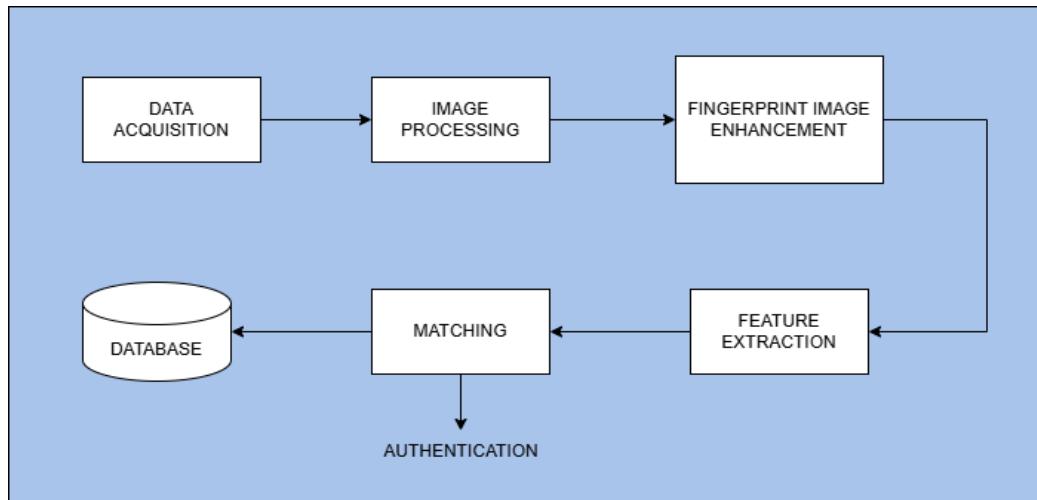
Keamanan Rumah merupakan aspek krusial dalam menunjang kualitas hidup. Setiap orang memerlukan sistem perlindungan yang memadai terhadap rumah mereka, sebagaimana pentingnya menjaga kondisi kesehatan. Oleh karena itu, berbagai kemajuan teknologi dirancang untuk meningkatkan perlindungan serta menjaga aset yang dimiliki, guna menciptakan lingkungan yang aman dan nyaman bagi penghuni (Ade Mubarok et al., 2020).

### **2.2.2 Sistem *Monitoring***

Sistem *monitoring* adalah proses pengawasan dan pemantauan suatu kondisi atau aktivitas secara terus-menerus guna mendeteksi perubahan, anomali, atau potensi gangguan. Sistem ini umumnya terdiri dari perangkat sensor, pengontrol, dan media notifikasi atau tampilan informasi. Dalam konteks keamanan rumah, sistem *monitoring* digunakan untuk mengawasi akses masuk, aktivitas mencurigakan, serta memberikan notifikasi secara *real-time* kepada pemilik rumah melalui perangkat seperti smartphone. Sistem *monitoring* yang baik tidak hanya memberikan informasi pasif, tetapi juga mampu merespons kejadian secara otomatis. Respons dapat berupa pengiriman notifikasi WhatsApp penyimpan rekaman dari CCTV saat terdeteksi ancaman.engan demikian, sistem tidak hanya bersifat informatif, tetapi juga proaktif dalam menjaga keamanan lingkungan yang diawasi.Sistem *monitoring* berbasis IoT memberikan efisiensi tinggi karena memungkinkan *user* memadukan kondisi lingkungan dari jarak jauh dengan tingkat keakuratan dan kecepatan yang lebih baik dibandingkan metode konvensional (Kusumah & Izzatul Islam, 2023).

### **2.2.3 *Fingerprint***

Teknologi *Fingerprint* atau sidik jari merupakan salah satu metode autentikasi biometrik yang memanfaatkan pola unik pada jari manusia untuk verifikasi identitas. Keunggulan utama dari metode ini adalah tingkat akurasi yang tinggi dan kesulitan dalam pemalsuan.



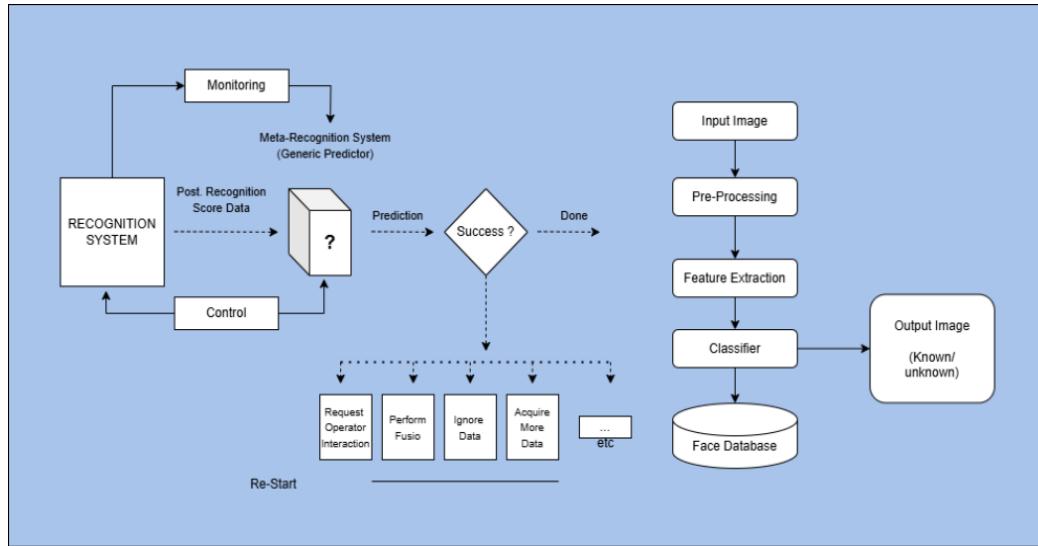
Gambar 2. 1 Arsitektur *Fingerprint*

Sumber: (Sumit Singh, 2015)

Sidik jari pada manusia adalah bukti fisik yang penting. Tidak ada dua sidik jari yang identik di dunia ini, bahkan antara saudara kembar sekalipun. Dalam kajian ilmiah, dikemukakan bahwa jika terdapat 5 juta orang di dunia, kemungkinan terjadinya dua sidik jari yang sama baru akan terjadi setelah 300 tahun. Untuk membandingkan sidik jari satu dengan lainnya, digunakan metode yang disebut *Minutiae Based Matching* yaitu garis yang terdapat pada permukaan kulit jari manusia yang membentuk pola sidik jari. *Minutia* ini memiliki berbagai jenis dan menjadi fitur penting yang diuji untuk menentukan kesesuaian pola sidik jari. Metode pencocokan yang digunakan adalah *Minutiae-Based Matching*, yaitu pencocokan berdasarkan fitur-fitur khusus seperti titik ujung garis (*ridge ending*) dan titik percabangan (*bifurcation*) (Anggya N D & Soetarmono, 2012).

#### 2.2.4 *Face Recognition*

*Face Recognition* merupakan hasil pengembangan dari teknologi pendeksiwan wajah. Teknologi ini mampu menghasilkan citra wajah dari tangkapan kamera, membandingkannya dengan data wajah yang telah tersimpan di sistem komputer untuk menemukan kesamaan. Dengan demikian, komputer dapat mengenali atau mendekksi keberadaan individu tersebut. Teknologi pengenalan wajah ini diklasifikasikan dalam tiga kategori utama, yaitu verifikasi, identifikasi, serta pengawasan (Andri Nugraha Ramdhon & Fadly Febriya, 2021).



Gambar 2. 2 Arsitektur *Face Recognition*

Sumber:(Medapati et al., 2020)

Pendeteksian wajah merupakan salah satu metode untuk mengidentifikasi dan memisahkan fitur area sekitar wajah untuk pengenalan wajah. Teknologi ini digunakan untuk menganalisis wajah dengan mengenali ciri dan sifat khas wajah, tanpa memperhatikan faktor lain seperti bangunan, pohon, atau tubuh manusia itu sendiri. Bidang penelitian terkait dengan pemrosesan wajah meliputi otentifikasi wajah, pelokasian wajah, pelacakan wajah, dan pengenalan ekspresi wajah. Pendekstnsian wajah merupakan salah satu tahap penting dalam preprosesing yang harus dilakukan sebelum tahap *Face Recognition* (Wiguna et al., 2022).

## 2.2.5 *Machine Learning*

*Machine learning* merupakan suatu cabang dari kecerdasan buatan (*Artificial Intelligence*) yang memungkinkan komputer belajar dari data untuk mengenali pola serta membuat keputusan tanpa diprogram secara eksplisit. Menurut (Ahmad Roihan et al., 2019), *machine learning* bekerja dengan memproses dataset untuk menghasilkan model yang dapat digunakan dalam proses prediksi atau klasifikasi. Dalam penelitian ini, *machine learning* digunakan untuk mengenali wajah berdasarkan dataset yang telah direkam sebelumnya, kemudian mencocokkannya dengan citra wajah yang baru diambil oleh kamera.

*Machine learning* merupakan suatu pendekatan dalam komputasi yang menggabungkan algoritma matematika dan data sebagai dasar pembelajaran untuk menghasilkan prediksi terhadap kejadian di masa mendatang (Goldberg & Holland, 1988). Proses pembelajarannya dilakukan melalui dua tahap utama, yaitu *training* untuk membangun model dari data yang tersedia, dan *testing* untuk mengevaluasi kemampuan model dalam mengenali pola baru secara otomatis. Penelitian terkini mengungkapkan bahwa *machine learning* terbagi menjadi tiga kategori utama berdasarkan cara sistem belajar dari data, yaitu:

### 1. *Supervised Learning*

Algoritma dilatih menggunakan *dataset* berlabel, di mana setiap data masukan memiliki label target. Pada penelitian ini, *face recognition* menggunakan *supervised learning* karena setiap citra wajah pada dataset dilabeli identitas pengguna saat proses registrasi. Model kemudian mempelajari hubungan antara citra wajah dan label identitas tersebut, sehingga saat menerima citra wajah baru dari kamera, sistem dapat mengklasifikasikannya sesuai identitas pada database.

### 2. *Unsupervised Learning*

Algoritma mempelajari pola atau struktur data yang tidak memiliki label, seperti pengelompokan wajah berdasarkan kemiripan tanpa mengetahui identitas. Pendekatan ini tidak digunakan dalam penelitian ini karena sistem membutuhkan pengenalan identitas untuk autentikasi pintu.

### 3. *Reinforcement Learning*

Algoritma belajar melalui umpan balik *reward* atau *punishment* dari interaksi dengan lingkungan. Pendekatan ini biasanya digunakan untuk pengendalian robot atau navigasi otonom, tidak relevan untuk pengenalan wajah pada sistem keamanan rumah ini (Ahmad Roihan et al., 2019).

Dalam sistem monitoirng keamanan rumah ini, proses *machine learning* dilakukan melalui beberapa tahapan yang saling berkaitan. Setiap tahapan berperan penting dalam memastikan sistem dapat mengenali wajah secara akurat dan efisien. Proses ini bertujuan untuk menyiapkan citra wajah dalam bentuk yang optimal sebelum dianalisis oleh algoritma. Berikut merupakan penjelasan lengkap untuk setiap tahap dalam proses machine learning pada face recognition.

## 1. Pengumpulan data

*Fase awal proses pengumpulan citra wajah saat pengguna melakukan registrasi*, yang kemudian disimpan di database sebagai data latih. Setiap individu memiliki beberapa citra wajah untuk keakuratan sistem.

## 2. *Pra-pemrosesan*

- a. *Grayscale conversion*, mengubah citra berwarna menjadi skala abu-abu agar pemrosesan lebih efisien.
- b. *Cropping region of interest (ROI)*, hanya area wajah yang diproses.
- c. *Normalization*, menyamakan ukuran citra, misalnya ke  $100 \times 100$  piksel.
- d. Penghapusan noise, memperbaiki kualitas citra agar fitur lebih jelas.

Pada penelitian (Safara Alfan et al., 2024) menggunakan proses *cropping*, *resizing*, dan *grayscale* sehingga data menjadi lebih konsisten dan kompatibel untuk ekstraksi fitur LBPH.

## 3. Ekstraksi *Fitur*

Setelah melalui tahap *pra-pemrosesan*, citra wajah dianalisis menggunakan algoritma *Local Binary Patterns Histogram* untuk melakukan ekstraksi fitur. LBPH bekerja dengan membandingkan intensitas piksel pusat terhadap piksel-piksel di sekitarnya, lalu mengubahnya menjadi pola biner yang merepresentasikan tekstur lokal citra. Pola ini kemudian dikonversi menjadi histogram untuk setiap blok gambar, yang selanjutnya digabungkan menjadi satu vektor fitur. Vektor inilah yang digunakan sebagai dasar pencocokan antara citra wajah *real-time* dan data yang telah tersimpan di database. Metode LBPH dikenal ringan, sehingga sangat sesuai diterapkan pada sistem berbasis IoT dengan keterbatasan sumber daya komputasi.

### a. Rumus Perhitungan LBP

Nilai LBP untuk sebuah piksel pusat  $(x_c, y_c)$  didefinisikan sebagai:

$$LBP(x_c, y_c) = \sum_{p=0}^{p-1} s(g_p - g_c) \cdot 2^p$$

dengan:

$g_c$  = nilai intensitas piksel pusat

$g_p$  = nilai intensitas piksel tetangga ke- $p$

$P$  = jumlah piksel tetangga (biasanya 8 dalam matriks  $3 \times 3$ )

Pada bentuk dasarnya, algoritma LBP memanfaatkan nilai piksel pusat dalam matriks  $3 \times 3$  sebagai ambang batas untuk membandingkan nilai intensitas dengan piksel di sekitar. Hasil perbandingan membentuk pola biner yang merepresentasikan tekstur lokal citra.

5	9	1
4	4	6
7	2	3

LBP

1	1	0
1		1
1	0	0

LBP Value: 11010011 = 211

#### b. Perhitungan Histogram LBPH

Setelah nilai LBP diperoleh untuk setiap piksel, citra dibagi menjadi beberapa cell dan dihitung setiap cell, lalu semua histogram digabungkan menjadi satu vektor fitur yang menjadi representasi citra wajah.

#### c. Perhitungan Jarak Histogram

Proses pencocokan dilakukan dengan membandingkan histogram citra uji ( $H1H\_1H1$ ) dan histogram citra latih ( $H2H\_2H2$ ) menggunakan *Euclidean Distance*:

$$d(H1, H2) = \sqrt{\sum_{i=1}^n (H1i - H2i)^2}$$

dengan:

a.  $n$  = jumlah bin pada histogram

b.  $H1i, H2i$  = nilai bin ke- $i$  dari histogram citra uji dan citra latih

Jika nilai jarak di bawah ambang batas tertentu, sistem menganggap kedua wajah cocok.

*Local Binary Pattern Histogram* merupakan salah satu metode yang banyak digunakan dalam proses pengenalan objek, khususnya wajah. Metode ini mampu mengenali wajah dari berbagai sudut pandang, baik frontal maupun menyamping. Prinsip kerja LBPH adalah dengan membedakan objek utama dari latar belakang berdasarkan pola tekstur lokal. LBPH merupakan kombinasi antara algoritma *Local Binary Pattern* dan pendekatan histogram seperti *Histogram of Oriented Gradients*. Citra wajah yang diambil secara *real-time* melalui kamera akan diekstraksi ciri-cirinya dengan histogram, lalu dibandingkan dengan data wajah yang telah tersimpan di database. Berikut ini adalah tahapan-tahapan utama dalam proses kerja algoritma LBPH.

### 1) Parameter Utama

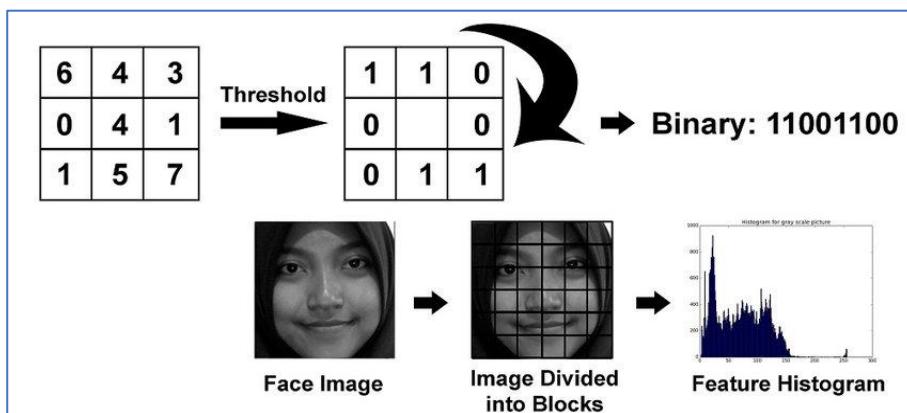
LBPH menggunakan empat parameter utama yang menentukan proses ekstraksi fitur, yaitu *radius*, *neighbors*, *grid X*, dan *grid Y*. Keempat parameter ini memengaruhi sensitivitas dan detail pola tekstur yang dihasilkan.

### 2) Proses Pelatihan (*Training*)

tahap pelatihan dilakukan untuk membangun model pengenalan wajah yang optimal. Dataset wajah dibaca oleh sistem, lalu digunakan untuk membentuk model yang akan digunakan dalam proses identifikasi selanjutnya.

### 3) Penerapan Operasi LBP

Algoritma LBP diterapkan pada citra wajah untuk menghasilkan citra yang mampu merepresentasikan karakteristik wajah lebih akurat. Hasil dari proses ini digunakan sebagai dasar pengenalan wajah tahap berikutnya.

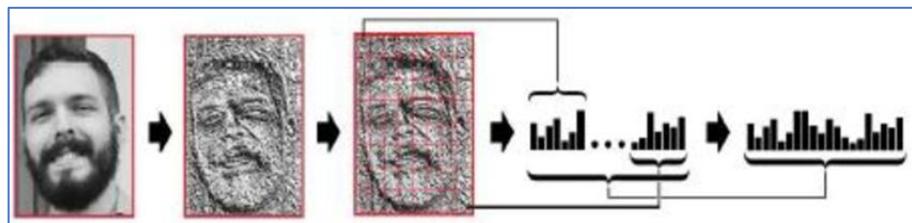


Gambar 2. 3 Operasi LBP

Sumber: (Sanjaya et al., 2017)

#### 4) Ekstrasi Histogram

Pada tahap ini, citra yang telah diproses akan dibagi menjadi sejumlah bagian berdasarkan parameter *grid X* dan *grid Y*. Setiap bagian akan diekstraksi histogramnya untuk merepresentasikan pola tekstur lokal secara lebih detail.



Gambar 2. 4 Proses Ekstrasi Histogram

Sumber: (Kosasih & Daomara, 2021)

Selanjutnya, untuk proses pengenalan wajah, citra uji akan dimasukkan ke dalam sistem serta melalui serangkaian tahapan yang sama seperti saat pelatihan. Hasil ekstraksi fitur dibandingkan dengan histogram wajah yang tersimpan dalam database untuk menentukan tingkat kecocokan. Dalam implementasinya, proses ini menggunakan OpenCV (*Open Source Computer Vision Library*), yang merupakan pustaka pemrograman populer dalam bidang *computer vision*. *OpenCV* memungkinkan sistem untuk melakukan analisis visual secara otomatis, menyerupai cara manusia mengenali objek melalui penglihatan, dan beberapa pengimplementasian dari computer vision ini yaitu diantaranya *face recognition*, *face detection*, *face/object tracking*, *road tracking*, dan lain-lain.

#### 4. Pelatihan Model

Fitur-fitur yang telah diekstraksi dari sejumlah besar citra wajah kemudian digunakan untuk melatih model *machine learning*. Model belajar mengenali pola-pola khas dari setiap wajah yang berbeda dan menyimpannya dalam bentuk representasi matematis. Dalam sistem ini, LBPH menyimpan histogram per pengguna sebagai referensi untuk proses identifikasi di tahap selanjutnya.

#### 5. Pengujian dan Identifikasi

Setelah model selesai dilatih, sistem akan menerima *input* wajah baru dari kamera secara *real-time*. Citra ini melalui tahapan *pra-pemrosesan* dan ekstraksi fitur, kemudian dibandingkan dengan data histogram yang tersimpan sebelumnya.

Proses ini untuk mengidentifikasi apakah wajah yang ditangkap merupakan wajah pengguna yang terdaftar atau bukan. Tahapan ini menjadi fondasi utama dalam proses pengembangan sistem *face recognition* berbasis *machine learning*. Dengan implementasi yang tepat, sistem dapat bekerja dengan *real-time* dan memberikan tingkat akurasi yang tinggi meskipun dijalankan pada perangkat dengan sumber daya terbatas seperti sistem *Internet of Things*.

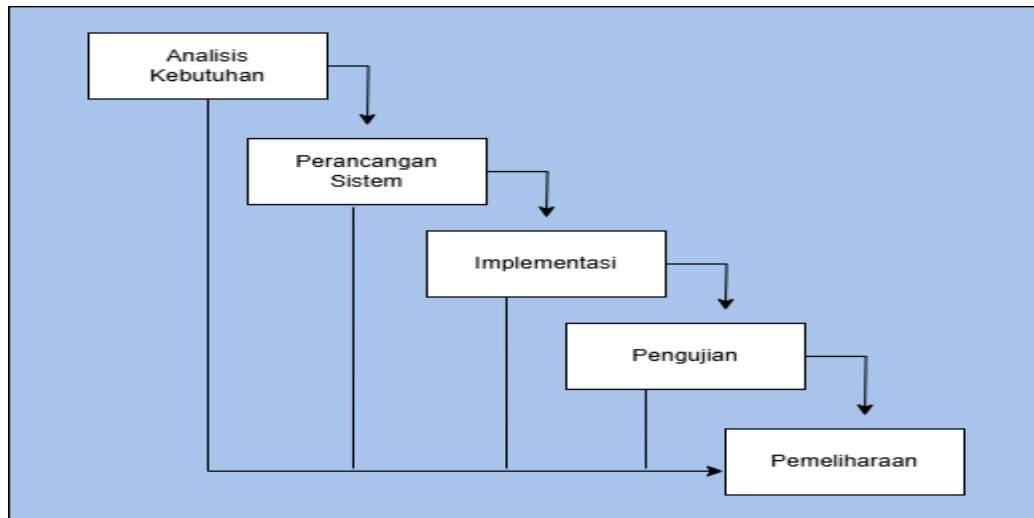
## 6. Integrasi *Machine Learning* dalam Sistem IoT

Integrasi *Machine Learning* ke dalam sistem IoT memungkinkan kamera CCTV untuk melakukan pengolahan visual secara otomatis dan responsif. Dalam sistem keamanan rumah, *Machine Learning* diterapkan pada alur berikut:

- a. Citra wajah ditangkap oleh kamera CCTV 360.
- b. Citra dikirim ke server/edge device untuk *pra-pemrosesan* dan ekstraksi fitur melalui algoritma *Machine Learning* seperti LBPH.
- c. Hasil ekstraksi dibandingkan dengan data wajah yang tersimpan di database untuk melakukan identifikasi.
- d. Berdasarkan hasil identifikasi, sistem memberi instruksi pada aktuator (berhasil atau gagal membuka pintu), serta dapat mengirim notifikasi melalui whatsapp jika terjadi kegagalan akses serta log aktivitasnya tersimpan pada dashboard web sistem monitoring keamanan rumah.

### 2.2.6 Model SDLC *Waterfall*

*Metode Waterfall* merupakan salah satu model dalam *Software Development Life Cycle* (SDLC) yang membagi proses pengembangan perangkat lunak ke dalam beberapa tahapan terstruktur seperti analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan. Pendekatan ini mengikuti alur kerja yang bersifat linear dan berurutan, di mana setiap tahapan harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahap berikutnya (Hilman Aziz & Imam Suharjo, 2024). *waterfall* menekankan pada perencanaan yang matang sejak awal serta pelaksanaan yang disiplin di setiap fase, sehingga menjadi metode penyelesaian masalah yang sistematis dan bertahap dalam pengembangan perangkat lunak.



Gambar 2. 5 Model SDLC Waterfall

Sumber: (Hilman Aziz & Imam Suharjo, 2024)

Model ini dipilih karena proses perancangan dan pengembangannya dilakukan secara bertahap dan sistematis.

### 1. Analisis Kebutuhan

Pada tahap ini, dilakukan studi literatur guna mengumpulkan informasi sebanyak mungkin mengenai data dan komponen elektronik yang relevan dengan sistem keamanan rumah. Pemilihan perangkat keras disesuaikan dengan kebutuhan sistem untuk dapat memberikan informasi kondisi aktual.

### 2. Perancangan Sistem

Tahap ini untuk merancang sistem berdasarkan spesifikasi kebutuhan. Desain sistem mencakup alur kerja autentikasi menggunakan pengenalan wajah dan sidik jari, serta integrasi dengan sistem notifikasi. Perancangan dilakukan menggunakan UML untuk memvisualisasikan struktur dan proses untuk memvisualisasikan struktur dan proses sistem, *Figma* untuk membuat tampilan antarmuka *user* yang sesuai dengan kebutuhan *user*.

### 3. Implementasi

Setelah proses perancangan, selanjutnya adalah pembangunan sistem yang mencakup pemasangan seluruh perangkat keras dan dilakukan proses pemrograman untuk mengintegrasikan perangkat keras dengan sistem perangkat lunak, guna mewujudkan fungsi pemantauan dan autentikasi sesuai dengan desain yang telah dibuat.

#### 4. Pengujian

Setiap bagian diuji terpisah untuk memastikan fungsinya berjalan sesuai dengan kebutuhan sistem. Verifikasi dilakukan untuk memastikan setiap bagian telah berfungsi dengan baik sebelum sinkronisasi menjadi satu sistem pemantauan keamanan rumah yang utuh.

#### 5. Pemeliharaan

Sistem yang telah selesai mulai dijalankan dan digunakan oleh *user*. Tahap pemeliharaan memungkinkan pengembang untuk melakukan perbaikan jika ditemukan kesalahan yang tidak terdeteksi pada tahap sebelumnya, serta memastikan sistem pemantauan keamanan tetap berjalan dengan baik dalam jangka panjang (Bagoes Satria & Ardiansyah, 2023).

### 2.2.7 UML (*Unified Modeling Language*)

*Unified Modeling Language* (UML) merupakan bahasa standar untuk merepresentasikan visual dari sistem perangkat lunak. UML digunakan untuk memvisualisasikan, merancang, membangun, dan mendokumentasikan sistem, sehingga mempermudah kerja pengembang. Beberapa jenis diagram UML yang umum digunakan antara lain *Use Case Diagram* untuk menunjukkan fungsi sistem dan aktornya. *Activity Diagram* untuk menggambarkan alur proses yang terjadi dalam sistem. *Sequence Diagram* untuk memperlihatkan interaksi antar objek berdasarkan urutan waktu. *Class Diagram* digunakan untuk menunjukkan struktur sistem, mencakup hubungan antar class, package, maupun objek di dalamnya (Khairunnisa et al., 2024).

#### 1. *Use Case Diagram*

*Use Case Diagram* adalah salah satu bentuk pemodelan dalam *Unified Modeling Language* (UML) yang merupakan representasi visual dari fungsionalitas yang diharapkan dari suatu sistem, yang menggambarkan interaksi antara aktor dan sistem tersebut. Aktor dalam *use case* merepresentasikan entitas, baik berupa individu (manusia) maupun sistem eksternal, yang berperan dalam menjalankan atau berinteraksi dengan fungsi-fungsi yang terdapat dalam sistem, setiap aktor berperan dalam menunjukkan interaksi entitas dengan layanan sistem.

Tabel 2. 2 Simbol *Use Case Diagram*

Sumber: (Ramdany et al., 2024)

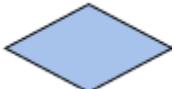
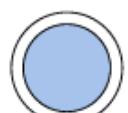
No	Simbol	Keterangan
1.	 Actor	Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i> .
2.	 Use Case	Abstraksi dan interaksi antara sistem dan <i>aktor</i> .
3.		Abstraksi dari penghubung antara <i>aktor</i> dengan <i>use case</i> .
4.	 Generalization	Menunjukkan spesialisasi <i>aktor</i> untuk dapat berpartisipasi dengan <i>use case</i> .
5.	 Include	Menunjukkan <i>use case</i> seluruhnya fungsionalitas dari <i>use case</i> lainnya.
6.	 Extend	Menunjukkan bahwa <i>use case</i> merupakan fungsional tambahan jika kondisi tertentu terpenuhi.

## 2. *Activity Diagram*

*Activity Diagram* digunakan untuk memvisualisasikan alur proses suatu sistem informasi. Diagram ini menjelaskan dari mana alur kerja dimulai, dan berakhir, aktivitas apa yang terjadi selama proses berlangsung, serta urutan dari setiap aktivitas. *Activity Diagram* memungkinkan pemodelan proses yang berjalan secara paralel hingga aktivitas yang kompleks digambarkan sistematis.

Tabel 2. 3 Simbol *Activity Diagram*

Sumber: (Ramdany et al., 2024)

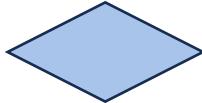
No	Simbol	Keterangan
1.	 Status awal	Sebuah diagram aktivitas memiliki status awal.
2.	 Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
3.	 Percabangan	Percabangan dimana ada pilihan aktivitas yang lebih dari satu
4.	 Status akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
5.	 Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

### 3. Class Diagram

*Class Diagram* merupakan representasi visual yang menggambarkan hubungan antar kelas serta detail atribut dan fungsi masing masing kelas dalam model perancangan suatu sistem. *Diagram* ini menunjukkan struktur sistem beserta aturan dan tanggung jawab dari setiap entitas yang membentuk perilaku sistem. *Class Diagram* berfungsi untuk menggambarkan struktur internal program berdasarkan jenis objek yang dibentuk, dan menunjukkan alur kerja yang berkaitan dengan desain *database* dalam sistem yang akan dikembangkan.

Tabel 2. 4 Simbol *Class Diagram*

Sumber: (Ramdany et al., 2024)

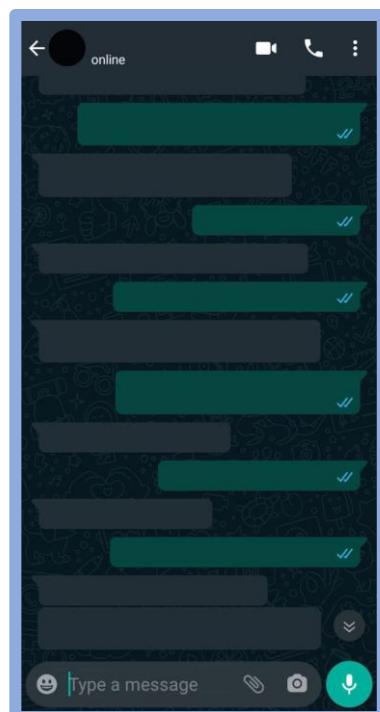
No	Simbol	Keterangan
1.	Generalization 	Hubungan dimana objek anak ( <i>descendant</i> ) berbagai perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> )
2.	Nary Association 	Upaya untuk menghindari asosiasi dengan lebih dari dua objek.
3.	Class 	Himpunan dari objek- objek yang berbagi atribut serta operasi yang sama.
4.	Collaboration 	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
5.	Realization 	Operasi yang benar benar dilakukan oleh suatu objek
6.	Dependency 	Hubungan dimana perubahan yang terjadi pada suatu elmen mandiri ( <i>independent</i> ) mempengaruhi elmen yang bergantung pada elmen yang tidak mandiri.
7.	Association 	Apa yang menghubungkan antara objek satu dengan objek lainnya.

### 2.2.8 Internet of Things (IoT)

*Internet of Things* (IoT) adalah konsep teknologi perangkat fisik seperti *sensor*, *akuator*, dan *mikrokontroler* untuk saling terhubung dan berkomunikasi melalui jaringan internet tanpa campur tangan manusia secara langsung. (Denta Widyapramana et al., 2021). Teknologi ini memungkinkan integrasi antara perangkat keras dan perangkat lunak untuk mengumpulkan, mengirim, dan bertukar data secara *real-time*. IoT dasar sistem, mengintegrasikan ESP32, sensor *Fingerprint*, kamera dengan server MQTT dan web *monitoring* untuk kendali pintu jarak jauh, serta tempat tersimpannya data log aktifitas akses rumah.

### 2.2.9 WhatsApp

WhatsApp merupakan aplikasi pesan instan yang dikembangkan oleh Jan Koum dan Brian Acton pada tahun 2009. Saat ini, WhatsApp digunakan oleh sekitar 124 juta *user* di Indonesia atau 83% *user* internet, dan berbeda dari SMS, aplikasi ini mengandalkan koneksi internet untuk mengirim pesan.



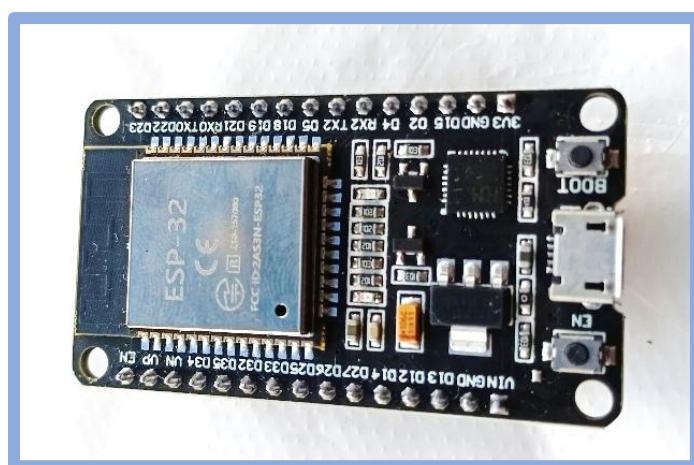
Gambar 2. 6 Aplikasi WhatsApp

Sumber: Tampilan Aplikasi WhatsApp

Notifikasi WhatsApp merupakan pesan otomatis yang dikirim secara langsung melalui aplikasi WhatsApp guna menyampaikan informasi penting kepada *user* secara cepat dan *real-time*. WhatsApp adalah aplikasi berbasis internet yang memiliki potensi besar untuk dimanfaatkan sebagai sarana komunikasi digital yang praktis, cepat, dan mudah diakses oleh berbagai kalangan *user* (Raharti, 2019). *User* an WhatsApp sebagai media notifikasi dalam sistem berbasis IoT dapat meningkatkan efektivitas komunikasi karena WhatsApp merupakan platform yang sangat umum digunakan di masyarakat. Keterbatasan *user* an platform Telegram yang kurang populer menimbulkan kendala dalam penerapannya, sehingga penting untuk mempertimbangkan alternatif seperti WhatsApp untuk meningkatkan aksesibilitas dan penerapan teknologi (Mardhatillah et al., 2024).

### 2.2.10 ESP32

ESP32 merupakan generasi lanjutan dari ESP8266 yang dikembangkan oleh Espressif Systems. Mikrokontroler ini dilengkapi dengan modul WiFi terintegrasi di dalam chip-nya, sehingga sangat mendukung dalam pengembangan aplikasi berbasis IoT. Pin pada ESP32 dapat difungsikan sebagai input maupun output, yang memungkinkan *user* untuk menghubungkannya dengan berbagai perangkat seperti LCD, lampu, atau bahkan motor DC (Ardiansah et al., 2024).



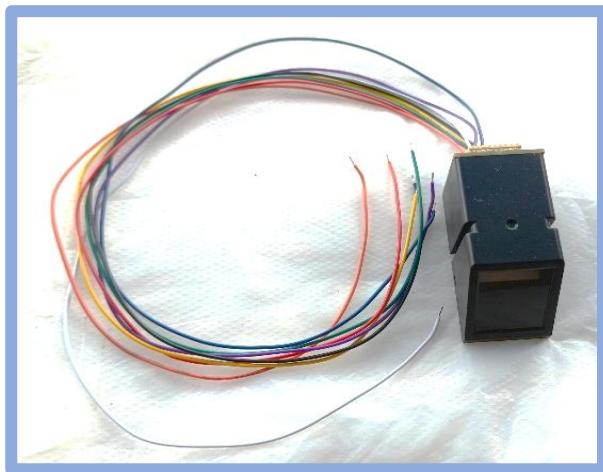
Gambar 2. 7 ESP32

Sumber: Penulis, 2025

Perangkat ini memiliki RAM dan kecepatan prosesor yang lebih tinggi dibanding pendahulunya, sehingga mampu menangani lebih banyak komponen secara efisien. ESP32 mendukung komunikasi serial dan memiliki GPIO (*General Purpose Input Output*) yang fleksibel, sehingga mudah diintegrasikan dengan perangkat seperti sensor sidik jari (*Fingerprint*), *relay*, LCD, dan komponen lainnya. Mikrokontroler ini juga telah banyak digunakan dalam proyek IoT *real-time*, termasuk yang berbasis *Flask*, MQTT, maupun HTTP API menjadikannya pilihan yang andal dan serbaguna dalam pengembangan sistem keamanan rumah.

### 2.2.11 Sensor *Fingerprint* AS608

Sensor *Fingerprint* AS608 diproduksi oleh perusahaan teknologi asal Tiongkok, *Hangzhou Synochip Company*. Modul ini terdiri dari sensor optik sidik jari, prosesor DSP, dan memori flash yang terintegrasi algoritma pengenalan sidik jari *Minutiae Based Matching*. AS608 mampu melakukan proses pengambilan gambar dan identifikasi sidik jari dengan efisien dan cepat, menjadikannya cocok untuk sistem autentikasi biometrik (Abroruddin et al., 2020).



Gambar 2. 8 Sensor *Fingerprint* AS608

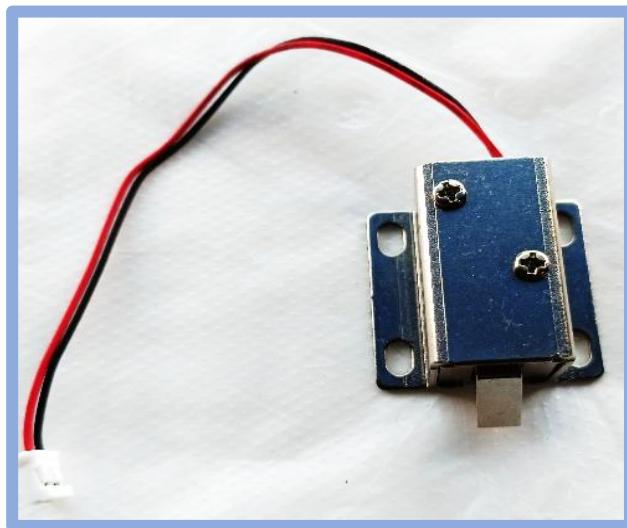
Sumber: Penulis, 2025

Dalam metode *Minutiae Based Matching*, fitur yang digunakan untuk membandingkan sidik jari meliputi jenis minutia, posisi koordinat relatif, serta sudut kemiringan dari minutia tersebut. Metode ini dianggap lebih fleksibel

dibandingkan *Pattern Based Matching* karena tetap mampu memberikan hasil yang optimal meskipun sebagian area sidik jari mengalami kerusakan seperti luka atau terkelupas (Anggya N D & Soetarmono, 2012).

### 2.2.12 *Solenoid Door Lock*

*Solenoid Door Lock* merupakan jenis *solenoid* yang dirancang khusus untuk sistem pengunci pintu elektronik. Mekanisme kerjanya adalah, dalam kondisi normal (tanpa aliran listrik), tuas berada dalam posisi terkunci atau memanjang. Namun, ketika dialiri tegangan listrik, tuas akan tertarik masuk sehingga posisi terkunci berubah menjadi terbuka. Di dalam solenoid ini terdapat gulungan kawat yang membungkus inti besi, saat arus listrik mengalir melalui gulungan tersebut, medan magnet terbentuk dan menghasilkan gaya tarik yang menarik inti besi ke dalam (Aulia Ramadini et al., 2025).



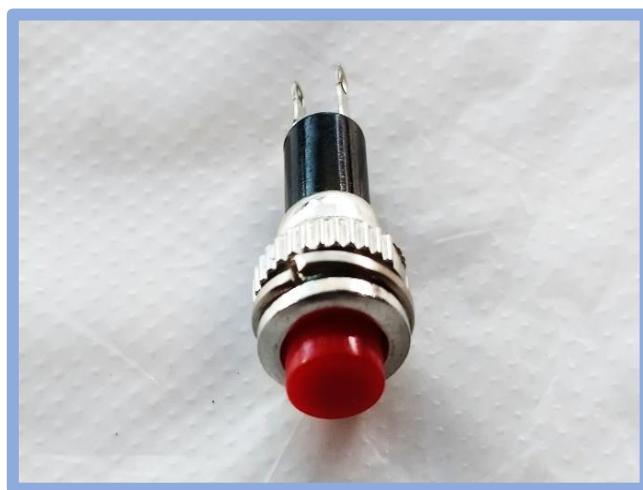
Gambar 2. 9 *Solenoid Door Lock*

Sumber: Penulis, 2025

*Solenoid Door Lock* memiliki dua jenis saklar, Yaitu Saklar Normally Open (NO), merupakan saklar yang dalam kondisi tidak aktif berada dalam keadaan terbuka, namun akan menutup saat diaktifkan, dan Saklar *Normally Closed* (NC) Merupakan saklar yang dalam kondisi tidak aktif berada dalam keadaan tertutup, namun akan terbuka saat diaktifkan (Royhan, 2021).

### 2.2.13 Push Button (Tombol Tekan)

*Push button* adalah jenis saklar tekan yang digunakan untuk menyambungkan atau memutuskan hubungan antar bagian dalam suatu instalasi listrik. Cara kerjanya dibedakan menjadi beberapa tipe, yaitu: *Normally Open (NO)* atau tombol start, dalam kondisi normal tidak terhubung, dan hanya akan mengalirkan arus listrik (*ON*) saat ditekan, lalu kembali terbuka ketika dilepas, *Normally Close (NC)* dikenal sebagai tombol stop, karena dalam keadaan normal terhubung, dan akan memutus aliran listrik (*OFF*) ketika ditekan, lalu kembali terhubung setelah dilepas, dan Tipe kombinasi NC dan *NO* memiliki empat terminal, di mana saat tombol tidak ditekan, satu pasangan kontak dalam kondisi NC dan pasangan lainnya NO. Ketika ditekan, posisi kontak akan berubah, kontak NC akan terbuka dan kontak NO akan tertutup (Tsalatsah & Ratama, 2024).



Gambar 2. 10 *Push Button* (Tombol Tekan)

Sumber: Penulis, 2025

Push button berfungsi sebagai kontrol manual dalam sistem *mikrokontroller*, seperti untuk mengunci atau membuka pintu, memicu input data memberikan sinyal interupsi. Saat ditekan, tombol menghasilkan sinyal logika digital, berupa tegangan *HIGH* (1) atau *LOW* (0) yang dibaca oleh pin input ESP32 atau Arduino. Untuk mencegah kesalahan akibat *bouncing*, push button dilengkapi dengan mekanisme *debouncing*, baik secara perangkat keras (dengan *kapasitor*) maupun perangkat lunak (melalui pemrograman *delay* atau *filtering*).

### 2.2.14 Kamera CCTV 360

CCTV 360° adalah jenis kamera pengawas yang memiliki kemampuan merekam dan menyatukan area secara menyeluruh dalam sudut pandang 360 derajat, baik menggunakan lensa fisheye maupun sistem pan-tilt. Teknologi ini memungkinkan *user* untuk melihat seluruh area di sekitarnya hanya dengan satu perangkat kamera, sehingga lebih efisien dibandingkan dengan *user* an banyak kamera konvensional. Dengan fitur pengawasan menyeluruh dan pemantauan *real-time*, CCTV 360° mendukung dokumentasi kejadian serta menjadi bukti visual jika terjadi percobaan akses yang mencurigakan.



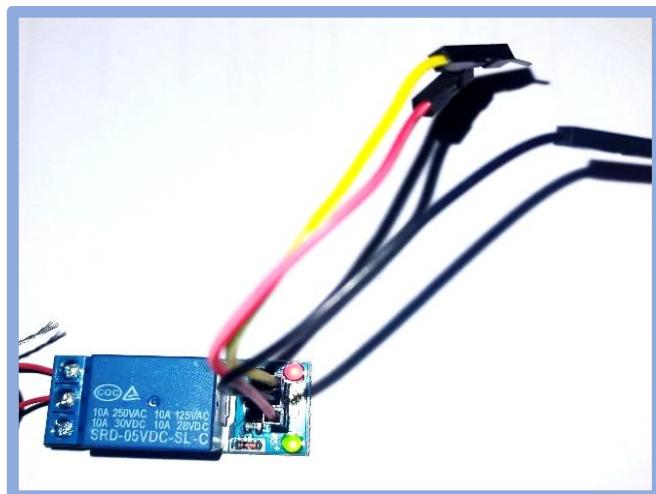
Gambar 2. 11 Kamera CCTV 360

Sumber: Penulis, 2025

CCTV yang terintegrasi dengan teknologi *Internet of Things* (IoT) memiliki keunggulan dalam hal keamanan yang lebih canggih dan kecerdasan sistem yang lebih tinggi. Selain itu, CCTV jenis ini dapat terhubung langsung dengan aplikasi pada smartphone, sehingga *user* dapat menjaga kondisi keamanan rumah secara fleksibel, kapan saja dan dari lokasi mana pun melalui jaringan internet (Pindarwati et al., 2022).

### 2.2.15 Relay Module

*Relay module* adalah komponen elektronika yang berfungsi sebagai saklar elektronik untuk mengontrol sirkuit arus tinggi (AC atau DC) dengan menggunakan sinyal arus rendah dari mikrokontroler seperti Arduino atau ESP32.



Gambar 2. 12 *Relay Module*

Sumber: Penulis, 2025

Relay bekerja berdasarkan prinsip elektromagnetik, yang ketika kumparan dalam relay dialiri arus, medan magnet yang dihasilkan akan menarik kontak saklar sehingga terjadi perubahan status (*ON* atau *OFF*). Relay adalah sebuah saklar yang dioperasikan secara elektrik. Umumnya, relay memanfaatkan prinsip elektromagnetik untuk mengaktifkan mekanisme sakelarnya. *Relay* sering digunakan untuk mengontrol suatu rangkaian menggunakan sinyal berdaya rendah, dengan tetap menjaga isolasi listrik antara rangkaian pengendali dan yang dikendalikan. *Relay* juga bermanfaat ketika diperlukan pengaturan rangkaian berbeda dengan satu sinyal pengontrol (Abraham Salihi et al., 2022).

### 2.2.16 Power Supply 12V 2A

*Power Supply* 12V 2A adalah perangkat catu daya yang mengubah tegangan AC (arus bolak-balik) dari jaringan listrik menjadi tegangan DC (arus searah) dengan besar 12 volt arus maksimum 2 ampere. Perangkat ini umum

digunakan dalam sistem elektronik dan IoT untuk menyalaikan komponen seperti kamera CCTV, relay, serta mikrokontroler. Dalam perancangan catu daya 12V 2A, digunakan IC regulator tegangan LM7812CT. Namun, karena kemampuan arus output IC regulator tidak mencapai 2A, diperlukan penambahan pass element berupa transistor (misalnya MJ2955 atau 2N3055) untuk meningkatkan kapasitas arus output. Konfigurasi ini memungkinkan catu daya menghasilkan arus hingga 2A dengan stabilitas tegangan yang baik (Fathoni, 2010).



Gambar 2. 13 *Power Supply* 12V 2A

Sumber: Penulis, 2025

*Power Supply* digunakan untuk memberikan daya utama pada *Solenoid Door Lock* (Jodi et al., 2022), serta dapat mendukung perangkat lain seperti mikrokontroler, sensor *Fingerprint*, dan modul kamera. *User* an *Power Supply* 12V 2A dirancang agar sistem tetap beroperasi stabil, meskipun terjadi fluktuasi kecil pada sumber daya, sehingga proses autentikasi dan kontrol akses pintu dapat berjalan dengan optimal tanpa gangguan. Tegangan sebesar 12 volt ini cukup untuk mengaktifkan mekanisme buka-tutup kunci secara dengan stabil.

#### 2.2.17 Kabel Micro USB (untuk NodeMCU)

Kabel *micro USB* digunakan sebagai media penghubung antara laptop dan board NodeMCU ESP32 untuk proses pemrograman atau pengunggahan kode (Wrastawa Ridwan et al., 2023).



Gambar 2. 14 Kabel Micro USB (untuk NodeMCU)

Sumber: Penulis, 2025

Fungsi utamanya adalah mentransfer program dari perangkat lunak seperti Arduino IDE ke memori NodeMCU, sehingga firmware dapat diunggah dan dijalankan sesuai instruksi yang telah ditulis.

#### 2.2.18 *Jumper wire*

*Jumper wire* adalah kabel untuk menghubungkan komponen dalam rangkaian elektronik. Kabel dapat digunakan tanpa penyolderan, sehingga sangat ideal untuk keperluan *prototyping* dan pengembangan sistem elektronik.



Gambar 2. 15 *Jumper Wire*

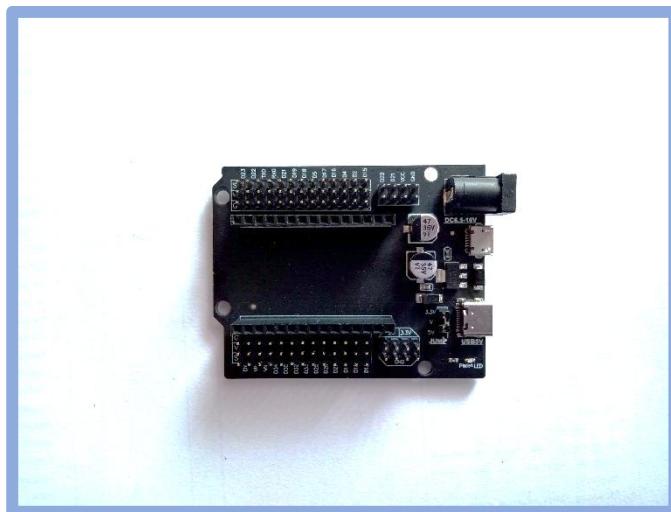
Sumber: Penulis, 2025

Kabel *jumper* merupakan kabel listrik yang memiliki konektor di kedua ujungnya dan berfungsi untuk menghubungkan dua komponen Arduino tanpa perlu proses penyolderan. Kabel ini berperan sebagai penghantar listrik yang memudahkan penyusunan dan pengujian rangkaian, khususnya pada breadboard.

atau perangkat *prototipe* lainnya. Ujung kabel tersedia dalam bentuk konektor jantan (*male*) yang dapat ditancapkan dan konektor betina (*female*) yang dapat menerima pin dari komponen lain (Hadriansa & Denis Prayogi, 2025).

### 2.2.19 *Base Plate* ESP32

*Base plate* ESP32 adalah papan ekspansi atau modul antarmuka yang dirancang khusus untuk memudahkan koneksi dan pengembangan sistem berbasis mikrokontroler ESP32. Base plate ini berfungsi sebagai media perantara antara board ESP32 dengan berbagai komponen eksternal, seperti sensor, aktuator, modul komunikasi, dan catu daya.



Gambar 2. 16 *Base Plate* ESP32

Sumber: Penulis, 2025

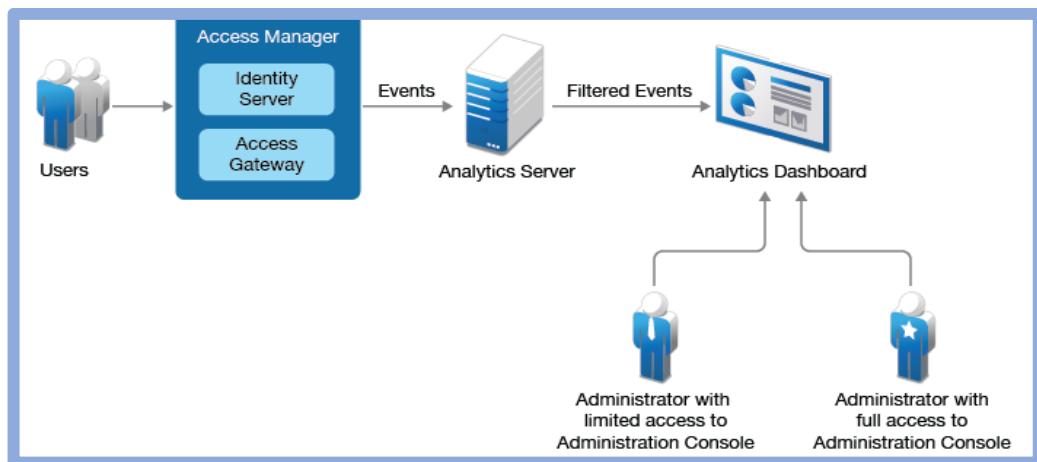
#### Fungsi utama *Base Plate* ESP32

- a. Mempermudah Koneksi: Menghadirkan susunan pin yang rapi untuk mengakses pin GPIO ESP32 tanpa perlu kabel *jumper* langsung ke *breadboard*.
- b. Ekspansi I/O: Memperluas jumlah port yang bisa digunakan untuk koneksi ke perangkat eksternal.
- c. Stabilitas dan Keamanan: Menyediakan konektor kuat dan rapi, meminimalkan kesalahan sambungan.

- d. Manajemen Daya: Beberapa base plate dilengkapi regulator tegangan, jack DC, atau port USB untuk suplai daya.
- e. *Debugging* Lebih Mudah: Dapat dilengkapi LED indikator dan port komunikasi untuk mempermudah proses pemrograman dan troubleshooting.

### 2.2.20 Website

*Website* merupakan aplikasi yang berisi dokumen-dokumen multimedia seperti gambar, teks, animasi, suara, dan video yang menggunakan protocol HTTP dan dapat diakses dengan *browser* (Arifin & Krisnadita, 2017).



Gambar 2. 17 Arsitektur Website

Sumber: (Wardhani et al., 2021)

Salah satu bagian dari Website Adalah *Dashboard* yang menampilkan tampilan visual untuk mempermudah pemahaman *user* pada sekumpulan informasi. *user* akan mendapatkan informasi lebih cepat dengan menggunakan *Dashboard*, kemudian *Dashboard* mengubah informasi yang disajikan dalam bentuk visual misalnya *grafik*, *log aktivitas*, dan status perangkat yang lebih mudah di mengerti oleh *user* (Dwi Bima Sakti et al., 2024). Terdapat tiga jenis *Dashboard* yang telah dikelompokkan menurut tingkatan manajemennya:

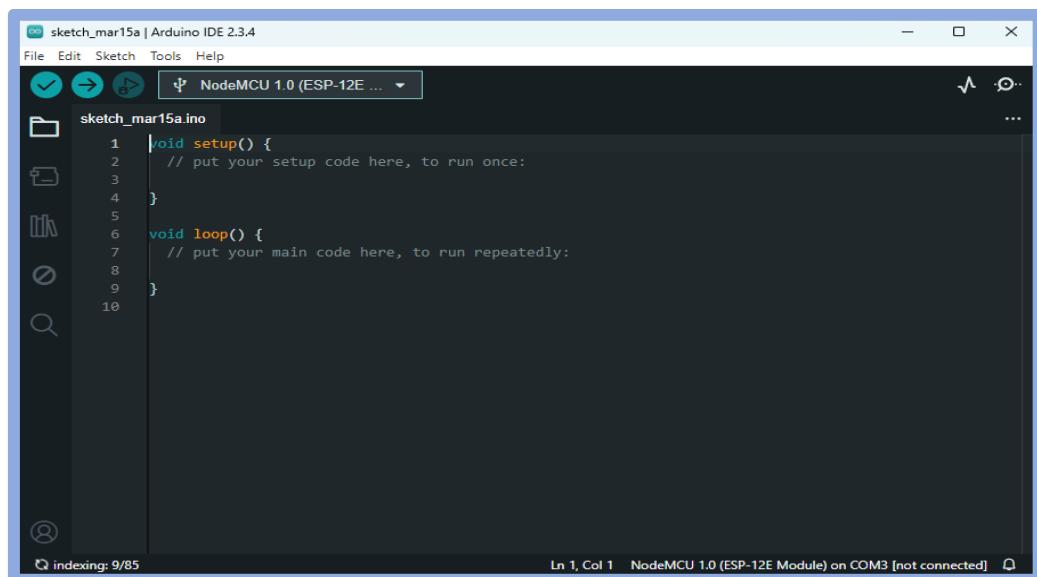
1. *Dashboard* strategis , yaitu *Dashboard* yang digunakan pada tingkat manajemen strategis untuk menyajikan informasi yang membantu dalam

pengambilan keputusan jangka Panjang, serta mengidentifikasi pelang bisnis, serta memberikan arah dalam mencapai tujuan.

2. *Dashboard* Taktis, *Dashboard* yang di fokuskan pada analisis guna mengidentifikasi sebab dari situasi atau kejadian tertentu.
3. *Dashboard* Oprasional, *Dashboard* untuk memantau aktivitas dan proses bisnis, aktivitas bisnis yang kompleks. *Dashboard* menampilkan pembaruan data berkala, harian, mingguan atau *real-time* guna menggambarkan status proses dalam suatu organisasi. (Rohmaniat & Heri Haerudin2, 2022).

### 2.2.21 Arduino IDE

*Arduino Integrated Development Environment* (IDE) adalah perangkat lunak lintas platform yang digunakan untuk menulis, mengompilasi, dan mengunggah kode program ke papan mikrokontroler.



Gambar 2. 18 Aplikasi Arduino IDE

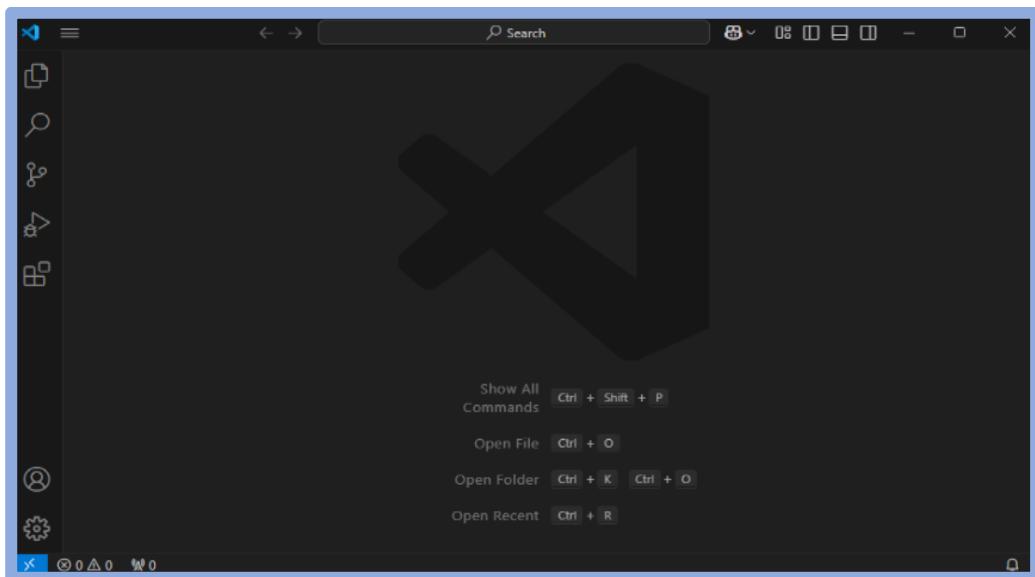
Sumber: Tampilan Aplikasi Arduino IDE

Untuk memprogram board Arduino, diperlukan *user* an aplikasi Arduino IDE (*Integrated Development Environment*). Aplikasi ini berfungsi untuk menulis, mengedit, dan membuka kode sumber yang digunakan dalam

pemrograman Arduino kode yang sering disebut sebagai representasi oleh para insinyur perangkat lunak. Kode sumber tersebut dikenal dengan istilah *sketch*, yaitu program yang memuat logika dasar serta algoritma yang akan diunggah ke dalam *chip mikrokontroler* untuk dijalankan (Abraham Salihi et al.,2022).

### 2.2.22 Visual Studio Code

*Visual Studio Code* adalah *Editor* kode sumber lintas platform yang dikembangkan oleh Microsoft. *Visual Studio Code* merupakan *Editor* kode sumber yang ringan namun memiliki kemampuan yang kuat, dapat dijalankan di desktop dan kompatibel dengan sistem operasi *Windows*, *macOS*, dan *Linux*. *Editor* ini dilengkapi dengan dukungan bawaan untuk bahasa pemrograman seperti *JavaScript*, *TypeScript*, dan *Node.js*, serta didukung oleh ekosistem ekstensi yang luas untuk bahasa lain seperti *C++*, *C#*, *Java*, *Python*, *PHP*, *Go*, maupun berbagai runtime seperti .NET dan *Unity* (Romzi & Kurniawan, 2020).



Gambar 2. 19 Aplikasi *Visual Studio Code*

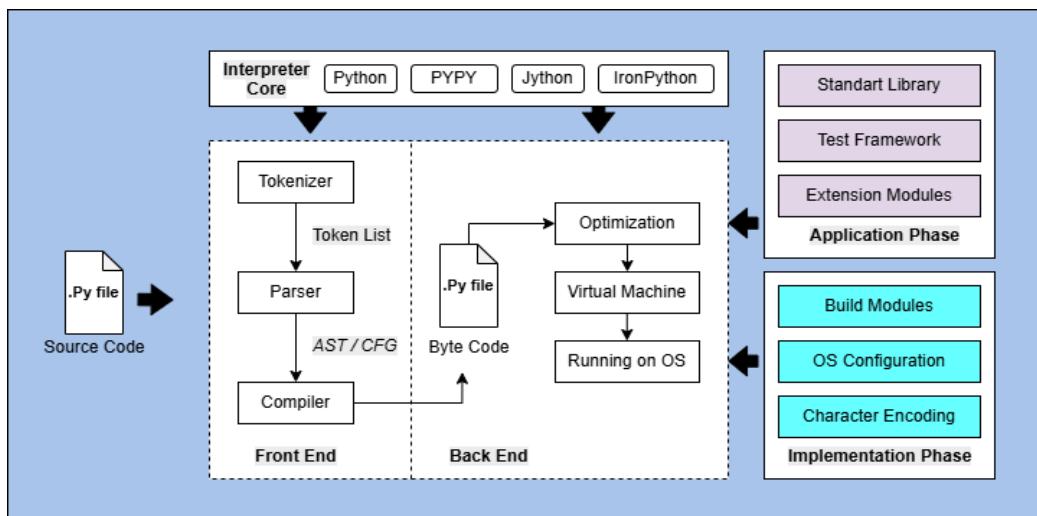
Sumber: Tampilan Aplikasi *Visual Studio Code*

*Visual Studio Code* (*VS Code*) salah satu *Editor* kode sumber yang banyak digunakan dalam pengembangan web karena kemampuannya yang fleksibel dan efisien. *VS Code* dilengkapi dengan berbagai fitur unggulan, termasuk *auto-*

*completion*, *syntax highlighting*, serta dukungan plugin dan ekstensi yang luas. Fitur *auto-completion* membantu dalam mempercepat proses penulisan kode dengan memberikan saran otomatis berdasarkan konteks, sementara *syntax highlighting* mempermudah pembacaan dan pemahaman struktur kode dengan membedakan elemen-elemen sintaks secara visual. Kedua fitur ini sangat berperan dalam meningkatkan efisiensi pemrograman dan mendukung proses pembelajaran, khususnya bagi *user* yang masih dalam tahap awal mempelajari pengembangan web (Priyoga Listyo Ananda et al., 2024).

### 2.2.23 Python

*Python* merupakan bahasa pemrograman yang memiliki kemampuan, mulai dari analisis data, perhitungan statistik yang rumit atau memakan waktu, pembuatan visualisasi, hingga penerapan algoritma *machine learning*.



Gambar 2. 20 Arsitektur *Python*

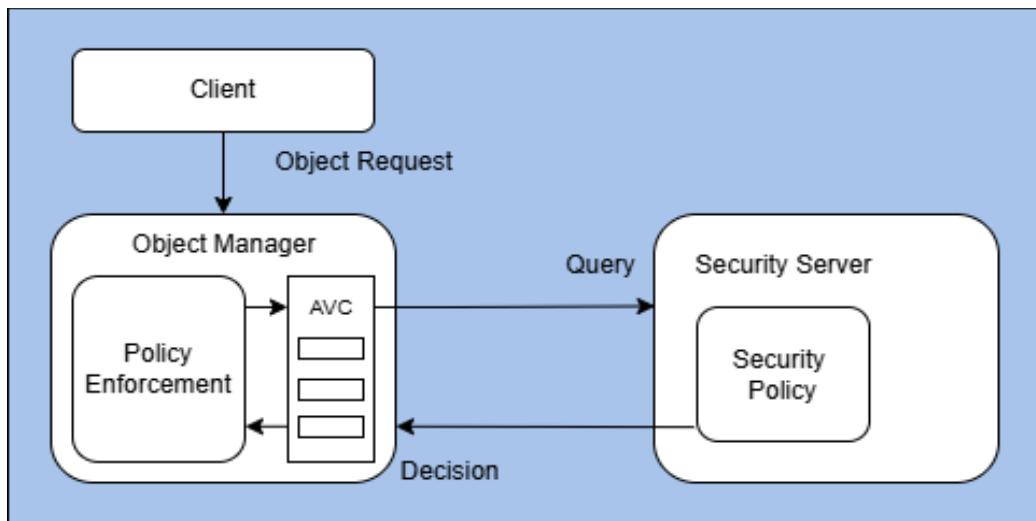
Sumber: Yang Feng, 2022

Keunggulan utama *Python* terletak pada efisiensinya dalam menghasilkan keluaran yang lebih cepat dan akurat dibandingkan pendekatan manual. Dalam konteks pembelajaran gambar, khususnya pada topik pengenalan wajah untuk *Machine Learning*, *Python* dapat digunakan untuk mengembangkan model yang mampu mengidentifikasi dan memverifikasi wajah berdasarkan data gambar.

Dengan menggunakan pustaka seperti *OpenCV* dan *TensorFlow*, *Python* memungkinkan pemrosesan gambar dan ekstraksi fitur wajah secara otomatis. Penggunaan *Python* dapat mempercepat proses pelatihan model *Machine Learning* dan menghasilkan akurasi yang lebih tinggi dalam mengenali wajah. Teknologi ini memberikan kontribusi besar dalam mempermudah pemahaman dan implementasi konsep-konsep pengolahan citra dan *Machine Learning*, serta meningkatkan kemampuan pengajaran pada topik yang berhubungan dengan pengenalan wajah dan aplikasi *Machine Learning* lainnya (Surbakti et al., 2024).

#### 2.2.24 *Flask*

*Flask* adalah *framework* web berbasis *Python* yang ringan, dirancang untuk membangun aplikasi web dengan cepat. *Flask* banyak digunakan dalam pengembangan sistem *monitoring* berbasis web karena kemudahannya dalam membangun *RESTful API*, mengelola *routing*, serta menyajikan antarmuka pengguna yang terhubung langsung dengan sistem *backend*.



Gambar 2. 21 Arsitektur Flask

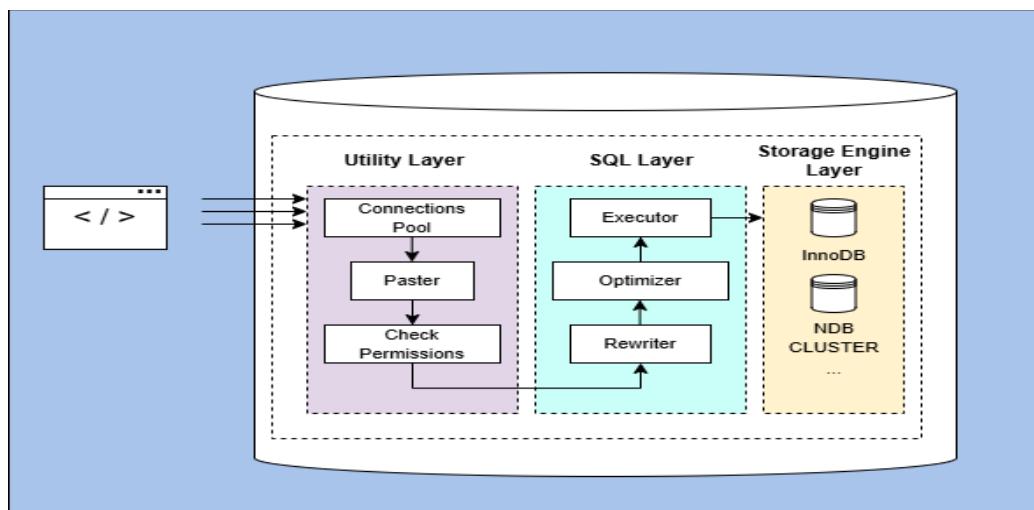
Sumber: (Ding et al., 2012)

*Flask framework* dapat digunakan sebagai *frontend* dan *backend* dalam pengembangan website. *Flask* dimanfaatkan sebagai *backend* utama untuk menangani komunikasi antara server lokal dengan sistem autentikasi biometrik

dan perangkat keras berbasis IoT. Penggunaan *Flask* sebagai *backend* memudahkan integrasi dengan model *Face Recognition* yang dikembangkan menggunakan *Python*, karena kesamaan bahasa pemrograman memungkinkan pemrosesan data biometrik dan pengiriman notifikasi dilakukan secara efisien dalam satu ekosistem (Giovanni Nathaniel et al., 2024).

### 2.2.25 MySQL

*MySQL* merupakan sistem manajemen basis data untuk melakukan proses pengaturan koleksi-koleksi struktur data baik yang meliputi proses pembuatan atau proses pengelolaan basis data (Suci et al., 2021). *MySQL* merupakan DBMS *open source* yang tersedia dalam dua jenis lisensi, yaitu perangkat lunak bebas (*Free Software*) dan perangkat lunak berpemilik *user* an terbatas (*Shareware*). Sebagai *database server* gratis, *MySQL* didistribusikan dengan lisensi GNU General Public License (GPL), memungkinkan penggunaan bebas untuk keperluan pribadi atau komersial tanpa biaya lisensi.



Gambar 2. 22 Arsitektur MySQL

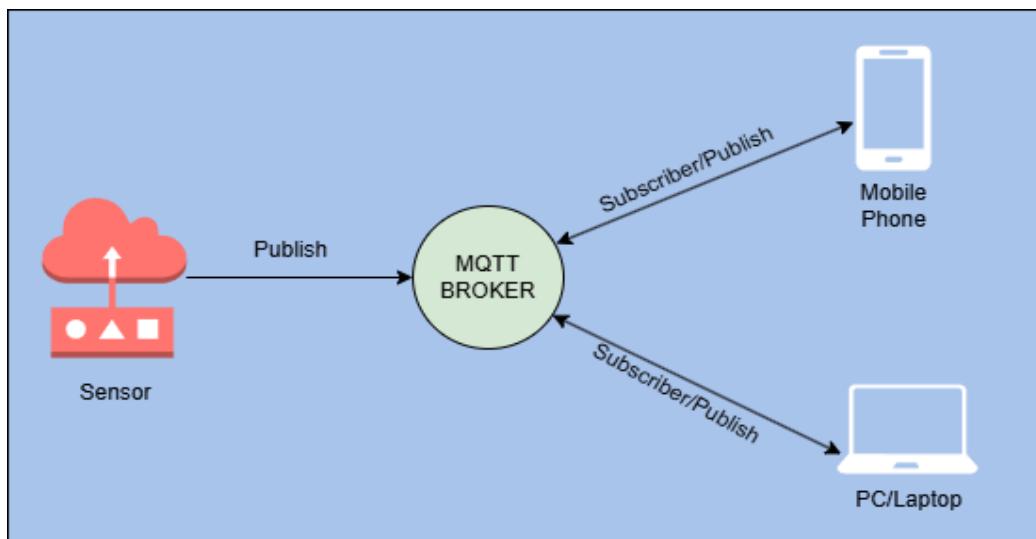
Sumber: Rishabh Gupta, 2023

MySQL dipilih oleh banyak pengembang karena kecepatan dan efisiensinya dalam menangani proyek kecil hingga menengah. Meskipun fiturnya tidak selengkap Oracle, MySQL tetap memenuhi kebutuhan banyak perusahaan

berkat fitur tingkat menengah yang memadai. Selain gratis dan bersifat *open-source*, MySQL mampu mengelola basis data besar hingga puluhan juta *record*, mendukung akses online dengan pengaturan hak akses, serta dilengkapi enkripsi kata sandi untuk menjaga keamanan. *MySQL* mendukung pengembangan aplikasi *desktop* maupun web menggunakan berbagai bahasa pemrograman seperti PHP, Java, dan C++. Tersedia pula plugin dan driver spesifik untuk integrasi dengan berbagai teknologi. *MySQL* dapat berjalan stabil di berbagai sistem operasi seperti *Windows*, *Linux*, dan *Unix*. Migrasi data antar sistem operasi pun dapat dilakukan dengan mudah dan tanpa hambatan (Kalsum Siregar et al., 2024).

### 2.2.26 MQTT (*Message Queuing Telemetry Transport*)

MQTT adalah protokol jaringan ringan berbasis model *publish-subscribe* yang dirancang khusus untuk komunikasi antar mesin (*Machine to Machine/M2M*) dalam kondisi terbatas, seperti perangkat dengan spesifikasi rendah atau jaringan dengan *bandwidth* minim. Memungkinkan pertukaran data tetap berjalan efisien pada perangkat berdaya rendah (Saiqul Umam et al., 2023).



Gambar 2. 23 Arsitektur MQTT

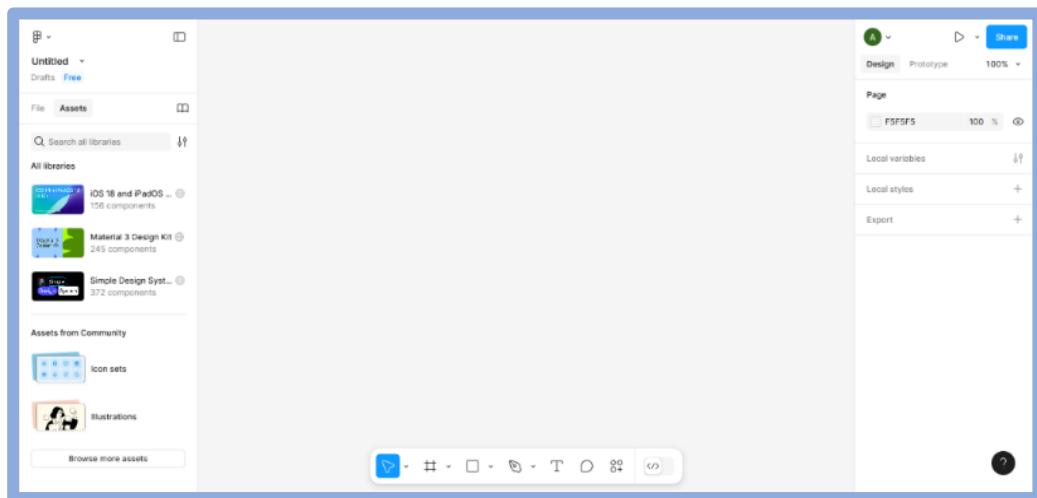
Sumber:(Mindriawan et al., 2018)

Protokol MQTT menggunakan metode komunikasi *publish/subscribe*, pesan yang dikirimkan oleh perangkat *publisher* akan diteruskan oleh *broker* ke

*subscriber* sesuai dengan topik yang telah ditentukan. Keunggulan MQTT adalah mampu menjamin pengiriman pesan meskipun koneksi empat terputus, sehingga komunikasi antar perangkat IoT tetap andal dan efisien (Abilovani et al., 2018).

### 2.2.27 Figma

Figma adalah alat desain antarmuka (UI/UX) berbasis web yang memungkinkan kolaborasi secara *real-time*. Figma sering digunakan untuk merancang tampilan aplikasi sebelum masuk ke tahap implementasi.



Gambar 2. 24 Aplikasi Figma

Sumber: Tampilan Aplikasi Figma

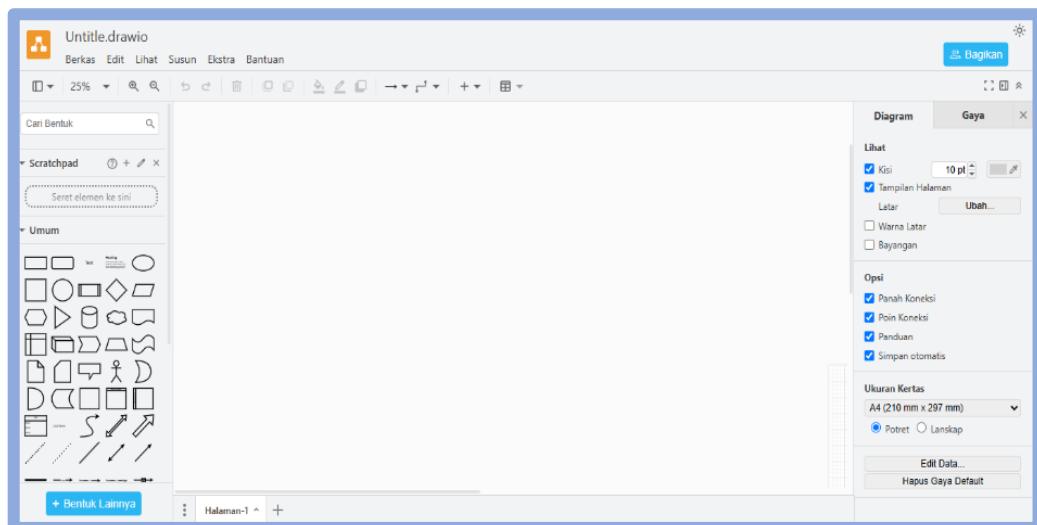
Figma memungkinkan kolaborasi tim langsung dalam proses desain. Para *desainer* dapat bekerja bersama secara *real-time*, saling memberikan komentar, masukan, dan mengembangkan desain secara simultan, sementara tim *front-end* dapat mengakses proyek melalui Figma. Keunggulan utama Figma antara lain:

- a. *Real-time functionality*, yaitu setiap perubahan disimpan otomatis dan dapat dilihat oleh anggota tim secara langsung.
- b. *Integrated prototyping*, memungkinkan pembuatan prototipe yang dapat langsung diuji pada perangkat tujuan.
- c. *Design library*, menyediakan kumpulan asset desain seperti komponen, gaya, variabel yang digunakan untuk proyek lainnya.

- d. *Easy sharing*, memberikan kemudahan dalam membagikan proyek desain kepada anggota tim atau pihak terkait (Ahmadiyah et al., 2024).

### 2.2.28 Draw.io

Draw.io adalah alat visualisasi berbasis web yang digunakan untuk membuat berbagai jenis diagram, seperti diagram UML, *flowchart*, dan *entity-relationship*.



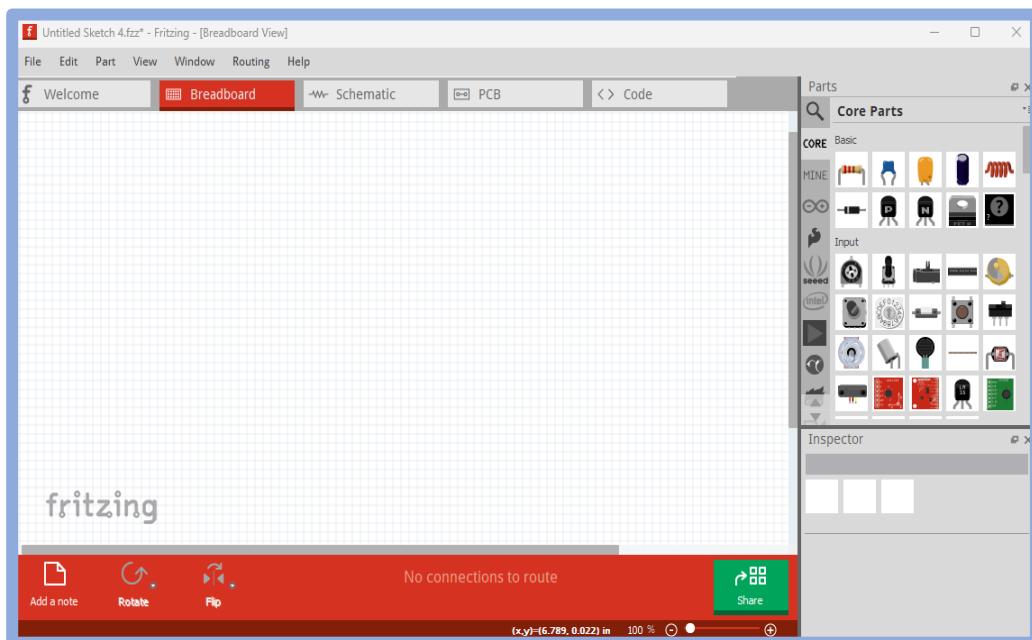
Gambar 2. 25 Aplikasi Draw.io

Sumber: Tampilan Aplikasi Draw.io

Aplikasi *Draw.io* memungkinkan pengguna membuat berbagai rancangan diagram tanpa perlu menginstal perangkat lunak tambahan, cukup dengan koneksi internet. Aplikasi ini bersifat fleksibel karena mendukung penyimpanan data di berbagai lokasi, termasuk media penyimpanan pribadi sesuai kebutuhan *user* (Yola Berliana Safira & Susi Wagiyati Purtiningrum, 2023). Dengan *Draw.io* *User* dapat menyimpan diagram yang telah dibuat dalam berbagai format file, seperti PNG, PDF, SVG, atau XML. *Draw.io* menyediakan integrasi dengan berbagai layanan *cloud*, seperti *Google Drive*, *OneDrive*, dan *Dropbox* sehingga memudahkan *user* untuk mengakses dan mengelola file. *Draw.io* mendukung kolaborasi *real-time* sehingga beberapa *user* dapat mengedit diagram bersamaan, ini sangat berguna dalam pengembangan sistem berbasis tim.

### 2.2.29 *Fritzing*

*Fritzing* merupakan perangkat lunak open source yang digunakan dalam perancangan perangkat keras elektronik. Yang berfungsi untuk memvisualisasikan rangkaian skematik, sehingga mempermudah proses perancangan, dokumentasi, serta implementasi sistem elektronik (Rizky et al., 2024). *Fritzing* ini tools yang efektif untuk pembelajaran elektronika. Perangkat lunak ini dapat dijalankan pada berbagai sistem operasi seperti GNU/Linux maupun *Microsoft Windows*. Setiap versi perangkat lunak memiliki keunggulan masing-masing tergantung pada kebutuhan pengguna. Adapun hal menarik dari *Fritzing*, yaitu menggabungkan pendekatan visual yang ramah *user* (human-oriented) dengan struktur teknis yang mendekati mesin (machine-oriented), sehingga mempermudah pemahaman dalam konsep perancangan rangkaian elektronik (Fajrin & Yenni, 2021).



Gambar 2. 26 Aplikasi *Fritzing*

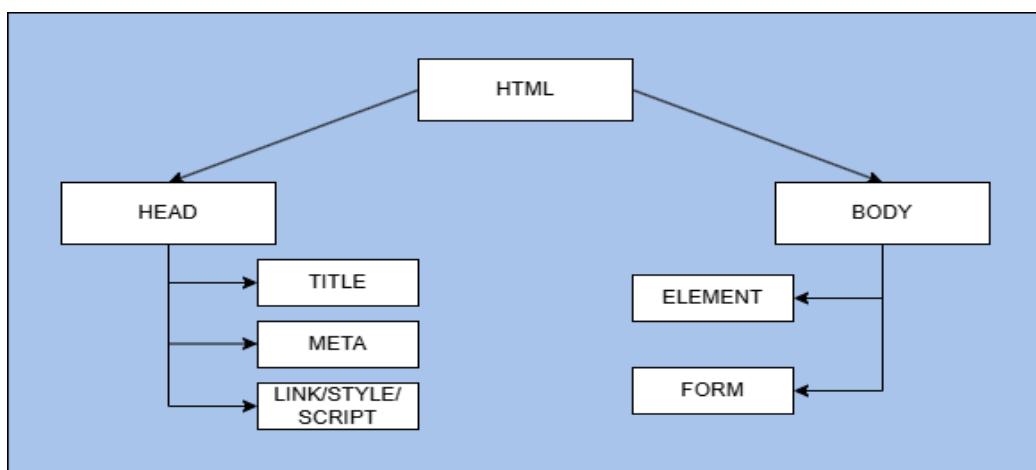
Sumber: Tampilan Aplikasi *Fritzing*

*Fritzing* banyak digunakan untuk pengembangan sistem *Internet of Things* (IoT), terutama pada tahap perancangan prototipe perangkat keras. Dalam IoT, *Fritzing* akan mempermudah visualisasi koneksi antara mikrokontroler seperti ESP32 atau Arduino dengan berbagai sensor serta aktuator. *User* dapat

merancang rangkaian breadboard, skematik elektronik, dan juga tata letak PCB secara intuitif, sehingga akan mempercepat proses pengembangan sistem IoT sebelum masuk pada tahap implementasi fisik.

### 2.2.30 HTML (*Cascading Style Sheets*)

HTML adalah bahasa untuk menyusun konten dan elemen halaman web, misalnya teks, gambar, tautan, tabel, dan lainnya. HTML sebagai fondasi utama dalam pengembangan web, yang memungkinkan browser untuk menampilkan informasi terstruktur dan dapat diakses oleh *user*. HTML bukanlah bahasa pemrograman, tapi bahasa markup yang memberi penanda atau label pada bagian-bagian dokumen untuk bisa ditampilkan dengan benar di *web browser*.



Gambar 2. 27 Arsitektur HTML

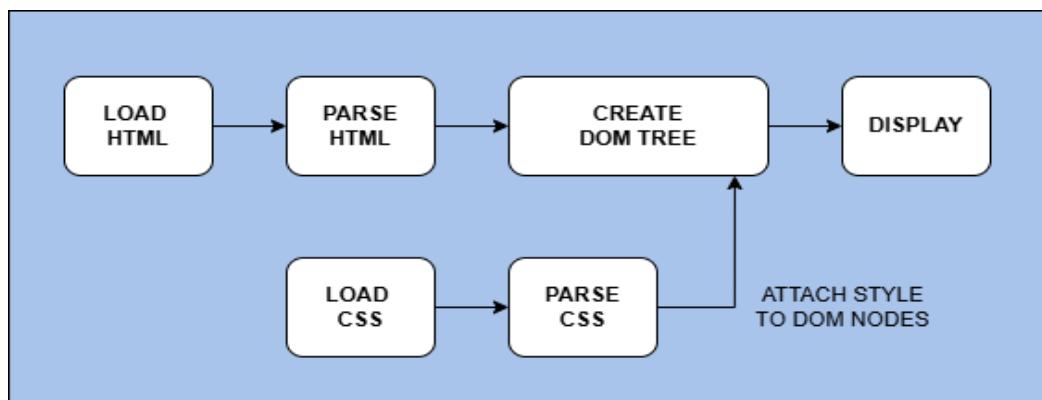
Sumber: Indra Tan, 2018

HTML terus mengalami perkembangan dari awal hingga versi terbaru, perkembangan HTML dimulai dari HTML v1.0, yang digunakan untuk mempermudah pertukaran dokumen antar ilmuwan. Versi ini mendukung elemen dasar seperti paragraf, heading, daftar, teks cetak tebal/miring, dan gambar tanpa pembungkus teks. Pada HTML v2.0 ada tambahan kemampuan yang menyajikan informasi dengan berbagai format dan formulir (form) yang memungkinkan *user* mengisi data seperti nama dan alamat. HTML v3.0 mulai dikenalkan elemen *figure* untuk menyisipkan gambar dan tabel dengan lebih fleksibel, dan

mendukung penulisan rumus matematika dalam dokumen. Berlanjut pada HTML v3.2 yang menambahkan teknologi pengaturan teks mengelilingi gambar, dukungan latar belakang, frame, style, tabel, dan *user an skrip* (script) untuk meningkatkan interaktivitas. Kemudian HTML v4.0 mengalami perubahan besar dengan peningkatan elemen gambar, tabel, teks, link, dan form. Versi ini menjadi standar resmi pada April 1998. Ada juga HTML v4.01 yang merupakan penyempurnaan dari versi 4.0 dengan memperbaiki berbagai kesalahan dan menetapkan standar elemen serta atribut HTML. Dan yang terakhir adalah HTML v5.0, ini merupakan versi modern dengan penyederhanaan sintaks, minim kesalahan, integrasi lebih baik dengan CSS dan JavaScript. Versi ini mendukung pengembangan web yang lebih dinamis dan interaktif sesuai dengan standar W3C dan IETF (Hendi Sama & Eric Hartanto, 2021).

### 2.2.31 CSS (*Cascading Style Sheets*)

CSS digunakan untuk mempercantik tampilan halaman web. Dengan CSS, pengembang dapat mengatur berbagai aspek visual seperti warna, ukuran *font*, jenis *font*, *margin*, *padding*, serta tata letak elemen pada halaman.



Gambar 2. 28 Arsitektur CSS

Sumber: My Skill, 2024

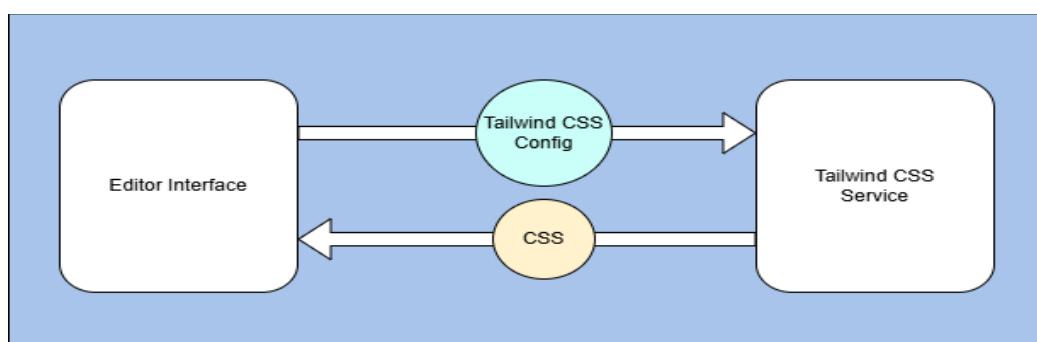
CSS adalah bahasa untuk mengelola tampilan dan tata letak pada elemen halaman web. CSS mempunyai konsep dasar yaitu *Selector* CSS yang digunakan dalam memilih elemen HTML yang ingin di percantik.

- a. *Selector* Elemen untuk memilih elemen berdasarkan pada nama elemennya, contohnya 'p' digunakan untuk paragraf.
- b. *Selector* Kelas, untuk memilih elemen berdasarkan dengan atribut kelas, ditandai dengan ( . ), misalnya : .kelas-saya.
- c. *Selector* ID, memilih elemen berdasarkan pada ID unik, pagar adalah tandanya (#), misalnya : #footer.

Konsep lainnya adalah Properti CSS untuk mengontrol aspek elemen terpilih. Pada properti mempunyai nama yang menggambarkan sifat yang dapat diubah serta diikuti nilai. Misal nilai CSS diberikan pada properti kemudian menentukan hasil akhir dari properti. Nilainya bisa berupa angka, warna, kata kunci tertentu seperti *left*, *center*, atau *right* untuk tata letak (Sinlae et al., 2024).

### 2.2.32 *Tailwind CSS*

*Tailwind CSS* merupakan kerangka kerja dengan mengutamakan utilitas *control ekstenatif* elemen desain, misalnya pada *layout*, warna, spasi, dan tipografi. Tailwind CSS dominan karena penyesuaian tingkat tinggi, memungkinkan pengembang menyesuaikan gaya standar sesuai kebutuhan desain spesifik. Tingkat kenyamanan ini memungkinkan pembuatan antarmuka pengguna unik dan personal dengan tetap menjaga konsistensi tampilan (Sree & Mohan, 2024).



Gambar 2. 29 Arsitektur *Tailwind CSS*

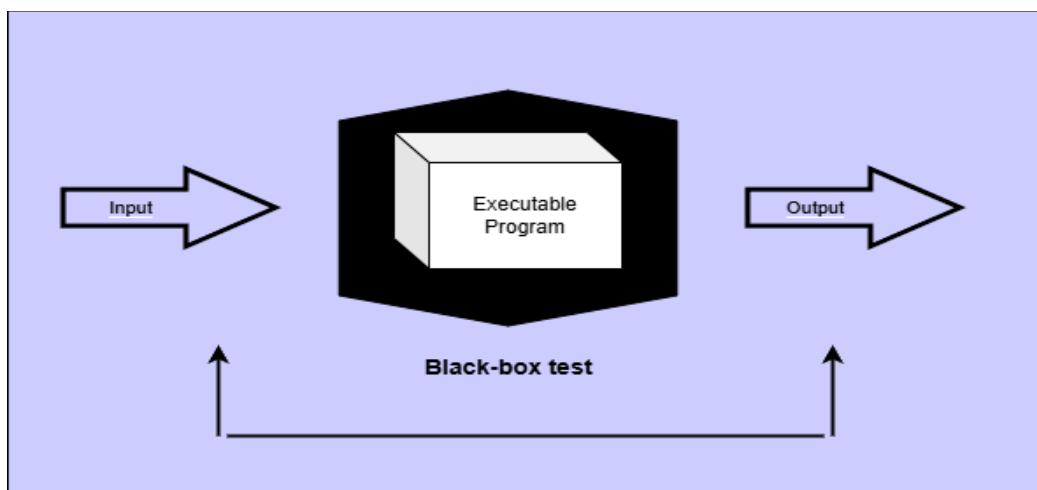
Sumber: Ivan Tajes Vidal, 2022

Pendekatan utilitas yang diterapkan oleh *Tailwind CSS* merepresentasikan pergeseran paradigma dalam pengembangan antarmuka *frontend*, karena

memberikan kebebasan tinggi bagi pengembang dalam menyesuaikan tampilan sesuai kebutuhan spesifik proyek. Dengan menekankan pada penggunaan kelas-kelas utilitas, *Tailwind CSS* memungkinkan pembuatan desain yang fleksibel tanpa dibatasi oleh struktur ketat *framework* konvensional (Dixit et al., 2024).

### 2.2.33 BlackBox Testing

*BlackBox Testing*, metode pengujian perangkat lunak yang berfokus pada fungsionalitas aplikasi tanpa memperhatikan struktur internal atau kode sumber. Pengujian dilakukan dengan memberikan input, mengamati *output* yang dihasilkan guna memastikan perangkat lunak berfungsi sesuai yang diharapkan.



Gambar 2. 30 Arsitektur *BlackBox Testing*

Sumber: (Laurie, 2006)

Tipe pengujian dalam metode *Black Box*:

1. *Equivalence Partitioning*, teknik membagi data input kelompok untuk mewakili kemungkinan masukan dan menghasilkan skenario pengujian.
2. *Comparison Testing*, menguji versi perangkat lunak menggunakan data yang sama untuk memastikan setiap versi menghasilkan output identik.
3. *Robustness Testing*, untuk memastikan perangkat lunak tidak mengalami kesalahan saat menerima masukan yang tidak sesuai.
4. *Performance Testing*, mengukur performa aplikasi terhadap tolok ukur tertentu, seperti kecepatan proses, penggunaan memori, dan aliran data.

5. *Requirement Testing*, menguji apakah perangkat lunak memenuhi kebutuhan atau spesifikasi yang telah ditentukan pada tahap awal analisis.
6. *Endurance Testing*, melibatkan pengujian berulang kali untuk menilai stabilitas dan ketahanan perangkat lunak dalam jangka waktu tertentu.
7. *Cause-Effect Relationship Testing*, menganalisis hubungan sebab-akibat dalam spesifikasi kebutuhan untuk mengidentifikasi bagian yang mungkin menimbulkan masalah dalam sistem (Hasibuan & Nurhaliza, 2024).

#### 2.2.34 Aplikasi V380 Pro

V380 Pro adalah aplikasi mobile untuk memantau CCTV berbasis IP melalui koneksi Wi-Fi. Aplikasi ini mendukung *live view*, pemutaran ulang, kontrol arah kamera, komunikasi dua arah, penyimpanan dengan *memory card*.



Gambar 2. 31 Aplikasi V380 Pro

Sumber: Tampilan Aplikasi V380 Pro

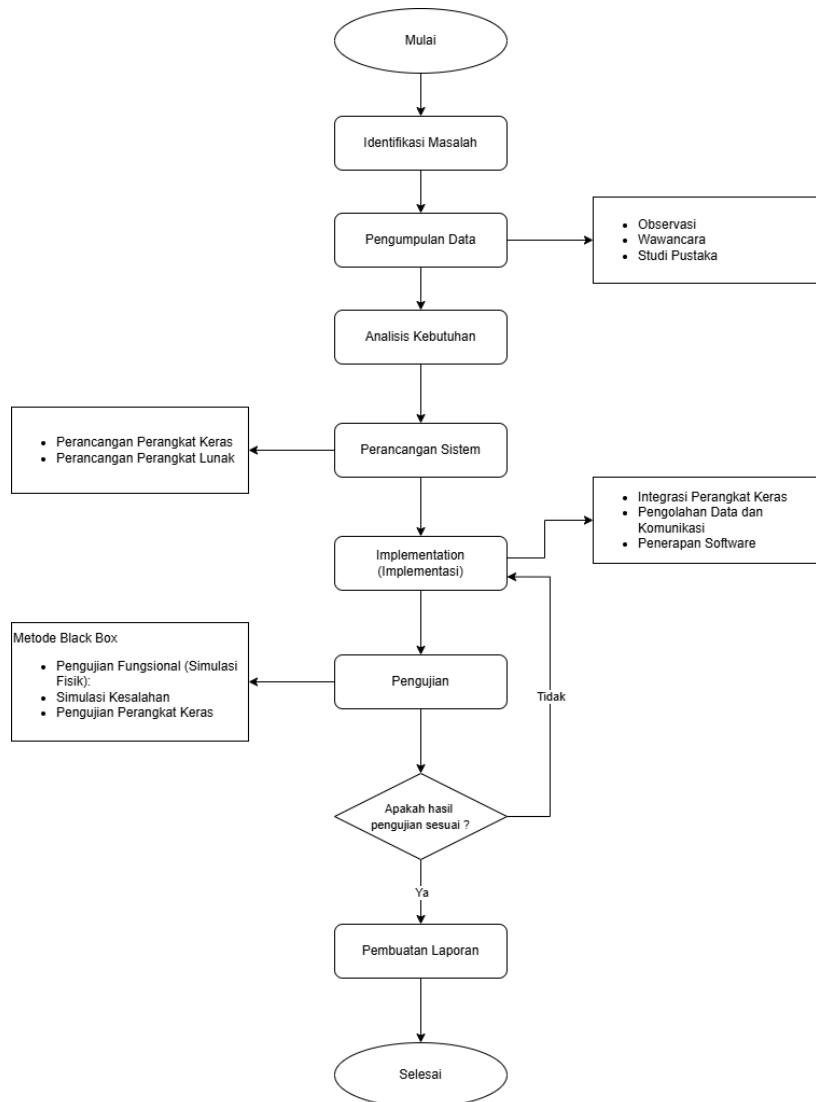
Aplikasi ini mendukung live view, pemutaran ulang, kontrol arah kamera, deteksi gerakan, notifikasi peringatan, komunikasi dua arah, dan penyimpanan *memory card*. Aplikasi bawaan ini tersedia untuk Android dan iOS, dilengkapi barcode pada perangkat untuk memudahkan pemindaian.

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Kerangka Berfikir

Kerangka pikir dalam penelitian ini disusun untuk menggambarkan alur pemikiran dan langkah sistematis yang akan ditempuh guna mencapai tujuan penelitian.



Gambar 3. 1 Kerangka Berfikir

### **3.2 Deskripsi**

Kerangka berpikir dalam penelitian ini disusun untuk menggambarkan tahapan sistematis dalam merancang dan membangun sistem *monitoring* keamanan rumah berbasis IoT. Tujuannya untuk menciptakan solusi keamanan yang modern, responsif, serta mampu memberikan notifikasi langsung kepada pemilik rumah saat terjadi akses gagal. Kerangka ini mencakup tahapan utama:

#### **3.2.1 Identifikasi Masalah**

Tahap awal pembuatan sistem *monitoring* keamanan rumah dimulai dengan identifikasi masalah. Penulis melakukan observasi lapangan, wawancara dengan perangkat Desa Wargaluyu, serta menyebarkan kuesioner kepada warga. Hasilnya menunjukkan adanya risiko pencurian, ketiadaan sistem log aktivitas, dan *user* an sistem keamanan konvensional yang dinilai kurang efektif.

#### **3.2.2 Pengumpulan Data**

Untuk mendukung kebutuhan sistem, peneliti menggunakan beberapa metode untuk mengumpulkan informasi melalui:

1. Observasi langsung terhadap sistem keamanan rumah warga Desa Wargaluyu, untuk mengetahui kondisi nyata di lapangan.
2. Wawancara dengan ketua perangkat desa, ketua rt, dan salah satu pemilik rumah guna menggali kebutuhan dalam pengelolaan akses rumah.
3. Penyebaran kuesioner kepada warga sebagai data pendukung untuk mengetahui persepsi dan harapan terhadap sistem keamanan.
4. Studi pustaka dari jurnal, artikel, terkait biometrik *Fingerprint*, *Face Recognition*, IoT ESP32, WhatsApp API, dan sistem *dashboard* web.

#### **3.2.3 Analisis Kebutuhan**

Dalam proses pembangunan sistem *Monitoring* keamanan rumah adalah melakukan analisis kebutuhan secara menyeluruh. Tahapan ini bertujuan untuk

mengidentifikasi dan memahami kebutuhan aktual *user* terhadap sistem keamanan pintu berbasis teknologi, yang akan menjadi dasar dalam perancangan dan pengembangan sistem yang aman dan responsif. Analisis kebutuhan dilakukan melalui beberapa metode, yaitu:

1. Analisis Sistem yang Berjalan

Analisis terhadap sistem keamanan rumah di Desa Wargaluyu menunjukkan bahwa masyarakat masih mengandalkan kunci manual yang rentan terhadap kehilangan, duplikasi, dan tidak memiliki fitur pemantauan akses. Hasil observasi dan wawancara mengungkapkan bahwa belum ada penerapan autentikasi biometrik maupun integrasi teknologi. Maka, dibutuhkan sistem keamanan berbasis IoT yang dilengkapi autentikasi biometrik, notifikasi otomatis, dan *monitoring real-time*.

2. Analisis Kebutuhan Sistem

Setelah melakukan analisis terhadap sistem yang berjalan, langkah selanjutnya adalah melakukan analisis kebutuhan sistem. Analisis ini digunakan sebagai dasar dalam perancangan dan pembuatan sistem.

- a. Kebutuhan Fungsionalitas

- 1) Autentikasi *Fingerprint*, digunakan untuk mengenali dan mencocokkan sidik jari *user* terdaftar untuk membuka pintu.
- 2) Autentikasi *Face Recognition*, mengenali wajah *user* melalui kamera yang terintegrasi *Deep Learning* untuk membuka pintu.
- 3) Notifikasi *Real-time*, sistem mengirimkan notifikasi melalui WhatsApp kepada pemilik rumah saat terjadi akses gagal.
- 4) *Monitoring Akses*, sistem menyediakan *dashboard web* untuk melihat riwayat akses, status kunci, dan manajemen *user*.
- 5) Manajemen Data *User*, sistem dapat menambahkan, menghapus, dan memperbarui data sidik jari dan wajah pengguna.

- b. Kebutuhan *Non-Fungsional*

- 1) Perangkat Keras, daftar kebutuhan perangkat keras yang digunakan dalam pengembangan sistem, terdiri dari laptop untuk pemrograman dan serta seluruh perangkat keras komponen IoT.

Tabel 3. 1 Kebutuhan Perangkat Keras

<i>Processor</i>	AMD Ryzen 3 3250U with Radeon Graphics
<i>Hard Disk</i>	500 GB
<i>Memory</i>	12 GB
<i>VGA</i>	2 GB
<i>Monitor</i>	Resolusi 1366x768px
<i>NodeMCU</i>	ESP32
<i>CCTV</i>	360
<i>Sensor Fingerprint</i>	R307
Pengunci Pintu	Selenoid door lock
<i>Relay</i>	Module 4 Channel
<i>Alrm</i>	<i>Buzzer</i>
<i>LCD</i>	16 x 2
Tombol	Tombol <i>Push on</i>
Kabel penghubung nodemcu	Kabel <i>micro USB</i>
Kabel penhubung komponen	<i>Jumper Wire</i>
<i>Power Supply</i>	12V 2A

## 2) Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan dalam perancangan sistem monitoring keamanan rumah meliputi berbagai tools untuk pemrograman dan komunikasi data.

Tabel 3. 2 Kebutuhan Perangkat Lunak

Sistem Operasi	Windows 11 64 bit
Bahasa Pemrograman	<i>Python</i>
<i>Code Editor (mikrokontroler)</i>	Arduino IDE
<i>Code Editor Aplikasi Web</i>	<i>Visual studio code</i>
<i>Framework antarmuka dashboard</i>	<i>Twilio</i>
<i>Framework backend</i>	<i>Flask</i>

<i>UI Design</i>	Figma
Desain Perangkat Keras	<i>Fritzing</i>
<i>Desain UML</i>	Draw.io
Manajemen basis data	<i>MySQL</i>
Applikasi notifikasi	<i>WhatsApp</i>

### 3.2.4 Perancangan sistem

Pada tahap perancangan sistem, digunakan *draw.io* dan Figma dengan pemodelan UML untuk menggambarkan alur, entitas, dan interaksi *user*. Sebagai acuan perancangan, disusun blok diagram dan desain sistem untuk menjelaskan alur kerja serta hubungan antar komponen.

#### A. Blok Diagram Perancangan Alat *Monitoring*

Blok diagram ini menunjukkan hubungan antar komponen utama sistem mulai dari input sensor, pemrosesan, output kontrol pintu, cctv, serta notifikasi.

#### B. Desain Sistem *Monitoring*

Desain sistem ini memetakan alur input–proses–output secara detail sebagai dasar integrasi rangkaian dan *dashboard web*.

#### C. Perancangan Perangkat Keras

##### 1. Perancangan Rangkaian dengan *Fritzing*

Skema rangkaian disusun dengan *Fritzing* untuk memetakan hubungan komponen utama, NodeMCU ESP32, sensor sidik jari, cctv, *solenoid door lock, relai, buzzer, lcd*, agar integrasi perangkat berjalan sistematis.

##### 2. Perancangan Fisik/ Alat

Setelah skema elektronik selesai, perangkat keras dirakit dan terintegrasi pada base plat sebagai penopang utama agar sistem tersusun rapi.

#### D. Perancangan Perangkat Lunak

##### 1. Pemodelan Sistem Menggunakan UML

Perancangan sistem ini mencakup pembuatan *Use Case Diagram, Activity Diagram, dan Class Diagram* untuk menggambarkan alur, aktivitas, serta hubungan antar entitas dalam sistem.

a) *Use Case Diagram*

*Use Case Diagram* sistem *monitoring* keamanan rumah ini terdapat dua aktor, yaitu *User* dan *Admin*, yang dimaksud *user* dan *Admin* adalah satu orang yang sama, hanya saja dalam aksesnya dibagi menjadi dua bagian, yang berinteraksi dengan sistem melalui berbagai fungsionalitas:

1) *User*

- a. Scan Wajah
- b. Scan Sidik Jari
- c. Membuka Pintu
- d. Mengirimkan Notifikasi

2) *Admin*

- a. *Login*
- b. *Dashboard*
- c. Kelola CCTV
- d. Akses Pintu
- e. Setting WA
- f. *Logout*

b) *Activity Diagram*

*Activity Diagram* digunakan untuk menggambarkan alur aktivitas sistem. Inti dari *Activity Diagram* yang akan dibuat meliputi:

1) *Activity Diagram User*

- a. *Scan Wajah*
- b. *Scan Sidik Jari*
- c. Membuka Pintu

2) *Activity Diagram Admin*

- a. *Login* ke *Dashboard*
- b. Kelola CCTV
- c. Manajemen Akses Pintu (Tambah, Hapus, *Update*).

c) *Class Diagram*

*Class Diagram* ini menggambarkan struktur data dan hubungan antar entitas dalam sistem *monitoring* keamanan rumah. Terdapat empat entitas utama yaitu: *User*, *DoorAccess*, *Image*, *Cctv*, *Number Phone*.

## 2. Perancangan *Dashboard Monitoring*

Pada tahap perancangan ini, antarmuka *Dashboard Monitoring* dirancang menggunakan Figma, *dashboard* ini untuk melakukan pemantauan aktivitas, kontrol akses pintu, memantau CCTV *real-time*, dan pengaturan notifikasi WhatsApp, sesuai fungsionalitas yang telah dijabarkan pada *Use Case Diagram* sehingga mendukung kemudahan dalam penggunaan.

### 3.2.5 Implementasi

Pada tahap ini, perancangan perangkat keras diprogram menggunakan *Arduino* IDE untuk mengontrol proses autentikasi dan kontrol pintu. Aplikasi web dibangun secara teknis menggunakan framework Flask yang terhubung dengan *server* untuk mengelola komunikasi IoT dan menyajikan data *monitoring* secara *real-time*. Sistem ini memungkinkan *user* memantau aktivitas dan mengontrol akses pintu melalui web yang terintegrasi dengan notifikasi otomatis.

### 3.2.6 Pengujian

Dalam penelitian ini, di gunakan metode *Black Box* untuk menguji fungsionalitas sistem melalui tiga skenario utama: autentikasi sidik jari dan wajah terdaftar untuk memastikan pintu terbuka, status LCD tampil, dan log web tercatat secara *real-time*, simulasi input tidak terdaftar untuk menguji penolakan akses, notifikasi peringatan, dan pencatatan aktivitas serta pengujian perangkat keras untuk menjamin stabilitas semua komponen fisik selama proses berlangsung.

### 3.2.7 Pembuatan Laporan

Tahapan akhir dalam penelitian ini adalah penyusunan laporan, yang berisi penjelasan secara rinci mengenai setiap langkah yang telah dilakukan selama proses penelitian serta hasil yang berhasil dicapai. Penyusunan laporan ini bertujuan untuk menyampaikan informasi secara sistematis kepada pembaca mengenai keseluruhan proses dan temuan dari penelitian yang telah dilaksanakan.

## **BAB IV**

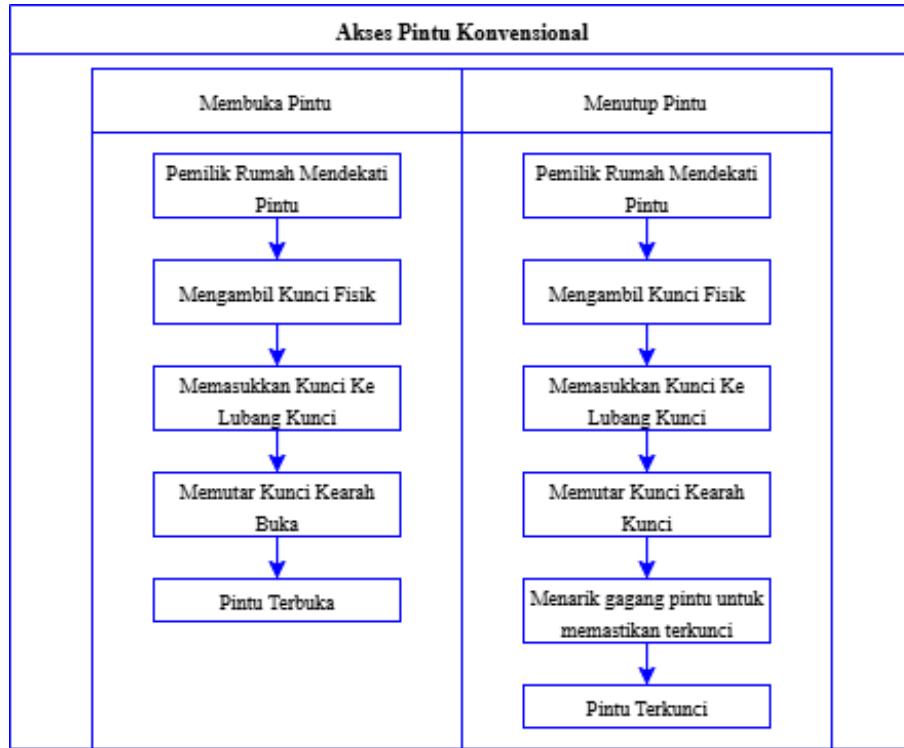
### **ANALISIS DAN PERANCANGAN**

#### **4.1 Analisis**

Analisis dilakukan sebagai tahapan awal dalam proses perancangan sistem untuk mengidentifikasi kebutuhan fungsional dan non-fungsional, baik dari sisi perangkat keras, perangkat lunak, maupun kebutuhan *user*. Tahapan ini untuk memahami sistem yang berjalan saat ini serta merumuskan solusi berbasis teknologi yang tepat. Analisis ini juga mencakup studi terhadap alur akses pintu secara manual, identifikasi permasalahan, serta kebutuhan integrasi sistem yang melibatkan sensor *Fingerprint*, *Face Recognition* berbasis *Machine Learning*, notifikasi WhatsApp, dan *dashboard web* untuk *monitoring secara real-time*. Dengan analisis yang tepat, sistem dapat dirancang lebih terstruktur dan responsif, sehingga dapat meningkatkan aspek keamanan, serta memastikan seluruh fitur yang dikembangkan benar-benar sesuai dengan kebutuhan dan perilaku pengguna.

##### **4.1.1 Analisis Masalah**

Langkah pertama pada tahap ini yaitu untuk mengidentifikasi permasalahan di lingkungan masyarakat Desa Wargaluyu terkait keamanan rumah. Proses dilakukan dengan mengamati fenomena yang terjadi, keterbatasan sistem keamanan konvensional, mengkaji seluruh kebutuhan sistem pengamanan *modern*. Sistem keamanan konvensional tidak memberikan respons cepat terhadap ancaman karena tidak terintegrasi dengan pemilik rumah. Tidak adanya fitur monitoring jarak jauh pada sistem konvensional menyulitkan pengguna untuk mengawasi kondisi rumah secara *real-time*. Setelah pintu dikunci menggunakan kunci manual, tidak tersedia pengawasan lanjutan yang terintegrasi dengan sistem keamanan otomatis. Untuk memperjelas alur akses pada sistem konvensional tersebut, berikut disajikan *flowmap* proses akses pintu manual yang umumnya masih digunakan oleh mayoritas masyarakat di Desa Wargaluyu.



Gambar 4. 1 *Flowmap* Akses Pintu Konvensional

*Flowmap* di atas menggambarkan proses akses pintu secara manual dengan interaksi langsung tanpa dukungan sistem keamanan otomatis. Dengan skenario akses manual, pengguna harus membuka pintu secara fisik tanpa adanya autentikasi biometrik dan kendali jarak jauh melalui sistem *monitoring*. Hal ini memiliki keterbatasan keamanan, sehingga dibutuhkan pengembangan sistem otomatis untuk meningkatkan perlindungan dan kenyamanan pengguna.

#### 4.1.2 Analisis Kebutuhan

Bagian ini membahas analisis kebutuhan sistem *monitoring* keamanan rumah, yang mencakup kebutuhan perangkat keras dan perangkat lunak.

##### 1. Kebutuhan *Hardware*

Sistem ini memerlukan perangkat keras untuk identifikasi, kontrol akses, dan pemantauan *real-time*. Komponen IoT terintegrasi melalui jaringan dan dikendalikan oleh sistem web. Di bawah ini merupakan *Hardware* yang digunakan:

- 1) ESP32, berfungsi sebagai pusat kendali yang mengatur komunikasi antara perangkat IoT, dan mengirim data ke *server* melalui koneksi wifi.

- 2) Sensor *Fingerprint*, digunakan untuk melakukan proses autentikasi *user* berdasarkan identifikasi sidik jari.
- 3) Kamera CCTV 360, CCTV terhubung ke wifi untuk *Face Recognition* digunakan untuk autentikasi wajah setelah *Fingerprint* berhasil.
- 4) *Relay*, Mengontrol kunci pintu elektronik (*solenoid door lock*).
- 5) *Power Supply* 12v, menyediakan daya untuk perangkat sistem.
- 6) *Buzzer*, Memberi feedback suara (akses ditolak/disetujui).
- 7) LCD 16x2, menampilkan informasi status.
- 8) *Jumper Wire*, menghubungkan antar komponen di PCB.

Tabel 4. 1 Kebutuhan *Hardware*

ESP32	1
Sensor <i>Fingerprint</i>	1
Kamera CCTV 360	1
<i>Relay</i> 4 channel	1
<i>Power Supply</i> 12v	2
<i>Buzzer</i>	1
<i>Push Button</i>	1
LCD 16x2	1
<i>Jumper Wire</i> 30 cm	50
<i>Base Plate</i>	1

## 2. Kebutuhan *Software*

Untuk mendukung pengembangan sistem, diperlukan *Software* yang berfungsi sebagai alat bantu dalam proses pemrograman dan pengelolaan data.

- 1) Arduino IDE, untuk memprogram ESP32 agar dapat membaca *Fingerprint*, berkomunikasi dengan *sensor*, dan mengaktifkan *relay*.
- 2) WhatsApp, digunakan untuk mengirim notifikasi ke *user* berupa pesan dan tautan yang tertuju pada web *monitoring*.
- 3) *Fritzing*, Alat visualisasi yang digunakan untuk merancang dan mendokumentasikan skema rangkaian elektronik.
- 4) *Flask*, *Framework backend* berbasis *Python* untuk membangun aplikasi web dan API *monitoring* IoT.

- 5) MySQL, Sistem manajemen basis data yang digunakan untuk menyimpan data dari sensor secara terstruktur.
- 6) MQTT, untuk komunikasi data *real-time* perangkat IoT dan *server*.
- 7) VS Code, *Editor* teks dan kode untuk menulis program antarmuka.
- 8) Draw.io, digunakan untuk membuat/merancang diagram UML.
- 9) Figma, *Platform* desain antarmuka *user* (UI) untuk merancang tampilan aplikasi *monitoring* .
- 10) Python, bahasa pemrograman *Machine Learning Face Recognition*.

#### **4.1.3 Analisis User**

Analisis *user* dalam sistem *monitoring* keamanan rumah ini merupakan hal tentang siapa yang akan menggunakan sistem serta bagaimana interaksi terhadap fitur yang tersedia. pengguna terdiri dari *user* yang melakukan proses autentikasi membuka pintu, serta *Admin* yang mengelola sistem melalui *dashboard* . Meskipun dibedakan berdasarkan fungsinya, kedua peran tersebut dapat dijalankan oleh orang yang sama. Maka sistem dirancang dengan praktis dari segi antarmuka. Perancangan sistem berfokus pada kemudahan akses, kejelasan tampilan *user interface*, dan fungsionalitas fitur yang mendukung keamanan rumah *real-time*. tujuannya menciptakan sistem yang memberi kenyamanan serta kontrol penuh *user* terhadap keamanan rumahnya.

#### **4.1.4 Interface User**

*Interface User* pada sistem *monitoring* keamanan rumah ini terdiri dari dua bagian utama, yaitu antarmuka perangkat fisik dan antarmuka berbasis web. Pembagian ini bertujuan untuk mendukung interaksi *user* secara langsung melalui perangkat, pengelolaan dan pemantauan jarak jauh melalui *dahsboard web*.

##### **1. Interface User Fisik**

Antarmuka perangkat fisik mencakup tampilan serta elemen interaktif yang terintegrasi langsung dengan alat, seperti:

- a. Tampilan LCD saat autentikasi
- b. *Buzzer* untuk notifikasi suara

- c. Sensor *Fingerprint*
- d. Kamera CCTV 360 untuk *Face Recognition*
- e. Push button untuk membuka pintu secara manual
- 2. *Interface User Digital (Dashboard Web)*

Antarmuka digital dirancang untuk memudahkan *Admin* dalam mengelola seluruh fungsi sistem melalui *dashboard web*, yang mencakup:

- a) *User Interface* Halaman *Login*
- b) *User Interface* Halaman *Dashboard Utama*
- c) *User Interface* Halaman *Monitoring CCTV*
- d) *User Interface* Halaman *User*
- e) *User Interface* Halaman Akses Pintu
- f) *User Interface* Halaman Whatsapp

#### **4.1.5 Fitur-Fitur**

Fitur-fitur dalam sistem *monitoring* keamanan rumah terbagi menjadi dua, yaitu fitur perangkat fisik dan fitur *Dashboard Web*, yang menunjukkan fungsi langsung pada alat saat uji coba di lapangan serta untuk kontrol jarak jauh.

##### **1. Fitur Perangkat Fisik**

Fitur pada perangkat fisik dirancang untuk mendukung autentikasi *user* serta memberikan respons otomatis terhadap akses pintu.

- a) Autentikasi *Fingerprint*
- b) *Face Recognition*
- c) LCD 16x2 *Display*
- d) *Buzzer*
- e) Tombol (*push button*)

##### **2. Fitur *Dashboard Web***

Fitur pada *dashboard digital* untuk mempermudah *Admin* dalam memantau sistem, mengelola data, serta menerima notifikasi secara *real-time*:

- a) Halaman *Login*
- b) Halaman utama
- c) Halaman untuk *monitoring CCTV*
- d) Halaman Pengguna

- e) Halaman log aktifitas akses pintu
- f) Manajemen *user*

#### **4.1.6 Analisis Data**

Untuk mendukung perancangan sistem *monitoring* keamanan rumah, penulis memerlukan data hasil pengujian perangkat serta interaksi *user*. Berikut jenis data yang digunakan dalam proses analisis dan pengembangan sistem:

1. Data Sistem, data sistem mencakup aktivitas yang tercatat otomatis pada *dashboard* seperti log autentikasi *Fingerprint ,face recognition*, dan notifikasi WhatsApp. Data ini digunakan untuk menganalisis performa sistem dalam mendeteksi, mencatat, merespon aktivitas secara *real-time*.
2. Data *User*, data ini diperoleh dari input *Admin* melalui *dashboard*, seperti tambah, hapus, dan pengaturan hak akses. Termasuk juga perilaku pengguna, seperti frekuensi autentikasi dan respons notifikasi, untuk menganalisis pola penggunaan sistem *monitoring* keamanan rumah.
3. Data Respons dan Kinerja, mencakup waktu respons sistem terhadap input *user* (*fingerprint* dan *face recognition*), kecepatan notifikasi, stabilitas koneksi ke *server*. Analisis ini untuk mengevaluasi performa sistem.

#### **4.1.7 Analisis Biaya**

Dalam perancangan dan pengembangan sistem *monitoring* keamanan rumah dibutuhkan sejumlah sumber daya, perangkat keras dan perangkat lunak. Biaya yang diperlukan meliputi komponen IoT, perangkat pendukung, layanan pengembangan, kebutuhan lainnya. Rincian biaya disajikan pada tabel berikut:

Tabel 4. 2 Analisis Biaya

No	Jenis Kebutuhan	Biaya
1.	Biaya ATK	Rp. 350.000
2.	Komponen IoT	Rp. 1. 500.000
3.	CCTV 360	Rp. 400.000
4.	Internet	Rp. 1.200.000

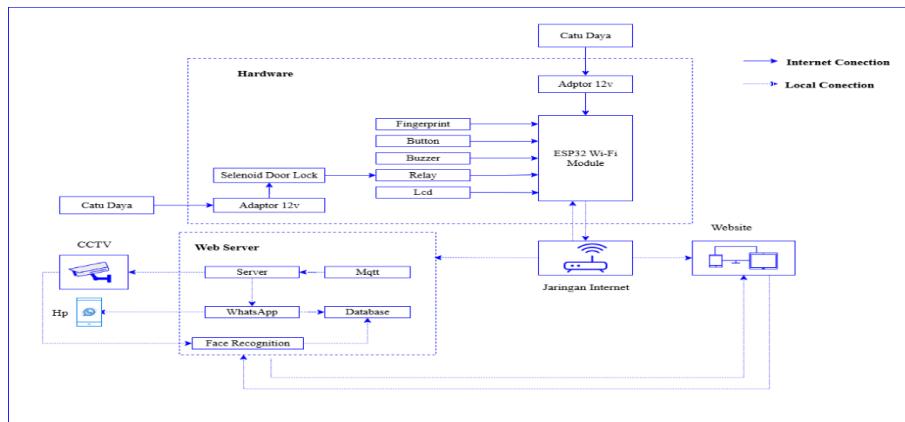
5.	programming	Rp. 5.000.000
6.	Desain	Rp. 1.000.000
Jumlah		Rp. 9.100.000

## 4.2 Perancangan

Sebelum tahap pembuatan sistem *monitoring* keamanan rumah, di perlukan perancangan sistem yang matang, meliputi blok diagram perancangan alat *monitoring* serta flowmap sitem dan dilanjutkan dengan rangkaian *Hardware* dengan aplikasi *Fritzing* dan perancangan *dashboard web* yang dimodelkan dengan UML, dan menyusun *Use Case Diagram*, *Class Diagram*, dan *Activity Diagram*.

#### 4.2.1 Blok Diagram Rancangan Alat *Monitoring* Keamanan Rumah

Blok diagram ini menunjukkan integrasi komponen sistem untuk autentikasi, kontrol akses, dan *monitoring* berbasis web.



Gambar 4. 2 Blok diagram perancangan alat *monitoring* keamanan rumah

Blok diagram ini menggambarkan arsitektur sistem *monitoring* keamanan rumah berbasis IoT dan *website*. Sistem ini terdiri dari tiga bagian utama yaitu perangkat keras (*Hardware*), *web server*, dan *website* sebagai antarmuka.

## 1. Perangkat Keras (*Hardware*)

Perangkat keras dikendalikan oleh ESP32 sebagai mikrokontroller utama yang terhubung dengan beberapa komponen yaitu:

- a. Sensor *Fingerprint*, untuk autentikasi sidik jari.

- b. *Push Button, Buzzer, Relay*, dan LCD untuk interaksi langsung antara sistem dengan pengguna.
- c. *Solenoid Door Lock* yang berfungsi sebagai pengunci pintu otomatis.

Semua perangkat mendapat *suplay* listrik melalui *adaptor* 12V yang terhubung pada listrik utama untuk memastikan operasional sistem tetap stabil.

## 2. Web Server

ESP32 berkomunikasi dengan server melalui jaringan Wi-Fi. Di sisi server terdapat beberapa proses penting, diantaranya adalah:

- a. MQTT sebagai protokol komunikasi ringan antara perangkat dan server.
- b. *Database* sebagai tempat penyimpanan data pengguna, log aktivitas akses pintu, dan juga untuk menyimpan hasil dari autentikasi.
- c. *Face Recognition*, yaitu salah satu fitur utama berbasis kamera CCTV .
- d. *WhatsApp Notification*, digunakan untuk mengirim pesan secara otomatis ketika terjadi kegagalan akses pintu.

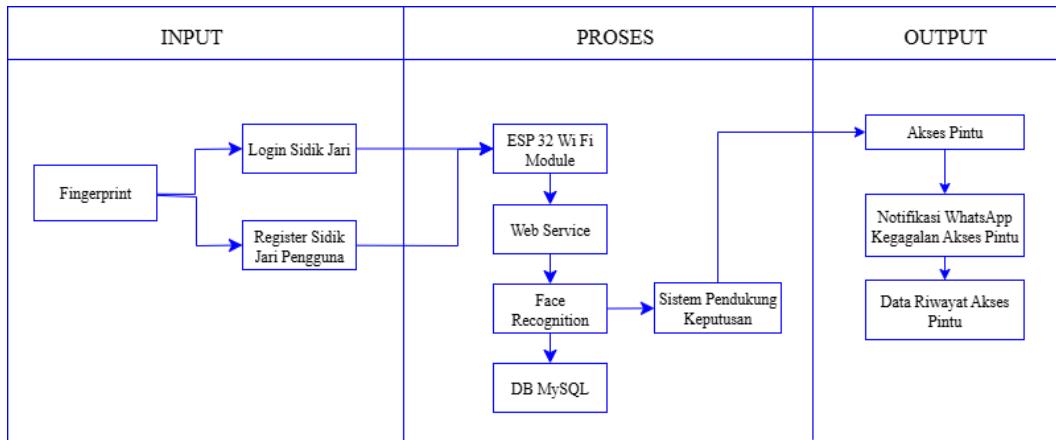
Pada sistem ini, *Face Recognition* berada dalam Web Server sebagai modul utama pengenalan wajah berbasis *machine learning*. Saat CCTV menangkap wajah seseorang di depan pintu, citra tersebut langsung dikirim ke server untuk diproses. Di sisi server, data wajah masuk ke modul *Face Recognition* yang menjalankan algoritma *Local Binary Patterns Histogram* (LBPH). Algoritma ini akan melakukan ekstraksi fitur berdasarkan tekstur lokal wajah dan membandingkan dengan dataset wajah yang tersimpan dalam database. Jika hasil perbandingan cocok maka server mengirimkan sinyal ke ESP32 melalui protokol MQTT agar mengaktifkan *relay* dan membuka kunci *solenoid door lock*. Jika wajah tidak dikenali, sistem menolak permintaan akses dan mengirimkan notifikasi kegagalan autentikasi. Proses ini dapat dipantau melalui dashboard web secara *real-time*, karena data status pintu dan autentikasi tersinkronisasi dengan sistem melalui jaringan internet.

## 3. Website

Website berfungsi sebagai dashboard administratif yang terhubung ke server melalui koneksi internet. Melalui antarmuka ini, *admin* dapat mengelola data pengguna, mengatur hak akses autentikasi, memantau aktivitas akses pintu secara *real-time*, mengendalikan buka/tutup pintu, serta mengakses tampilan langsung dari kamera CCTV.

#### 4.2.2 Flowmap Sistem Monitoring Keamanan Rumah

Perancangan ini menggambarkan alur sistem *monitoring* keamanan rumah dari input sidik jari hingga keputusan akses pintu melalui proses autentikasi ESP32, *Face Recognition*, dan *database*.



Gambar 4. 3 Flowmap Sistem Monitoring Keamanan Rumah

Gambar ini merupakan flowmap menggambarkan alur proses sistem *monitoring* keamanan rumah berbasis *Fingerprint* dan *Face Recognition*.

##### 1. Input

Sistem diawali dari sensor *Fingerprint* yang digunakan untuk autentikasi pengguna. Dan terdapat dua jalur input, *Login* dan register sidik jari.

- Login* Sidik Jari, digunakan saat pengguna ingin membuka akses pintu.
- Register Sidik Jari *user* , digunakan oleh *Admin* untuk menambahkan pengguna baru ke dalam sistem untuk mendapat hak akses pintu.

##### 2. Proses

Data dari sensor dikirim ke ESP32 Wi-Fi Module yang berfungsi sebagai pengendali utama. ESP32 terhubung dengan:

- Web Service* untuk meneruskan data ke *server*.
- Face Recognition*, sistem akan mengambil data wajah pengguna dan menganalisisnya menggunakan model *Machine Learning*.
- Hasil analisis disimpan di DB MySQL sebagai referensi verifikasi wajah.
- Output pengenalan wajah akan masuk ke Sistem Pendukung Keputusan yang menentukan apakah akses akan diberikan atau ditolak.

### 3. Output

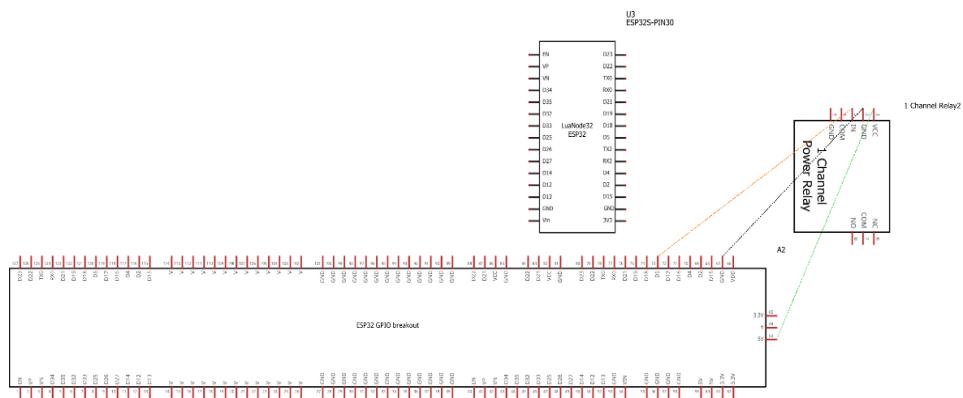
Sistem ini menghasilkan tiga output utama, yaitu Jika proses autentikasi berhasil, maka sistem akan membuka pintu dan mencatat keberhasilan akses. Namun jika gagal, pintu tetap terkunci, dan sistem akan mengirimkan notifikasi kegagalan dengan juga mencatat log aktivitas percobaan akses gagal tersebut.

#### 4.2.3 Perancangan Rangkaian Sistem

Sebelum sistem berjalan, diperlukan rancangan rangkaian perangkat keras agar semua komponen terintegrasi sesuai fungsinya. Pada tahap ini, dilakukan penyusunan rangkaian elektronik menggunakan aplikasi *Fritzing* untuk memvisualisasi koneksi antar komponen IoT. Rancangan ini berfungsi sebagai acuan dalam proses perakitan fisik sistem *monitoring* keamanan rumah.

##### 1. Rangkaian sensor *Fingerprint* terhubung dengan ESP32

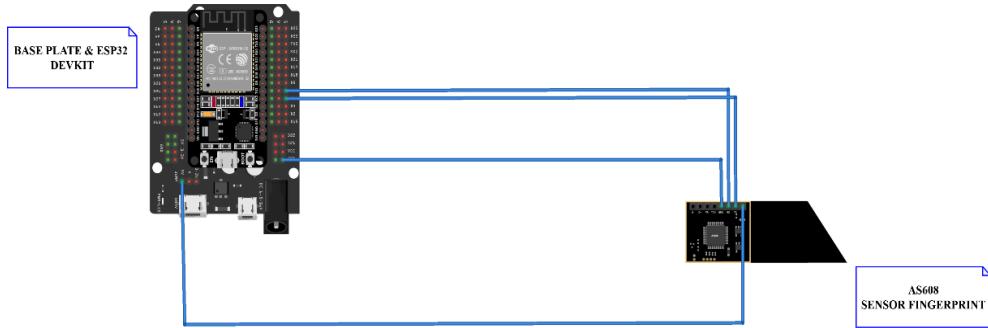
Gambar di bawah menunjukkan skema rangkaian dan implementasi wiring sensor *fingerprint* AS608 yang terhubung langsung ke ESP32 melalui base plate.



Gambar 4. 4 Skema Rangkaian Koneksi Sensor

*Fingerprint* AS608 ke ESP32

Gambar 4.4 menunjukkan skema koneksi antara ESP32 dan sensor *fingerprint* AS608. Pin TX dan RX sensor dihubungkan ke pin RX dan TX ESP32 secara menyilang, sedangkan pin VCC dan GND terhubung ke sumber daya dan ground dari ESP32. Skema ini digunakan untuk memastikan komunikasi serial berjalan dengan baik.

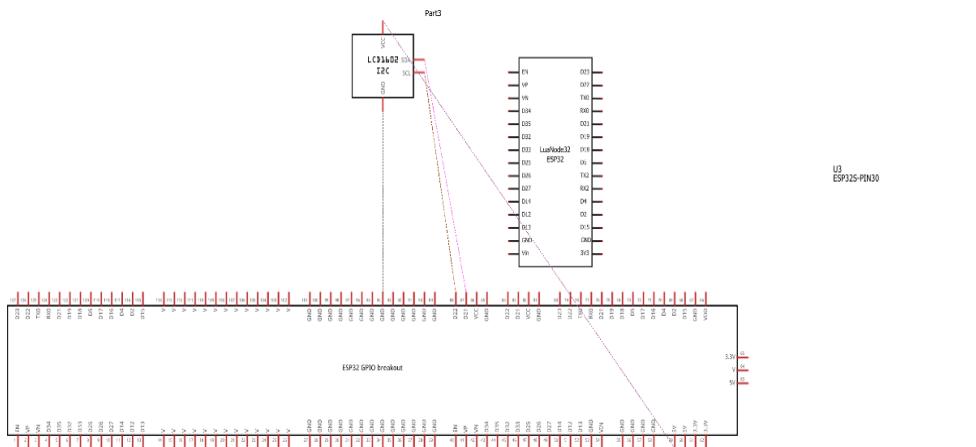


Gambar 4. 5 Implementasi *Wiring* Sensor *Fingerprint* AS608 ke ESP32

Pada Gambar 4.6 adalah implementasi *wiring* secara fisik, ESP32 dipasang pada *base plate* dan dihubungkan ke sensor *fingerprint* menggunakan kabel *jumper*. Meskipun kabel terpasang pada *base plate*, secara teknis tetap terhubung langsung ke pin-pin ESP32 yang berfungsi sesuai skema.

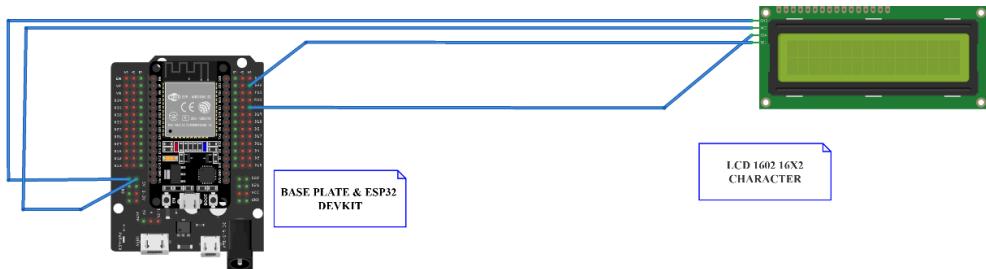
## 2. Rangkaian LCD terhubung dengan ESP32

Gambar di bawah menunjukkan skema rangkaian dan implementasi *wiring* LCD 1602 I2C yang terhubung langsung ke ESP32 melalui *base plate*.



Gambar 4. 6 Skema Rangkaian LCD dengan ESP32

Pada Gambar 4.6 menampilkan skema koneksi antara ESP32, LCD 1602, dan modul I2C *backpack*. Pada skema ini, LCD terhubung ke ESP32 menggunakan protokol komunikasi I2C, yang ditandai dengan sambungan pin SDA (data) dan SCL (clock). Penggunaan I2C mempermudah koneksi karena hanya membutuhkan dua pin, lebih efisien dibandingkan koneksi paralel.

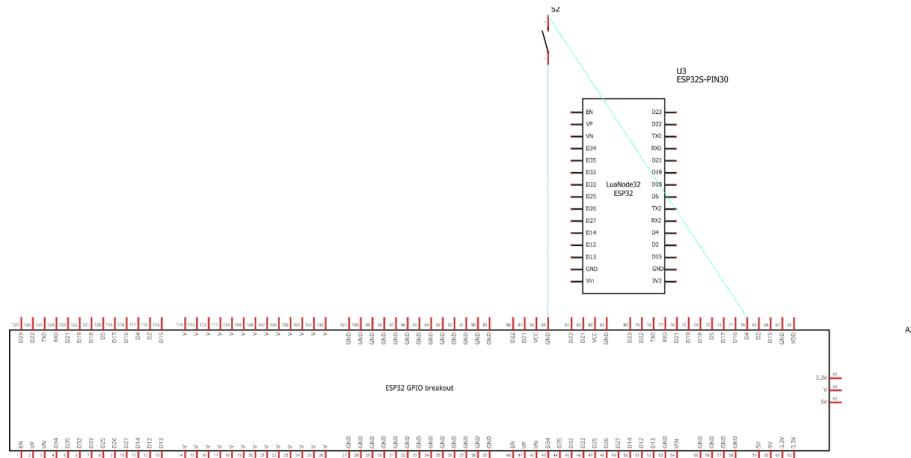


Gambar 4. 7 Implementasi *Wiring* LCD ke ESP32

Pada Gambar 4.7 di atas merupakan implementasi *wiring* antara modul LCD 1602 I2C dengan ESP32. Modul LCD dihubungkan langsung menggunakan antarmuka I2C, di mana pin VCC dan GND dihubungkan ke 3.3V dan GND pada ESP32 untuk catu daya. Pin SDA dan SCL dari LCD terhubung ke pin D21 (SDA) dan D22 (SCL) pada ESP32 untuk komunikasi data. *Wiring* ini memungkinkan ESP32 menampilkan data teks dengan *real-time* ke LCD.

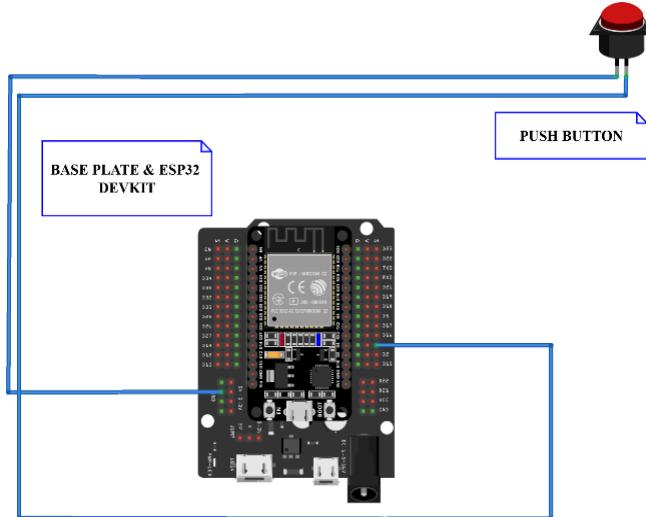
### 3. Rangkaian *Push Button* terhubung dengan ESP32

Gambar di bawah menunjukkan skema rangkaian dan implementasi *wiring* *Push Button* yang terhubung langsung ke ESP32 melalui *base plate*.



Gambar 4. 8 Skema Rangkaian *PushButton* terhubung dengan ESP32

Pada Gambar 4.8 adalah skema rangkaian *push button* dengan ESP32. *Push button* dihubungkan ke salah satu pin GPIO dan ground. Konfigurasi ini untuk memberikan input digital ke mikrokontroler saat tombol ditekan.

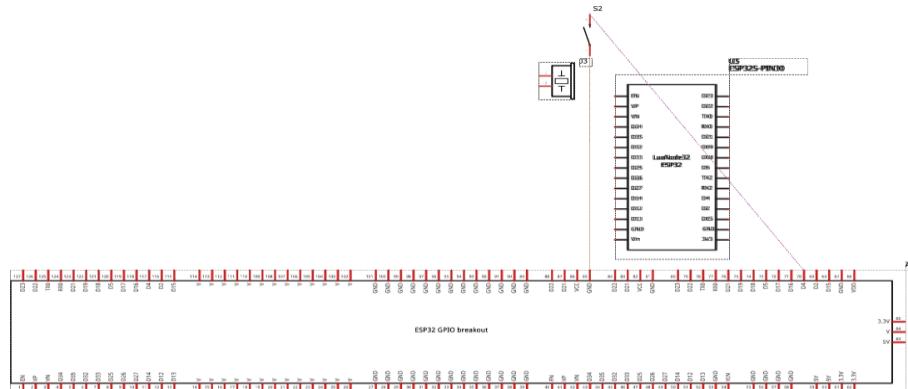


Gambar 4. 9 Implementasi *Wiring LCD ke ESP32*

Gambar 4.9 ini memperlihatkan implementasi *wiring push button* secara fisik pada board ESP32. Koneksi dilakukan melalui *base plate*, kabel terhubung dari *push button* ke pin GPIO dan GND pada board ESP32, sesuai skema.

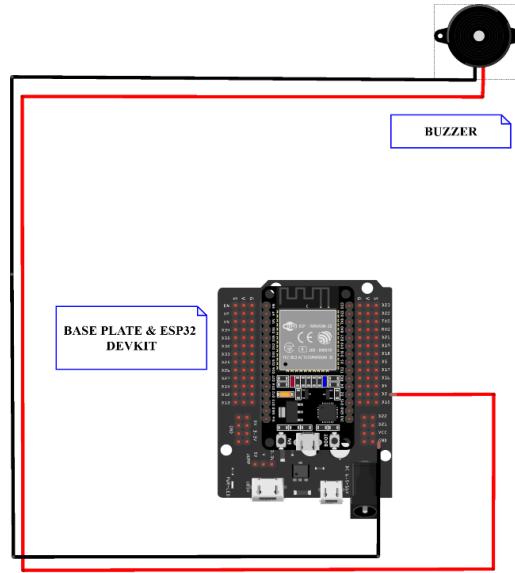
#### 4. Rangkaian *Buzzer* terhubung dengan ESP32

Gambar di bawah menunjukkan skema rangkaian dan implementasi *wiring Buzzer* yang terhubung langsung ke ESP32 melalui *base plate*.



Gambar 4. 10 Skema Rangkaian *Buzzer* terhubung dengan ESP32

Gambar 4.10 adalah skema koneksi *buzzer* ke ESP32. *Buzzer* dihubungkan pada pin digital ESP32 melalui jalur sinyal, serta ke sumber tegangan dan ground agar bekerja saat diberi perintah logika *HIGH*, untuk memberikan sinyal suara sebagai respons dari sistem, seperti saat akses ditolak atau saat pintu terbuka.

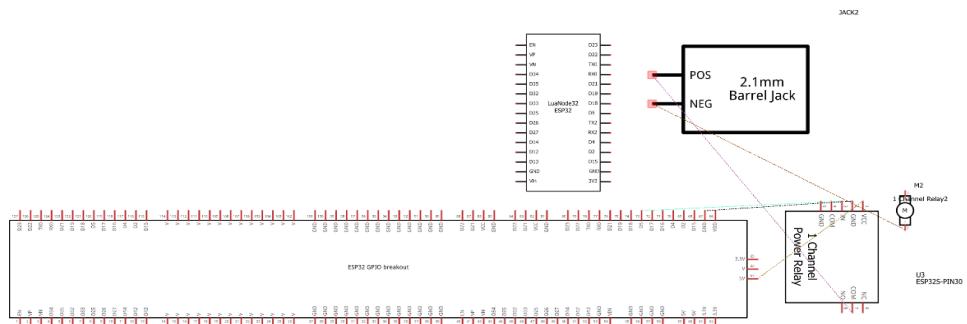


Gambar 4. 11 Implementasi *Wiring Buzzer* terhubung dengan ESP32

Gambar 4.11 berikut menunjukkan implementasi *wiring* secara fisik menggunakan ESP32 dan *buzzer*. Kabel merah menunjukkan koneksi positif (VCC dan sinyal kontrol), sedangkan kabel hitam mengarah ke GND. Implementasi ini bertujuan untuk memberikan umpan balik berupa bunyi ketika sistem mendeteksi peringatan atau konfirmasi keberhasilan akses pintu.

##### 5. Rangkaian *Solenoid Door Lock* terhubung dengan *Relay Adaptor* dan ESP32

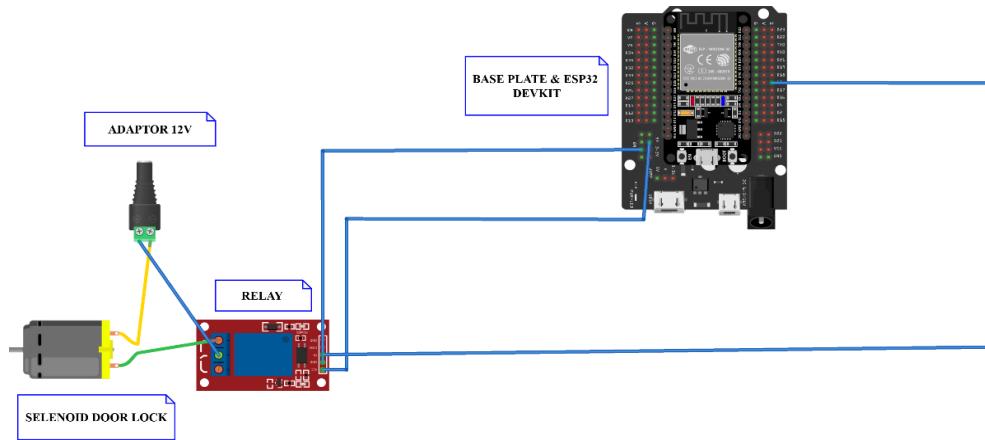
Gambar ini menunjukkan skema rangkaian dan implementasi *wiring* untuk *Solenoid Door Lock, Relay, Adaptor* yang terhubung ke ESP32 melalui *base plate*.



Gambar 4. 12 Skema Rangkaian *Buzzer* terhubung dengan ESP32

Gambar 4.12 adalah rangkaian ESP32, relay, *adaptor*, dan *solenoid door lock*. *Adaptor* menjadi sumber daya yang dialirkan melalui *relay* yang berfungsi

sebagai saklar elektronik yang dikendalikan pin digital ESP32, sehingga instruksi membuka kunci, *relay* mengaktifkan *solenoid door lock* untuk membuka pintu.

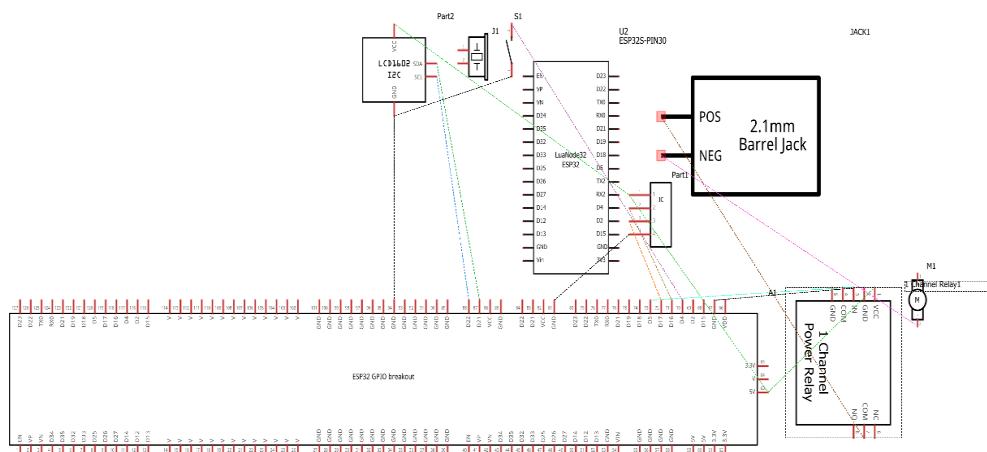


Gambar 4. 13 Implementasi *Wiring Selenoid , Relay*, Adaptor terhubung ke ESP32

Gambar 4.13 adalah implementasi wiring rangkaian sebelumnya. Modul ESP32 terpasang pada *base plate* dan terhubung dengan kabel ke modul *relay*. Modul *relay* menerima daya dari adaptor 12V dan terhubung langsung ke *solenoid doorlock*. Implementasi ini memudahkan proses *debugging* dan pengujian sistem.

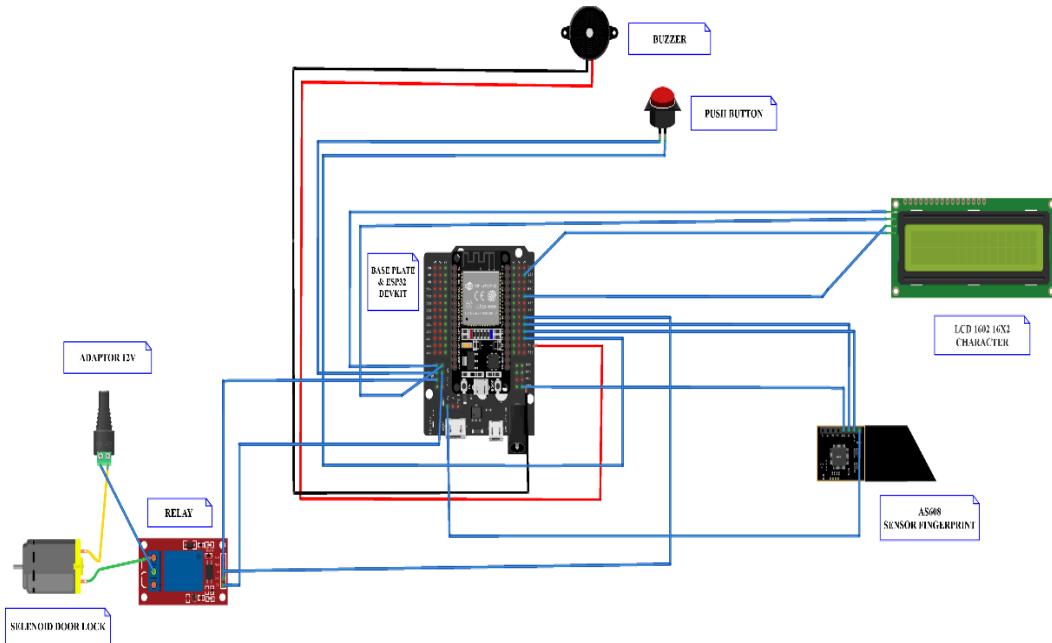
## 6. Rangkaian alat keseluruhan

Gambar ini menunjukkan skema dan implementasi *wiring* untuk rangkaian dari keseluruhan komponen iot sistem *monitoring keamanan*.



Gambar 4. 14 Skema Rangkaian Alat Keseluruhan Sistem

Skema pada Gambar 4.14 ini memperlihatkan rangkaian elektronik secara detail, termasuk pin GPIO yang digunakan pada NodeMCU ESP32, koneksi I2C pada LCD, sambungan *relay* untuk pengendalian *solenoid door lock*, jalur catu daya, serta sambungan *push button*. Diagram ini untuk memahami detail *pinout* dan rangkaian elektronik sebelum diimplementasikan ke rangkaian fisik.



Gambar 4. 15 Implementasi Wiring Seluruh Alat Sistem

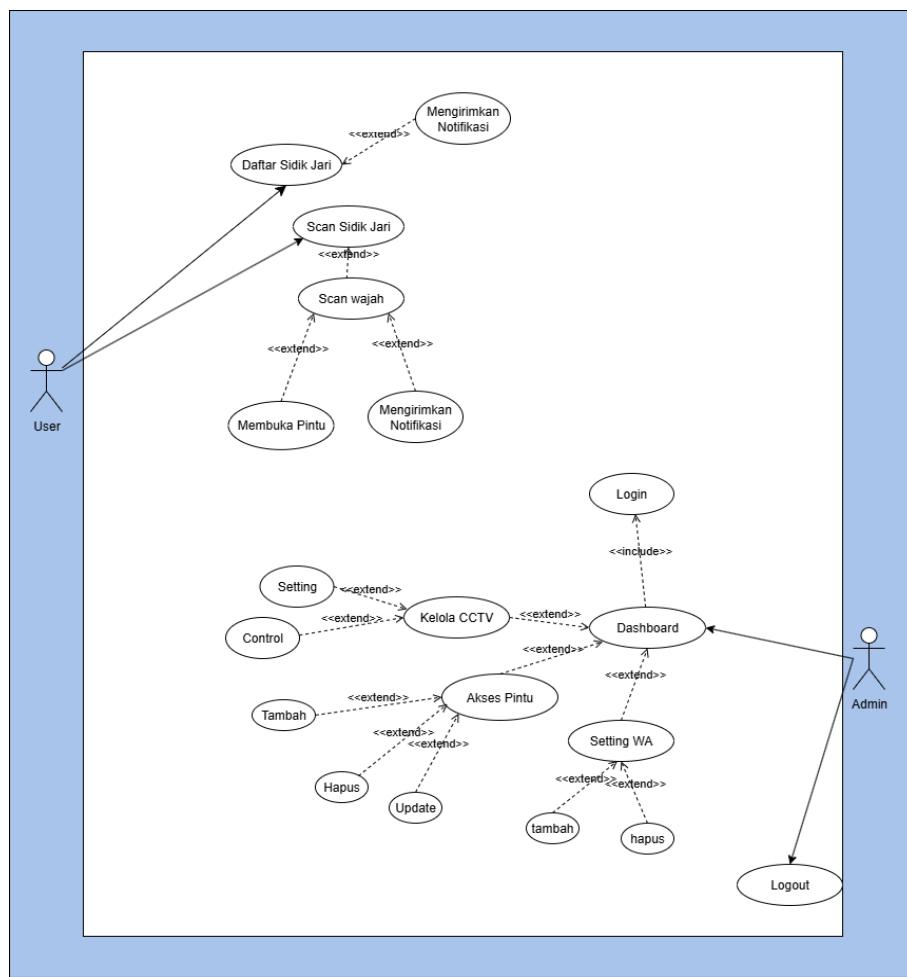
Pada Gambar 4.15 menunjukkan rangkaian fisik keseluruhan alat dari sistem *monitoring* keamanan rumah. Rangkaian ini menghubungkan ESP32, sensor *fingerprint*, *relay*, *solenoid door lock*, LCD, *buzzer*, dan *push button* dengan kabel *jumper* sesuai pin. Jalur biru berfungsi sebagai data/sinyal, sedangkan merah dan hitam untuk catu daya. Wiring ini memastikan seluruh komponen terintegrasi dan dapat dipantau secara *real-time* melalui Internet.

#### 4.2.4 Perancangan Sistem UML

Untuk menggambarkan proses serta struktur dari sistem secara konseptual dan digunakan pendekatan UML. Uml dapat membantu memodelkan bagaimana *user* berinteraksi dengan sistem, alur aktivitas sistem, dan struktur data serta objek yang digunakan. Pada tahap ini ditampilkan beberapa diagram UML, yaitu:

## 1. Use Case Diagram

*Use Case Diagram* berikut menggambarkan interaksi antara *aktor* (*User* dan *Admin*) dengan sistem *monitoring* keamanan rumah berbasis IoT. Diagram ini dirancang untuk memperlihatkan fungsi utama yang dapat dilakukan oleh masing-masing aktor, seperti autentikasi *user*, pengelolaan akses pintu, pemantauan CCTV, serta pengaturan sistem melalui *dashboard* web. Untuk mempermudah pemahaman, diagram dibagi menjadi dua bagian terpisah untuk *user* serta *Admin*.



Gambar 4. 16 Use Case Diagram

Penjelasan mengenai *Use Case Diagram* di atas dapat dijabarkan secara lebih terperinci melalui tabel deskripsi *use case* yang telah disusun. Tabel ini bertujuan untuk memberikan gambaran yang lebih jelas mengenai alur fungsi, aktor yang terlibat, serta interaksi yang terjadi dalam sistem.

1) Deskripsi *aktor*

Berikut ini merupakan tabel untuk mendeskripsikan aktor pada use case sistem *monitoring* keamanan rumah.

Tabel 4. 3 Deskripsi Aktor

Aktor	Deskripsi
<i>User</i>	<i>User</i> merupakan <i>user</i> umum dari sistem keamanan rumah yang dapat melakukan pendaftaran dan pemindaian sidik jari serta wajah untuk akses buka pintu. Sistem juga akan mengirimkan notifikasi saat autentikasi.
<i>Admin</i>	<i>Admin</i> merupakan aktor yang memiliki hak akses penuh terhadap sistem, <i>Login</i> ke <i>dashboard</i> , mengelola CCTV, mengakses dan mengatur pintu, serta mengatur notifikasi WhatsApp. <i>Admin</i> juga dapat melakukan pengaturan, penambahan, penghapusan, dan pembaruan data.

2) Deskripsi *Use Case*

Di bawah ini merupakan tabel-tabel yang akan menjelaskan detail use case yang menggambarkan interaksi antara aktor dan sistem *monitoring* keamanan rumah.

a. Skenario *Use Case* Daftar Sidik Jari

Dibawah ini adalah tabel untuk mendeskripsikan *use case* daftar sidik jari pada sistem *monitoring* keamanan rumah.

Tabel 4. 4 Deskripsi *Use Case* Daftar Sidik Jari

Nama	Daftar Sidik Jari
Aktor	<i>User</i>
<b>Skenario Utama</b>	
Kondisi Awal	<i>User</i> belum memiliki data sidik jari yang terdaftar di sistem.
<b>Aktor</b>	<b>Sistem</b>

Mengakses perangkat sistem	Menampilkan menu pendaftaran sidik jari
Meletakkan jari di sensor <i>Fingerprint</i>	Menangkap dan menyimpan data <i>Fingerprint</i>
Kondisi Akhir	Data sidik jari <i>user</i> tersimpan dalam sistem.

b. Skenario *Use Case Scan Sidik Jari*

Dibawah ini adalah tabel yang digunakan untuk mendeskripsikan *use case scan* sidik jari pada sistem *monitoring* keamanan rumah.

Tabel 4. 5 Deskripsi *Use Case Scan Sidik Jari*

Nama	Scan Sidik Jari
Aktor	<i>User</i>
<b>Skenario Utama</b>	
Kondisi Awal	<i>User</i> telah memiliki data sidik jari yang terdaftar.
<b>Aktor</b>	<b>Sistem</b>
Meletakkan jari di sensor <i>Fingerprint</i>	Mendeteksi dan mencocokkan sidik jari, Jika cocok, lanjut proses pembukaan pintu, Jika tidak cocok, tampilkan pesan gagal
Kondisi Akhir	Autentikasi berhasil atau gagal.

c. Skenario *Use Case Scan Wajah*

Dibawah ini adalah tabel yang digunakan untuk mendeskripsikan *use case scan* wajah pada sistem *monitoring* keamanan rumah.

Tabel 4. 6 Deskripsi *Use Case Scan Wajah*

Nama	Scan Wajah
Aktor	<i>User</i>
<b>Skenario Utama</b>	

Kondisi Awal	Wajah <i>user</i> terdaftar dalam sistem <i>monitoring</i> keamanan rumah.
<b>Aktor</b>	<b>Sistem</b>
Berdiri menghadap kamera cctv.	Mencocokkan wajah; jika cocok, pintu dibuka, jika gagal, sistem mengirim notifikasi kegagalan.
Kondisi Akhir	Autentikasi berhasil atau gagal.

d. Skenario *Use Case* Membuka Pintu

Dibawah ini adalah tabel yang digunakan untuk mendeskripsikan *use case* membuka pintu pada sistem *monitoring* keamanan rumah.

Tabel 4. 7 Deskripsi *Use Case* Membuka Pintu

Nama	Membuka Pintu
Aktor	<i>User</i>
<b>Skenario Utama</b>	
Kondisi Awal	<i>User</i> melakukan autentikasi sidik jari dan wajah.
<b>Aktor</b>	<b>Sistem</b>
Scan sidik jari dan wajah	Autentikasi dicocokkan, jika cocok, pintu terbuka, jika tidak, sistem mengirim notifikasi, mencatat log.
Kondisi Akhir	Autentikasi berhasil atau gagal.

e. Skenario *Use Case Login*

Dibawah ini adalah tabel yang digunakan untuk mendeskripsikan *use case Login* pada sistem *monitoring* keamanan rumah.

Tabel 4. 8 Deskripsi *Use Case Login*

Nama	<i>Login</i>
Aktor	<i>Admin</i>

<b>Skenario Utama</b>	
Kondisi Awal	Aktor belum punya akses aplikasi
<b>Aktor</b>	<b>Sistem</b>
Mengakses halaman <i>Login</i>	Menampilkan halaman <i>Login</i>
Mengisi form <i>Login</i>	Validasi login, lalu tampilkan halaman utama.
Kondisi Akhir	Menampilkan halaman utama.

f. Skenario *Use Case Dashboard*

Dibawah ini adalah tabel yang digunakan untuk mendeskripsikan *use case dashboard* pada sistem *monitoring* keamanan rumah.

Tabel 4.9 Deskripsi *Use Case dashboard*

Nama	<i>dashboard</i>
Aktor	<i>Admin</i>
<b>Skenario Utama</b>	
Kondisi Awal	<i>Admin</i> <i>Login</i> dan mengakses tampilan utama, mengelola sistem
<b>Aktor</b>	<b>Sistem</b>
Mengakses menu <i>dashboard</i>	Menampilkan fitur manajemen pintu, <i>Fingerprint</i> , CCTV, WA.
Kondisi Akhir	<i>Admin</i> dapat mengakses dan mengelola sistem sesuai hak akses

g. Skenario *Use Case Kelola CCTV*

Dibawah ini adalah tabel yang digunakan untuk mendeskripsikan *use case kelola CCTV* pada sistem *monitoring* keamanan rumah.

Tabel 4. 10 Deskripsi *Use Case Kelola CCTV*

Nama	Kelola CCTV
Aktor	<i>Admin</i>

<b>Skenario Utama</b>	
Kondisi Awal	<i>Admin</i> berada di dalam <i>dashboard</i> untuk memantau dan mengelola cctv.
<b>Aktor</b>	<b>Sistem</b>
Memilih menu halaman cctv	Menampilkan tampilan kamera
Mengatur setting/control	Merekam data
Kondisi Akhir	<i>Admin</i> berhasil mengelola atau mengontrol kamera

h. Skenario Use Case Halaman Pengguna

Dibawah ini adalah tabel yang digunakan untuk mendeskripsikan use case pengguna pada sistem *monitoring* keamanan rumah

Tabel 4. 11 Deskripsi Use Case Halaman Pengguna

Nama	Halaman Pengguna
Aktor	<i>Admin</i>
<b>Skenario Utama</b>	
Kondisi Awal	<i>Admin</i> berada di <i>dashboard</i> , dan membuka halaman pengguna untuk mengatur daftar <i>user</i> yang bisa mengakses pintu.
<b>Aktor</b>	<b>Sistem</b>
Memilih menu akses pintu	Menampilkan daftar <i>user</i> yang telah akses.
Melakukan tambah, hapus, update <i>user</i>	Menyimpan perubahan data akses.
Kondisi Akhir	Daftar <i>user</i> yang memiliki ijin akses pintu terupdate

i. Skenario Use Case halaman Akses Pintu

Dibawah ini adalah tabel yang digunakan untuk mendeskripsikan use case akses pintu pada sistem *monitoring* keamanan rumah

Tabel 4. 12 Deskripsi *Use Case* Akses Pintu

Nama	Akses Pintu
Aktor	<i>Admin</i>
<b>Skenario Utama</b>	
Kondisi Awal	<i>Admin</i> membuka halaman akses pintu untuk melihat log aktifitas dan melakukan buka tutup pintu melalui sistem.
<b>Aktor</b>	<b>Sistem</b>
Memilih menu akses pintu	Menampilkan daftar <i>user</i> yang telah akses.
Klik tombol buka/tutup	Menunjukkan pintu terbuka atau tertutup
Kondisi Akhir	Daftar log aktifitas tercatat.

j. Skenario *Use Case Setting WhatsApp*

Dibawah ini adalah tabel yang digunakan untuk mendeskripsikan *use case setting WhatsApp* pada sistem *monitoring* keamanan rumah

Tabel 4. 13 Deskripsi *Use Case Setting WA*

Nama	Setting WA
Aktor	<i>Admin</i>
<b>Skenario Utama</b>	
Kondisi Awal	<i>Admin</i> mengatur nomor wa penerima notifikasi WhatsApp.
<b>Aktor</b>	<b>Sistem</b>
Memilih menu Setting WA	Menampilkan forum pengaturan
Menambah atau menghapus nomor	Menyimpan data ke sistem
Kondisi Akhir	Nomor WhatsApp berhasil ditambahkan.

k. Skenario *Use Case* Notifikasi WhatsApp

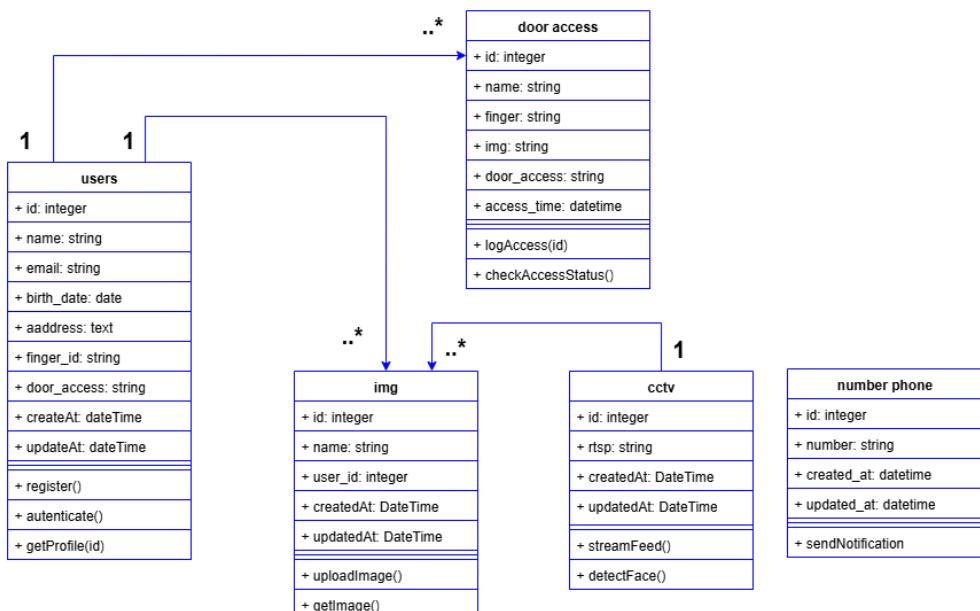
Dibawah ini adalah tabel untuk mendeskripsikan *use case* notifikasi WhatsApp pada sistem *monitoring* keamanan rumah.

Tabel 4. 14 Deskripsi Use Case Notifikasi WhatsApp

Nama	Notifikasi WhatsApp
Aktor	Admin
<b>Skenario Utama</b>	
Kondisi Awal	Pengguna gagal autentikasi untuk akses pintu
<b>Aktor</b>	<b>Sistem</b>
Melakukan kegagalan autentikasi sidik jari atau wajah	Mengirimkan notifikasi kegagalan akses ke WhatsApp.
Kondisi Akhir	WhatsApp menerima pesan kegagalan.

## 2. Class Diagram

Class Diagram sistem *monitoring* keamanan rumah berikut menunjukkan struktur logis, relasi, dan fungsi antar kelas sebagai acuan perancangan sistem.



Gambar 4. 17 Class Diagram

*Class Diagram* diatas memperlihatkan relasi antar komponen sistem monitoring keamanan rumah. Diagram ini terdiri dari, *Users*, *DoorAccess*, *Img*, *CCTV*, dan *NumberPhone*. *Class Users* terhubung dengan log akses pintu melalui *DoorAccess*, serta terhubung dengan data gambar melalui *Img* untuk verifikasi wajah. *Class CCTV* berfungsi sebagai perangkat pemantau *real-time* dan mendukung proses *face recognition*. *NumberPhone* digunakan untuk menyimpan nomor penerima notifikasi keamanan. Struktur ini mendukung manajemen akses, pemantauan, dan notifikasi otomatis untuk keamanan rumah.

#### A. Struktur Tabel

Struktur tabel pada sistem *monitoring* keamanan rumah dirancang untuk menyimpan data-data penting terkait proses autentikasi, aktivitas *user* , dan pengaturan sistem. Basis data digunakan sebagai penyimpanan utama guna mendukung fungsionalitas *dashboard* web. Berikut tabel-tabel yang digunakan:

##### 1. Perancangan Tabel *User*

Tabel berikut dirancang untuk menyimpan informasi detail user yang memiliki hak akses untuk membuka pintu, seperti data identitas, kontak, serta data autentikasi yang mendukung proses kontrol dan monitoring pintu secara aman.

Tabel 4. 15 Perancangan tabel *user*

Field	Type	Size	Index	Deskripsi
<i>id</i>	<i>integer</i>	20	<i>Primary key</i>	ID unik <i>user</i>
<i>name</i>	<i>varchar</i>	100	-	Nama <i>user</i>
<i>email</i>	<i>varchar</i>	100	<i>Unique</i>	Email <i>user</i>
<i>birth_date</i>	<i>date</i>	-	-	Tanggal lahir <i>user</i>
<i>address</i>	<i>text</i>	-	-	Alamat <i>user</i>
<i>finger_id</i>	<i>varchar</i>	50	-	Id <i>Fingerprint user</i>
<i>door_acce ss</i>	<i>varchar</i>	100	-	Akses pintu oleh <i>user</i>
<i>updateAt</i>	<i>datetime</i>		-	Waktu data diubah
<i>createdAt</i>	<i>datetime</i>		-	Waktu data dibuat

## 2. Perancangan Tabel *Door Access*

Tabel di bawah ini untuk mencatat riwayat aktivitas akses pintu oleh setiap pengguna. Data yang disimpan mencakup ID fingerprint, nama pengguna, gambar pendukung, validasi akses, serta waktu akses dilakukan.

Tabel 4. 16 Perancangan tabel door access

Field	Type	Size	Index	Deskripsi
<i>id</i>	<i>integer</i>	20	Primary key	Id <i>user</i> yang mengakses pintu.
<i>name</i>	<i>varchar</i>	100	-	Nama yang mengakses pintu
<i>finger</i>	<i>varchar</i>	100	-	Data id <i>Fingerprint</i>
<i>img</i>	<i>varchar</i>	100	-	Gambar <i>user</i> yang akses pintu
<i>door_acces</i>	<i>varchar</i>	100	-	Validasi akses <i>user</i>
<i>access_time</i>	<i>datetime</i>	-	-	Waktu akses pintu

## 3. Perancangan Tabel CCTV

Tabel di bawah ini berfungsi untuk menyimpan data konfigurasi CCTV yang digunakan dalam sistem. Informasi yang disimpan mencakup ID CCTV, serta waktu pembuatan dan pembaruan data.

Tabel 4. 17 Perancangan tabel CCTV

Field	Type	Size	Index	Deskripsi
<i>id</i>	<i>integer</i>		Primary key	Id CCTV
<i>rtsp</i>	<i>varchar</i>	255	-	Link akses CCTV
<i>creatseAt</i>	<i>datetime</i>		-	Waktu dibuat nya data
<i>updateAt</i>	<i>datetime</i>		-	Waktu diubah nya data

#### 4. Perancangan Tabel IMG

Tabel ini digunakan untuk menyimpan file gambar hasil tangkapan wajah atau bukti autentikasi pengguna. Tabel ini juga menyimpan relasi ke *user* yang terkait serta informasi waktu pembuatan dan pembaruan data..

Tabel 4. 18 Perancangan tabel *Image*

Field	Type	Size	Index	Deskripsi
<i>id</i>	<i>integer</i>		<i>Primary key</i>	Id gambar
<i>name</i>	<i>varchar</i>	100	-	Nama file gambar
<i>user_id</i>	<i>integer</i>		<i>Foreign key</i>	Id <i>user</i> pemilik image
<i>updateAt</i>	<i>datetime</i>		-	Waktu data diubah
<i>createdAt</i>	<i>datetime</i>		-	Waktu data dibuat

#### 5. Perancangan Tabel *Number Phone*

Tabel ini digunakan untuk menyimpan nomor WhatsApp yang menjadi tujuan notifikasi sistem.

Tabel 4. 19 Perancangan tabel *number phone*

Field	Type	Size	Index	Deskripsi
<i>id</i>	<i>integer</i>		<i>Primary key</i>	Id gambar
<i>number</i>	<i>varchar</i>	20	-	Nama file gambar
<i>updateAt</i>	<i>datetime</i>		-	Waktu data diubah
<i>createdAt</i>	<i>datetime</i>		-	Waktu data dibuat

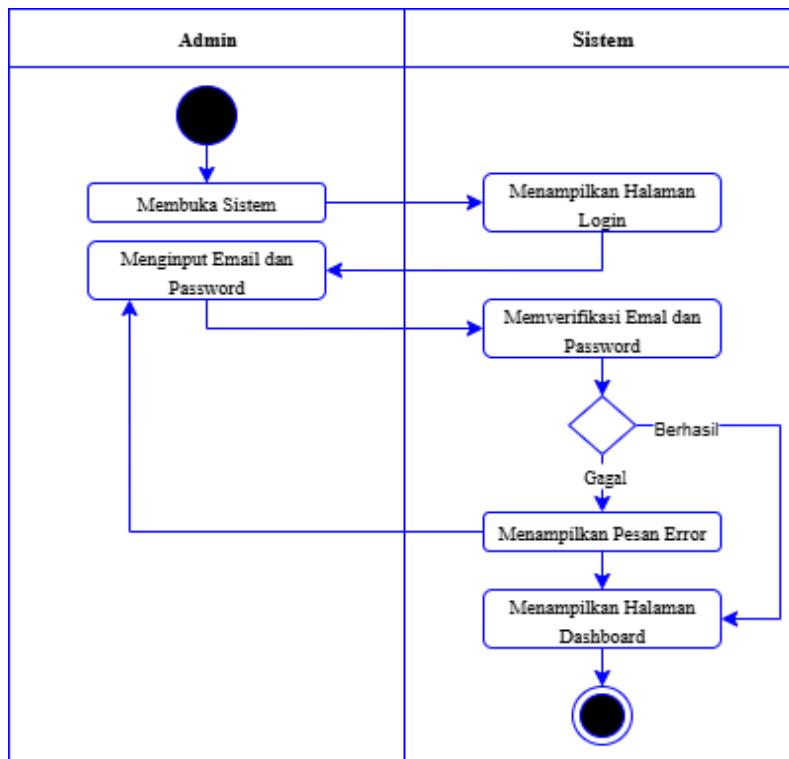
#### 3. *Activity Diagram*

*Activity diagram* digunakan untuk menggambarkan urutan aktivitas dalam suatu sistem serta alur kerja dari satu proses ke proses lainnya. Diagram ini memperlihatkan bagaimana aliran aktivitas berjalan, termasuk pengambilan

keputusan dan transisi antar aktivitas. Dengan menggunakan simbol-simbol standar UML, *Activity Diagram* membantu memvisualisasikan proses yang kompleks menjadi lebih mudah dipahami. *Activity diagram* berguna dalam memodelkan proses, alur kerja, dan interaksi antar komponen secara *visual*.

### 1. *Activity Diagram* Menampilkan Halaman *Login*

*Activity Diagram* dibawah ini menggambarkan alur *Login Admin*, mulai dari *input* data *Login* hingga verifikasi dan hasil akses ke *dashboard* sistem.

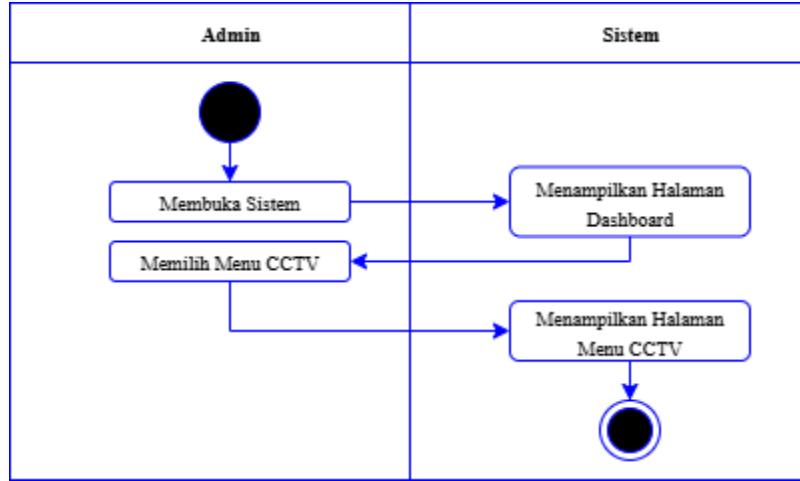


Gambar 4. 18 *Activity Diagram* *Login*

*Activity Diagram* ini menunjukkan alur aktivitas saat *Admin* mengakses halaman *login*. *Admin* mengisi form *login* dengan *input email* dan *password*. Sistem akan melakukan validasi, jika data valid, sistem akan membuka halaman utama *dashboard*. Namun, jika tidak valid, sistem menampilkan pesan kesalahan dan tetap akan berada di halaman *login* untuk mengulang proses masuk.

### 2. *Activity Diagram* Menampilkan Halaman menu CCTV

*Activity diagram* ini menggambarkan alur interaksi *Admin* saat mengakses fitur *CCTV* pada *dashboard* sistem *monitoring* keamanan rumah.

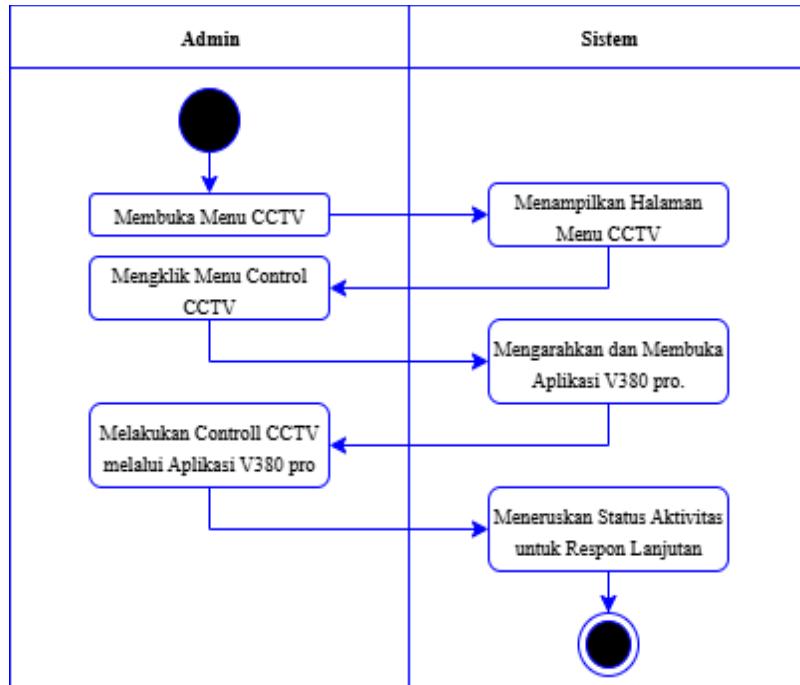


Gambar 4. 19 *Activity Diagram* Menampilkan Halaman CCTV

*Activity Diagram* pada halaman ini terdapat empat aktivitas yang dapat dilakukan, mengontrol CCTV, menampilkan *full screen*, melakukan pengaturan, serta mengedit link rtsp. Seluruh aktivitas diakses dengan mengklik menu yang kemudian akan diarahkan ke aplikasi mobile bawaan CCTV, yaitu V380 Pro.

### 3. *Activity diagram* membuka menu kontrol CCTV.

*Activity Diagram* di bawah ini menggambarkan alur saat *Admin* mengelola CCTV melalui dashboard dan aplikasi V380 Pro.

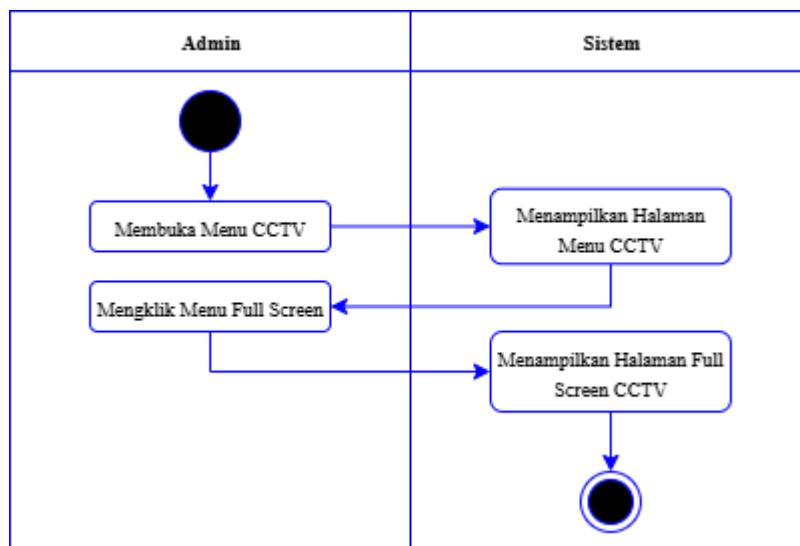


Gambar 4. 20 *Activity Diagram* membuka menu kontrol CCTV

*Activity Diagram* pada gambar 4.20 di atas menggambarkan alur interaksi antara *Admin* dengan sistem dalam mengakses dan mengontrol CCTV melalui aplikasi bawaan dari CCTV nya yaitu V380 Pro. Proses dimulai saat *Admin* membuka sistem kemudian memilih menu CCTV, sistem menampilkan halaman *dashboard* dan menu CCTV yang tersedia. Setelah *Admin* mengklik menu kontrol CCTV, sistem otomatis mengarahkan dan membuka aplikasi V380 Pro sebagai aplikasi resmi untuk pengelolaan kamera. Dan kontrol CCTV di lakukan langsung melalui aplikasi tersebut, sementara sistem meneruskan status aktivitas yang dilakukan untuk kebutuhan pemantauan atau respon lanjutan.

#### 4. *Activity diagram* membuka menu Full Screen

*Activity diagram* ini menggambarkan alur sistem melakukan aktifitas Full Screen melalui halaman CCTV di *dashboard monitoring* keamanan rumah.

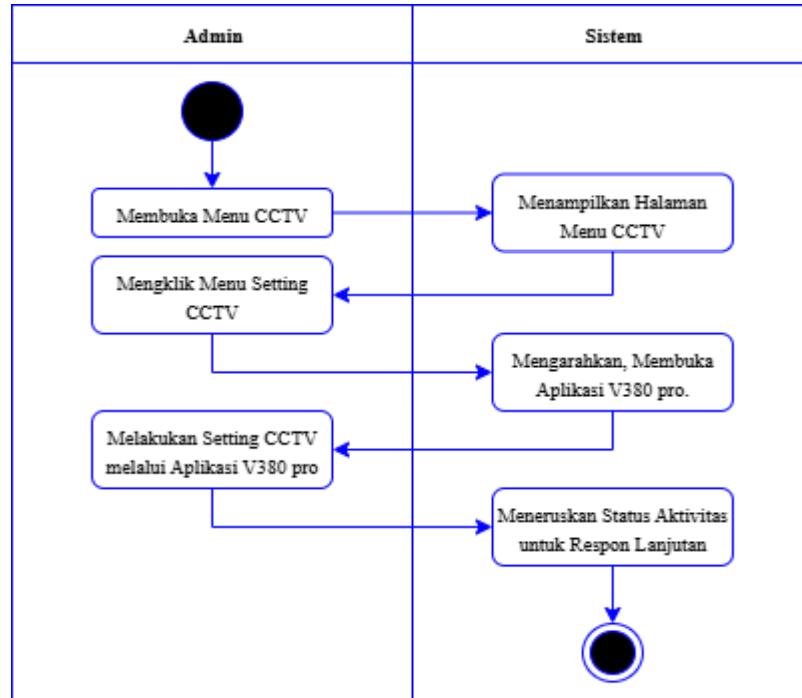


Gambar 4. 21 *Activity Diagram* membuka menu *Full Screen*

*Activity Diagram* diatas menjelaskan alur saat *Admin* menampilkan CCTV dalam mode layar penuh. Proses dimulai ketika *Admin* membuka sistem, memilih menu CCTV, lalu mengklik opsi full screen untuk menampilkan tampilan kamera secara penuh.

#### 5. *Activity diagram* membuka menu Setting CCTV

*Activity diagram* ini menggambarkan alur sistem melakukan aktifitas Setting CCTV dari halaman CCTV di *dashboard monitoring* keamanan rumah.

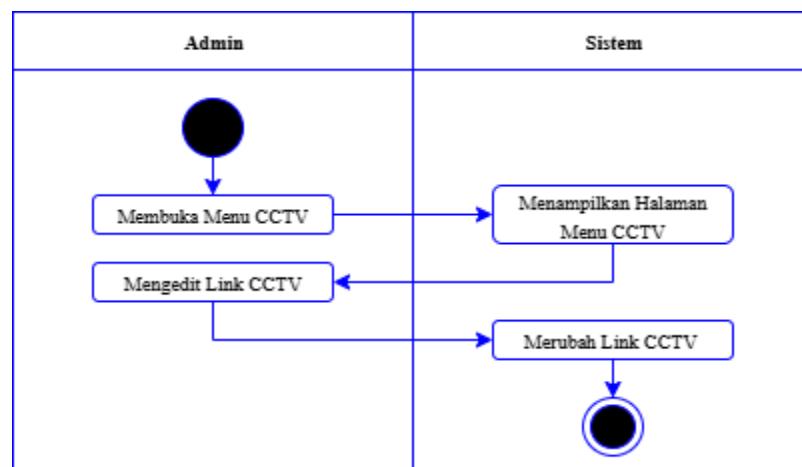


Gambar 4. 22 Activity diagram membuka menu Setting CCTV

*Activity Diagram* di atas menunjukkan alur *Admin* saat mengontrol CCTV melalui sistem. Proses dimulai dengan membuka dan menampilkan menu halaman CCTV, lalu *Admin* memilih setting CCTV yang akan mengarahkan ke aplikasi V380 Pro untuk mengatur jaringan, posisi kamera, serta kualitas video.

#### 6. *Activity Diagram* Mengedit Link CCTV

*Activity diagram* ini menggambarkan alur sistem melakukan aktivitas edit link CCTV dari halaman CCTV di *dashboard monitoring* keamanan rumah.

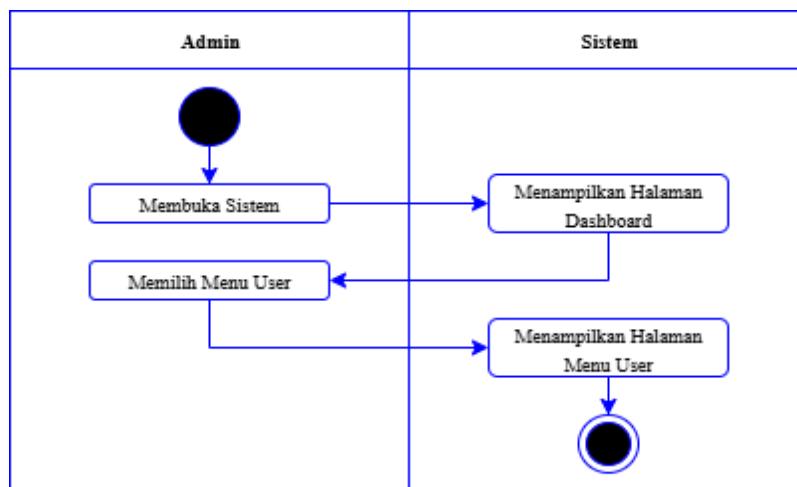


Gambar 4. 23 Activity Diagram Mengedit Link CCTV

*Activity Diagram* diatas menggambarkan alur saat *Admin* melakukan pengeditan link CCTV pada sistem. Proses diawali saat *Admin* membuka sistem dan menampilkan halaman *dashboard*. *Admin* memilih menu CCTV, lalu sistem menampilkan halaman menu CCTV. *Admin* mengakses fitur edit link pada CCTV, dan sistem akan mengganti link sesuai perubahan oleh *Admin*. Proses ini untuk memperbarui jalur streaming CCTV jika terjadi perubahan alamat atau konfigurasi kamera. Dengan adanya pengaturan ini, *Admin* dapat memastikan link RTSP selalu valid dan CCTV dapat diakses secara *real-time* tanpa gangguan, sehingga akan mendukung kelancaran sistem *monitoring* keamanan rumah.

#### 7. *Activity Diagram* Menampilkan Halaman *User*

*Activity Diagram* dibawah ini menggambarkan alur interaksi membuka menu *user* untuk melihat daftar pengguna yang terdaftar dalam sistem.

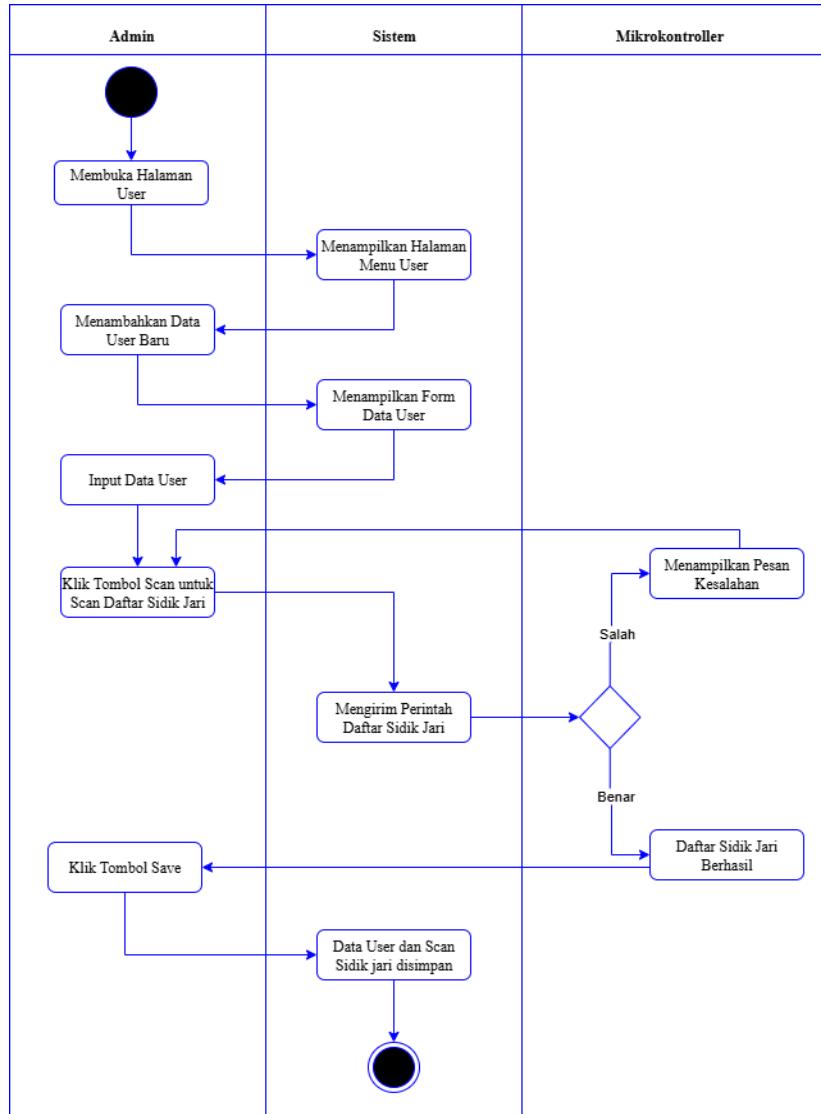


Gambar 4. 24 *Activity Diagram* Menampilkan Halaman *User*

*Activity Diagram* diatas menjelaskan alur aktivitas *Admin* saat mengakses halaman *User* pada sistem. Proses dimulai saat *Admin* membuka sistem, dan sistem menampilkan halaman *dashboard*. *Admin* memilih menu *pengguna*, sistem menampilkan halaman menu *pengguna* untuk pengelolaan data *user*, seperti menambah, mengedit, atau menghapus informasi *user* yang terdaftar.

#### 8. *Activity Diagram* Menambahkan *User*

*Activity Diagram* ini menggambarkan alur interaksi *Admin* saat akan menambah *user* untuk mendapatkan ijin autentikasi akses pintu.

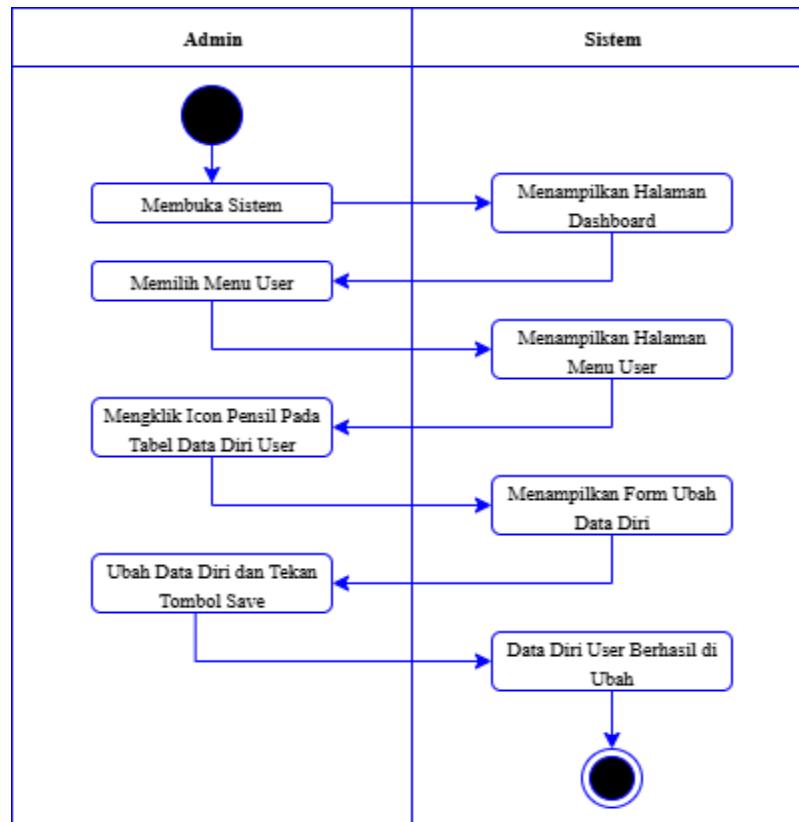


Gambar 4. 25 Activity Diagram Menambahkan User

Activity Diagram diatas menjelaskan alur yang dilakukan *Admin* untuk menambahkan data *user* baru sekaligus mendaftarkan sidik jari ke dalam sistem. Proses dimulai saat *Admin* membuka halaman menu pengguna pada dashboard. *Admin* memilih tombol Tambah untuk menambahkan user baru. Sistem menampilkan *form input* data *user* yang harus diisi, kemudian *Admin* menekan tombol Scan untuk proses pendaftaran sidik jari. Sistem akan mengirimkan perintah ke mikrokontroler agar memproses *input* sidik jari dari sensor *fingerprint*. Mikrokontroler memverifikasi data sidik jari. Jika berhasil, data disimpan ke database. Jika gagal, sistem menampilkan pesan kesalahan agar *Admin* dapat mengulang. Terakhir, *Admin* menekan Save untuk menyelesaikan..

## 9. Activity Diagram Mengedit Data Diri Pengguna

*Activity Diagram* menggambarkan alur interaksi yang dilakukan oleh *Admin* saat melakukan proses pengeditan data diri pengguna dalam sistem.

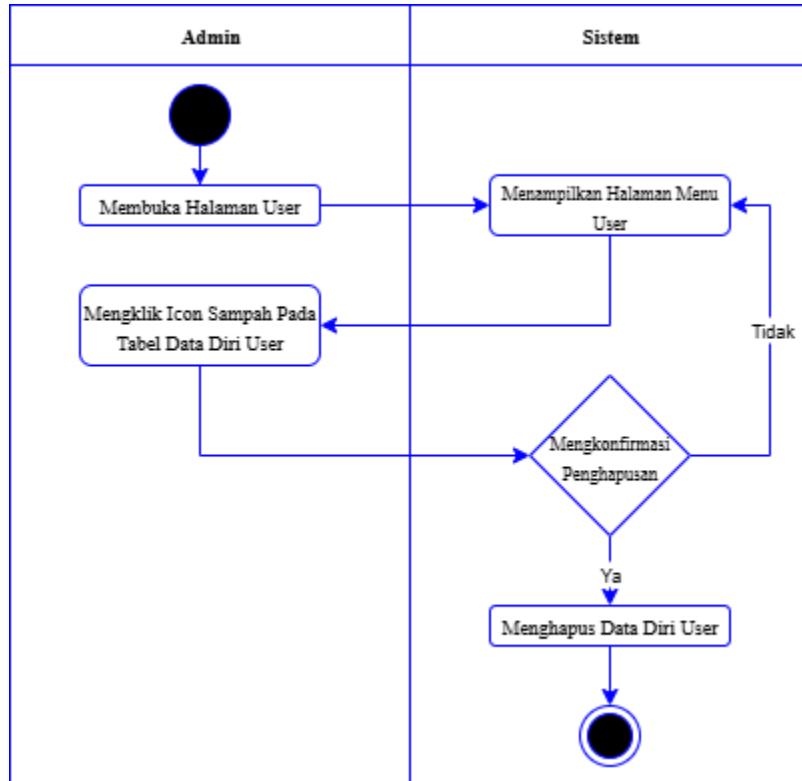


Gambar 4. 26 *Activity Diagram* Mengedit Data Diri *User*

*Activity Diagram* diatas menggambarkan alur aktivitas Admin saat melakukan pengeditan data diri user melalui system. Proses diawali ketika Admin membuka halaman menu pengguna. Setelah halaman terbuka, Admin memilih ikon pensil pada baris data user yang ingin diperbarui. Sistem kemudian menampilkan *form* ubah data yang memuat detail informasi. Admin dapat memperbarui data sesuai kebutuhan. Setelah selesai, Admin menekan tombol Save untuk menyimpan perubahan. Selanjutnya, sistem akan memproses pembaruan dan menyimpan data terbaru ke dalam database.

## 10. Activity Diagram Untuk Menghapus Pengguna

*Activity Diagram* untuk alur interaksi yang dilakukan oleh *Admin* saat melakukan proses penghapusan data diri pengguna dalam sistem.

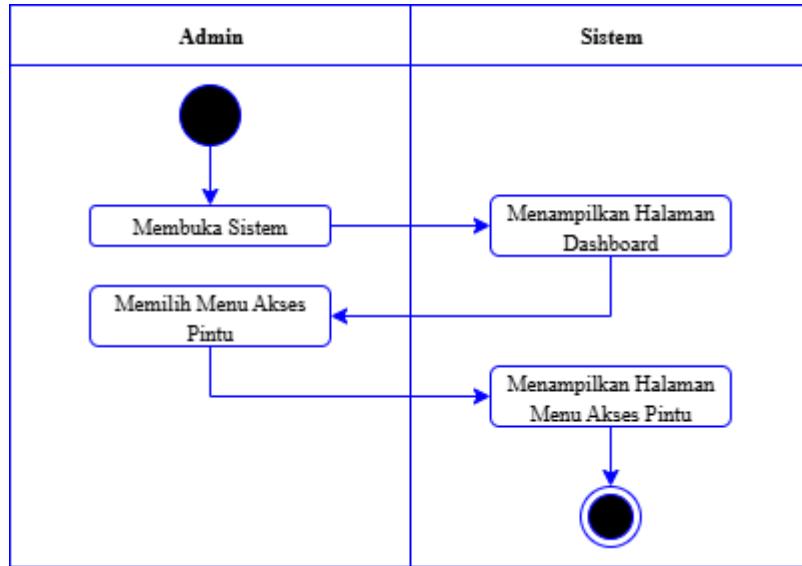


Gambar 4. 27 *Activity Diagram* Untuk Menghapus *User*

*Activity Diagram* pada Gambar 4.27 di atas menggambarkan alur aktivitas yang dilakukan oleh Admin saat akan menghapus data diri user. Proses dimulai saat Admin membuka halaman menu pengguna dan melihat daftar data *user* yang sudah tersimpan di sistem. Admin memilih data pengguna yang akan dihapus dengan mengklik ikon tempat sampah pada baris data tersebut. Setelah itu, sistem akan menampilkan dialog konfirmasi untuk memastikan bahwa admin benar-benar ingin menghapus data tersebut secara permanen. Jika admin menekan tombol Ya, sistem melanjutkan proses penghapusan dengan menghapus data pengguna dari basis data dan memperbarui tampilan daftar pengguna. Jika admin memilih Tidak, proses dibatalkan otomatis dan sistem akan kembali ke halaman sebelumnya tanpa melakukan perubahan apa pun. Dengan adanya alur ini, sistem membantu meminimalkan risiko penghapusan data yang tidak disengaja oleh admin.

#### 11. *Activity Diagram* Menampilkan Halaman Akses Pintu

*Activity Diagram* ini menggambarkan alur interaksi yang dilakukan oleh Admin saat akan melakukan proses membuka halaman akses pintu.

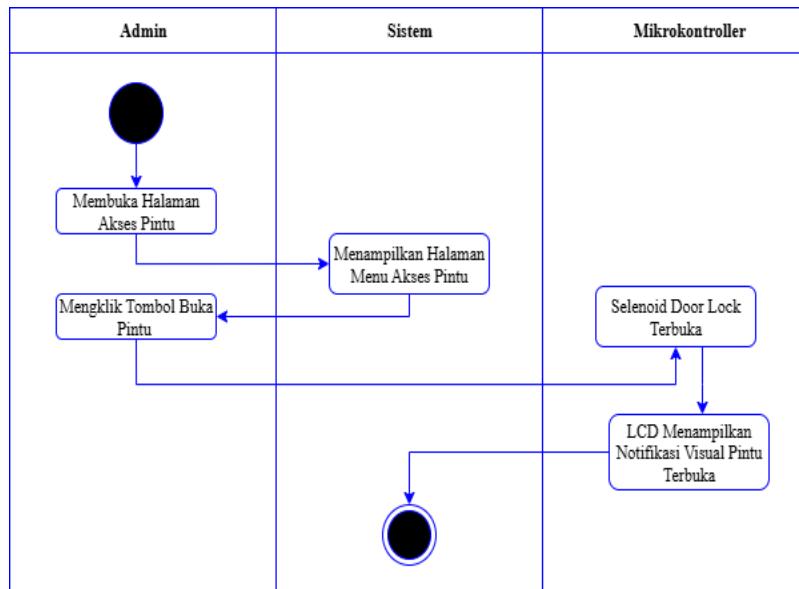


Gambar 4. 28 *Activity Diagram* Menampilkan Halaman Akses Pintu

*Activity Diagram* pada Gambar 4.28 menunjukkan alur saat *Admin* membuka sistem, sistem menampilkan dashboard utama, lalu *Admin* memilih menu Akses Pintu. Sistem kemudian menampilkan halaman Akses Pintu yang digunakan untuk memantau dan mengelola riwayat akses pintu pada rumah.

#### 12. *Activity Diagram* Membuka Pintu dari Sistem

*Activity Diagram* ini menggambarkan alur interaksi yang dilakukan pada saat akan membuka pintu melalui *dashboard* pada halaman akses pintu .

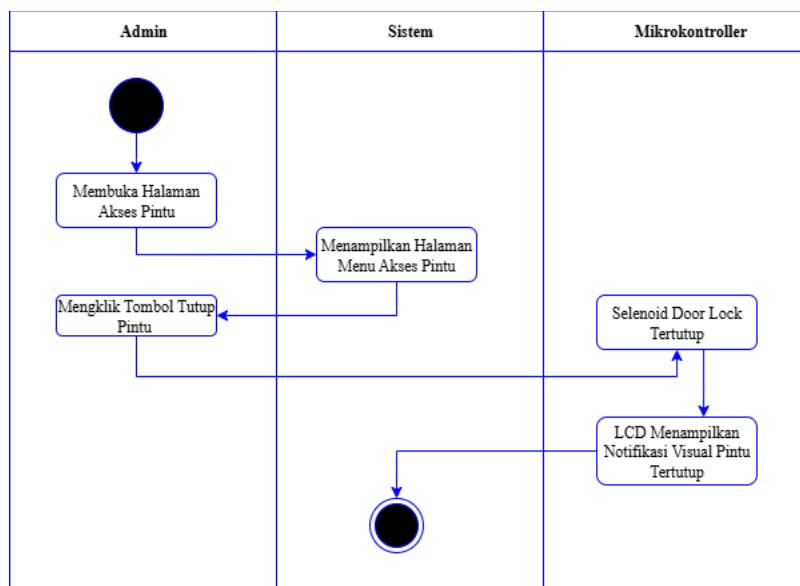


Gambar 4. 29 *Activity Diagram* Membuka Pintu dari sistem

*Activity Diagram* pada Gambar 4.29 ini menunjukkan alur saat *Admin* membuka halaman Akses Pintu. Kemudian sistem menampilkan halaman akses pintu. Setelah *Admin* menekan tombol buka, sistem mengirimkan perintah pada mikrokontroler untuk membuka *solenoid door lock*, lalu LCD juga merespon dengan menampilkan notifikasi pintu terbuka.

### 13. *Activity Diagram* Menutup Pintu dari Sistem

*Activity Diagram* ini menggambarkan alur interaksi yang dilakukan pada saat akan menutup pintu melalui *dashboard* pada halaman akses pintu .

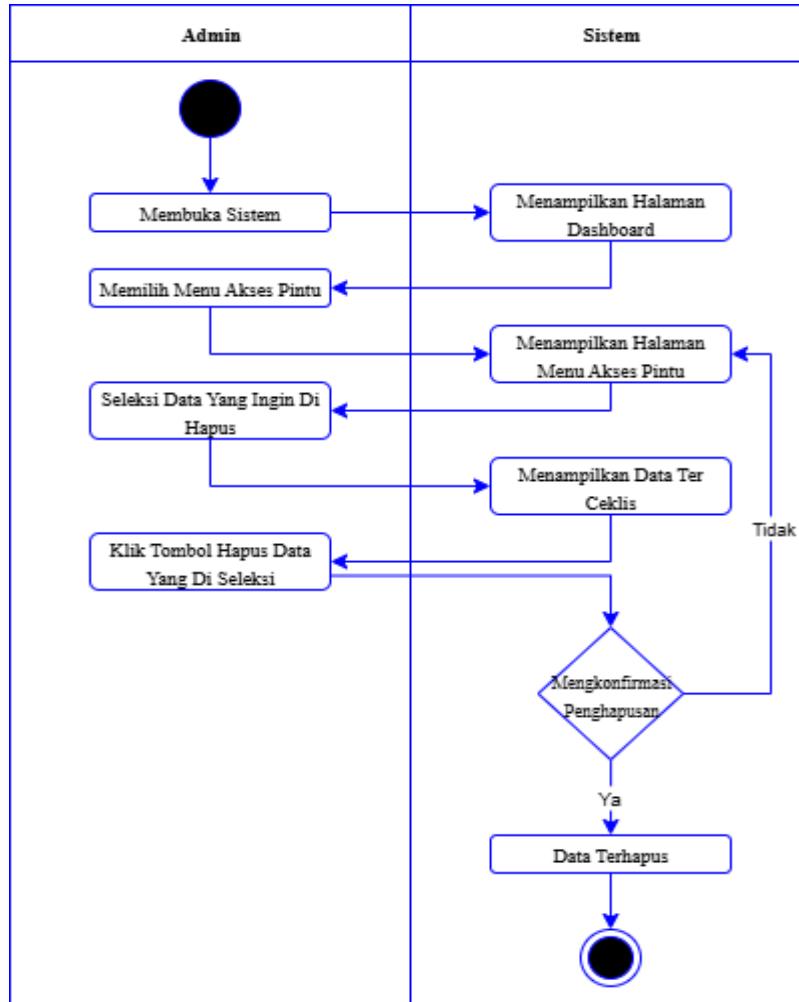


Gambar 4. 30 *Activity Diagram* Menutup Pintu dari Sistem

*Activity Diagram* pada Gambar 4.30 diatas menunjukkan alur saat *Admin* akan membuka halaman Akses Pintu. Pada halaman tersebut *Admin* dapat menekan tombol tutup dan sistem mengirimkan perintah ke mikrokontroler untuk menonaktifkan *solenoid doorlock*, dan lcd menampilkan notifikasi bahwa pintu telah tertutup. Proses ini memastikan pintu terkunci kembali dengan aman dan sistem siap menerima perintah berikutnya.

### 14. *Activity Diagram* Menghapus Data Hasil Akses Pintu

*Activity Diagram* ini menggambarkan alur interaksi yang dilakukan oleh *Admin* pada saat akan menghapus data riwayat akses pintu melalui *dashboard* pada halaman akses pintu .



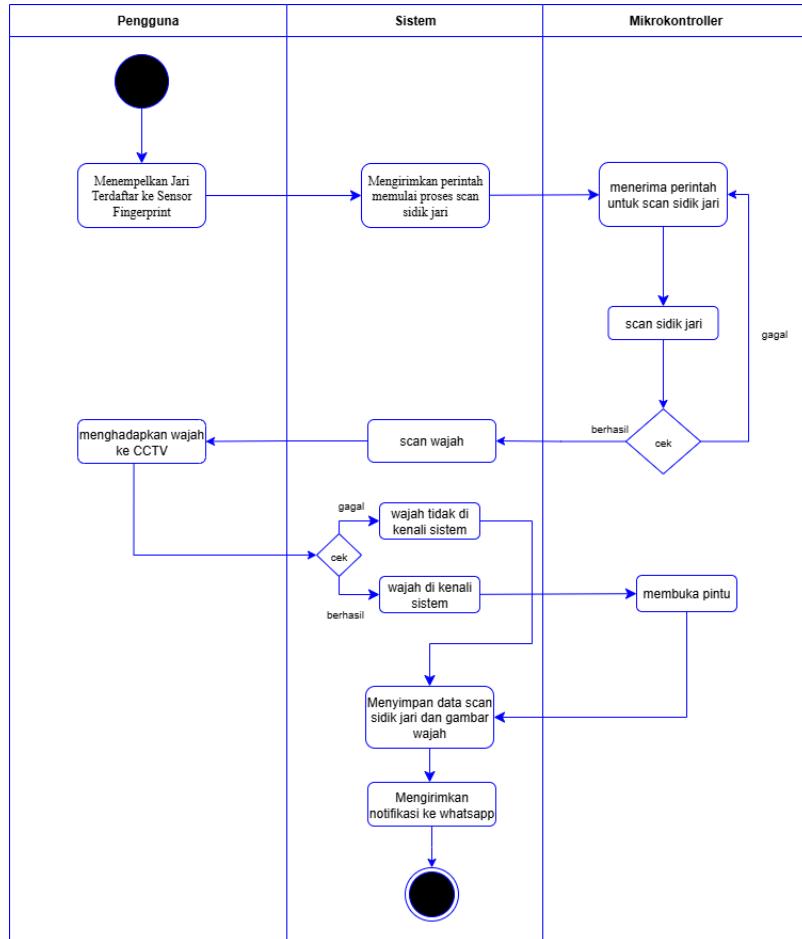
Gambar 4. 31 *Activity Diagram* Menghapus Data

Hasil Akses Pintu

*Activity Diagram* ini menunjukkan alur saat *Admin* menghapus data akses pintu. Proses dimulai ketika *Admin* membuka sistem dan sistem menampilkan halaman *dashboard*. *Admin* kemudian memilih menu Akses Pintu, dan sistem merespons dengan menampilkan halaman tersebut. Selanjutnya, *Admin* menyeleksi data yang ingin dihapus. Sistem menampilkan data yang telah dicentang dan meminta konfirmasi penghapusan. Jika *Admin* menyetujui, maka sistem menghapus data tersebut dan proses berakhir.

### 15. *Activity Diagram* Akses Pintu dengan Autentikasi

*Activity Diagram* dibawah ini menggambarkan alur interaksi yang dilakukan user pada saat akan membuka pintu melalui autentikasi menggunakan sidik jari dan *face recognition*.

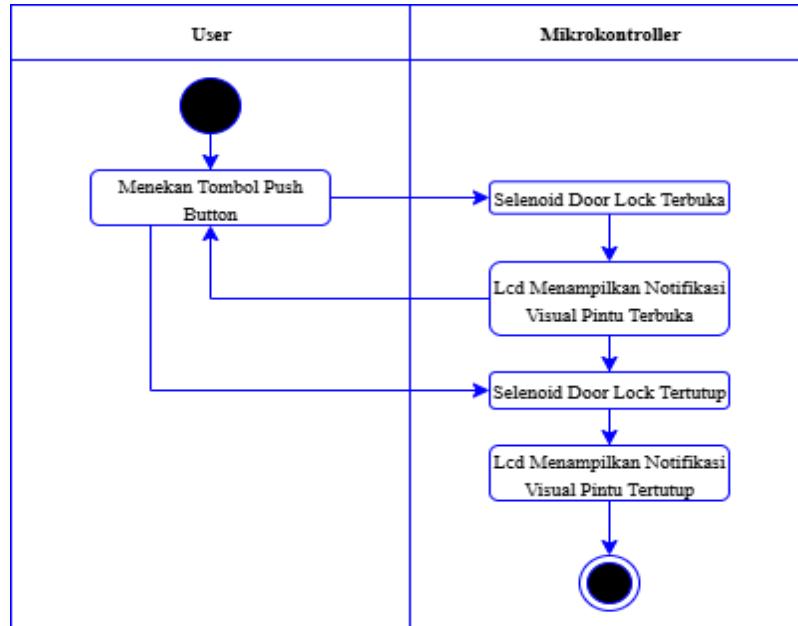


Gambar 4. 32 *Activity Diagram* Akses Pintu dengan Autentikasi

*Activity Diagram* diatas menjelaskan alur autentikasi pengguna pada sistem *monitoring* keamanan rumah dengan sensor *fingerprint* dan *face recognition*. Proses diawali saat pengguna menempelkan jari yang terdaftar ke sensor *fingerprint*. Sistem kemudian mengirimkan perintah ke mikrokontroler untuk memulai proses pemindaian sidik jari. Jika pemindaian sidik jari berhasil, pengguna diarahkan untuk menghadapkan wajah ke CCTV, lalu sistem melakukan pemindaian wajah. Jika wajah dikenali, sistem membuka pintu secara otomatis. Sebaliknya, jika autentikasi sidik jari atau wajah gagal, sistem akan mengirimkan notifikasi ke WhatsApp pemilik rumah. Selain itu, sistem juga menyimpan data gambar wajah untuk dicatat pada log aktivitas.

#### 16. *Activity Diagram* Akses Pintu dengan Push Button atau Manual

*Activity Diagram* ini menggambarkan alur interaksi yang dilakukan oleh *user* saat akses pintu dengan Push Button atau Manual.

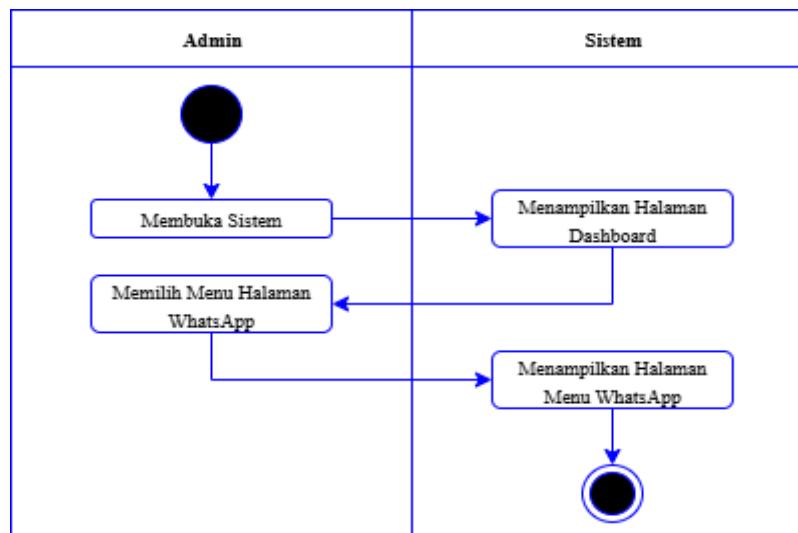


Gambar 4. 33 *Activity Diagram* Akses Pintu dengan *Push Button*

*Activity Diagram* pada Gambar 4.33 menjelaskan alur buka-tutup pintu dengan *push button*. User menekan tombol, solenoid aktif, pintu terbuka, dan LCD menampilkan notifikasi. Tekan tombol lagi, solenoid mati, pintu tertutup, LCD menampilkan notifikasi, lalu sistem siap menerima perintah baru.

#### 17. *Activity Diagram* Menampilkan Halaman WhatsApp

*Activity Diagram* dibawah ini menggambarkan alur interaksi yang dilakukan oleh admin untuk menampilkan halaman menu whatsApp.

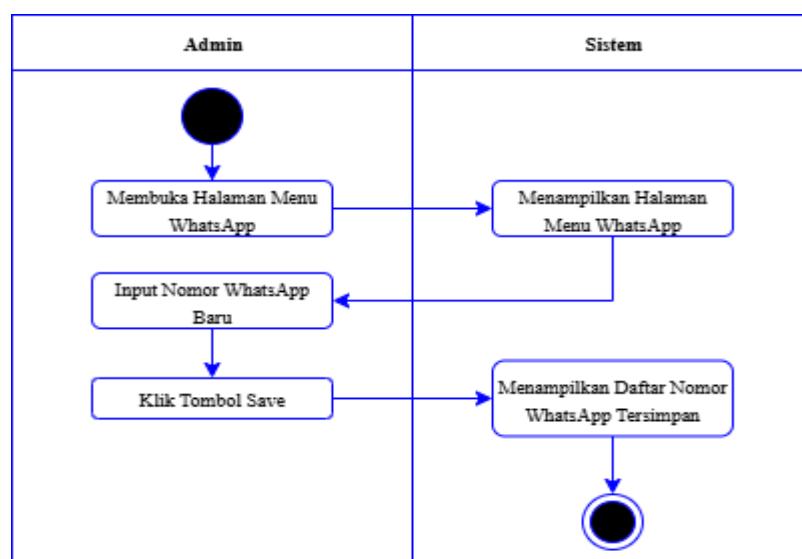


Gambar 4. 34 *Activity Diagram* Menampilkan WhatsApp

*Activity Diagram* diatas menunjukkan alur saat *Admin* membuka sistem monitoring keamanan rumah. Setelah dashboard ditampilkan, *Admin* memilih menu WhatsApp untuk mengatur nomor penerima notifikasi. Sistem kemudian menampilkan halaman WhatsApp untuk mengelola nomor dan pengaturan pesan yang akan dikirim otomatis saat terjadi kegagalan akses pintu.

#### 18. *Activity Diagram* Menambahkan Nomor WhatsApp

*Activity Diagram* ini menggambarkan alur interaksi untuk menambahkan nomor WhatsApp sebagai penerima notifikasi kegagalan akses.

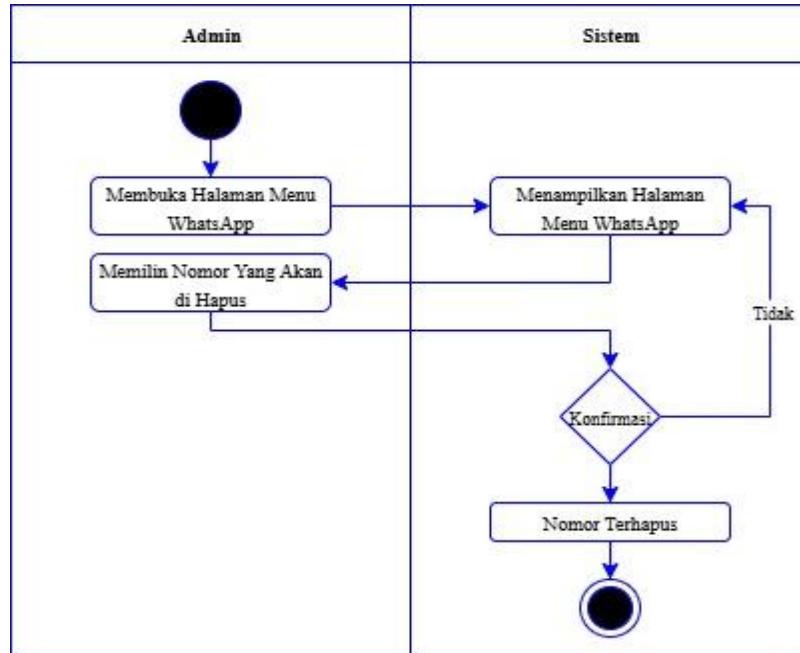


Gambar 4. 35 *Activity Diagram* Menambahkan Nomor WhatsApp

*Activity Diagram* diatas menggambarkan alur aktivitas *Admin* saat akan menambahkan nomor WhatsApp baru pada sistem. Proses dimulai saat *Admin* membuka halaman menu WhatsApp, dan sistem menampilkan halaman tersebut. Lalu *Admin* menginput nomor WhatsApp baru untuk menerima notifikasi keamanan. Setelah itu, *Admin* menekan tombol Save untuk menyimpan. Sistem kemudian menampilkan daftar nomor WhatsApp yang sudah tersimpan sebagai konfirmasi bahwa nomor berhasil ditambahkan.

#### 19. *Activity Diagram* Menghapus Nomor WhatsApp

*Activity Diagram* dibawah ini menggambarkan alur interaksi yang dilakukan oleh admin untuk Menghapus nomor WhatsApp agar berhenti sebagai penerima notifikasi kegagalan akses.

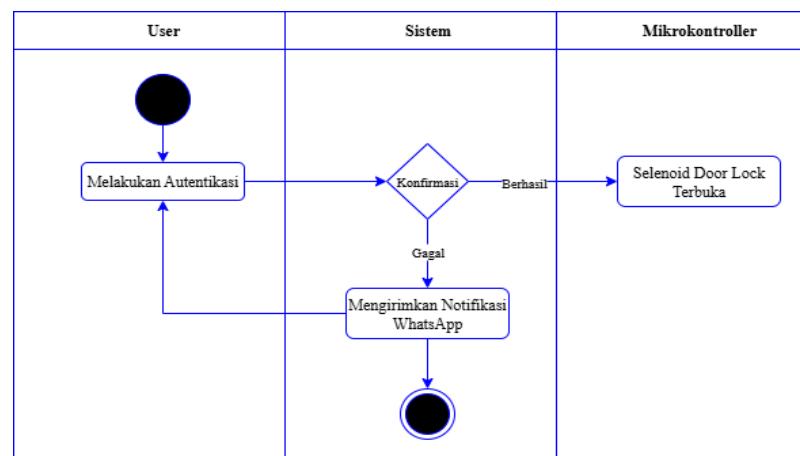


Gambar 4. 36 *Activity Diagram* Menghapus Nomor WhatsApp

*Activity Diagram* di atas menggambarkan alur hapus nomor WhatsApp pada sistem. Proses diawali saat Admin membuka menu WhatsApp, pilih nomor yang akan dihapus, pada dialog konfirmasi, jika memilih Ya, sistem menghapus nomor, jika Tidak, penghapusan dibatalkan. Alur ini memastikan pengelolaan daftar penerima notifikasi aman dan terkontrol.

#### 20. *Activity Diagram* Notifikasi WhatsApp

*Activity Diagram* di bawah ini menggambarkan alur proses yang dilakukan sistem untuk mengelola pengiriman notifikasi melalui WhatsApp.

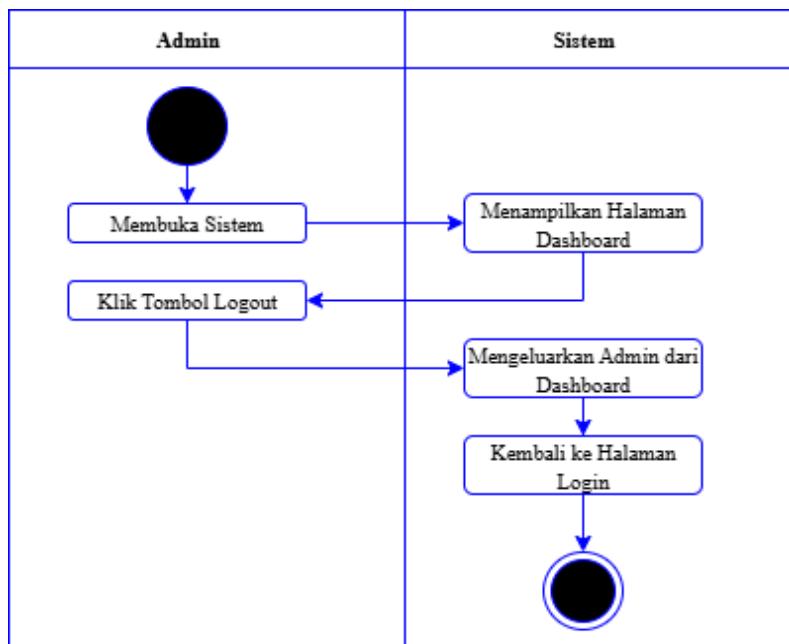


Gambar 4. 37 *Activity Diagram* Notifikasi WhatsApp

*Activity Diagram* diatas menunjukkan autentikasi untuk membuka pintu. Autentikasi yang berhasil memicu sistem membuka *solenoid door lock*. Jika gagal, sistem otomatis mengirim notifikasi WhatsApp kegagalan akses.

## 21. *Activity Diagram Logout*

*Activity Diagram* dibawah ini menjelaskan alur aktivitas saat Admin akan melakukan proses *logout* dari sistem monitoring keamanan rumah..



Gambar 4. 38 *Activity Diagram Logout*

*Activity Diagram* diatas menggambarkan alur proses saat Admin membuka sistem dan berhasil mengakses sistem. Setelah selesai menggunakan sistem, Admin menekan tombol *Logout* untuk keluar. Sistem memproses perintah dengan mengeluarkan Admin dari dashboard dan mengarahkan kembali ke halaman login. Dengan alur ini, keamanan akses tetap terjaga karena seseorang harus ulang untuk dapat kembali menggunakan sistem..

### 4.2.5 Perancangan Antarmuka *Dashboard Web*

Perancangan antarmuka *dashboard web* pada sistem *monitoring* keamanan rumah menghasilkan tampilan halaman yang informatif, terstruktur, dan mudah digunakan. Perancangan ini dibagi dalam dua tahapan, yaitu pembuatan *wireframe* dan perancangan antarmuka (UI). *Wireframe* digunakan untuk

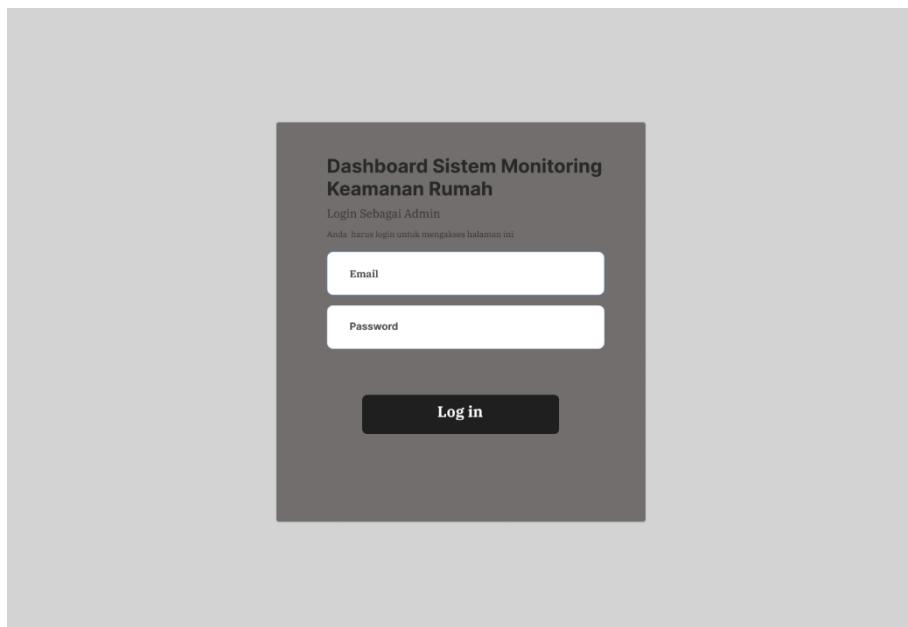
memetakan kerangka tata letak halaman sederhana, sebagai pedoman awal dalam menentukan posisi dan elemen penting. Perancangan antarmuka (UI) menampilkan visual akhir yang telah dilengkapi warna, ikon, tipografi, serta elemen grafis pendukung yang mendukung identitas sistem. Kedua tahapan ini untuk memastikan tampilan *dashboard web* dapat digunakan dengan konsisten, nyaman diakses, serta sesuai dengan kebutuhan penggunanya.

#### A. *Wireframe*

*Wireframe* digunakan untuk menggambarkan kerangka tata letak halaman pada sistem monitoring keamanan rumah dengan sederhana dan terstruktur. Fungsinya sebagai panduan awal pemetaan posisi elemen-elemen penting, mulai dari halaman *login*, *dashboard*, CCTV, pengguna, akses pintu, dan pengaturan notifikasi WhatsApp. Melalui *wireframe*, alur navigasi, menu, tombol, formulir, dan area informasi dapat divisualisasikan dengan jelas sebelum dikembangkan menjadi desain antarmuka final, sehingga proses pengembangan antarmuka lebih terarah, konsisten, dan sesuai dengan kebutuhan pengguna.

##### 1) *Wireframe* Halaman *Login*

Gambar di bawah ini merupakan *wireframe* untuk halaman *login* pada sistem monitoring keamanan rumah.

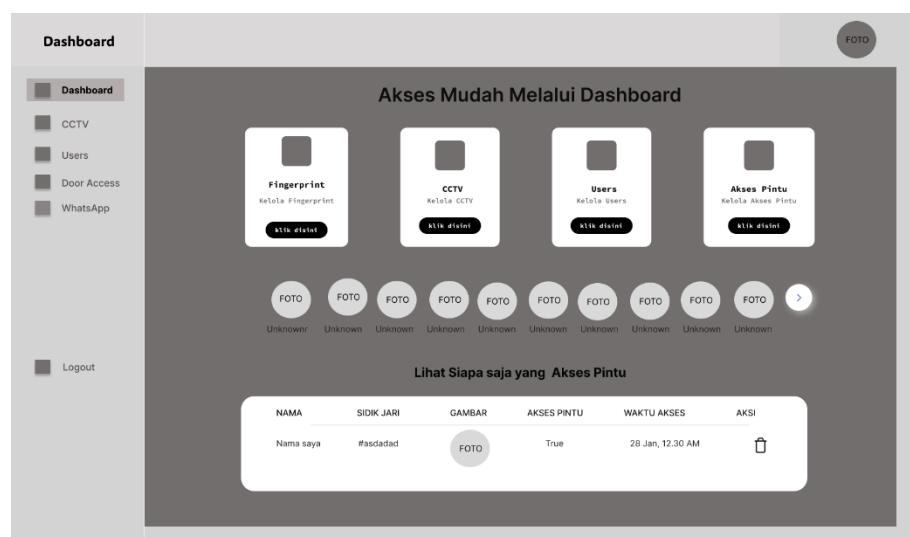


Gambar 4. 39 *Wireframe* Halaman *Login*

*Wireframe* pada halaman *Login* menggambarkan struktur dasar antarmuka *login* bagi *admin*. Perancangan fokus pada penempatan elemen penting, seperti judul halaman, kolom *input email* dan *password*, dan tombol *login*. Tujuannya untuk memetakan alur *login* dengan jelas sebelum diterapkan pada perancangan visual, sehingga dapat memberikan kemudahan *admin* dalam mengakses sistem.

## 2) *Wireframe* Halaman *Dashboard*

Gambar di bawah ini merupakan *wireframe* untuk halaman *Dashboard* pada sistem monitoring keamanan rumah.

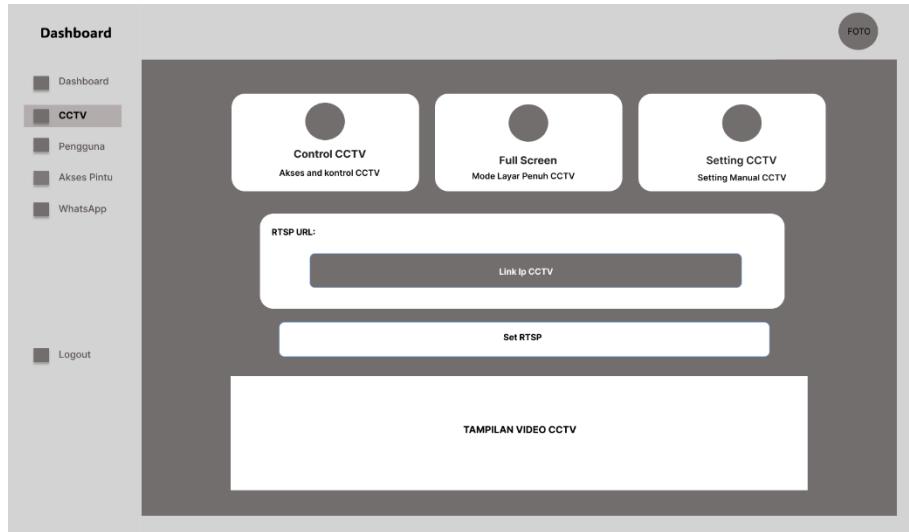


Gambar 4. 40 *Wireframe* Halaman *Dashboard* Utama

*Wireframe* Halaman *Dashboard* menampilkan kerangka dasar tata letak elemen pada halaman utama sistem monitoring keamanan rumah. *Wireframe* pada rancangan ini berfungsi sebagai panduan visual sebelum pengembangan antarmuka finalnya. Menu navigasi utama yaitu (*Dashboard*, *CCTV*, pengguna, akses pintu, dan *WhatsApp*) ditata dengan konsisten pada sisi kiri untuk memudahkan perpindahan setiap halaman. Pada bagian utama halaman digunakan sebagai area kontrol dan informasi, menampilkan konten dinamis seperti status pintu, log aktivitas, dan tampilan *CCTV* secara *real-time*.

## 3) *Wireframe* Halaman Menu *CCTV*

Gambar di bawah ini merupakan *wireframe* untuk halaman *Dashboard* pada sistem monitoring keamanan rumah.

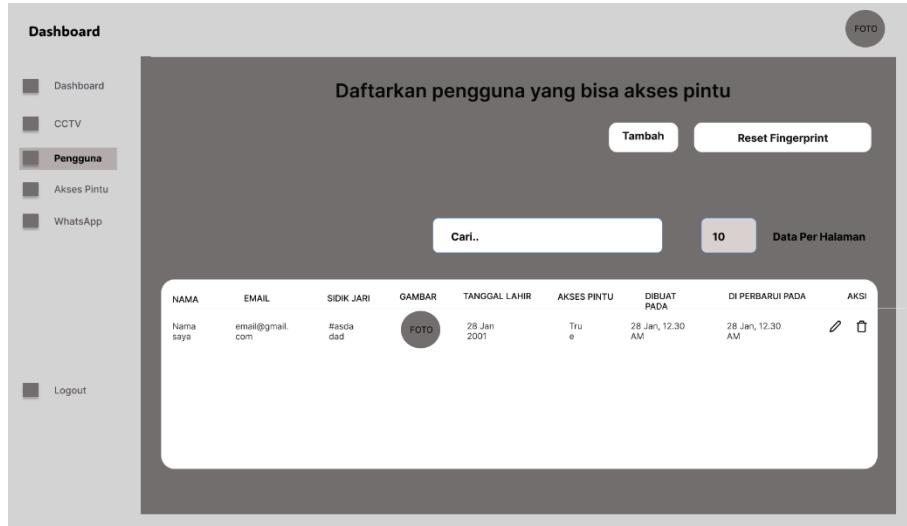


Gambar 4. 41 *Wireframe* Halaman Menu CCTV

*Wireframe* halaman menu CCTV diatas menggambarkan kerangka tata letak menu navigasi (Dashboard, CCTV, Pengguna, Akses Pintu, WhatsApp) di sisi kiri, area utama yang berisi fitur kontrol CCTV, mode layar penuh, pengaturan manual, input RTSP URL, tombol Set RTSP, dan tampilan video *real-time*. Seluruh elemen divisualisasikan sederhana dalam bentuk blok dan teks penanda tanpa detail visual, hingga mempermudah perancangan antarmuka secara terarah.

#### 4) *Wireframe* Halaman Pengguna

Gambar di bawah ini merupakan *wireframe* untuk halaman pengguna pada sistem monitoring keamanan rumah.



Gambar 4. 42 *Wireframe* Halaman Pengguna

*Wireframe* diatas menunjukkan struktur dasar halaman menu Pengguna. Pada sisi kiri terdapat menu navigasi vertikal dengan pilihan menu *Dashboard*, *CCTV*, *Pengguna*, *Akses Pintu*, *WhatsApp*, dan *Logout*. Halaman ini juga telah dilengkapi tombol Tambah yang digunakan untuk menambah data pengguna baru, tombol *Reset Fingerprint*, kolom pencarian, serta opsi jumlah data per halaman. Pada bagian bawah ditampilkan tabel data user yang memuat informasi penting guna mempermudah *admin* dalam mengelola akses user ke dalam sistem.

### 5) *Wireframe* Halaman Tambah Pengguna

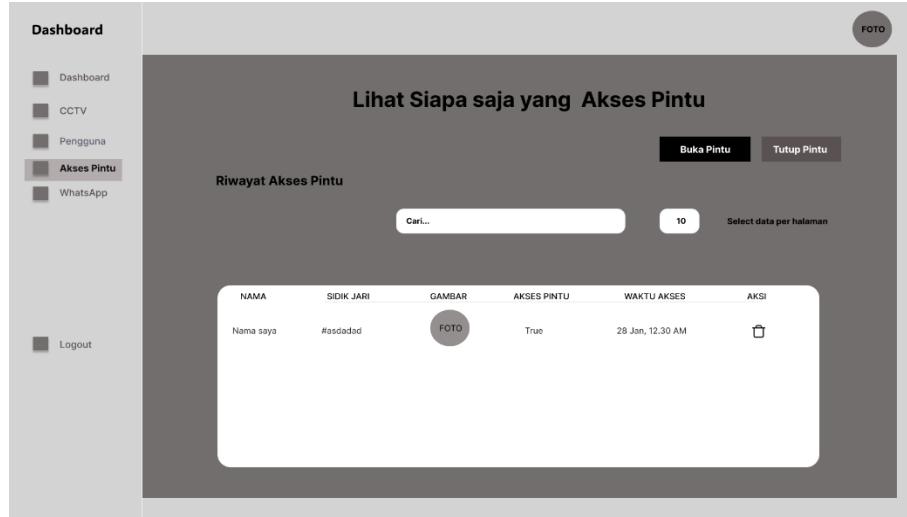
Gambar di bawah ini merupakan *wireframe* untuk halaman tambah pengguna pada sistem monitoring keamanan rumah.

Gambar 4. 43 *Wireframe* Halaman Tambah pengguna

*Wireframe* ini menunjukkan struktur formulir tambah data pengguna yang diberikan akses membuka pintu. Tata letak elemen dirancang terstruktur, dengan navigasi vertikal di sisi kiri untuk mempermudah perpindahan antar halaman. Pada area utama, terdapat formulir input Nama, *Email*, Tanggal Lahir, Alamat, *Finger ID*, dan status Akses Pintu. Tersedia tombol *Scan* untuk pengambilan data sidik jari, tombol *Save* untuk menyimpan data pengguna dalam sistem.

### 6) *Wireframe* Halaman Akses Pintu

Gambar di bawah ini merupakan *wireframe* untuk halaman Dashboard pada sistem monitoring keamanan rumah

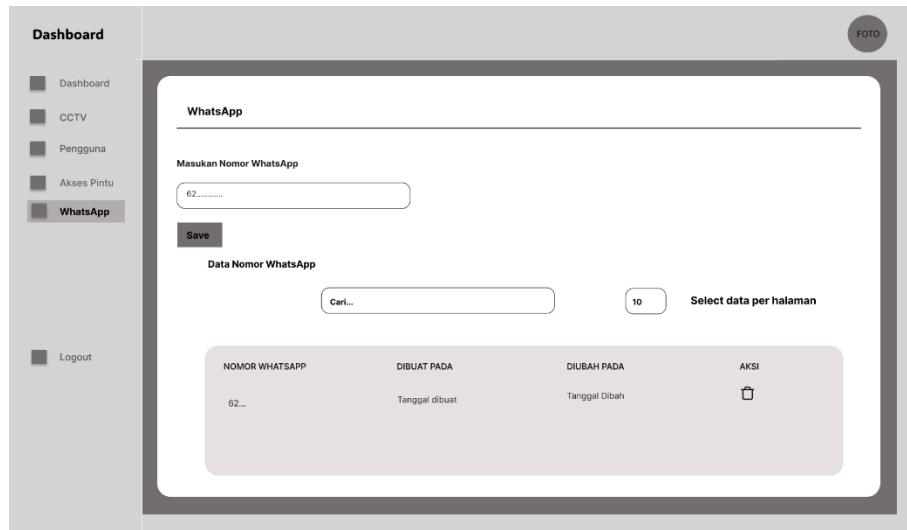


Gambar 4. 44 *Wireframe* Halaman Akses Pintu

*Wireframe* ini dirancang untuk menampilkan informasi riwayat akses pengguna terhadap sistem. Pada sisi kiri terdapat menu navigasi (Dashboard, CCTV, Pengguna, Akses Pintu, WhatsApp, dan *Logout*). Pada area utama, ditampilkan judul “Lihat Siapa Saja yang Akses Pintu” beserta dua tombol aksi utama, yaitu Buka Pintu dan Tutup Pintu, yang memungkinkan admin untuk dapat mengontrol pintu secara langsung melalui dashboard sistem.

## 7) *Wireframe* Halaman WhatsApp

Gambar di bawah ini merupakan *wireframe* untuk halaman WhatsApp pada sistem monitoring keamanan rumah.



Gambar 4. 45 *Wireframe* Halaman WahtsApp

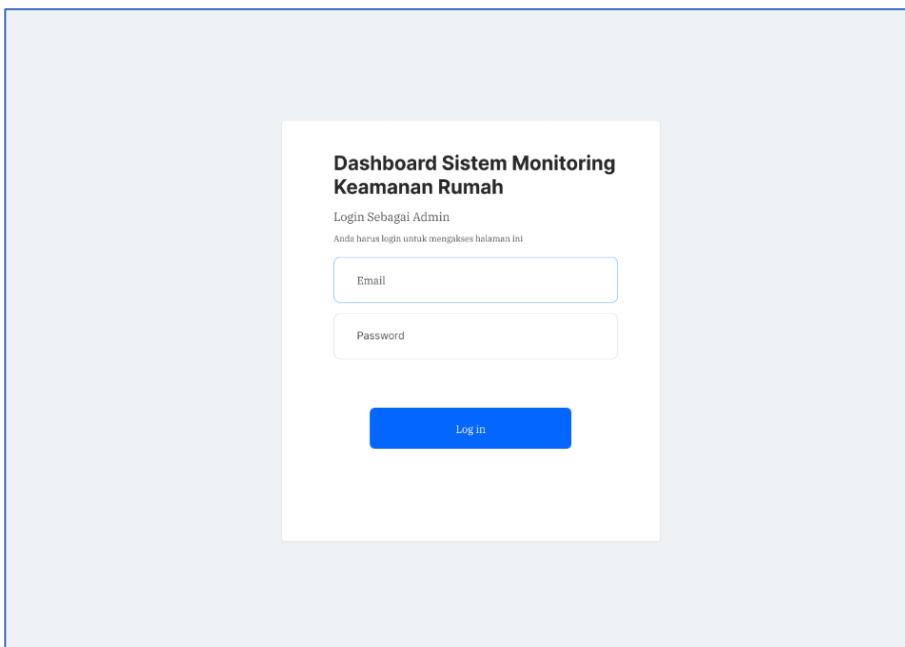
*Wireframe* pada gambar di atas menunjukkan rancangan halaman WhatsApp pada sistem monitoring keamanan rumah yang dirancang untuk mempermudah pengelolaan nomor WhatsApp sebagai media notifikasi. Tampilan disusun sederhana dengan elemen *form input* nomor, tombol Simpan, kolom pencarian, opsi jumlah data per halaman, tabel yang menampilkan detail nomor, tanggal pembuatan atau perubahan, dan *ikon* hapus untuk menghapus data nomor.

#### B. Perancangan Antarmuka (UI)

Perancangan antarmuka (UI) pada sistem monitoring keamanan rumah dikembangkan berdasarkan *wireframe* yang telah disusun sebelumnya untuk setiap halaman. Rancangan final ini menampilkan elemen visual secara mendetail, seperti penerapan warna, *ikon*, dan tipografi yang mendukung identitas sistem. Penggunaan elemen visual ini bertujuan untuk memperjelas tata letak, memperindah tampilan, serta mempermudah admin sebagai pengguna dalam memahami navigasi di seluruh halaman, sehingga proses pengelolaan fitur dapat dilakukan dengan lebih mudah dan nyaman.

##### b Perancangan Antarmuka (UI) Halaman *Login*

Perancangan antarmuka halaman *login* pada sistem monitoring keamanan rumah ditunjukkan pada gambar di bawah.

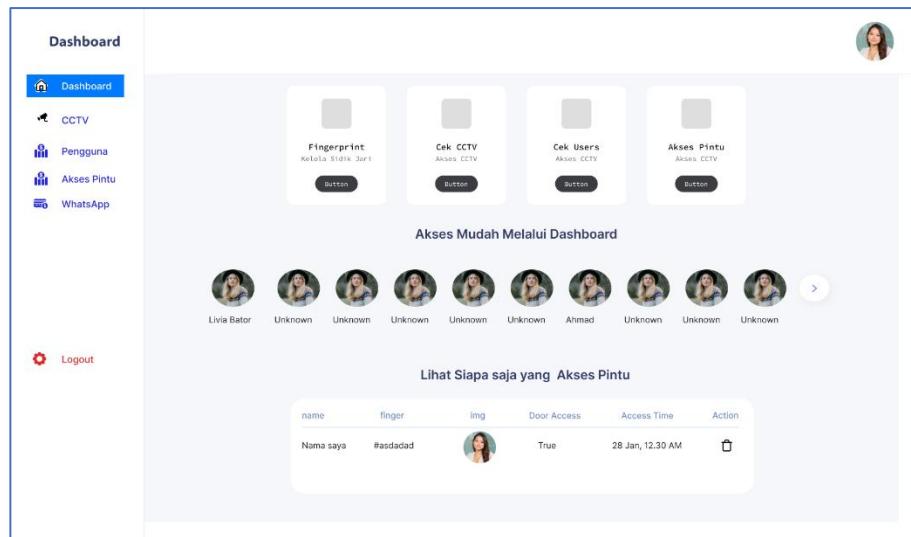


Gambar 4. 46 Perancangan Antarmuka (UI) Halaman *Login*

Perancangan Antarmuka (UI) halaman *Login* merupakan pengembangan dari *Wireframe* dengan elemen visual yang lebih lengkap. Tampilan ini menggunakan warna latar putih dan aksen biru untuk memberikan kesan profesional dan bersih. *Elemen input* dan tombol dibuat responsif, sehingga memudahkan *Admin* saat mengakses sistem. Rancangan ini mendukung prinsip *user-friendly* dan fokus pada pengalaman pengguna yang ringkas dan nyaman.

### c Desain Antarmuka (UI) *Dashboard*

Perancangan antarmuka halaman *Dashboard* pada sistem monitoring keamanan rumah ditunjukkan pada gambar di bawah.

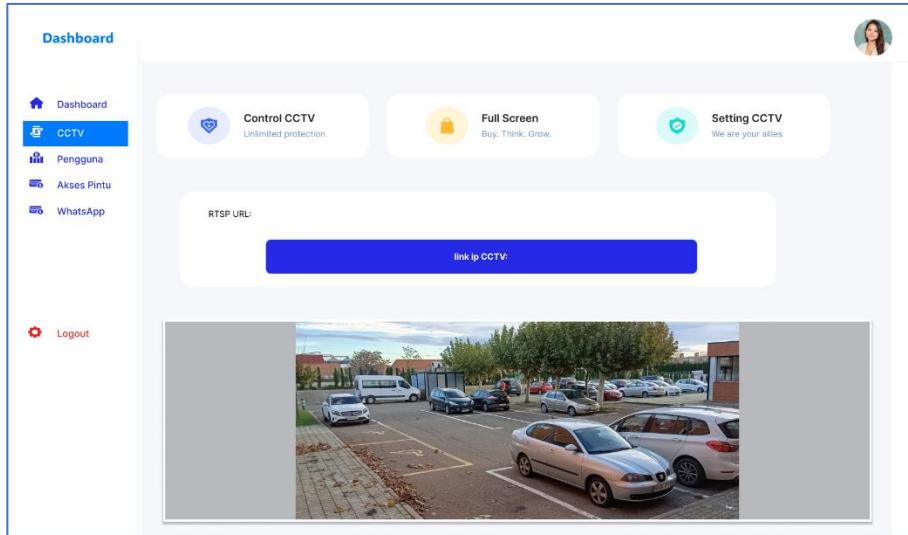


Gambar 4. 47 Desain Antarmuka (UI) *Dashboard*

Perancangan Antarmuka (UI) halaman *dashboard* memperjelas hasil dari *Wireframe* dengan warna, *ikon*, gambar profil, tipografi, dan elemen interaktif yang menarik. Pada bagian *Shortcut* di *dashboard* dibuat dengan visual yang mudah dipahami, bagian profil pengguna dibuat tampilan menggunakan gambar dan tabel akses pintu serta dilengkapi dengan aksi hapus untuk mempermudah pengelolaan. Perancangan ini mempermudah admin sebagai pengguna dalam mengelola sistem keamanan dengan praktis dan informatif.

### d Perancangan Antarmuka (UI) Halaman Menu *CCTV*

Perancangan antarmuka halaman Menu *CCTV* pada sistem monitoring keamanan rumah ditunjukkan pada gambar di bawah.

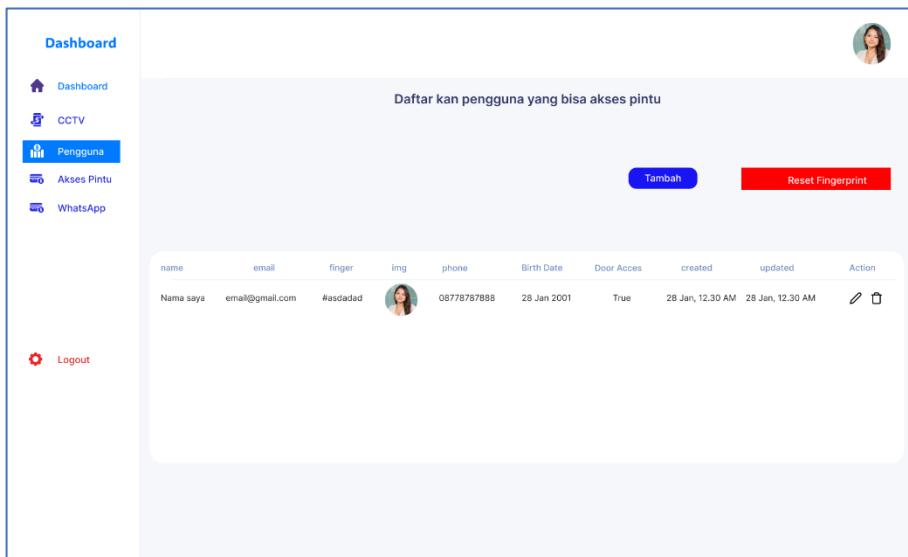


Gambar 4. 48 Perancangan Antarmuka (UI) Halaman Menu CCTV

Perancangan antarmuka Halaman menu CCTV yang telah diberi warna yang lebih menarik, penggunaan *ikon* berwarna untuk memperjelas fungsi tiap fitur, tata letak yang rapi sehingga memudahkan *admin* dalam mengoperasikan CCTV. Area *input* rtsp url dibuat menonjol dengan tombol link berwarna cerah, dan hasil tampilan CCTV divisualisasikan dalam bentuk *feed* video yang nyata.

#### e Perancangan Antarmuka (UI) Halaman Pengguna

Perancangan antarmuka halaman Pengguna pada sistem monitoring keamanan rumah ditunjukkan pada gambar di bawah.



Gambar 4. 49 Perancangan Antarmuka (UI) Halaman Pengguna

Perancangan ini menampilkan visual akhir yang telah disesuaikan dengan elemen warna, ikon, tipografi, dan komponen interaktif. Menu navigasi di sisi kiri menggunakan ikon berwarna biru untuk fitur yang sedang aktif, menu *Logout* diberi aksen merah agar terlihat. Di area utama, judul halaman ditampilkan dengan tipografi jelas, tombol Tambah berwarna biru dan tombol *Reset Fingerprint* berwarna merah untuk membedakan fungsi aksi penting. Tabel data pengguna dirancang dengan susunan baris dan kolom yang terstruktur, dilengkapi foto profil sebagai pendukung fitur face recognition pada sistem keamanan rumah. Tabel dilengkapi dengan ikon pensil untuk mengedit data pengguna dan ikon tempat sampah untuk menghapus data yang tidak lagi digunakan. Rancangan ini untuk meningkatkan kemudahan pengelolaan dan menjaga akurasi data dalam sistem.

#### f Perancangan Antarmuka (UI) Halaman Tambah Pengguna

Perancangan antarmuka halaman Tambah Pengguna pada sistem monitoring keamanan rumah ditunjukkan pada gambar di bawah.

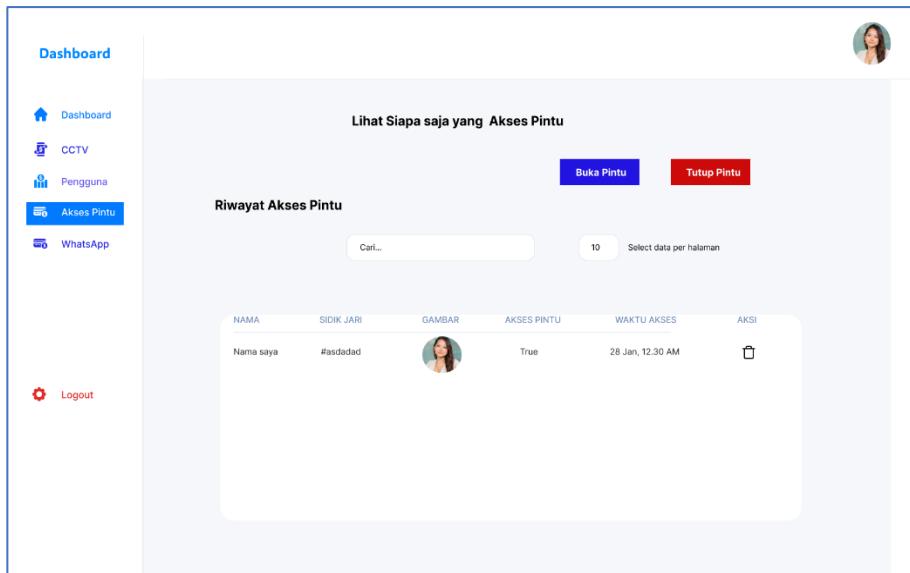
Gambar 4. 50 Perancangan Antarmuka (UI) Halaman Tambah Pengguna

Perancangan antarmuka halaman Tambah Pengguna menampilkan tampilan yang bersih, *modern*, dan ramah pengguna. Perpaduan warna biru dan putih memberikan kesan profesional serta memudahkan fokus pada isi formulir. Elemen input seperti kotak teks, *dropdown*, serta tombol interaktif dibuat dengan jarak antar elemen yang proporsional untuk meningkatkan keterbacaan dan

meminimalkan kesalahan *input* data. Penempatan foto profil di sisi kiri atas formulir memberikan identitas visual bagi pengguna yang sedang ditambahkan, sedangkan *placeholder* pada setiap kolom berfungsi sebagai panduan bagi *admin* dalam mengisi data. Selain itu, rancangan ini juga memperhatikan alur penggunaan yang praktis, mulai dari pengisian data hingga penyimpanan, sehingga dapat mendukung proses manajemen pengguna terintegrasi dengan sistem keamanan rumah secara keseluruhan.

#### g Perancangan Antarmuka (UI) Halaman Akses Pintu

Perancangan antarmuka halaman Akses Pintu pada sistem monitoring keamanan rumah ditunjukkan pada gambar di bawah.



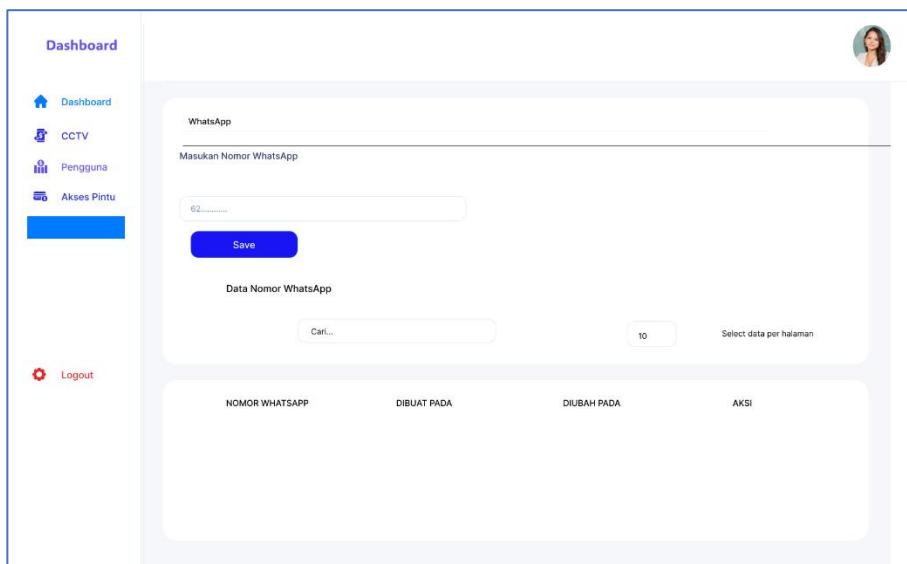
Gambar 4. 51 Perancangan Antarmuka (UI) Halaman Akses Pintu

Perancangan antarmuka halaman Akses Pintu dirancang dengan tata letak yang bersih, terstruktur, dan mudah dipahami. Pada bagian atas, terdapat tombol Buka Pintu berwarna biru dan Tutup Pintu berwarna merah yang memudahkan admin untuk mengontrol akses pintu secara *real-time*. Selain itu, disediakan fitur pencarian data dan *dropdown* untuk memilih jumlah data yang ditampilkan per halaman, sehingga mempermudah pengguna dalam menelusuri riwayat akses. Tabel di bagian bawah menampilkan informasi penting, seperti Nama, ID Sidik Jari, Gambar Wajah, Status Akses, Waktu Akses, dan Aksi (hapus), sehingga

mendukung pemantauan aktivitas secara detail. Penataan elemen ini dirancang agar *admin* dapat dengan cepat meninjau log aktivitas, mengambil keputusan, serta mengontrol akses pintu dengan lebih aman, praktis, dan terorganisir.

#### h Perancangan Antarmuka (UI) halaman WhatsApp

Perancangan antarmuka halaman WhatsApp pada sistem monitoring keamanan rumah ditunjukkan pada gambar di bawah.



Gambar 4. 52 Perancangan Antarmuka (UI) Halaman Whatsapp

Perancangan antarmuka (UI) halaman WhatsApp menggunakan rancangan yang minimalis dan bersih, dengan dominasi warna putih serta aksen biru pada elemen interaktif seperti tombol *Save* untuk memberikan kesan profesional dan memudahkan fokus penggunaannya. Navigasi di sisi kiri memberikan akses cepat ke menu lain seperti *Dashboard*, *CCTV*, *Pengguna*, dan *Akses Pintu*, sehingga mendukung perpindahan antar halaman dengan tepat. Penempatan elemen yang konsisten, ikon yang intuitif, serta tata letak *form input* nomor WhatsApp, kolom pencarian, opsi jumlah data per halaman, dan tabel data disusun sedemikian rupa untuk mendukung alur kerja yang jelas dan terstruktur. Rancangan ini diharapkan dapat meningkatkan kenyamanan *admin* dalam menggunakan sistem untuk memonitor dan mengelola jalur komunikasi melalui WhatsApp sebagai media notifikasi keamanan rumah, sehingga mendukung terciptanya sistem monitoring yang responsif, praktis, dan mudah dioperasikan.

## BAB V

### IMPLEMENTASI DAN PENGUJIAN

#### 5.1 Implementasi

Setelah melalui tahap analisis dan perancangan sistem secara menyeluruh, selanjutnya adalah implementasi. Implementasi dilakukan untuk menerapkan hasil dari perancangan sistem *monitoring* keamanan rumah yang telah dibuat, baik dari sisi perangkat keras (*Hardware*) maupun perangkat lunak (*Software*). Tahap ini menjadi langkah nyata dalam merealisasikan sistem agar dapat berfungsi sesuai dengan tujuan dan kebutuhan yang telah ditentukan pada fase perancangan.

##### 5.1.1 Listing Program

###### 1. Listing Program App.py

Listing program app.py berfungsi sebagai *entry point* untuk menjalankan *server web*, komunikasi *real-time* menggunakan *SocketIO*, serta layanan *background* untuk MQTT dan autentikasi *fingerprint*.

```
from app import create_app, socketio
import os
import threading
from app.services.mqtt_service import run_mqtt_service
from app.services.finger_service
import start_finger_Login_loop
app = create_app()
def start_background_services():
    # Langsung jalankan MQTT thread
    threading.Thread(target=lambda: run_mqtt_service(app),
daemon=True).start()
    # Jalankan loop pengiriman Login setiap 10 detik
    start_finger_Login_loop()
if __name__ == "__main__":
```

```
# Mulai background services
start_background_services()
socketio.run(app, debug=True, host="0.0.0.0", port=5000,
use_reloader=False)
```

## 2. Listing Program *Face Recognition*

Listing program di bawah menunjukkan proses memuat dataset wajah menggunakan metode face encoding, mengambil snapshot CCTV, serta streaming video real-time ke dashboard. Bagian ini memanfaatkan library OpenCV dan face\_recognition dengan cache encoding untuk efisiensi.

```
import cv2
import face_recognition
import numpy as np
import datetime
import os
import time
from camera_stream import frame_buffer, frame_lock, known_face_encodings, known_face_names, faces_loaded, threading, streaming_active
import pickle
import hashlib

def load_known_faces():
    global known_face_encodings, known_face_names
    known_face_encodings = []
    known_face_names = []
    base_path = "app/static/train model/snapshots"
    cache_path = "app/camera/face_cache.pkl"
    cache_data = {}

    # Load cache if exists
    if os.path.exists(cache_path):
        with open(cache_path, 'rb') as f:
            cache_data = pickle.load(f)
```

```

updated_cache = {}

try:
    for folder_name in os.listdir(base_path):
        folder_path = os.path.join(base_path, folder_name)
        if os.path.isdir(folder_path):
            user_name = folder_name
            for filename in os.listdir(folder_path):
                if filename.lower().endswith('.jpg', '.jpeg', '.png'):
                    file_path = os.path.join(folder_path, filename)

# Buat hash berdasarkan isi file untuk mendeteksi perubahan
    with open(file_path, 'rb') as img_file:
        file_hash = hashlib.md5(img_file.read()).hexdigest()
        cache_key = f'{user_name}/{filename}'

# Cek apakah sudah di-cache dan tidak berubah
    if cache_key in cache_data:
        if cache_data[cache_key]['hash'] == file_hash:
            encoding = cache_data[cache_key]['encoding']
            print(f"♻️ Cache digunakan untuk: {cache_key}")

    else:
        image = face_recognition.load_image_file(file_path)
        face_encodings = face_recognition.face_encodings(image)
        if not face_encodings:
            print(f"[!] Tidak ada wajah di file: {file_path}")
            continue
        encoding = face_encodings[0]
        print(f"✅ Wajah dimuat ulang: {cache_key}")

# Simpan ke data runtime dan cache baru
        known_face_encodings.append(encoding)
        known_face_names.append(user_name)
        updated_cache[cache_key] = {
            "hash": file_hash,

```

```

        "encoding": encoding
    }

except Exception as e:
    print(f"[!] Gagal memuat wajah: {e}")

# Simpan cache
    with open(cache_path, 'wb') as f:
        pickle.dump(updated_cache, f)
    return known_face_encodings, known_face_names

#* Screenshot Foto CCTV

def take_snapshot(RTSP_URL):
    print("[INFO] Connecting to CCTV...")
    cap = cv2.VideoCapture(RTSP_URL)
    if not cap.isOpened():
        print("[ERROR] Failed to connect to CCTV.")
        return None
    print("[INFO] Reading frame...")
    ret, frame = cap.read()
    cap.release()
    if not ret:
        print("[ERROR] Failed to capture frame.")
        return None
    # Buat folder jika belum ada
    output_dir = "app/static/snapshots/captured"
    os.makedirs(output_dir, exist_ok=True)

# Simpan frame ke file
    timestamp = datetime.datetime.now().strftime("%Y%m%d_%H%M%S")
    filename = f"{output_dir}/snapshot_{timestamp}.jpg"
    cv2.imwrite(filename, frame)
    print(f"[INFO] Snapshot saved at {filename}")
    return filename

def start_capture_thread(RTSP_URL):

```

```

global streaming_active

cap = cv2.VideoCapture(RTSP_URL)

cap.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
cap.set(cv2.CAP_PROP_BUFFERSIZE, 1)

if streaming_active:
    return

streaming_active = True

# Jalankan thread capture

threading.Thread(target=capture_thread, args=(RTSP_URL,),
daemon=True).start()

def capture_thread(RTSP_URL):
    global frame_buffer

    cap = cv2.VideoCapture(RTSP_URL, cv2.CAP_FFMPEG)

    if not cap.isOpened():

        print("[ERROR] Failed to open RTSP stream.")

        return

    while True:

        success, frame = cap.read()

        if not success:

            print("[WARNING] Failed to read frame. Retrying...")

            time.sleep(0.5) # kasih delay biar gak 100% CPU usage

            continue

        with frame_lock:

            frame_buffer = frame

#* ===== Video Frame Generator Stream=====

def gen_frames():

    timeout = 10 # detik

    start_time = time.time()

    while True:

        if frame_buffer is None:

            if time.time() - start_time > timeout:

```

```
        print("[INFO] Timeout: Tidak ada frame yang
diterima.")
        break
    time.sleep(1/40)
    continue
    start_time = time.time()
    try:
        with frame_lock:
            frame = frame_buffer.copy()
            ret, buffer = cv2.imencode('.jpg', frame)
            if not ret:
                continue
            frame = buffer.tobytes()
            yield (b'--frame\r\n'
                   b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')
    except Exception as e:
        print(f"[ERROR] Streaming frame failed: {e}")
        continue
```

Listing program di atas merupakan cuplikan dari keseluruhan kode yang digunakan. Adapun source code lengkap dapat dilihat pada Lampiran 4 untuk memberikan gambaran menyeluruh terkait implementasi sistem.

### 5.1.2 Implementasi Modul Elektronika

Implementasi modul elektronika merupakan tahap penerapan hasil dari analisis dan perancangan sistem *monitoring* keamanan rumah berbasis IoT dan aplikasi web. Meskipun sistem belum diimplementasikan secara luas di lingkungan nyata, pengujian dilakukan dalam bentuk prototipe sebagai perwakilan skenario penggunaan sesungguhnya. Waktu dan tempat penerapan sistem sebagai berikut:

### Waktu dan Tempat Implementasi:

Tempat : Rumah Ketua RT 02/RW 08  
 Alamat : Kp. Cimomplo rt02/rw08 Desa Wargaluyu Kecamatan Arjasari  
           Kabupaten Bandung  
 Waktu : Juli 2025

Implementasi ini bertujuan untuk memvalidasi fungsionalitas perangkat keras dan perangkat lunak sistem, termasuk proses autentikasi melalui *Face Recognition* dan *Fingerprint*, notifikasi otomatis ke WhatsApp, serta tampilan dan manajemen data melalui *dashboard* web.

#### 5.1.3 Spesifikasi Sistem

Spesifikasi sistem menjelaskan kebutuhan perangkat keras dan perangkat lunak yang digunakan dalam pengembangan dan implementasi sistem *monitoring* keamanan rumah. Informasi ini penting agar sistem berjalan dengan optimal serta mendukung seluruh fungsi dan fitur yang dirancang, seperti autentikasi biometrik, kontrol pintu, notifikasi otomatis, dan *monitoring* melalui *dashboard* web.

Spesifikasi dibagi menjadi dua bagian, yaitu perangkat keras dan perangkat lunak.

##### 1. Spesifikasi Perangkat Keras

Berikut perangkat keras yang digunakan untuk mengakses sistem:

Tabel 5. 1 Spesifikasi Perangkat Keras

Processor	AMD Ryzen 3250
RAM	8 GB
SSD	512 GB
Microcontroller	NodeMCU ESP32
Sensor <i>Fingerprint</i>	AS068
Kamera CCTV	Kamera IP 360°
Modul	Relay 2 Channel, Solenoid Door Lock
Layar	LCD 16x2
Power Supply	12V 2A

## 2. Spesifikasi Perangkat Lunak

Berikut adalah perangkat lunak yang digunakan dalam pengaksesan sistem ini adalah sebagai berikut:

Tabel 5. 2 Spesifikasi Perangkat Lunak

Sistem Operasi	Windows 10
Database	MySQL
Bahasa Pemrograman	Python, Arduino IDE, VS Code
Framework	Flask, Tailwind CSS

### 5.1.4 Instalasi Sistem

Instalasi sistem memuat penjelasan mengenai tahapan yang harus dilakukan untuk menyiapkan lingkungan pengembangan dan menjalankan sistem *monitoring* keamanan rumah. Proses instalasi mencakup instalasi aplikasi pendukung seperti Arduino IDE, Laragon (yang mencakup MySQL dan interpreter Python), serta *Visual Studio Code* sebagai *editor* kode. bagian ini juga menjelaskan instalasi *database* untuk kebutuhan penyimpanan data sistem.

#### 1. Instalasi Aplikasi

##### 1) Arduino IDE

Arduino IDE digunakan untuk memprogram serta mengunggah kode ke mikrokontroller ESP32. Langkah instalasinya sebagai berikut:

- a Buka web browser dan kunjungi situs resmi Arduino IDE:  
<https://www.arduino.cc/>
- b Jalankan installer Arduino IDE yang telah diunduh
- c Klik Next untuk melanjutkan proses instalasi.
- d Pilih komponen yang ingin diinstal, biarkan default, lalu klik Next.
- e Tentukan lokasi instalasi, lalu klik Install.
- f Tunggu hingga proses instalasi selesai, kemudian klik Finish.
- g Buka Arduino IDE, masuk ke Preferences >Additional Board URLs.
- h Tambahkan URL boardESP32: [https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\\_esp32\\_index.json](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json).

- i Buka Boards Manager melalui Tools > Board > Boards Manager, cari dan instal ESP32 by Espressif Systems.
- j Setelah instalasi board selesai, Arduino IDE siap digunakan untuk pemrograman ESP32.

## 2) Instalasi Laragon

Laragon digunakan sebagai server lokal yang mencakup Apache, MySQL, dan interpreter Python untuk kebutuhan backend sistem. Berikut adalah Langkah instalasinya:

- a. Buka web browser dan kunjungi situs resmi Laragon: <https://laragon.org>.
- b. Jalankan file installer laragon yang telah diunduh.
- c. Klik next untuk memulai proses instalasi.
- d. Pada pilihan komponen, biarkan saja pada pengaturan default atau disesuaikan dengan kebutuhan, kemudian klik install
- e. Tunggu proses instalasi selesai, kemudian klik finish.
- f. Setelah Laragon terbuka, klik tombol Start All untuk menjalankan Apache dan MySQL.
- g. Jika menggunakan backend berbasis Python, pastikan interpreter Python aktif melalui menu Menu > Terminal > python.
- h. Laragon siap untuk digunakan.

## 3) Instalasi *Visual Studio Code*

*Visual Studio Code* digunakan untuk mengembangkan dan mengimplementasikan program dengan berbagai bahasa pemrograman. Tahapan instalasi *Visual Studio Code* dijelaskan sebagai berikut.:

- a. Buka web browser dan kunjungi situs resmi Visual Studio Code: <https://code.visualstudio.com/>
- b. Lalu pilih *Download* for Windows untuk mengunduh installer *Visual Studio Code* versi Windows.
- c. Setelah unduhan selesai, cari file instalasi (biasanya bernama "VSCODESetup.exe") kemudian jalankan.
- d. Ikuti instruksi pada layar untuk menyelesaikan setiap langkah instalasi.

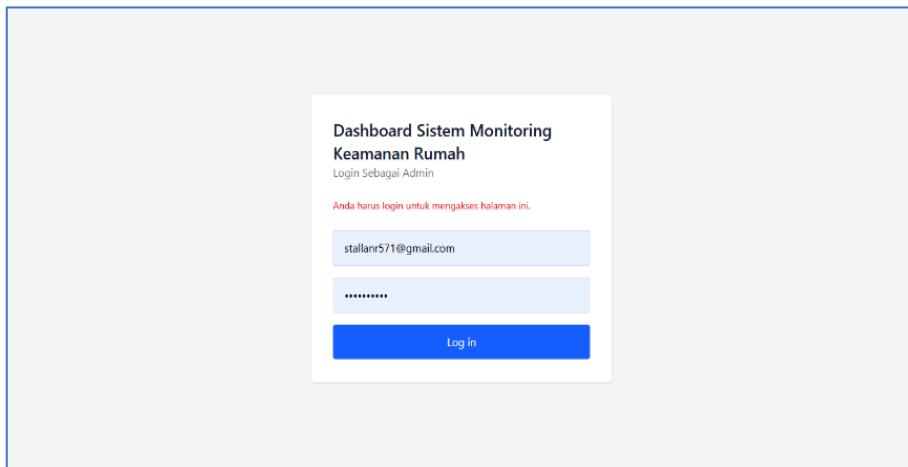
- e. Setelah selesai, buka VS Code dan instal ekstensi pendukung seperti Python, Flask Snippets.
  - f. Setelah instalasi selesai, *Visual Studio Code* siap digunakan.
2. Instal *Database*
- Database* yang digunakan adalah MySQL dan telah terintegrasi dalam Laragon. Untuk menginstal dan mengelola *database*, berikut langkahnya:
- a. Buka Laragon, aktifkan Apache dan MySQL.
  - b. Akses <http://localhost/phpmyAdmin> melalui *browser*.

### 5.1.5 Menjalankan Sistem

Pada bagian ini akan dijelaskan mengenai bagaimana langkah-langkah menjalankan sistem *monitoring* keamanan rumah.

#### 1. Halaman *Login*

Sebelum mengakses sistem *monitoring* keamanan rumah, *admin* melakukan *Login* menggunakan *email* dan *password* yang telah terdaftar.



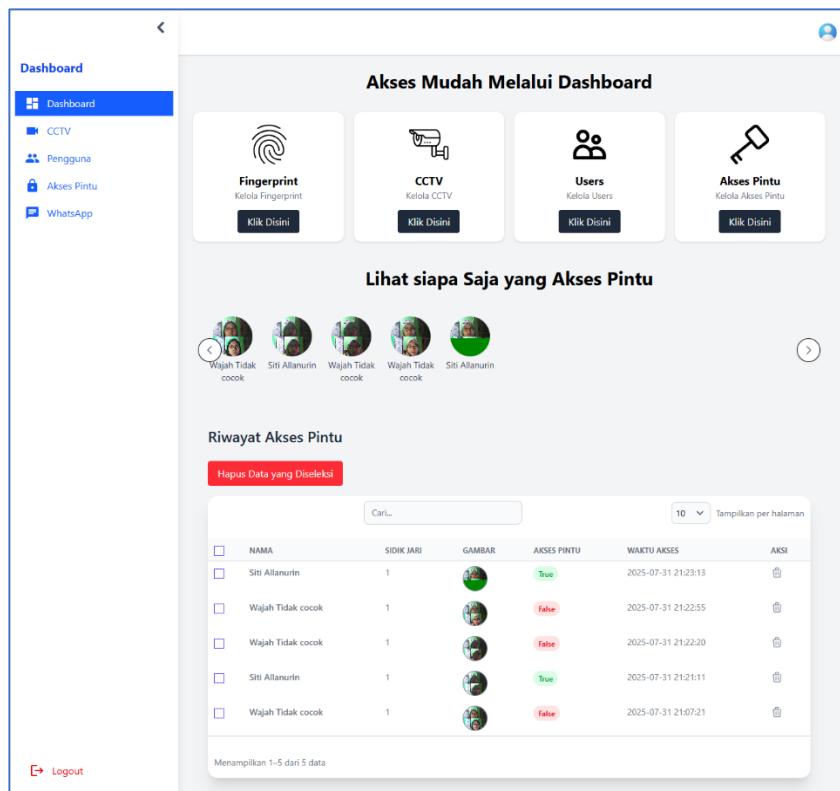
Gambar 5. 1 Halaman *Login*

Pada gambar di atas, ditampilkan antarmuka halaman *login* sistem *monitoring* keamanan rumah. Sistem ini digunakan oleh *admin* dan sebelum dapat menggunakan fitur pada sistem, *admin* harus melakukan *login* terlebih dahulu dengan memasukkan *email* dan *password* terdaftar, kemudian menekan

tombol *Masuk*. Proses ini memastikan bahwa hanya *admin* saja yang memiliki identitas login terdaftar untuk mengakses dan mengelola seluruh fitur sistem.

## 2. Halaman Dashboard

Pada halaman dashboard, admin dapat melakukan semua kontrol CCTV, dan memantau semua pergerakan yang tertangkap oleh kamera.

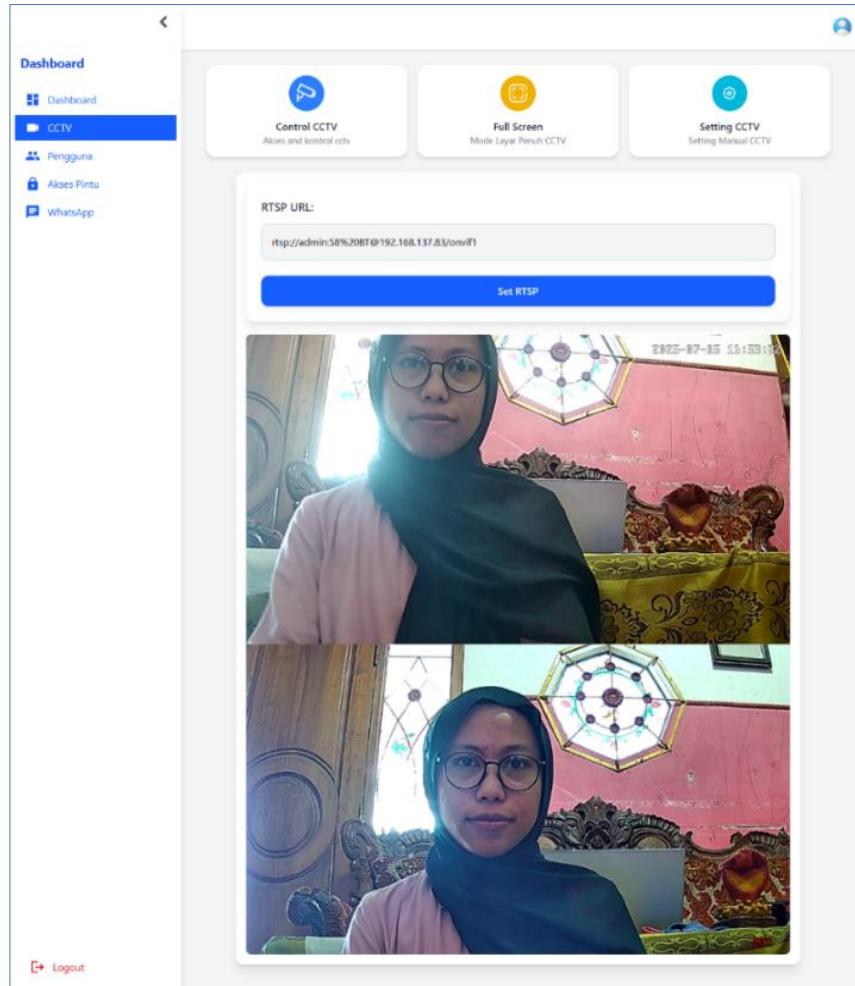


Gambar 5. 2 Halaman Dashboard

Halaman Dashboard disini merupakan tampilan utama yang menjadi pusat kontrol seluruh fitur pada sistem monitoring keamanan rumah. Melalui halaman ini, admin dapat dengan mudah mengakses berbagai menu. Penyajian pintasan (shortcut) untuk mempermudah navigasi, mempersingkat waktu operasional, serta memberikan kemudahan dalam memantau aktivitas keamanan rumah secara *real-time* melalui antarmuka yang terstruktur dan informatif.

## 3. Halaman Menu CCTV

Pada halaman menu CCTV, admin dapat melakukan semua kontrol CCTV, dan memantau semua pergerakan yang tertangkap oleh kamera.

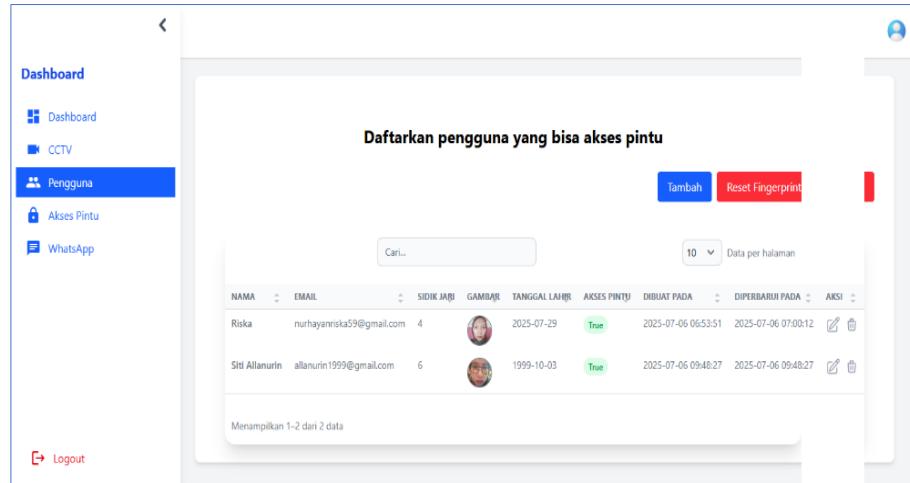


Gambar 5. 3 Halaman *Monitoring* CCTV

Gambar di atas menunjukkan tampilan halaman *monitoring* pada menu CCTV pada *dashboard* sistem keamanan rumah. Pada halaman ini *admin* dapat mengakses menu kontrol CCTV, mode layar penuh, dan pengaturan CCTV melalui tombol navigasi. Terdapat kolom input RTSP URL untuk mengatur link *streaming* kamera. Di bawahnya, hasil tangkapan kamera CCTV ditampilkan *real-time* sehingga *admin* dapat memantau kondisi rumah langsung melalui *dashboard* web. Melalui halaman ini, pengguna dapat mengontrol aktivitas kamera CCTV, baik melalui *dashboard* atau dengan aplikasi bawaan dari CCTV, yaitu V380 Pro.

#### 4. Halaman Pengguna

Pada halaman pengguna, ditampilkan data diri *user* yang memiliki akses membuka pintu. Halaman ini juga tempat menambahkan *user* baru yang diizinkan membuka pintu melalui autentikasi.



Gambar 5. 4 Halaman Pengguna

Gambar di atas menampilkan halaman pengguna. Pada halaman ini *admin* dapat melihat daftar pengguna yang memiliki akses untuk membuka pintu, lengkap dengan informasi seperti nama, email, ID sidik jari, foto wajah, tanggal lahir, status akses pintu, serta waktu pembuatan dan pembaruan data. Terdapat juga tombol *Tambah* untuk mendaftarkan *user* baru dan tombol *Reset Fingerprint* untuk mengatur ulang data sidik jari. Fitur ini memudahkan pengguna dalam mengelola dan memperbarui data akses pengguna secara cepat dan terorganisir.

##### 5. Halaman Tambah Pengguna

Halaman ini terdapat pada halaman pengguna, dan *admin* dapat menambahkan data diri dan registrasi sidik jari dan wajah untuk dapat akses pintu

Gambar 5. 5 Halaman Tambah Pengguna

Pada halaman tambah pengguna ini *admin* dapat menambahkan pengguna baru untuk ijin akses pintu melalui registrasi wajah dan *Fingerprint* untuk autentikasi, berikut merupakan tampilan proses registrasi data diri dan data untuk autentikasi fingerprint dan face recognition autentikasi.



Gambar 5. 6 Proses Registrasi *Fingerprint*



Gambar 5. 7 Perintah Registrasi *Fingerprint* dari LCD

Pada sistem smart door lock yang dibangun, autentikasi sidik jari menggunakan sensor AS608, yang bekerja berdasarkan pencocokan ciri khas sidik jari (*minutiae*). Dibawah ini adalah alur kerja pengenalan sidik jari dalam sistem:

1. Akuisisi Sidik Jari (*Fingerprint Capture*)

Pengguna menempelkan jari pada sensor AS608. Sensor akan mengambil citra sidik jari dengan menggunakan sensor optik. Citra sidik jari kemudian diproses dan disimpan sementara dalam memori modul sensor.

2. *Pra-Pemrosesan Citra (Image Preprocessing)*

Setelah citra diperoleh, dilakukan serangkaian proses untuk meningkatkan kualitas gambar, seperti:

- a) Konversi ke *grayscale*.
- b) *Enhancement* untuk menajamkan garis-garis pola sidik jari.
- c) *Noise reduction* untuk menghilangkan gangguan pada gambar.

3. Ekstraksi Ciri Khas (*Feature Extraction*)

Algoritma akan mencari dan mengekstrak titik-titik khas (minutiae) dari pola sidik jari, seperti:

- a) *Bifurkasi* (percabangan garis)
- b) *Ridge endings* (ujung garis)

Data minutiae ini akan diubah menjadi representasi digital (template), bukan disimpan sebagai gambar asli, sehingga lebih aman secara privasi.

- a. Pendaftaran Sidik Jari (*Enrollment*)

Pada saat registrasi, template sidik jari disimpan ke dalam database internal sensor AS608. Setiap template dikaitkan dengan ID pengguna.

- b. Pencocokan Sidik Jari (*Matching*)

Ketika pengguna menempelkan jari kembali (misalnya saat membuka pintu), sistem akan:

- b) Mengambil citra baru
- c) Mengekstrak *minutiae* kembali
- d) Membandingkan hasilnya dengan template yang tersimpan menggunakan metode *minutiae-based matching*

Pencocokan ini dilakukan oleh sensor AS608 secara internal, dan hasilnya berupa nilai kesesuaian (*match score*) atau status verifikasi (berhasil/gagal).

#### 4. Keputusan Akses

Jika hasil pencocokan sesuai dengan salah satu template yang terdaftar, sistem memberikan akses untuk menuju scan wajah. Jika tidak cocok, akses ditolak dan sistem dapat memberikan respons notifikasi ke whatsapp.

Sensor AS608 memiliki kelebihan berupa kemampuan menyimpan hingga 1000 sidik jari (tergantung konfigurasi) dan melakukan pencocokan langsung tanpa harus memproses gambar di mikrokontroler.

#### 6. Halaman Akses Pintu

Halaman Akses Pintu menampilkan riwayat autentikasi *user* yang mengakses pintu. Data ditampilkan dalam tabel, dan fitur pencarian, buka tutup pintu, serta penghapusan data disediakan untuk mendukung kemudahan *monitoring* dan manajemen sistem keamanan.

NAMA	SIDIK JARI	GAMBAR	AKSES PINTU	WAKTU AKSES	AKSI
Wajah Tidak cocok	6		False	2025-07-06 16:53:06	
nr	2		True	2025-07-06 16:37:28	
nr	2		True	2025-07-06 14:59:45	
nr	2		True	2025-07-06 14:50:54	
Tidak diketahui	0		False	2025-07-06 14:50:22	
nr	2		True	2025-07-06 14:49:31	
Wajah Tidak cocok	2		False	2025-07-06 14:49:01	
adam	1		True	2025-07-06 14:48:01	
Tidak diketahui	0		False	2025-07-06 14:47:21	
adam	1		True	2025-07-06 14:46:55	

Gambar 5. 8 Halaman Akses Pintu

Akses pintu dapat dilakukan dengan beberapa cara yaitu dengan autentikasi *Fingerprint* dan wajah melalui perangkat IoT, atau dengan tombol otomatis di *dashboard* web, dibawah ini merupakan proses autentikasi *Fingerprint* dan wajah pada saat mengakses pintu.



Gambar 5. 9 Proses Autentikasi

Dalam sistem ini, penerapan machine learning untuk fitur face recognition mengikuti pendekatan supervised learning dengan algoritma Local Binary Patterns Histogram (LBPH). Berikut ini adalah alur kerjanya:

1. Pengumpulan Data (Data Acquisition)

Pada tahap ini, sistem melakukan *registrasi wajah* pengguna. Kamera mengambil citra wajah yang kemudian disimpan ke dalam database. Setiap citra wajah diberi label sesuai dengan identitas pengguna.

2. Pra-Pemrosesan (Preprocessing)

Citra wajah yang diperoleh dari kamera akan diproses agar lebih optimal untuk pelatihan dan pengenalan. Tahapan pra-pemrosesan meliputi:

- a) Mengubah citra ke dalam format grayscale.
- b) Melakukan resizing agar semua citra memiliki dimensi yang seragam.
- c) Melakukan cropping area wajah untuk menghilangkan latar belakang yang tidak relevan.
- d) Noise removal untuk memperjelas fitur wajah yang penting.

3. Setelah melalui pra-pemrosesan, sistem akan mengekstrak ciri khas dari wajah menggunakan algoritma LBPH. Algoritma ini bekerja dengan cara

menganalisis pola tekstur lokal dari gambar wajah dan mengubahnya menjadi histogram deskriptif.

#### 4. Pelatihan Model (Training)

Dataset wajah yang telah diekstrak fiturnya digunakan untuk melatih model machine learning. Karena data telah dilabeli (misalnya dengan nama pengguna), maka proses pelatihan bersifat supervised.

#### 5. Pengenalan Real-Time (*Recognition*)

Ketika pengguna mencoba membuka pintu, kamera akan menangkap wajah pengguna. Sistem kemudian melakukan proses yang sama seperti sebelumnya (pra-pemrosesan → ekstraksi fitur), dan hasilnya dibandingkan dengan data pada model yang telah dilatih untuk menentukan apakah wajah tersebut cocok dengan salah satu entri di database.

#### 6. Keputusan dan Tindakan

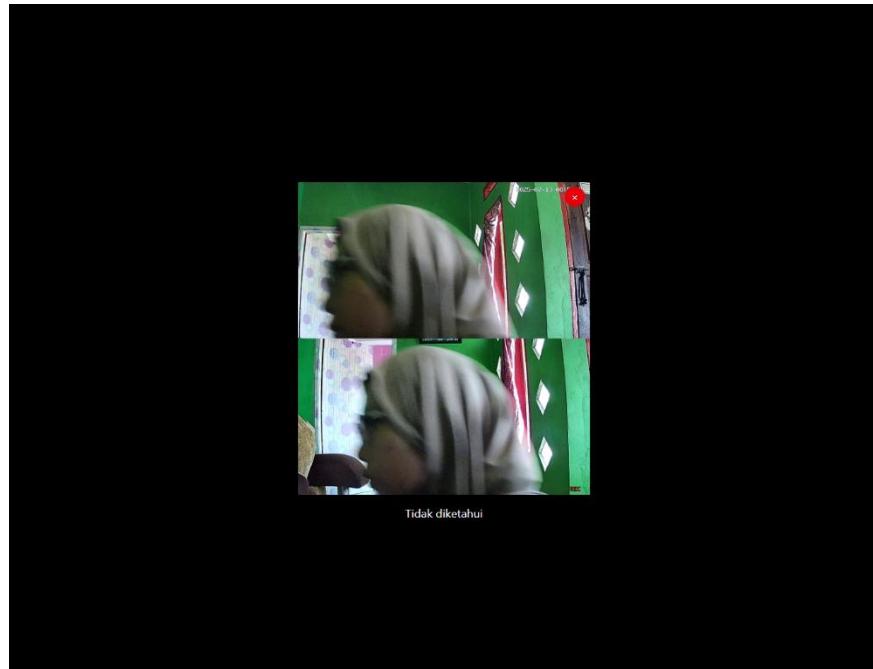
Jika wajah dikenali dan cocok dengan data pengguna yang valid, maka sistem akan memberikan izin akses (membuka pintu). Jika tidak dikenali, akses ditolak dan sistem dapat memberikan notifikasi (misalnya via WhatsApp).

#### 7. Evaluasi Kinerja (*Performance Evaluation*)

Untuk mengukur seberapa baik sistem bekerja, dilakukan pengujian menggunakan metrik seperti:

- a) Accuracy – Persentase pengenalan yang benar.
- b) False Acceptance Rate (FAR) – Kemungkinan wajah yang tidak dikenal diterima sebagai valid.
- c) False Rejection Rate (FRR) – Kemungkinan wajah yang valid ditolak oleh sistem.

Dibawah ini merupakan foto dimana saat akses pintu menggunakan autentifikasi fingerprint dan face recognition dan akses gagal. Dengan status False pada tabel sistem *monitoring* keamanan rumah pada menu halaman akses pintu



Gambar 4. 53 Tampilan Foto dengan Status False

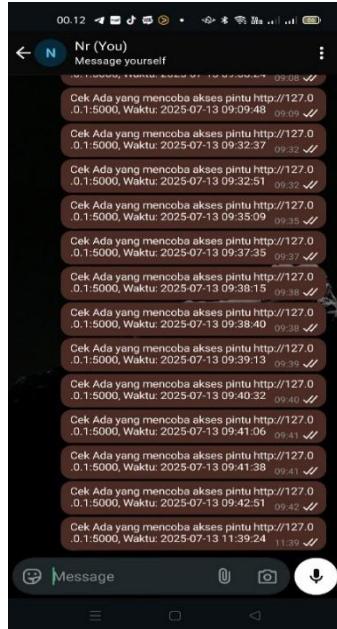
#### 7. Halaman *WhtasApp*

Halaman ini untuk memasukkan nomor *WhatsApp* yang akan menerima notifikasi dari kegagalan akses pintu. Nomor yang ditambahkan pada halaman ini akan dapat menerima notifikasi peringatan akses gagal.

NOMOR WHATSAPP	DIBUAT PADA	DIUBAH PADA	AKSI
6282124847221	2025-06-21 04:09:35	2025-06-21 04:09:35	

Gambar 5. 10 Halaman *WhatsApp*

Penggunaan whtasapp pada sistem ini yaitu untuk mengirimkan notifikasi otomatis pada saat terdapat kegagalan pada akses pintu, notifikasi ini merupakan salah satu fitur utama dalam sistem *monitoring* keamanan rumah.



Gambar 4. 54 Tampilan Notifikasi WhtasApp

Gambar diatas merupakan isi pesan dari notifikasi kegagalan akses pintu dengan autentikasi, pesan berisi keterangan bahwa terdeteksi ada yang mencoba akses pintu disertai link dari *dashboard* web sistem *monitoring* keamanan rumah.

## 5.2 Pegujian *Prototipe*

Setelah tahap perancangan dan pembuatan prototipe sistem selesai, dilakukan pengujian menggunakan metode *black-box testing* untuk memastikan bahwa fitur-fitur utama berfungsi sesuai dengan spesifikasi. Pengujian dilakukan pada alat mikrokontroller dan sistem *dashboard monitoring*, di sajikan dalam dua tabel. Hasil pengujian sistem ditampilkan pada tabel berikut.

### 1. Pengujian Alat *Mikrokontroller*

Pada bagian ini menjelaskan hasil pengujian pada perangkat mikrokontroler yang berfungsi sebagai pengendali utama pada sistem monitoring keamanan rumah. Pengujian dilakukan untuk memastikan bahwa seluruh komponen yang terhubung dengan *mikrokontroler* dapat bekerja sesuai dengan rancangan.

Tabel 5. 3 Pengujian Alat *Mikrokontroller*

No.	Item	Skenario	Hasil yang diharapkan	Hasil Pengujian	Keterangan
1.	Sensor <i>Fingerprint</i>	Tempelkan jari pada sensor <i>Fingerprint</i>	Sensor merespon dengan notifikasi di LCD.		Valid
2.	Kamera CCTV	Sambungkan CCTV dengan listrik	CCTV aktif dan memberikan respon gerakan		Valid
2.	CCTV	Hadapkan wajah ke arah kamera	Sistem mengenali wajah		Valid
3.	LCD	Lakukan aktifitas akses pintu	Menampilkan hasil dilayar dari aktifitas akses pintu.		Valid
4.	Buzzer	Lakukan akses pintu, percobaan berhasil dan gagal	Buzzer bunyi 1x jika berhasil dan 2 kali jika gagal.		Valid
5.	Tombol akses	Buka pintu dengan tombol.	Pintu akan membuka atau menutup		Valid

6.	Selenoid door lock	Memberi respon buka tutup ketika ada yang akses pintu.	Selenoid door lock membuka dan menutup.		Valid
7.	Relay	Uji autentikasi pada pintu.	Pintu terbuka/ tertutup melalui kendali relay		Valid

## 2. Pengujian *Dashboard Monitoring*

Bagian ini, hasil pengujian Dashboard Monitoring untuk memastikan fungsionalitas tampilan dan fitur dapat berjalan sesuai kebutuhan pengguna.

Tabel 5. 4 Pengujian *Dashboard Monitoring*

No	Item	Skenario	Hasil yang diharapkan	Hasil Pengujian
1.	<i>Login Admin</i>	Masukkan email dan password dan klik <i>Login</i>	<i>Dashboard</i> akan terbuka jika data benar	Valid
2.	Registrasi <i>User</i> Baru	klik tombol Tambah di halaman <i>user</i> Isi data diri, foto, rekam sidik jari, lalu tekan <i>Save</i> .	<i>User</i> baru berhasil ditambahkan dan disimpan di tabel <i>user</i> .	Valid
3.	Registrasi <i>Fingerprint</i>	Tempelkan jari pada sensor <i>Fingerprint</i> sebanyak dua kali	<i>Fingerprint</i> berhasil registrasi dengan munculnya ID <i>Fingerprint</i> di dalam form registrasi <i>user</i> baru.	Valid
4.	<i>Monitoring CCTV</i>	Memilih menu CCTV	Tampilan video live CCTV muncul	Valid
5.	Scan <i>Fingerprint</i>	Menempelkan jari yang telah terdaftar pada sensor <i>Fingerprint</i> .	Jika berhasil LCD akan menampilkan perintah scan wajah untuk autentikasi selanjutnya.	Valid

6.	Scan Wajah	<i>User</i> menghadap pada kamera agar wajahnya terdeteksi.	Scan wajah berhasil dan pintu akan terbuka.	Valid
	Notifikasi LCD	<i>User</i> membuka pintu dengan autentikasi, tombol manual, ataupun melalui <i>dashboard</i> .	LCD akan menampilkan notifikasi di layar dari setiap hasil yang didapat dari proses akses pintu.	Valid
8.	Akses Pintu	Klik nukleus pintu atau tutup pintu di halaman akses pintu	Pintu terbuka dan terutup serta log tercatat pada tabel akses.	Valid
9.	Notifikasi WhatsApp	<i>Admin</i> menambahkan nomor WhatsApp dan melakukan akses pintu	Notifikasi terkirim ke nomor terdaftar jika terjadi akses gagal	Valid
10.	Log Aktivitas	Buka halaman riwayat akses	Log data tampil lengkap sesuai dengan data apa yang ada saat register.	Valid
11.	Pencarian Data	Gunakan fitur search di tabel <i>user</i> / akses	Data yang dicari muncul sesuai input	Valid
12.	Edit/Hapus <i>User</i>	Klik ikon pensil / tempat sampah	Data berhasil diperbarui atau dihapus	Valid
13.	Logout	Klik tombol “Logout”	Kembali ke halaman <i>Login</i>	Valid

### 3. Pengujian Sistem Monitoring Keamanan Rumah Keseluruhan

Berdasarkan hasil pengujian yang telah dilakukan terhadap seluruh komponen sistem, dapat disimpulkan bahwa mikrokontroler ESP32, *relay*, *solenoid door lock*, *buzzer*, *push button*, serta LCD 16x2 berfungsi sesuai dengan rancangan. ESP32 mampu mengolah perintah autentikasi dan mengirimkan sinyal kendali dengan baik, *relay* dan *solenoid door lock* merespons secara tepat saat akses diajukan maupun ditolak, *buzzer* aktif memberikan peringatan ketika autentikasi gagal, serta *push button* dan LCD dapat mendukung proses interaksi pengguna dengan sistem. Dengan demikian, keseluruhan perangkat keras terbukti bekerja secara integratif dalam mendukung prototipe dari sistem monitoring keamanan rumah.

1. Pengujian *Fingerprint*

Tabel 5. 5 Pengujian *Fingerprint*

No	Jari yang digunakan	Jumlah percobaan	Berhasil	Gagal	Persentase keberhasilan (%)
1.	Ibu Jari	10	10	0	100%
2.	Jari Telunjuk	10	10	0	100%
3.	Jari Tengah	10	9	1	90%
4.	Jari Manis	10	8	2	80%
5.	Jari Kelingking	10	8	2	80%

2. Pengujian *Face Recognition* Berdasarkan Jarak

Tabel 5. 6 Pengujian *Face Recognition* Berdasarkan Jarak

No	Jarak (cm)	Jumlah Percobaan	Berhasil	Gagal	Persentasi Keberhasilan (%)
1.	20	10	6	4	60%
2.	30	10	9	1	90%
3.	50	10	5	5	50%
4.	70	10	1	9	10%
5.	100	10	0	10	0%

3. Pengujian *Face Recognition* Berdasarkan Kondisi Pengguna

Tabel 5. 7 Pengujian *Face Recognition* Berdasarkan Kondisi Pengguna

No	Kondisi Pengguna	Jumlah Percobaan	Berhasil	Gagal	Persentasi Keberhasilan (%)
1.	Tanpa Jilbab	10	8	2	80%
2.	Jilbab Hitam Polos	10	1	9	10%
3.	Jilbab Warna Terang	10	9	1	90%
4.	Jilbab Bermotif	10	5	5	50%

5.	Menggunakan Masker	10	0	10	0%
----	--------------------	----	---	----	----

#### 4. Pengujian *Face Recognition* Berdasarkan Cahaya

Tabel 5. 8 Pengujian *Face Recognition* Berdasarkan Cahaya

No	Kondisi Cahaya	Jumlah Percobaan	Berhasil	Gagal
1.	Terang Indoor	10	9	1
2.	Sedang (alami)	10	9	1
3.	Rendah (low-light))	10	5	5
4.	Backlight (melawan cahaya)	10	4	6
5.	Cahaya Tidak Merata	10	3	7

#### 5. Pengujian Notifikasi WhatsApp

Tabel 5. 9 Pengujian Notifikasi WhatsApp

No	Skenario	Jumlah Percobaan	Terkirim	Tidak Terkirim	Rata-rata Delay (Detik)
1.	Autentikasi Berhasil	10	0	10	1
2.	Autentikasi Gagal	10	10	0	1
3.	Internet Tidak Stabil (kegagalan)	10	10	0	6

#### 6. Pengujian Penyimpanan Data ke *Dashboard Web*

Tabel 5. 10 Pengujian Penyimpanan Data ke *Dashboard Web*

No	Skenario	Jumlah Percobaan	Data Masuk	Data Hilang	Persentase Keberhasilan (%)
1.	Autentikasi Berhasil	10	10	0	100%
2.	Autentikasi Gagal	10	10	0	100%
3.	Internet Tidak Stabil (kegagalan)	10	10	0	100%

7. Pengujian ESP32 (komunikasi dan kontrol)

Tabel 5. 11 Pengujian ESP32 (Komunikasi dan Kontrol)

No	Skenario	Aksi Uji	Indikator Keberhasilan	Hasil Uji	Status
1.	Koneksi Wifi	Hubungkan ESP32 ke SSID lokal	ESP32 terhubung (IP muncul di serial)	Terhubung (IP: 192.168.1.8)	Berhasil
2.	Kirim Log ke Dashboard	Kirim Payload Autentikasi	Data Tampil di tabel "Door Access"	10/10 kali	Berhasil
3.	Terima Perintah Buka Pintu	Trigger dari Web > API > ESP32	ESP32 set pin relay = HIGH	HIGH Terbaca dan Relay Aktif	Berhasil

8. Pengujian *Relay + Selenoid Door Lock*

Tabel 5. 12 Pengujian *Relay + Selenoid Door Lock*

No	Kondisi	Perintah Sistem	Hasil Yang Diharapkan	Hasil Uji	Status
1.	Autentikasi Berhasil	Relay ON (HIGH)	Selenoid Aktif > Pintu Terbuka	Terbuka dalam $\pm 1$ s	Berhasil
2.	Autentikasi Gagal	Relay OFF (LOW)	Selenoid NonAktif > Pintu Terkunci	Tetap Terkunci	Berhasil
3.	Manual : Open/ Close	Tombol buka/ Tutup ditekan	Relay ON, Pintu Terbuka, dan Relay OFF, Pintu Tertutup	Membuka dan Menutup Normal	Berhasil

9. Pengujian *Buzzer*

Tabel 5. 13 Pengujian *Buzzer*

No	Skenario	Trigger	Respon Diharapkan	Hasil Uji	Durasi Bunyi (Detik)
----	----------	---------	-------------------	-----------	----------------------

1.	Autentikasi Berhasil	Valid Access	Bunyi Panjang 1 kali	Sesuai	1,0
2.	Autentikasi Gagal	Fingerprint & Face Recognition Gagal	Buzzer Bunyi 2 kali	Sesuai	1,5
3.	Autentikasi Gagal	Only Fingerprint	Buzzer Bunyi Keras 1 kali	Sesuai	1,0
3.	Akses Ditolak (ID tidak terdaftar)	Input ID Unknown	Buzzer Bunyi Keras 1 kali	Sesuai	1,0

#### 10. Pengujian *Push Button* (buka/tutup manual)

Tabel 5. 14 Pengujian *Push Button* (buka/tutup manual)

No	Fungsi	Aksi Uji	Respon Diharapkan	Hasil Uji	Debounce OK
1.	Buka Pintu	Tekan 1x	Relay ON > Selenoid Aktif	Aktif Dalam 1,5 Detik	Ya
2.	Tutup Pintu	Tekan 1x	Relay OFF > Selenoid Nonaktif	Terkunci Kembali	Ya

#### 11. Pengujian ESP32 (komunikasi dan kontrol)

Tabel 5. 15 Pengujian ESP32 (Komunikasi dan Kontrol)

No	Skenario	Pesan Diharapkan	Waktu Muncul (Detik)	Keterangan
1.	Pintu Terbuka	”Pintu Terbuka”	0,5	Aktif Dalam 1,5 Detik
2.	Pintu Tertutup	”Pintu Tertutup”	0,5	Terkunci Kembali
3.	Akses Gagal	“Akses Gagal”	0,5	Sinkron dengan Buzzer

## **BAB VI**

### **KESIMPULAN**

#### **6.1 Kesimpulan**

Berdasarkan dari hasil penelitian yang telah dilakukan oleh penulis melalui beberapa tahapan pada bab-bab sebelumnya, maka penulis dapat menyimpulkan bahwa:

1. Sistem monitoring keamanan rumah berbasis web dengan autentikasi *fingerprint* dan *face recognition* berbasis *deep learning* berhasil dibuat dalam bentuk prototipe, dilengkapi perangkat keras pendukung untuk meningkatkan keamanan rumah secara otomatis dan real-time.
2. Integrasi mikrokontroler IoT, sensor Fingerprint, dan kamera CCTV berfungsi sesuai rancangan serta terhubung dengan dashboard berbasis web yang memudahkan pemilik rumah memantau kondisi rumah dari jarak jauh.
3. Fitur notifikasi melalui aplikasi WhatsApp berjalan efektif untuk memberikan peringatan apabila terdeteksi percobaan akses tidak sah atau kondisi mencurigakan, sehingga pemilik rumah dapat segera merespons.
4. Sistem yang dikembangkan dapat membantu meminimalisir risiko tindak kejahatan dengan proses autentikasi biometrik ganda yang valid serta dukungan notifikasi instan.
5. Hasil pengujian menunjukkan bahwa seluruh perangkat keras (ESP32, *relay*, *solenoid door lock*, *fingerprint* sensor, CCTV, *buzzer*, *push button*, dan LCD) berfungsi integratif dan responsif, sementara dashboard berhasil menjalankan fitur utama seperti registrasi pengguna, autentikasi biometrik, *monitoring* CCTV, log aktivitas, serta notifikasi *real-time* dengan tingkat keberhasilan tinggi. Sensor *fingerprint* memiliki akurasi >90%, dan *face recognition* optimal pada jarak 30 cm dengan keberhasilan 90% meski menurun pada penggunaan masker dan kondisi pencahayaan yang rendah. Notifikasi WhatsApp dan pencatatan log berhasil 100% dengan *delay* ±1 detik. Secara umum sistem telah memenuhi spesifikasi fungsional dan layak dijadikan solusi keamanan rumah yang lebih *modern*.

## 6.2 Saran

Berdasarkan hasil penelitian sistem monitoring keamanan rumah menggunakan *Fingerprint* dan *Face Recognition* berbasis *, machine learning*, aplikasi ini memiliki kekurangan yang dapat di perbaiki atau dikembangkan untuk hasil lebih baik. Penulis memberikan beberapa saran pengembangan, antara lain:

1. Penggunaan kamera dengan resolusi lebih tinggi dan sudut pandang lebih luas direkomendasikan agar hasil *Face Recognition* semakin akurat, terutama dalam kondisi pencahayaan rendah.
2. Fitur notifikasi dapat diperluas, misalnya dengan integrasi ke platform lain seperti Telegram atau aplikasi mobile, agar pengguna memiliki opsi alternatif jika WhatsApp mengalami gangguan.
3. Diperlukan uji coba sistem dalam kondisi lingkungan yang lebih bervariasi (siang, malam, cuaca ekstrem) untuk memastikan stabilitas koneksi IoT dan akurasi sensor tetap terjaga.
4. Disarankan agar di Desa Wargaluyu dilakukan sosialisasi kepada warga mengenai penggunaan sistem monitoring keamanan rumah ini, khususnya dalam hal autentikasi biometrik dan *monitoring* melalui *website*. Dengan adanya pelatihan sederhana dan pendampingan teknis, masyarakat dapat memanfaatkan sistem secara optimal sekaligus meningkatkan kesadaran akan pentingnya keamanan rumah berbasis teknologi IoT.
5. Pada penelitian ini, dashboard web belum diimplementasikan pada *hosting* ke server publik karena keterbatasan infrastruktur jaringan di lokasi penelitian. Untuk peneliti selanjutnya disarankan dapat mengembangkan sistem ini dengan melakukan hosting pada server publik serta menambahkan fitur pendukung, seperti autentikasi multi-faktor, sebagai lapisan keamanan tambahan. Selain itu, evaluasi kinerja sistem sebaiknya dilakukan pada berbagai lingkungan berbeda, misalnya di kawasan perkotaan atau fasilitas publik, agar diperoleh gambaran performa yang lebih komprehensif. Mengingat sistem ini sangat bergantung pada koneksi internet, penambahan cadangan daya (UPS) juga disarankan agar perangkat tetap berfungsi meskipun terjadi pemadaman listrik.

## DAFTAR PUSTAKA

- Abilovani, Z. B., Yahya, W., & Bakhtiar, F. A. (2018). *Implementasi Protokol MQTT Untuk Sistem Monitoring Perangkat IoT* (Vol. 2, Issue 12). <http://j-ptiik.ub.ac.id>
- Abraham Salihi, I., Chanda Pelangi, K., & Mokoginta, N. (2022). *Sistem Pengontrol Pintu Otomatis Ruangan Fakultas Ilmu Komputer Berbasis IoT.* 1(1).
- Abroruddin, M., Ramadhan, F., & Roihan, A. (2020). Perancangan Sistem Pengaman Pintu Rumah menggunakan Sidik Jari berbasis Arduino. *Jurnal Teknologi Informasi Indonesia (JTII)*, 5(1), 18–23. <https://doi.org/10.30869/jtii.v5i1.520>
- Ade Mubarok, Ivan Sofyan, Ali Akbar Rismayadi, & Ina Naiyah. (2020). *Sistem Keamanan Rumah Menggunakan RFID, Sensor PIR dan Modul GSM Berbasis Mikrokontroler.*
- Ahmad Roihan, Po Abas Sunarya, & Ageng Setiani Rafika. (2019). IJCIT (Indonesian Journal on Computer and Information Technology) Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper. In *IJCIT (Indonesian Journal on Computer and Information Technology)* (Vol. 5, Issue 1).
- Ahmadiyah, A. S., Sarno, R., Hidayati, S. C., Anggraini, R. N. E., Sungkono, K. R., & Munif, A. (2024). Pelatihan Desain Antarmuka Mobile Application dengan Figma untuk Meningkatkan Kompetensi Guru MGMP TIK Surabaya. *Sewagati*, 8(4), 1931–1942. <https://doi.org/10.12962/j26139960.v8i4.1216>
- Andri Nugraha Ramdhon, & Fadly Febriya. (2021). Penerapan Face Recognition Pada Sistem Presensi. *Journal of Applied Computer Science and Technology*, 2(1), 12–17. <https://doi.org/10.52158/jacost.v2i1.121>
- Anggya N D, & Soetarmono, S. K. (2012). *Identifikasi Sidik Jari Dengan Menggunakan Struktur Minutia.*

- Ardiansah, A., Nuraeni, M., & Ridwang<sup>3</sup>, A. (2024). *Rancang Bangun Akses Kunci Pintu Otomatis menggunakan Fingerprint Berbasis Internet of Things (IoT)*.
- Arifin, S., & Krisnadita, Y. (2017). Aplikasi Plugin Transfer Domain di PT Beon Intermedia. In *Jurnal Teknologi Informasi* ISSN (Vol. 8, Issue 1). www.namaanda.com
- Aulia Ramadini, D., Negeri Padang Jl Hamka, U., Tawar Barat, A., Padang, K., & Barat, S. (2025). Sistem Kunci Elektronik Pintu Kos Menggunakan IoT Berbasis E-Ktp. *Jurnal Informatika Dan Teknik Elektro Terapan*, 13(1), 2830–7062. <https://doi.org/10.23960/jitet.v13i1.6177>
- Bagoes Satria, M., & Ardiansyah, H. (2023). Analisis dan Perancangan Sistem Raport Digital Metode Waterfall. *Journal on Education*, 05(02), 5143–5151.
- Denta Widyapramana, M., Dewantoro, G., Handoko, dan, Kristen Satya Wacana Jl Diponegoro, U., & Tengah, J. (2021). *Perancangan Sistem Cerdas untuk Keamanan dan Pemantauan Pintu Rumah Berbasis IoT* (Vol. 4, Issue 1).
- Ding, B., Yao, F., Wu, Y., & He, Y. (2012). Improving flask implementation using hardware assisted in-VM isolation. *IFIP Advances in Information and Communication Technology*, 376 AICT, 115–125. [https://doi.org/10.1007/978-3-642-30436-1\\_10](https://doi.org/10.1007/978-3-642-30436-1_10)
- Dixit, N., Shrivastava, V., Pandey, A., & Sharma, E. R. (2024). International Journal Of Research Publication And Reviews Revolutionizing Web Design With Tailwind Css: A Comprehensive Exploration. In *International Journal of Research Publication and Reviews* (Issue 5). www.ijrpr.com
- Dwi Bima Sakti, R., Lestanti, S., Nur Budiman, S., Balitar Jl Majapahit No, I., Sananwetan, K., & Blitar, K. (2024). Perancangan Dashboard Monitoring Penjualan Pada Website Pateron.Id Menggunakan Framework Laravel Dan Vue Js. In *Jurnal Mahasiswa Teknik Informatika* (Vol. 8, Issue 2).
- Fajrin, R., & Yenni, Y. (2021). Rancang Bangun Alat Pengusir Hama Tanaman Menggunakan Arduino Dan Pengontrol Berbasis Arduino. *Jurnal Comasie*, 04(0).
- Fathoni. (2010). Unjuk Kerja Catu Daya 12 Volt 2a Dengan Pass Element Transistor Npn Dan Pnp. In *Jurnal* (Vol. 3, Issue 1).

- Giovanni Nathaniel, Casie Setianingsih, & Meta Kallista. (2024). *Penerapan Framework Flask sebagai API Dalam Pengembangan Website Prediksi Kebakaran Hutan dan Lahan di Indonesia*.
- Hadriansa, & Denis Prayogi. (2025). *Penerapan IoT pada Keamanan Lingkungan Berbasis Android*.
- Hasibuan, N. H., & Nurhaliza, Z. (2024). *Oktal: Jurnal Ilmu Komputer dan Science Metode Black Box Pada Pengujian Sistem Informasi Surat Keluar Masuk*. 3(6).
- Hilman Aziz, & Imam Suharjo. (2024). Pengembangan Sistem Keamanan Gerbang Rumah Smart Home Berbasis IoT dengan Metode RnD. *JEKIN - Jurnal Teknik Informatika*, 5, 13–23. <https://doi.org/10.58794/jekin.v5i1.839>
- Jodi, S., Siregar, M., Asmira, A., & Kusumawati, N. (2022). Prototype Sistem Keamanan Pintu Rumah Menggunakan Tag Card dan PIN Berbasis Arduino Uno. *SIMKOM*, 7(2), 82–91. <https://doi.org/10.51717/simkom.v7i2.83>
- Joevan Maulana Florian, Lintang Wahyu Aji Saputro, Muhammad Anugrah Putra, Muhammad Hanif Hilmi, & Rudi Susanto. (2024). *Rancangan Sistem Keamanan Pintu Rumah Joglo Menggunakan Fringerprint*.
- Kalsum Siregar, U., Arbaim Sitakar, T., Haramain, S., Nur Salamah Lubis, Z., Nadhirah, U., & Sains dan Teknologi, F. (2024). *Pengembangan database Management system menggunakan My SQL* (Vol. 1, Issue 1).
- Khairunnisa, G., Voutama Sistem Informasi, A., & Singaperbangsa Karawang Jalan Ronggo Waluyo Karawang, U. H. (2024). Penerapan Uml Dalam Perancangan Sistem Informasi Peminjaman Inventaris Berbasis Web Di Bem Fasilkom Unsika. In *Jurnal Mahasiswa Teknik Informatika* (Vol. 8, Issue 3).
- Kosasih, R., & Daomara, C. (2021). Pengenalan Wajah dengan Menggunakan Metode Local Binary Patterns Histograms (LBPH). *Jurnal Media Informatika Budidarma*, 5(4), 1258. <https://doi.org/10.30865/mib.v5i4.3171>
- Kusumah, R., & Izzatul Islam, H. (2023). Sistem Monitoring Suhu dan Kelembaban Berbasis Internet of Things (IoT) Pada Ruang Data Center. In *Journal of Applied Informatics and Computing (JAIC)* (Vol. 7, Issue 1). <http://jurnal.polibatam.ac.id/index.php/JAIC>
- Laurie. (2006). *Testing Overview and Black-Box Testing Techniques*.

- Lionar Putra, & Indah Fenriana. (2024). *Rancang Bangun Smart Home System Berbasis IoT Dengan Integrasi Sidik Jari (Fingerprint) Dan Otomasi Elektronik*.
- Mardhatillah, A., Binawidya Jl Soebrantas KM, K. H., & Baru, S. (2024). *Sistem Notifikasi Dan Kontrolling Smart Home Berbasis Internet of Things*. 9(1), 2024.
- Medapati, P. K., Tejo Murthy, P. H. S., & Sridhar, K. P. (2020). Lamstar: For IoT-based face recognition system to manage the safety factor in smart cities. *Transactions on Emerging Telecommunications Technologies*, 31(12). <https://doi.org/10.1002/ett.3843>
- Mindriawan, Z., Wayan, I., Arimbawa, A., Pasek, G., & Wijaya, S. (2018). *Implementasi Internet of Things Pada Sistem Monitoring Suhu dan Kontrol Air Pada Kandang Burung Puyuh Petelur dengan Menggunakan Protokol MQTT (Implementation of Internet of Things on Temperature Monitoring Systems and Water Control in Quail Farms Using the MQTT Protocol)*. <https://1sheeld.com/mqtt-protocol/pure-javascript-mqtt-broker/>
- Muhammad Nasir, & Zainul Al Gifari. (2024). Rancang Bangun Sistem Keamanan Pintu Rumah Menggunakan Solenoid Door Lock Dan Magnetic Switch Sensor Dengan Notifikasi Dan Kontrol Melalui Telegram. In *Seminar Nasional Industri dan Teknologi (SNIT)*.
- Nasir, M., & Al Gifari, Z. (2024). Rancang Bangun Sistem Keamanan Pintu Rumah Menggunakan Solenoid Door Lock Dan Magnetic Switch Sensor Dengan Notifikasi Dan Kontrol Melalui Telegram. In Seminar Nasional Industri dan Teknologi (SNIT).
- Pindarwati, A., Nurfebrian, A., Ray, B. H., Hidayat, R., Mahira Salsabillah, A., Dwiyanti, R., Anisa, S., Damayanti, E., L, S. I., & Artikel, R. (2022). Jurnal Multidisiplin Indonesia Implementasi Penggunaan Cctv Berbasis Internet Of Things (Iot) Sebagai Smart Security Untuk Menanggulangi Angka Kejahatan Studi Kasus: Smk Insan Cita. *Jurnal Multidisiplin Indonesia*, 1(2). <https://jmi.rivierapublishing.id/>
- Priyoga Listyo Ananda, Neisy Wardhani, & Eni Nurhayati. (2024). *Pemanfaatan Bahasa Pemograman Web Untuk Meningkatkan Pemahaman*

- Teknologi Informasi: Studi Kasus Penggunaan Visual Studio Code Di Program Studi Informatika Upn Veteran Jawa Timur.*
- Raharti. (2019). “Whatsapp” Media Komunikasi Efektif Masa Kini (Studi Kasus Pada Layanan Jasa Informasi Ilmiah Di Kawasan Puspiptek).
- Ramdany, S. W., Aulia Kaidar, S., Aguchino, B., Amelia, C., Putri, A., & Anggie, R. (2024). Penerapan UML Class Diagram dalam Perancangan Sistem Informasi Perpustakaan Berbasis Web. In *Journal of Industrial and Engineering System* (Vol. 5, Issue 1).
- Rizky, M., Fadhillah, N., Program, J., & Komputer, S. I. (2024). Rancang Bangun Sistem Pemantau Kualitas Air Pada PT Abacus Dana Pensiuntama Berbasis Arduino Uno. *Jurnal Teknologi Informasi*, 10. <https://ejournal.urindo.ac.id/index.php/TI/index>
- Rohmaniati, & Heri Haerudin2. (2022). *Perancangan Dashboard Monitoring Painting Defect Berbasis Website*. 1(10).
- Romzi, M., & Kurniawan, B. (2020). Implementasi Pemrograman Python Menggunakan Visual Studio Code. In *JIK: Vol. XI* (Issue 2). www.python.org
- Royhan, M. (2021). Fingerprint Untuk mengunci Pintu Terintegrasi Dengan Arduino. *Jurnal Teknik Informatika Unis*, 9(1), 2252–5351. <https://www.arduino.cc>
- Safara Alfan, D., Rochman, A., Firdaus, M., Setiawan, N., & Rosyani, P. (2024). Penerapan Metode Haar-Cascade Dan LBPH Untuk Face Detection dan Recognition. *Jurnal Artificial Inteligent Dan Sistem Penunjang Keputusan*, 2(1).
- Saiqul Umam, M., Adi Wibowo, S., & Agus Pranoto, Y. (2023). Implementasi Protokol Mqtt Pada Aplikasi Smart Garden Berbasis Iot (Internet Of Things). In *Jurnal Mahasiswa Teknik Informatika* (Vol. 7, Issue 1).
- Sanjaya, W. S. M., Anggraeni, D., Zakaria, K., Juwardi, A., & Munawwaroh, M. (2017). *The design of face recognition and tracking for human-robot interaction*. *Proceedings - 2017 2nd International Conferences on Information Technology, Information Systems and Electrical Engineering, ICITISEE 2017, 2018-January*, 315–320. <https://doi.org/10.1109/ICITISEE.2017.8285519>

- Sinlae, F., Kalmany, L., Setiaji, R., & Syahrul, M. (2024). *Menjelajahi Dunia Web: Panduan Pemula Untuk Pemrograman Web.* 2(2). <https://doi.org/10.38035/jsmd.v2i2>
- Sree, U. R., & Mohan, P. (2024). Comparison of Utility-First CSS Framework. In *Journal Of Innovation And Technology* (Vol. 2024, Issue 32).
- Suci, D., Trimarsiah, Y., & Informatika JurnalInformatika dan Komputer, J. (2021). Sistem Informasi Kepegawaian Madrasah Aliyah Al-Azhar Center Baturaja Menggunakan *Embarcadero Xe2* Berbasis Client Server. In *JIK* (Vol. 12, Issue 2).
- Sumit Singh, D. K. Y. (2015). *Fingerprint Based Attendance System Using Microcontroller and LabView. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 04(06), 5111–5121. <https://doi.org/10.15662/ijareeie.2015.0406029>
- Surbakti, N. M., Angelyca Angelyca, Anita Talia, Cecilia Br Perangin-Angin, Dina Olivia Nainggolan, Nia Devi Friskauly, & Sikap Ruth Br Tumorang. (2024). Penggunaan Bahasa Pemrograman Python dalam Pembelajaran Kalkulus Fungsi Dua Variabel. *Algoritma : Jurnal Matematika, Ilmu Pengetahuan Alam, Kebumian Dan Angkasa*, 2(3), 98–107. <https://doi.org/10.62383/algoritma.v2i3.67>
- Tsalatsah, I. E., & Ratama, N. (2024). *Otomatisasi Sistem Keamanan Dan Monitoring Pada Pintu Gerbang Rumah Dengan Pengenalan Wajah Menggunakan Arduino.* <https://journal.mediapublikasi.id/index.php/logic>
- Wardhani, W., Hadi, S., & Budiarto, J. (2021). Rancang Bangun Sistem Monitoring Suhu dan Kelembaban Udara Pada Ruang Server Berbasis Wireless Sensor Network. *JTT (Jurnal Teknologi Terpadu)*, 9(2), 115–125. <https://doi.org/10.32487/jtt.v9i2.1155>
- Wiguna, C. W., Dedy Irawan, J., & Orisa, M. (2022). Penerapan Metode *Convolutional Neural Network* Pada Aplikasi Deteksi Wajah Buronan Berbasis Web. In *Jurnal Mahasiswa Teknik Informatika* (Vol. 6, Issue 2).
- Wijaya Setiady, K., & Amanda Ginting, J. (2023). *Perancangan Dan Implementasi Security Dan Sistem Kendali Otomatis Smart Home Menggunakan Nodemcu Design And Implementation Of Security And Smart*

*Home Automatic Control Systems Using Nodemcu.* VI(1), 543–552.  
<https://doi.org/10.30813/j-alu.v2i2.3756>

Wrastawa Ridwan, Fahriansyah S. Dg. Parebba, Iskandar Z. Nasibu, & Ifan Wiranto. (2023). *Sistem Pengamanan Rumah dan Pengendali Penerangan Menggunakan ESP8266 dan Blynk.*

Yola Berliana Safira, & Susi Wagiyati Purtiningrum. (2023). *Sistem Pendukung Keputusan Penilaian Ketidakdisiplinan Siswa Menggunakan Metode SAW Berbasis Web (Studi Kasus : MA Al-Muddatsiriyah)*. <https://journals.upi-yai.ac.id/index.php/ikraith-informatika/issue/archive>

## LAMPIRAN

### Lampiran 1: Hasil Wawancara

#### A. Wawancara Dengan Sekertaris Desa Wargaluyu

Narasumber : Ayi Suhendar  
Jabatan : Sekertaris Desa  
Hari / Tanggal : Sabtu, 8 Maret 2025  
Instansi : Pemerintahan Desa

Wawancara dilakukan untuk mendukung pengumpulan data dalam penelitian “Sistem *Monitoring* Keamanan Rumah Menggunakan *Fingerprint* dan *Face Recognition* Berbasis *Deep Learning*”. Narasumber adalah pegawai desa Wargaluyu yang memahami kondisi keamanan lingkungan. Wawancara ini bertujuan menggali informasi terkait kebutuhan dan potensi penerapan sistem keamanan berbasis teknologi. Berikut disajikan pertanyaan dan jawaban yang diperoleh.

No.	Pertanyaan	Jawaban
1.	Apa bentuk perhatian pemerintah desa terhadap aspek keamanan rumah warga, khususnya dalam hal pemanfaatan teknologi?	Dengan menerapkan siskamling. Dan untuk teknologi baru pada <i>User</i> an CCTV itu pun untuk beberapa rumah saja.
2.	Mengapa pemerintah desa belum memiliki program atau wacana terkait sistem keamanan digital seperti CCTV atau kunci otomatis??	Karena desa masih tergolong wilayah pedesaan dengan kondisi akses jalan yang terbatas atau jalan buntu, sehingga potensi kejahatan relatif rendah
3.	Siapa saja pihak luar yang pernah bekerja sama dengan desa dalam	Belum ada, jika untuk pengembangan keamanan, namun kedepannya akan ada rencana

	pengembangan teknologi untuk mendukung keamanan?	untuk pemasangan wifi gratis di dua titik, hal ini bisa menjadikan awal berkembangnya teknologi di pedesaan.
4.	Bagaimana pendapat Bapak mengenai pengembangan sistem smart door lock berbasis <i>Fingerprint</i> dan <i>Face Recognition</i> dengan <i>monitoring</i> melalui website dan notifikasi WhatsApp?	Sangat bagus, apalagi di daerah perdesaan.
5.	Siapa saja di kalangan masyarakat desa yang kemungkinan besar akan terbuka terhadap <i>user</i> an teknologi biometrik untuk keamanan rumah?	Untuk seluruh masyarakat dirasa akan terbuka, terlebih lagi teknologi ini untuk keamanan rumah.
6.	Apa saja tantangan yang mungkin dihadapi jika sistem ini diterapkan di lingkungan desa?	Tantangannya adalah ada pada masalah biaya.

Pewawancara

Siti Allanurin

Sekertaris Desa



Ayi Suhendar

## B. Wawancara Dengan Ketua RT 02/RW 08 Desa Wargaluyu

Narasumber : Eden Rohandi  
Jabatan : Ketua RT 02  
Hari / Tanggal : Sabtu, 15 Maret 2025  
Instansi : Lembaga Kepemerintahan

Wawancara ini merupakan bagian dari proses pengumpulan data penting untuk mendukung penelitian yang berjudul “Sistem *Monitoring Keamanan Rumah Menggunakan Fingerprint dan Face Recognition Berbasis Deep Learning*”. Narasumber dalam wawancara ini adalah Ketua RT 02 RW 08 Desa Wargaluyu, yang dipilih karena dinilai memahami kondisi keamanan lingkungan setempat secara langsung. Berikut disajikan daftar pertanyaan beserta jawaban hasil dari wawancara yang berkaitan dengan permasalahan keamanan rumah di wilayah tersebut.

No.	Pertanyaan	Jawaban
1.	Apa sistem keamanan yang umumnya digunakan oleh warga di wilayah ini saat ini?	Sistem keamanannya menggunakan kunci atau gembok biasa.
2.	Mengapa <i>user</i> an kunci konvensional masih menjadi pilihan utama di masyarakat?	Karena yang memang umum digunakan dan tergolong murah sesuai dengan keuangan.
3.	Kapan terakhir kali terjadi percobaan pembobolan atau pencurian di lingkungan RT ini?	2 tahun llau ada motor yang hilang, dan beebrapa bulan lalu terdapat pintu yang kebobolan.
4.	Di mana saja titik-titik rawan keamanan atau rumah yang pernah mengalami upaya pembobolan?	Di kp. Cihonje, kp. Cimomplo.
4.	Siapa saja yang sudah mulai menggunakan teknologi	Beberapa orang saja yang telah menggunakan CCTV di

	keamanan seperti CCTV, dan kunci pintar di lingkungan ini?	rumahnya, namun untuk kunci pintar belum ada yang menggunakan.
5.	Apakah <i>user</i> an cctv ini telah menggunakan pendekksi wajah? Jadi nantinya hanya orang rumah saja yang bisa masuk!	Belum, cctv digunakan berdasarkan standar dari bawaan pabrik, dan tidak terhubung pada keamanan lainnya.
6.	Bagaimana pandangan Bapak terkait kesiapan masyarakat dalam menerima teknologi keamanan berbasis IoT seperti sistem smart door lock dan <i>monitoring</i> web?	Karena kebutuhan keamanan maka sangat siap untuk teknoLoginya, namun belum siap dalam hal pembiayaannya.

**Pewawancara**



Siti Allanurin

**Ketua RT 02**



Eden Rohandi

### C. Wawancara Dengan Salah Satu Warga (korban pencurian).

Narasumber : Haryanti  
 Status : Warga kp. Cihonje Desa Wargaluyu  
 Hari / Tanggal : Sabtu, 22 Maret 2025

Wawancara ini dilakukan dengan salah satu warga Desa Wargaluyu yang pernah menjadi korban pencurian. Tujuannya adalah untuk memperoleh gambaran langsung mengenai kondisi keamanan rumah warga, sebagai bagian dari data pendukung dalam penelitian “Sistem Monitoring Keamanan Rumah Menggunakan *Fingerprint* dan *Face Recognition* Berbasis *Deep Learning*”. Berikut disajikan pertanyaan beserta jawaban yang diperoleh.

No.	Pertanyaan	Jawaban
1.	Apa yang terjadi saat pencurian berlangsung dan bagaimana kronologinya menurut Ibu??	Pencurian terjadi pada waktu malam saat semua orang tertidur, pencuri masuk kedalam rumah melewati jendela yang berdekatan dengan pintu.
2.	Siapa yang mengetahui atau terdampak saat kejadian pencurian berlangsung?	Saya dan keluarga, karena selain merusak properti juga mengambil barang berharga.
3.	Kapan kira kira kejadian itu berlangsung?	Sekitar 5 tahun lalu, namun kemarin tetangga ada yang mengalami hal yang sama, yaitu pencurian emas.
4.	Mengapa sistem keamanan ibu tidak mampu mencegah kejadian tersebut?	Karena menggunakan kunci biasa jadinya mudah untuk di bobol. Dan tidak ada bukti juga jika akan di laporkan pada pihak yang berwajib.

5.	Bagaimana perubahan yang Ibu lakukan pada sistem keamanan rumah?	Untuk pintu di tambahkan kunci ganda berupa kunci selot, dan untuk dijendela di tambahkan besi tralis.
6.	Bagaimana pendapat Ibu mengenai <i>user</i> an sistem keamanan berbasis biometrik, seperti <i>Fingerprint</i> dan <i>Face Recognition</i> ?	Bagus, karena akan sangat meningkatkan keamanan rumah.
7.	Jika tersedia sistem yang dapat mendeteksi upaya pencurian dan mengirim notifikasi secara langsung ke ponsel, apakah Ibu bersedia menggunakannya di rumah?	Bersedia sekali.

**Pewawancara**

**Siti Allanurin**

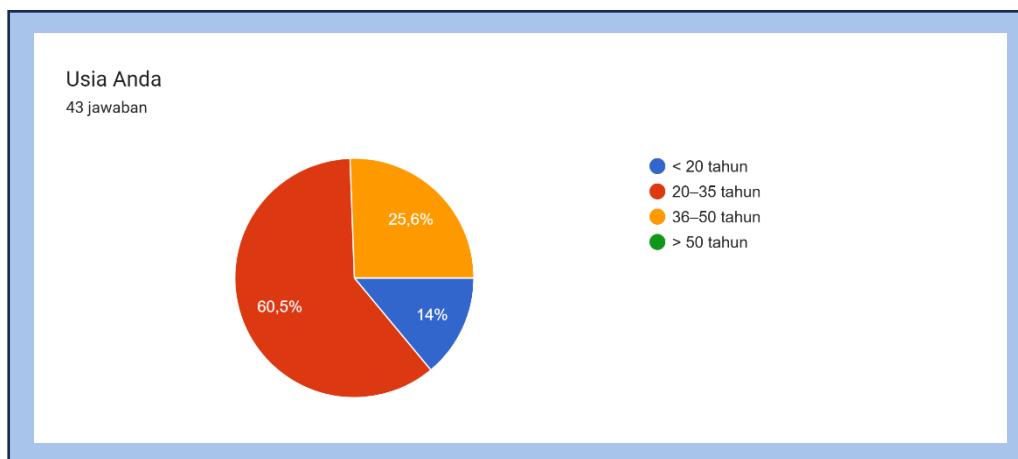
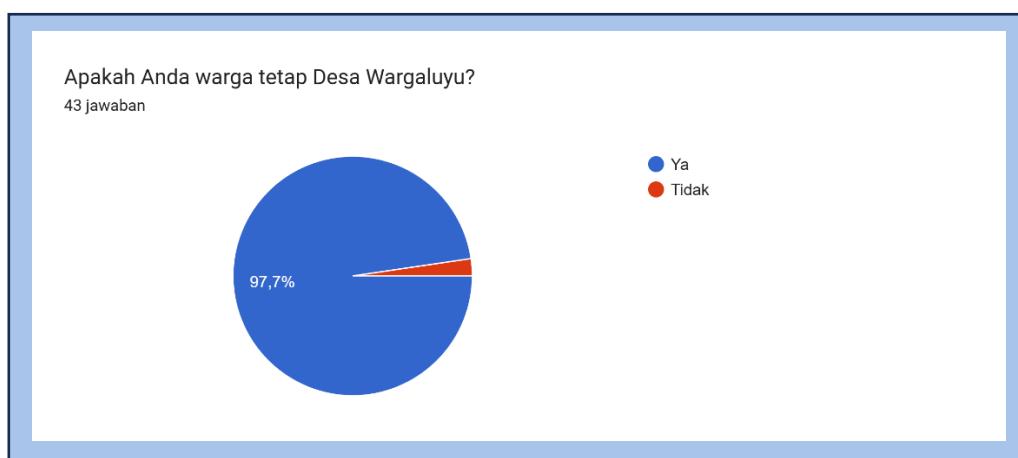
**Warga Desa Wargaluyu**

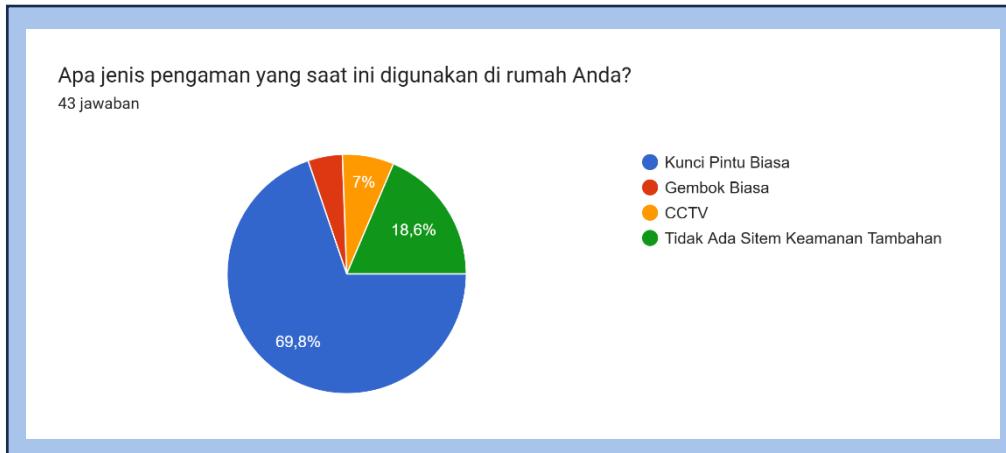
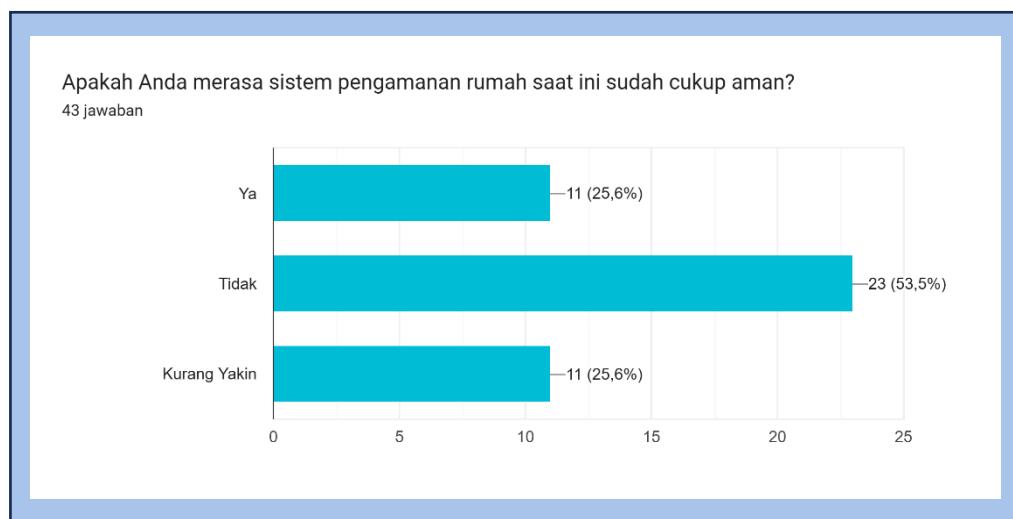
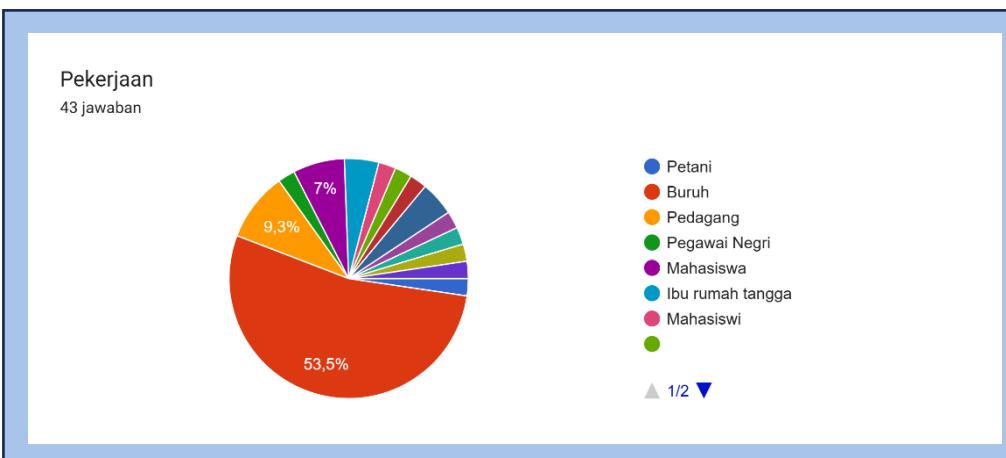
**Haryanti**

## Lampiran 2 : Hasil Kuisioner Warga Desa Wargaluyu

Jumlah Responden : 43  
Media Kuisioner : Google Forms  
Waktu Pengisian : Jum'at, 7 Maret 2025  
Lokasi Responden : Desa Wargaluyu

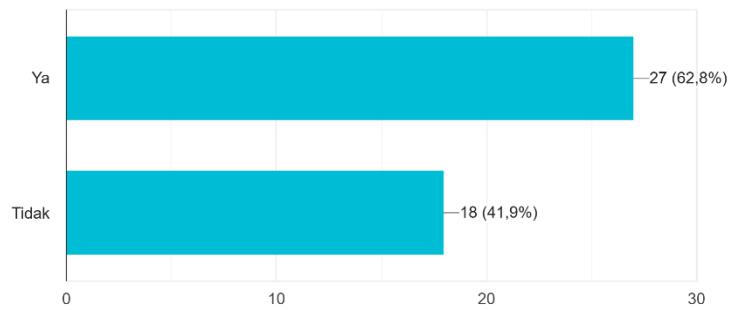
Kuisisioner ini untuk mengumpulkan data mengenai pengalaman, persepsi, serta minat warga terhadap sistem keamanan rumah, serta potensi penerapan teknologi IoT. Data yang terkumpul diolah dan disajikan dalam bentuk diagram batang dan diagram lingkaran untuk menggambarkan distribusi jawaban secara visual. Berikut hasil rekapitulasi kuisioner warga:





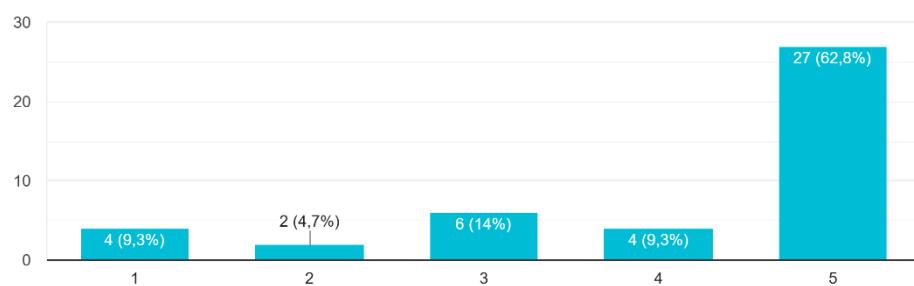
Pernahkah Anda mengalami kasus pencurian?

43 jawaban



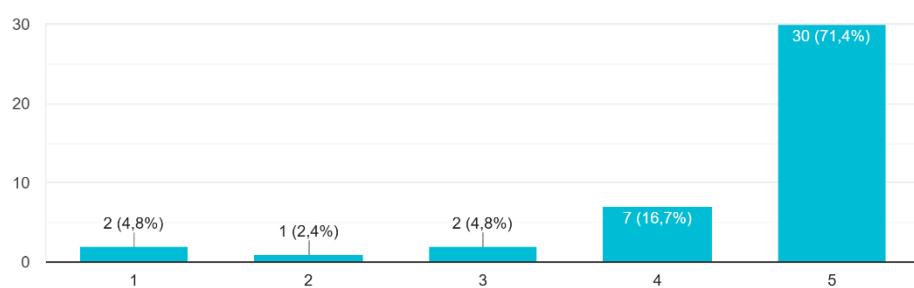
Apakah Anda pernah merasa khawatir meninggalkan rumah dalam keadaan kosong?

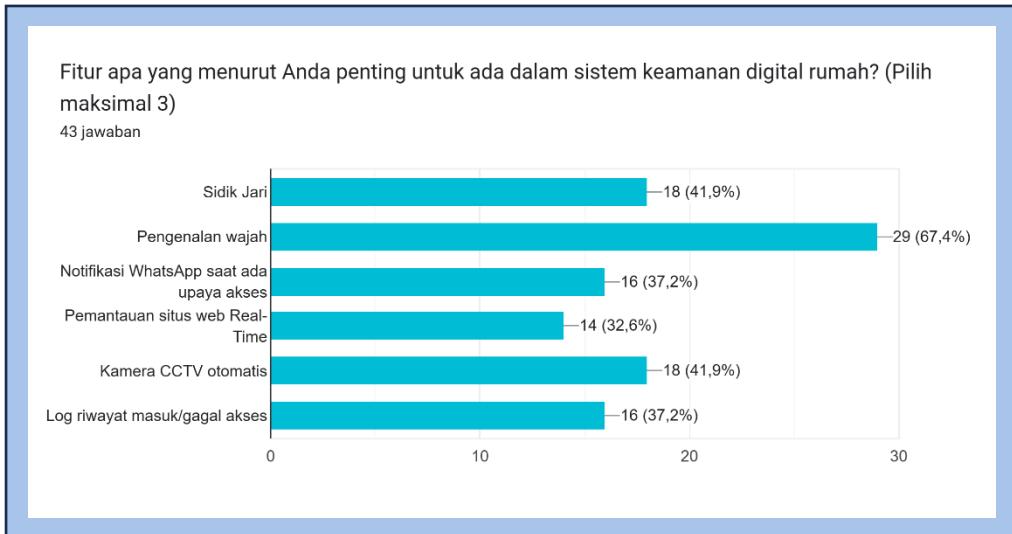
43 jawaban



Jika tersedia, apakah Anda tertarik menggunakan sistem monitoring keamanan rumah dengan Smart Door lock di rumah Anda?

42 jawaban





Kuesioner ini diisi oleh 43 responden, mayoritas (93%) merupakan warga tetap Desa Wargaluyu. Sebagian besar berusia 20–35 tahun (60,5%) dengan pekerjaan terbanyak buruh (53,5%), diikuti pedagang (9,5%) dan mahasiswa (7%). Terkait persepsi keamanan rumah, 53,5% responden merasa tidak aman, 25,6% kurang yakin, dan 25,6% merasa cukup aman. Pengamanan paling umum adalah kunci konvensional; belum ada yang menggunakan teknologi keamanan modern, sementara CCTV digunakan oleh sekitar 7% responden. Sebanyak 62,8% pernah mengalami pencurian dan mayoritas sering khawatir saat meninggalkan rumah kosong.

Mayoritas responden menunjukkan minat tinggi terhadap teknologi keamanan modern, dengan 71,4% tertarik pada Smart Door Lock. Fitur yang dianggap penting meliputi pengenalan wajah (67,4%), sidik jari dan CCTV otomatis (41,9%), notifikasi WhatsApp dan log aktivitas (37,2%), serta pemantauan real-time via website (32,6%).

Secara keseluruhan, hasil kuesioner menunjukkan bahwa sistem keamanan rumah berbasis teknologi, seperti yang dirancang dalam penelitian ini, berpotensi tinggi diterima masyarakat Desa Wargaluyu. Dengan mempertimbangkan biaya, kemudahan penggunaan, dan edukasi, solusi ini dapat menjadi jawaban atas kebutuhan keamanan rumah yang lebih optimal.

### **Lampiran 3 : Dokumentasi Wawancara**

#### **1. Dokumentasi Wawancara Dengan Sekertaris Desa**

**Lokasi : Kp. Carirang Desa Wargaluyu**



## 2. Dokumentasi Wawancara Dengan Ketua RT02/ RW08 Desa Wargaluyu

Lokasi : Kp. Cimomplo rt 02/ rw08 Desa Wargaluyu



### **3. Dokumentasi Wawancara Dengan Salah Satu Warga**

**Lokasi : Kp. Cihonje rt 05/ rw02 Desa Wargaluyu**



#### **Lampiran 4: TOR (*Term of Reference*)**

Penelitian ini dilakukan di Desa Wargaluyu dengan tujuan merancang sistem *monitoring* keamanan rumah menggunakan autentikasi *Fingerprint* dan *Face Recognition* berbasis *Deep Learning*. Sistem ini bertujuan meningkatkan keamanan dengan membatasi akses masuk hanya kepada individu yang terdaftar, serta secara otomatis mencatat dan memantau aktivitas melalui kamera CCTV 360 yang terhubung dengan server. Perangkat keras yang digunakan mencakup sensor sidik jari, kamera CCTV, dan mikrokontroler (ESP32), sementara algoritma *Deep Learning* untuk fitur *Face Recognition* diimplementasikan menggunakan Python. Sistem ini juga terintegrasi dengan notifikasi WhatsApp untuk memberikan peringatan secara *real-time* ketika terdeteksi upaya akses ilegal. Sistem *monitoring* ini berbasis web, yang memungkinkan pemantauan riwayat aktivitas dan status keamanan rumah secara langsung melalui *dashboard*. Metode penelitian yang digunakan meliputi observasi, wawancara, dan studi pustaka, yang dilakukan setelah lokasi penelitian disetujui.

Untuk memastikan fokus dan ruang lingkup penelitian, ditetapkan beberapa batasan sebagai berikut:

1. Fokus pada pembangunan sistem *monitoring* keamanan rumah berbasis *Fingerprint* dan *Face Recognition*.
2. Sistem hanya mengenali wajah dan sidik jari yang telah terdaftar dalam *database*.
3. Kamera CCTV digunakan untuk mengambil citra wajah dalam proses identifikasi.
4. Notifikasi dikirim melalui WhatsApp jika terdeteksi wajah yang tidak dikenali.
5. Sistem *monitoring* dilengkapi dengan website yang menampilkan riwayat aktivitas dan status keamanan rumah secara *real-time*.
6. Pengujian dilakukan dalam bentuk simulasi, dengan sistem web yang berjalan sepenuhnya, sedangkan perangkat keras IoT masih berupa prototipe dan belum diimplementasikan secara fisik pada pintu rumah.

7. Peneliti bertanggung jawab atas pembuatan prototipe untuk keperluan uji coba, sementara biaya implementasi lebih lanjut berada di luar tanggung jawab peneliti.

Bandung, April 2025

Disetujui Oleh:

**Pewawancara**



**Siti Allanurin**

**Ketua RT 02**



**Eden Rohandi**

## Lampiran 5 : Listing Program

### A. Listing Program Mikrokontroller

```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <PubSubClient.h>
#include <Adafruit_Fingerprint.h>
#include <Wire.h>
#include <hd44780.h>
#include <hd44780ioClass/hd44780_I2Cexp.h>

// ===== KONFIGURASI WIFI & MQTT =====
const char* ssid = "NR";
const char* password = "gataulupa";
constchar*mqtt_server=
"f0b9c2d7c89643e6858d3868ae0fe474.s1.eu.hivemq.cloud";
const int mqtt_port = 8883;
const char* mqtt_user = "hivemq.webclient.1749797854399";
const char* mqtt_password = "Ye8b>3V?p.64Ud,FAgBt";

// ===== TOPIK MQTT =====
const char* t_reset_fp = "door/reset/fingerprint";
const char* t_reset_fp_id = "door/reset/fingerprint/id";
const char* t_login_status = "door/login/status";
const char* t_register_status = "door/register/status";
const char* t_message = "door/message";
const char* t_status = "door/status";
const char* t_register_fp = "door/register/fingerprint";
const char* t_login_fp = "door/login/fingerprint";

// ===== PIN =====
#define RELAY_PIN 4
```

```

#define BUZZER_PIN 2
#define BUTTON_PIN 5
#define RXD2 16
#define TXD2 17

// ===== LCD =====
hd44780_I2Cexp lcd;
const int lcdColumns = 16;
const int lcdRows = 2;

// ===== OBJEK =====
WiFiClientSecure espClient;
PubSubClient client(espClient);
HardwareSerial fingerSerial(2);
Adafruit_Fingerprint finger(&fingerSerial);

// ===== VARIABEL GLOBAL =====
volatile bool modeLogin = false;
volatile bool modeRegister = false;
bool relayState = false;
bool lastButtonState = HIGH;
bool loginInProgress = false;
bool registerInProgress = false;
unsigned long registerStepTime = 0;
int registerStep = 0;
int enrollID = -1;

// ===== FUNGSI BANTUAN =====
void printLCD(const char* msg) {
    lcd.clear();
    String line1 = "", line2 = "";
    String input = String(msg);

```

```

int newLineIndex = input.indexOf('\n');
if (newLineIndex != -1) {
    line1 = input.substring(0, newLineIndex);
    line2 = input.substring(newLineIndex + 1);
} else {
    line1 = input;
}
while (line1.length() < 16) line1 += ' ';
while (line2.length() < 16) line2 += ' ';
lcd.setCursor(0, 0); lcd.print(line1);
lcd.setCursor(0, 1); lcd.print(line2);
}

void beep(bool success) {
    tone(BUZZER_PIN, success ? 2000 : 500, 200);
    delay(200);
    noTone(BUZZER_PIN);
}

// ===== WIFI =====
void connectWiFi() {
    printLCD("Connecting WiFi");
    WiFi.begin(ssid, password);
    for (int i = 0; i < 20 && WiFi.status() != WL_CONNECTED; i++) delay(500);
    printLCD(WiFi.status() == WL_CONNECTED ? "WiFi Connected" : "WiFi Failed");
    beep(WiFi.status() == WL_CONNECTED);
    espClient.setInsecure();
}

// ===== MQTT CALLBACK =====
void callback(char* topic, byte* payload, unsigned int length) {

```

```

String msg;
for (unsigned int i = 0; i < length; i++) msg += (char)payload[i];
msg.trim();
String topicStr = String(topic);
if (topicStr == t_login_status && msg == "1") {
    modeLogin = true; modeRegister = false;
    printLCD("Login: Scan jari"); beep(true);
}
else if (topicStr == t_register_status && msg == "1") {
    modeRegister = true; modeLogin = false;
    printLCD("Regis: Scan jari"); beep(true);
}
else if (topicStr == t_message) {
    printLCD(msg.c_str()); beep(true);
}
else if (topicStr == t_status) {
    relayState = (msg == "1");
    digitalWrite(RELAY_PIN, relayState);
    printLCD(relayState ? "Pintu Tertutup" : "Pintu Terbuka");
    beep(true);
}
else if (topicStr == t_reset_fp && msg == "1") {
    resetFingerprintData();
}
else if (topicStr == t_reset_fp_id && msg.length() > 0) {
    int idToDelete = msg.toInt();
    resetFingerprintByID(idToDelete);
}
}

// ===== MQTT CONNECT =====
void reconnect() {

```

```

while (!client.connected()) {
    printLCD("MQTT connect...");
    if (client.connect("ESP32_FINGER_CLIENT", mqtt_user,
    mqtt_password)) {
        client.subscribe(t_login_status);
        client.subscribe(t_register_status);
        client.subscribe(t_message);
        client.subscribe(t_status);
        client.subscribe(t_register_fp);
        client.subscribe(t_login_fp);
        client.subscribe(t_reset_fp);
        client.subscribe(t_reset_fp_id);
        printLCD("MQTT Connected");
        beep(true);
    } else {
        printLCD("MQTT Failed");
        delay(2000);
    }
}

// ===== ID KOSONG =====
int getFreeID() {
    for (int id = 1; id < 127; id++) {
        if (finger.loadModel(id) != FINGERPRINT_OK)
            return id;
    }
    return -1;
}

// ===== HANDLE LOGIN =====
void handleLogin() {

```

```

if (modeLogin && finger.getImage() == FINGERPRINT_OK)
{
    if (finger.image2Tz(1) == FINGERPRINT_OK &&
    finger.fingerSearch() == FINGERPRINT_OK) {
        client.publish(t_login_fp, String(finger.fingerID).c_str());
        printLCD("Scan Wajah");
        beep(true);
    } else {
        printLCD("Login Gagal");
        client.publish(t_login_fp, "0");
        beep(false);
    }
    loginInProgress = false;
    modeLogin = false;
}
}

// ===== HANDLE REGISTER =====
void handleRegister() {
    if (modeRegister && !registerInProgress) {
        enrollID = getFreeID();
        if (enrollID == -1) {
            printLCD("Memori penuh!");
            beep(false);
            modeRegister = false;
        } else {
            printLCD("Regis: Jari #1");
            registerInProgress = true;
            registerStep = 1;
        }
    }
}

if (!registerInProgress) return;

```

```

    if (registerStep == 1 && finger.getImage() ==
FINGERPRINT_OK) {
    if (finger.image2Tz(1) == FINGERPRINT_OK) {
        printLCD("Angkat jari");
        registerStepTime = millis();
        registerStep = 2;
    } else {
        printLCD("Gagal scan #1");
        beep(false);
        registerInProgress = false;
        modeRegister = false;
    }
}
else if (registerStep == 2 && millis() - registerStepTime > 2000)
{
    printLCD("Tempel jari #2");
    registerStep = 3;
}
else if (registerStep == 3 && finger.getImage() ==
FINGERPRINT_OK) {
    if (finger.image2Tz(2) == FINGERPRINT_OK &&
        finger.createModel() == FINGERPRINT_OK &&
        finger.storeModel(enrollID) == FINGERPRINT_OK) {
        client.publish(t_register_fp, String(enrollID).c_str());
        printLCD("Regis Berhasil");
        beep(true);
    } else {
        printLCD("Regis Gagal");
        beep(false);
    }
}
registerInProgress = false;
modeRegister = false;

```

```

        }

    }

// ===== HANDLE RESET FINGER =====

void resetFingerprintData() {
    if (finger.emptyDatabase() == FINGERPRINT_OK) {
        printLCD("DB Finger Reset");
        beep(true);
    } else {
        printLCD("Reset Gagal");
        beep(false);
    }
}

void resetFingerprintByID(int id) {
    if (id < 1 || id > 126) {
        printLCD("ID tidak valid");
        beep(false);
        return;
    }
    if (finger.deleteModel(id) == FINGERPRINT_OK) {
        printLCD(("Hapus ID: " + String(id)).c_str());
        beep(true);
    } else {
        printLCD(("Gagal hapus ID: " + String(id)).c_str());
        beep(false);
    }
}

// ===== TOMBOL MANUAL =====

void button_manual() {
    bool nowBtn = digitalRead(BUTTON_PIN);
    if (lastButtonState == HIGH && nowBtn == LOW) {

```

```

relayState = !relayState;
digitalWrite(RELAY_PIN, relayState);
printLCD(relayState ? "Manual: Tutup" : "Manual: Buka");
beep(true);
delay(2000);
}

lastButtonState = nowBtn;
}

// ===== SETUP =====
void setup() {
Serial.begin(115200);
Serial2.begin(57600, SERIAL_8N1, RXD2, TXD2);
finger.begin(57600);
finger.getTemplateCount();
pinMode(RELAY_PIN, OUTPUT);
pinMode(BUZZER_PIN, OUTPUT);
pinMode(BUTTON_PIN, INPUT_PULLUP);
digitalWrite(RELAY_PIN, LOW);
digitalWrite(BUZZER_PIN, LOW);
lcd.begin(lcdColumns, lcdRows);
lcd.print("Init...");
connectWiFi();
client.setServer(mqtt_server, mqtt_port);
client.setCallback(callback);
}

// ===== LOOP =====
void loop() {
if (!client.connected()) reconnect();
client.loop();
button_manual();
}

```

```
    handleLogin();  
    handleRegister();  
}
```

## B. Listing Program Face Recognition

```
import cv2  
import face_recognition  
import numpy as np  
import datetime  
import os  
import time  
from .camera_stream import  
frame_buffer, frame_lock, known_face_encodings, known_face_n  
ames, faces_loaded, threading, streaming_active  
import pickle  
import hashlib  
def load_known_faces():  
    global known_face_encodings, known_face_names  
    known_face_encodings = []  
    known_face_names = []  
    base_path = "app/static/train model/snapshots"  
    cache_path = "app/camera/face_cache.pkl"  
    cache_data = {}  
    # Load cache if exists  
    if os.path.exists(cache_path):  
        with open(cache_path, 'rb') as f:  
            cache_data = pickle.load(f)  
    updated_cache = {}  
    try:  
        for folder_name in os.listdir(base_path):  
            folder_path = os.path.join(base_path, folder_name)
```

```

if os.path.isdir(folder_path):
    user_name = folder_name
    for filename in os.listdir(folder_path):
        if filename.lower().endswith('.jpg', '.jpeg', '.png'):
            file_path = os.path.join(folder_path, filename)
    # Buat hash berdasarkan isi file untuk mendeteksi perubahan
            with open(file_path, 'rb') as img_file:
                file_hash = hashlib.md5(img_file.read()).hexdigest()
                cache_key = f'{user_name}/{filename}'
    # Cek apakah sudah di-cache dan tidak berubah
    if cache_key in cache_data and cache_data[cache_key]['hash'] == file_hash:
        encoding = cache_data[cache_key]['encoding']
        print(f'♻️ Cache digunakan untuk: {cache_key}')
        else:
            image = face_recognition.load_image_file(file_path)
            face_encodings = face_recognition.face_encodings(image)
            if not face_encodings:
                print(f'[!] Tidak ada wajah di file: {file_path}')
                continue
            encoding = face_encodings[0]
            print(f'☑ Wajah dimuat ulang: {cache_key}')
    # Simpan ke data runtime dan cache baru
            known_face_encodings.append(encoding)
            known_face_names.append(user_name)
            updated_cache[cache_key] = {
                "hash": file_hash,
                "encoding": encoding
            }
    except Exception as e:
        print(f'[!] Gagal memuat wajah: {e}')

```

```

# Simpan cache
with open(cache_path, 'wb') as f:
    pickle.dump(updated_cache, f)
return known_face_encodings, known_face_names

#* Screenshot Foto CCTV

def take_snapshot(RTSP_URL):
    print("[INFO] Connecting to CCTV...")
    cap = cv2.VideoCapture(RTSP_URL)
    if not cap.isOpened():
        print("[ERROR] Failed to connect to CCTV.")
        return None
    print("[INFO] Reading frame...")
    ret, frame = cap.read()
    cap.release()
    if not ret:
        print("[ERROR] Failed to capture frame.")
        return None
    # Buat folder jika belum ada
    output_dir = "app/static/snapshots/captured"
    os.makedirs(output_dir, exist_ok=True)
    # Simpan frame ke file
    timestamp=datetime.datetime.now().strftime("%Y%m%d_%H%M%S")
    filename = f'{output_dir}/snapshot_{timestamp}.jpg'
    cv2.imwrite(filename, frame)
    print(f'[INFO] Snapshot saved at {filename}')
    return filename

def start_capture_thread(RTSP_URL):
    global streaming_active
    cap = cv2.VideoCapture(RTSP_URL)
    cap.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
    cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
    cap.set(cv2.CAP_PROP_BUFFERSIZE, 1)

```

```

if streaming_active:
    return
streaming_active = True
# Jalankan thread capture
threading.Thread(target=capture_thread, args=(RTSP_URL,), daemon=True).start()
def capture_thread(RTSP_URL):
    global frame_buffer
    cap = cv2.VideoCapture(RTSP_URL, cv2.CAP_FFMPEG)
    if not cap.isOpened():
        print("[ERROR] Failed to open RTSP stream.")
        return
    while True:
        success, frame = cap.read()
        if not success:
            print("[WARNING] Failed to read frame. Retrying...")
            time.sleep(0.5) # kasih delay biar gak 100% CPU usage
            continue
        with frame_lock:
            frame_buffer = frame

#* ===== Video Frame Generator Stream=====
def gen_frames():
    timeout = 10 # detik
    start_time = time.time()
    while True:
        if frame_buffer is None:
            if time.time() - start_time > timeout:
                print("[INFO] Timeout: Tidak ada frame yang diterima.")
                break
            time.sleep(1/40)

```

```

        continue
        start_time = time.time()
        try:
            with frame_lock:
                frame = frame_buffer.copy()
                ret, buffer = cv2.imencode('.jpg', frame)
                if not ret:
                    continue
                frame = buffer.tobytes()
                yield (b'--frame\r\n'
                       b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')
        except Exception as e:
            print(f"[ERROR] Streaming frame failed: {e}")
            continue

```

### C. Listing Program Login

```

from flask import Blueprint, render_template, redirect, request,
url_for, session, flash
import app.config as config
from dotenv import load_dotenv
load_dotenv()

auth = Blueprint('auth', __name__)

# Hardcoded credentials
users = {
    'email': config.ADMIN_EMAIL,
    'password': config.ADMIN_PASSWORD
}

@auth.route('/login', methods=['GET', 'POST'])

```

```

def login():

    if request.method == 'POST':
        email = request.form['email']
        password = request.form['password']
        print(email, password)
        print(users)

        if users.get('email') == email and users.get('password') == password:
            session['user'] = users
            flash('Login successful', 'success')
            return redirect(url_for('main.index'))

        else:
            flash('Invalid email or password', 'danger')
            return redirect(url_for('auth.login'))

    return render_template('pages/auth/login.html')

@auth.route('/logout')
def logout():
    session.pop('user', None)
    return redirect(url_for('auth.login'))

```

#### D. Listing Program MQTT (komunikasi alat dan sistem)

```

import paho.mqtt.client as paho
import os
import time
import ssl
import certifi
from dotenv import load_dotenv
from app import socketio # Untuk emit ke client

```

```

from datetime import datetime, timedelta
load_dotenv()
# 🔑 Konfigurasi
BROKER = os.environ.get('MQTT_BROKER', '')
PORT = 8883
USERNAME = os.environ.get('MQTT_USERNAME')
PASSWORD = os.environ.get('MQTT_PASSWORD')
TOPICS = [
    ("door/register/status", 1),
    ("door/login/status", 1),
    ("door/message", 1), # ✅ ini sudah tuple
    ("door/register/fingerprint", 1),
    ("door/login/fingerprint", 1), # ✅ ini sudah tuple
    ("door/status", 1),
    ("door/reset/fingerprint", 1),
    ("door/reset/fingerprint/id", 1),
]
latest_finger_data = {
    "login": None,
    "register": None,
}
finger_last_seen = {
    "login": None,
    "register": None,
}
finger_last_value = {
    "login": None,
    "register": None,
}
finger_last_change = {
    "login": None,
}

```

```

    "register": None,
}

finger_alert_sent = {
    "login": None,
    "register": None,
}

check_times = [1, 2, 3, 4, 5, 10, 24] # jam
SENSOR_RESET_SECONDS = 60 # Reset fingerprint setiap
120 detik
client = None

# 🔍 Callback saat koneksi ke broker MQTT
def on_connect(client, userdata, flags, rc, properties=None):
    if rc == 0:
        print("✅ MQTT terhubung ke broker")
        for topic, qos in TOPICS:
            client.subscribe(topic, qos)
            print(f"🔗 Berlangganan ke topik: {topic}")
        send_login_command()
    else:
        print(f"❌ Gagal koneksi ke MQTT, kode: {rc}")

# 🛡️ Penanganan data register dan login
def handle_fingerprint_data_factory(app):
    def handle_fingerprint_data(client, userdata, message):
        topic = message.topic
        value = message.payload.decode('utf-8').strip()
        mapping = {
            "door/register/fingerprint": "register",
            "door/login/fingerprint": "login",
        }
        label = mapping.get(topic)
        if label:

```

```

try:
    value_parsed = int(value)
except ValueError:
    print(f" ✗ Gagal parsing nilai untuk {label}: {value}")
    return

now = datetime.now()
last_val = finger_last_value[label]
if last_val != value_parsed:
    finger_last_change[label] = now
    finger_last_value[label] = value_parsed
    print(f" ✅ Data {label} terkirim: {value_parsed}")
    latest_finger_data[label] = value_parsed
    finger_last_seen[label] = now
    socketio.emit("finger_update", latest_finger_data)

#  Bungkus bagian yang mengakses Flask context

if label == "login":
    with app.app_context():
        from app.services.finger_service import fingerprint_auth
        fingerprint_auth(value_parsed,app)

    return handle_fingerprint_data

def check_finger_status():
    now = datetime.now()
    updated = False
    for label in latest_finger_data:
        last_seen = finger_last_seen.get(label)
        val = latest_finger_data[label]
        # Reset jika tidak ada pembaruan selama 120 detik
        if last_seen and (now - last_seen).total_seconds() >
SENSOR_RESET_SECONDS:
            if val is not None:
                latest_finger_data[label] = None

```

```

        finger_last_seen[label] = None
        finger_last_value[label] = None
        print(f" ✘ Fingerprint '{label}' direset setelah 60 detik
tidak ada update.")

    updated = True

    if updated:
        socketio.emit("finger_update", latest_finger_data)

# 🔍 Fungsi untuk menjalankan client MQTT
def run_mqtt_service(app=None):
    global client

    print(f" 🚀 Menghubungkan ke MQTT broker di
{BROKER}:{PORT}")

    client=paho.Client(client_id="smart_door_lock",
protocol=paho.MQTTv5)

    client.username_pw_set(USERNAME, PASSWORD)
    client.tls_set(ca_certs=certifi.where(),
tls_version=ssl.PROTOCOL_TLS_CLIENT)

    client.on_connect = on_connect
    client.message_callback_add("door/login/fingerprint",
handle_fingerprint_data_factory(app))

    client.message_callback_add("door/register/fingerprint",
handle_fingerprint_data_factory(app))

    try:
        client.connect(BROKER, PORT)
        client.loop_start()
    except Exception as e:
        print(f" ✘ Gagal koneksi: {e}")
        socketio.emit("mqtt_error", {"error": str(e)})
        return

    try:
        while True:

```

```

check_finger_status()
time.sleep(30) # cek status fingerprint setiap 30 detik

except KeyboardInterrupt:
    print("🔴 Memutuskan koneksi MQTT...")
    client.disconnect()
    client.loop_stop()

# 📡 Kirim perintah login ke ESP32

def send_login_command(msg="1"):
    if client:
        try:
            client.publish("door/login/status", msg)
            print("📡 Kirim perintah 'login/status' ke ESP32")
        except Exception as e:
            print(f"✗ Gagal kirim perintah login: {e}")

    else:
        print("✗ Client MQTT belum tersedia")

# 📡 Kirim perintah register ke ESP32

def send_register_command(msg="0"):
    if client:
        try:
            client.publish("door/register/status", msg)
            print("📡 Kirim perintah 'register/status' ke ESP32")
        except Exception as e:
            print(f"✗ Gagal kirim perintah register: {e}")

    else:
        print("✗ Client MQTT belum tersedia")

def send_message_command(msg):
    if client:
        try:
            client.publish("door/message", msg)

```

```

    print(f"📤 Kirim pesan ke ESP32: {msg}")
except Exception as e:
    print(f"❌ Gagal kirim pesan: {e}")
else:
    print("❌ Client MQTT belum tersedia")
# 📤 Kirim perintah status ke ESP32

def send_door_status_command(msg):
    if client:
        try:
            client.publish("door/status", msg)
            print(f"📤 Kirim status pintu ke ESP32: {msg}")
            socketio.emit("door_status", {"status": msg})
        except Exception as e:
            print(f"❌ Gagal kirim status pintu: {e}")
    else:
        print("❌ Client MQTT belum tersedia")

def send_reset_fingerprint_command(msg):
    if client:
        try:
            client.publish("door/reset/fingerprint", msg)
            print(f"📤 Kirim perintah reset fingerprint ke ESP32: {msg}")
        except Exception as e:
            print(f"❌ Gagal kirim perintah reset fingerprint: {e}")
    else:
        print("❌ Client MQTT belum tersedia")

def send_reset_fingerprintID_command(msg):
    if client:
        try:
            client.publish("door/reset/fingerprint/id", msg)

```

```

print(f"👉 Kirim perintah reset fingerprint ke ESP32:  

{msg}")  

except Exception as e:  

    print(f"❌ Gagal kirim perintah reset fingerprint: {e}")  

else:  

    print("❌ Client MQTT belum tersedia")

```

#### E. Listing Program Akses Pintu (on/ off)

```

from app.repositories.door_access_repositories  

import  

create_door_access, delete_door_access, get_all_door_access, get_  

door_access_by_id, update_door_access  

import os  
  

def create_door_access_service(data):  

    try:  

        door = create_door_access(data)  

        return {"message": "door access created successfully"}, 201  

    except Exception as e:  

        print(f"[!] Terjadi error di service create_door_access: {e}")  

        return {"error": str(e)}, 500  
  

def delete_door_access_service(door_id):  

    try:  

        # Ambil data door access  

        data = get_door_access_by_id(door_id)  

        if not data:  

            return {"status": "error", "message": "Data tidak ditemukan"}  

            # Ambil path gambar dari field img (misal:  

            'static/snapshots/captured/snapshot_20250620_101423.jpg')

```

```

    img_path = data.img
    full_path = os.path.join("app", img_path) if not
    img_path.startswith("app/") else img_path
    # Hapus file gambar jika ada
    if os.path.exists(full_path):
        os.remove(full_path)
        print(f"⚠️ Gambar dihapus: {full_path}")
    else:
        print(f"⚠️ Gambar tidak ditemukan: {full_path}")
    # Hapus data dari database
    delete_door_access(door_id)
    return {"status": "success", "message": "Door access dan
    gambar berhasil dihapus"}
except Exception as e:
    print(f"[!] Terjadi error di service delete_door_access: {e}")
    return {"status": "error", "message": str(e)}
def get_all_door_access_service():
    try:
        door = get_all_door_access()
        return door
    except Exception as e:
        print(f"[!] Terjadi error di service get_all_door_access: {e}")
        return {"error": str(e)}, 500
def get_door_access_by_id_service(door_id):
    try:
        door = get_door_access_by_id(door_id)
        return door
    except Exception as e:
        print(f"[!] Terjadi error di service get_door_access_by_id:
        {e}")
        return None

```

## F. Listing Program Pengelola Data Pengguna

```
from app.repositories import create_user,
    get_user_by_id_with_images,      update_user_finger_id      as
        repo_update_user_finger_id, get_user_by_finger_id, delete_user
        as delete_users, get_user_by_id_with_images_all,
            get_user_by_id, update_user, add_image_for_user
from werkzeug.utils import secure_filename
import os
import shutil
from app.services.mqtt_service import
    send_reset_fingerprintID_command
from app.repositories.user_repositories import
    delete_image_by_id, get_img_by_id, get_images_by_user
import re
UPLOAD_FOLDER = 'app/static/train model/snapshots'
from app.camera.snapshot import take_snapshot,
load_known_faces
def add_user_service(data):
    try:
        # Cek Finger ID
        if get_user_by_finger_id(data['finger']):
            return {"error": "Finger ID already exists"}, 400

        # Buat folder tujuan
        folder_name = f'{data["name"].replace(' ', '_')}{data["finger"]}'
        save_path = os.path.join(UPLOAD_FOLDER, folder_name)
        os.makedirs(save_path, exist_ok=True)
        image_paths = []
        # Simpan semua gambar
        for idx, img_file in enumerate(data['image']):
            ext = img_file.filename.rsplit('.', 1)[-1].lower()
```

```

        filename = secure_filename(f'{data['name'].replace(' ', '_')}{data['finger']}{idx + 1}.{ext}')
        image_full_path = os.path.join(save_path, filename)
        img_file.save(image_full_path)
        relative_path = image_full_path.split('app/')[1][:-1].replace('\\', '/')
        image_paths.append(relative_path)
    # Simpan user ke database
    user_data = data.copy()
    del user_data['image']
    user_data['image'] = image_paths # simpan list path relatif
    create_user(user_data)
    print('✓ Gambar berhasil disimpan:', image_paths)
    return {"message": "User created successfully"}, 201
except Exception as e:
    print(f'✗ Error add_user: {e}')
    return {"error": str(e)}, 500
def update_user_finger_service(user_id, finger_id):
    try:
        print(f'[INFO] Updating finger_id for user {user_id} to {finger_id}')
        repo_update_user_finger_id(user_id, finger_id)
        return {"message": "Finger ID updated successfully"}, 200
    except Exception as e:
        return {"error": str(e)}, 500
def find_user_by_id_service(user_id):
    try:
        # Get the user and images data
        user, images = get_user_by_id_with_images(user_id)
        # Check if the user exists
        if not user:

```

```

        return {"error": "User not found"}, 404
    # Return user and images in the response
    return user, images
except Exception as e:
    # Return error message with exception details for debugging
    return {"error": f"An error occurred: {str(e)}"}, 500
def delete_user_service(user_id):
    try:
        # Ambil data user dan gambar terkait
        user, images = get_user_by_id_with_images_all(user_id)
        if not user:
            return False
        # Hapus file gambar dari sistem file
        if images: # Pastikan ada gambar terkait
            # Ambil folder dari path gambar pertama
            first_image_path = os.path.join('app', images[0].name)
            folder_path = os.path.dirname(first_image_path) #
Dapatkan path folder
        if os.path.exists(folder_path):
            shutil.rmtree(folder_path) # Hapus folder beserta isinya
            print(f"✅ Folder {folder_path} berhasil dihapus.")
        else:
            print(f"⚠️ Folder {folder_path} tidak ditemukan.")
        # Hapus finger_id dari user
        if user.finger_id:
            print(f"🔴 Menghapus finger_id {user.finger_id} dari user
{user_id}.")
            send_reset_fingerprintID_command(user.finger_id)
        # Hapus user dari database
        delete_users(user_id)
    return True

```

```

except Exception as e:
    print(f" ✗ Error saat menghapus user: {e}")
    return False

def update_user_service(user_id, data):
    try:
        # Ambil data user
        user = get_user_by_id(user_id)
        if not user:
            return {"error": "User not found"}, 400
        # Perbarui data user (name, email, dll)
        update_user(user_id, data)
        # Ambil gambar yang sudah ada dari database
        existing_images = get_images_by_user(user_id)
        existing_paths = [img.name for img in existing_images]
        # Tangani gambar baru
        new_images = data.get("image")
        if not new_images or isinstance(new_images, bool):
            return {"message": "User updated successfully (tanpa gambar)"}, 200
        if not isinstance(new_images, list):
            new_images = [new_images]
        # Siapkan folder penyimpanan berdasarkan nama + finger ID
        folder_name = f'{data["name"].replace(' ', '_')}{data["finger"]}'
        save_path = os.path.join(UPLOAD_FOLDER, folder_name)
        os.makedirs(save_path, exist_ok=True)
        # Tentukan prefix nama file dan indeks selanjutnya
        prefix = f'{data["name"].replace(' ', '_')}{data["finger"]}'
        next_index = get_next_index(existing_paths, prefix)
        for img_file in new_images:
            if not hasattr(img_file, 'filename') or not img_file.filename:
                print(" ⚠ File tidak valid, dilewati.")

```

```

        continue

        ext = img_file.filename.rsplit('.', 1)[-1].lower()
        filename = secure_filename(f'{prefix}_{next_index}.{ext}')
        image_full_path = os.path.join(save_path, filename)
        # Tentukan relative path
        if "app/" in image_full_path:
            relative_path = image_full_path.split("app/", 1)[-1].replace("\\", "/")
        else:
            relative_path = image_full_path.replace("\\", "/")
        # Skip jika path sudah ada di DB
        if relative_path in existing_paths:
            print(f'⚠️ Gambar sudah ada di database: {relative_path}')
            continue
        # Simpan file fisik
        if not os.path.exists(image_full_path):
            img_file.save(image_full_path)
            print(f'✅ Gambar disimpan: {image_full_path}')
        else:
            print(f'⚠️ File fisik sudah ada: {image_full_path}')
        # Simpan path ke database
        add_image_for_user(user.id, relative_path)
        print(f'✅ Path gambar ditambahkan ke database: {relative_path}')

    next_index += 1
    return {"message": "User updated successfully"}, 200
except Exception as e:
    print(f'❌ Error saat memperbarui user: {e}')
    return {"error": str(e)}, 500

```

```

def get_next_index(existing_paths, prefix):
    """Cari index tertinggi dari gambar yang sudah ada, lalu return
index selanjutnya."""
    max_index = 0
    pattern = re.compile(rf"^{re.escape(prefix)}_(\d+)\.$")
    for path in existing_paths:
        fname = os.path.basename(path)
        match = pattern.search(fname)
        if match:
            try:
                idx = int(match.group(1))
                if idx > max_index:
                    max_index = idx
            except ValueError:
                continue
    return max_index + 1

def delete_user_image_service(img_id):
    try:
        # Full path ke file gambar
        img = get_img_by_id(img_id)
        full_image_path = os.path.abspath(os.path.join('app',
img.name))
        if not img.name:
            return {"error": "User not found"}, 404
        # Hapus file gambar
        if os.path.exists(full_image_path):
            os.remove(full_image_path)
            delete_image_by_id(img.id)
            print(f"✓ Gambar {full_image_path} berhasil dihapus.")
    else:
        print(f"⚠ Gambar {full_image_path} tidak ditemukan.")

```

```

        return {"message": "Image deleted successfully"}, 200
    except Exception as e:
        print(f" ✗ Error saat menghapus gambar: {e}")
        return {"error": "Terjadi kesalahan saat menghapus
gambar."}, 500

```

## G. Listing Program Notifikasi WhatsApp

```

import os
import requests
from app.repositories.number_phone_repositories import get_all_number_
phone_repositories
from dotenv import load_dotenv
load_dotenv()
def notify_whatsapp_Service(msg, app=None):
    wa_server_url = os.getenv('WA_SERVER_URL')
    session_id = os.getenv('WA_SESSION_ID')
    if not wa_server_url:
        print(" ✗ WA_SERVER_URL tidak ditemukan di .env")
        return
    if not session_id:
        print(" ✗ WA_SESSION_ID tidak ditemukan di .env")
        return
    with app.app_context():
        try:
            numbers = get_all_number_phone_repositories()
            if not numbers:
                print(" ⚠ Tidak ada nomor WA yang ditemukan di database.")
                return
            for record in numbers:

```

```

    phone_number = record.number if hasattr(record, 'number') else
    record['number']

    payload = {
        "number": phone_number,
        "message": msg,
        "sessionId": session_id }

    headers = {
        "Content-Type": "application/json" }

    try:
        response = requests.post(wa_server_url, json=payload,
headers=headers)

        print(f"✉ Mengirim pesan ke {phone_number}...")
        print(f"Payload: {payload}")
        response.raise_for_status()
        res_json = response.json()
        if res_json.get("status") == "success":
            print(f"✓ Pesan berhasil dikirim ke {phone_number}")
        else:
            print(f"✗ Gagal kirim ke {phone_number}: {res_json}")
    except requests.exceptions.RequestException as err:
        print(f"✗ Error koneksi ke {phone_number}: {err}")
    except Exception as e:
        print(f"✗ Error saat mengambil nomor dari database: {e}")

```

Kode program yang ditampilkan pada tabel-tabel diatas merupakan sebagian potongan dari implementasi sistem *monitoring* keamanan rumah menggunakan *Fingerprint* dan *Face Recognition* berbasis *Machine Learning*. Untuk menghindari keterbatasan ruang penulisan, *source code* secara lengkap disediakan melalui repositori GitHub yang dapat diakses pada tautan berikut: [https://github.com/allanurin/Sistem\\_monitoring-keamanan\\_rumah.git](https://github.com/allanurin/Sistem_monitoring-keamanan_rumah.git).

## RIWAYAT HIDUP PENULIS



Data Diri Nama : Siti Allanurin

Tempat/ Tgl Lahir : Lampung, 03 Oktober 1999

Jenis Kelamin : Perempuan

Agama : Islam

Status : Belum Menikah

Nama Ayah : Doddy Fermana Setia

Nama Ibu : Haryanti

### Hobby

- Membaca buku & Novel
- Membuat Bouquet
- Nonton Film China & Korea
- Storytelling

### Riwayat Pendidikan

1. SD Negeri 03 Watuagung	2005 - 2011
2. SMP Negeri 02 Kalirejo	2011 - 2014
3. SMK Muhammadiyah 01 Kalirejo	2015 - 2018
4. Universitas Bale Bandung (S1- Teknik Informatika)	2021 - 2025

### Motto

Do what you can do, and don't do what you can't do

### Contact

- Instagram : stallanr\_
- E-mail : stallanr571@gmail.com