

# RANCANG BANGUN SISTEM KEAMANAN RUMAH BERBASIS IOT MENGGUNAKAN ALGORITMA FUZZY LOGIC MAMDANI DENGAN PEMANTAUAN REAL-TIME MELALUI APLIKASI ANDROID

*by* Yodha Nabih Rafif

---

**Submission date:** 08-Jul-2025 03:59PM (UTC+0700)

**Submission ID:** 2711872808

**File name:** Turnitin\_-\_Yodha\_Nabiha\_Rafif\_Widiyono\_Putra.docx (12.77M)

**Word count:** 18272

**Character count:** 121346

#### Abstrak

Kejahat pencurian di wilayah padat penduduk seperti Jakarta Timur mendorong kebutuhan sistem keamanan rumah yang responsif dan mudah diakses. Penelitian ini merancang dan mengembangkan sistem keamanan rumah berbasis *Internet of Things* (IoT) dengan penerapan algoritma Fuzzy Logic Mamdani, serta pemantauan secara real-time melalui aplikasi *Android*. Sistem ini mengintegrasikan sensor *Passive Infrared* (PIR) untuk deteksi gerakan, sensor magnetik untuk status pintu, kamera Raspberry Pi untuk pengawasan visual, serta buzzer sebagai alarm. Proses klasifikasi status keamanan ("Aman", "Waspada", "Bahaya") dilakukan melalui metode *fuzzy logic* berbasis input intensitas gerakan, kondisi pintu, dan waktu. Data dikirim ke aplikasi *Android* menggunakan protokol MQTT, sementara video dipantau melalui RTSP. Aplikasi *Android* dibangun dengan arsitektur MVVM dan Jetpack Compose, menyediakan fitur notifikasi, *livestream*, kontrol alarm, dan pengaturan jaringan. Berdasarkan hasil pengujian *blackbox* dan simulasi pada miniatur rumah, sistem mampu bekerja secara efektif dalam mendeteksi dan merespons ancaman keamanan. Sistem ini diharapkan dapat menjadi solusi praktis dan terjangkau dalam meningkatkan keamanan rumah di lingkungan masyarakat.

**Kata kunci:** Sistem Keamanan Rumah, Fuzzy Logic Mamdani, MQTT, RTSP, *Android*.

#### Abstract

The high rate of burglary in densely populated areas such as East Jakarta drives the need for a responsive and easily accessible home security system. This study designs and develops a home security system based on the *Internet of Things* (IoT) with the implementation of the Mamdani Fuzzy Logic algorithm and real-time monitoring through an *Android* application. The system integrates a *Passive Infrared* (PIR) sensor for motion detection, a magnetic sensor for door status, a Raspberry Pi camera for visual surveillance, and a buzzer as an alarm. Security status classification ("Safe", "Alert", "Danger") is determined using fuzzy logic based on motion intensity, door condition, and time. Data is sent to the *Android* application via the MQTT protocol, while video monitoring is conducted using RTSP. The *Android* application is built using the MVVM architecture and Jetpack Compose, featuring notifications, *livestreaming*, alarm control, and network configuration. Based on *blackbox* testing and simulations using a miniature house, the system effectively detects and responds to security threats. This system is expected to provide a practical and affordable solution to enhance home security within communities.

**Keywords:** Home Security System, Mamdani Fuzzy Logic, MQTT, RTSP, *Android*.

## BAB 1. PENDAHULUAN

### 1.1 Latar Belakang

Kejahatan pencurian merupakan salah satu permasalahan keamanan yang sering terjadi di berbagai daerah di Indonesia. Kejahatan ini tidak hanya menimbulkan kerugian materi, tetapi juga rasa tidak aman di kalangan masyarakat. Berdasarkan data dari Badan Pusat Statistik (BPS, 2023, hlm. 34), DKI Jakarta menjadi salah satu provinsi dengan angka kejadian pencurian tanpa kekerasan tertinggi di Indonesia. Pada tahun 2023, sebanyak 57,30% desa/kelurahan di provinsi ini melaporkan kasus pencurian tanpa kekerasan, yang menunjukkan bahwa ancaman terhadap keamanan masyarakat masih cukup signifikan.

DKI Jakarta sebagai ibu kota negara, memiliki tingkat kejahatan pencurian yang cukup tinggi, terutama di wilayah Jakarta Timur Berdasarkan hasil laporan dari Polda Metro Jaya (Gultom, 2024), Jakarta Timur menduduki peringkat ke-3 sebagai wilayah dengan potensi kerawanan tindak kejahatan, seperti pencurian dengan pemberatan (curat), pencurian dengan kekerasan (curas), pencurian kendaraan bermotor (curanmor), dan tawuran pada tahun 2024. Salah satu contoh kasus yang terjadi di Jakarta Timur adalah insiden pencurian brankas di sebuah rumah di Ciracas, yang dilaporkan oleh Kompas (Ramadhian & Maulana, 2024). Dalam kejadian tersebut, pelaku melakukan aksi dengan memanjat pagar rumah korban dan berhasil membawa kabur brankas yang berisi barang berharga. Kasus ini menjadi salah satu gambaran nyata mengenai ancaman keamanan di wilayah Jakarta Timur, khususnya terhadap kejahatan pencurian.

Berdasarkan hasil wawancara dengan warga di Rt 06 Rw 03 Ciracas, terungkap bahwa sebagian besar rumah di daerah ini belum dilengkapi dengan sistem keamanan yang memadai. Banyak warga yang belum menggunakan perangkat keamanan modern seperti CCTV atau alat pemantauan berbasis teknologi lainnya. Mereka umumnya masih mengandalkan pengawasan manual melalui pos ronda atau kegiatan patroli lingkungan. Dalam beberapa kasus, warga mengalami kehilangan barang berharga, seperti hewan peliharaan, barang elektronik, hingga kendaraan bermotor, akibat kurangnya pengawasan.

Dari kondisi tersebut, solusi yang dapat diusulkan adalah pengembangan sistem keamanan berbasis Internet of Things (IoT) yang dirancang untuk mendeteksi dan mencegah tindak kejahatan, khususnya pencurian. Sistem ini dirancang untuk mengintegrasikan perangkat seperti kamera, sensor, dan alarm, yang memungkinkan pemilik rumah memantau kondisi rumah secara real-time melalui aplikasi Android. Dengan adanya sistem ini, warga dapat menerima notifikasi langsung saat terdeteksi aktivitas mencurigakan, memantau lingkungan secara fleksibel dari jarak jauh, serta meningkatkan keamanan rumah secara keseluruhan. Untuk mengoptimalkan kinerja sistem keamanan berbasis IoT ini, digunakan fuzzy logic karena hasil dari algoritma tersebut dapat memberikan keluaran berupa klasifikasi tingkat ancaman, seperti "aman," "waspada," atau "bahaya," yang mudah dipahami pengguna.

Penelitian ini bertujuan untuk merancang dan mengembangkan sistem keamanan berbasis IoT dengan fitur seperti deteksi gerakan, pemantauan kamera real-time, kontrol alarm, serta pengambilan keputusan berbasis fuzzy logic. Diharapkan, solusi ini dapat

membantu mengatasi masalah keamanan di daerah seperti Ciracas, dan juga memberikan rasa aman bagi masyarakat yang terdampak.

39

### 1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah yang akan dijawab dalam penelitian ini adalah:

1. Bagaimana membangun perangkat IoT sebagai sistem keamanan rumah dengan menerapkan algoritma fuzzy logic mamdani?
2. Bagaimana membangun aplikasi Android untuk integrasi perangkat IoT yang dapat menerima notifikasi dan pemantauan real-time?

### 1.3 Batasan Masalah

Ruang lingkup penelitian ini mencakup:

1. Pengembangan sistem keamanan rumah berbasis IoT menggunakan Raspberry Pi, sensor PIR, sensor magnetik, kamera, dan buzzer.
2. Pengembangan aplikasi Android yang terhubung dengan perangkat IoT.
3. Pengambilan latar permasalahan dilakukan di lingkungan warga Rt 6 Rw 3 Kelurahan Ciracas Kecamatan Ciracas Jakarta Timur, melalui wawancara dan observasi kondisi keamanan rumah di daerah tersebut.
4. Pengujian dan simulasi sistem dilakukan menggunakan miniatur rumah sebagai representasi lingkungan rumah, untuk memastikan fungsionalitas sistem berjalan sesuai rancangan.
5. Aplikasi Android berjalan pada versi Android Nougat atau versi di atasnya.
6. Fitur yang tersedia dalam Aplikasi berupa pemantauan kamera, kontrol jaringan, status keamanan rumah, kontrol alarm dan kontrol notifikasi.
7. Perangkat IoT dapat membunyikan alarm, mengirim data status keamanan rumah, dan mengambil gambar dengan kamera.

75

### 1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Membangun sistem keamanan rumah berbasis IoT dengan menerapkan fuzzy logic Mamdani.
2. Membangun aplikasi Android yang terintegrasi dengan perangkat IoT agar pengguna dapat memantau keamanan rumah dari jarak jauh.

### 3 1.5 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah:

#### 1. Bagi Peneliti :

- a. Penelitian ini dapat menjadi referensi untuk pengembangan sistem *iot* lainnya yang fokus pada keamanan.
- b. Menyediakan solusi keamanan rumah yang lebih pintar dan efektif menggunakan teknologi IoT.

#### 2. Bagi Pengembang :

- a. Hasil penelitian ini dapat menjadi kontribusi dalam pengembangan teknologi IoT dan keamanan berbasis digital.
- b. Mengembangkan pemahaman dan pengetahuan tentang metode *Fuzzy Logic* dalam konteks sistem keamanan rumah.

#### 3. Bagi Masyarakat :

- a. Sistem ini dapat memberikan solusi keamanan rumah yang lebih efektif dan responsif.
- b. Dapat membantu memberikan peringatan dini apabila terjadi pencurian disekitar rumah
- c. Pengaplikasian fungsi metode *Fuzzy Logic* pada kehidupan nyata untuk memantau rumah.

### 78 1.6 Luaran Penelitian

Hasil yang diharapkan dari penelitian ini adalah menghasilkan aplikasi sistem keamanan rumah dengan memanfaatkan perangkat *IoT* dan mengintegrasikannya pada aplikasi Android sehingga pengguna dapat selalu memantau keamanan rumahnya.

### 6 1.7 Sistematika Penulisan

Sistematika penulisan dalam penelitian ini adalah sebagai berikut:

#### BAB I PENDAHULUAN

Bab ini akan mengulas latar belakang penelitian, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan laporan.

#### BAB II LANDASAN TEORI

Bab ini akan membahas teori pendukung yang terkait dengan judul penelitian, termasuk sistem keamanan rumah, perangkat IoT, metode extreme programming, aplikasi android, serta rangkuman penelitian terdahulu yang relevan dengan masalah penelitian.

### BAB III METODE PENELITIAN

Bab ini akan menguraikan tahapan penelitian, alat pendukung penelitian, dan jadwal penelitian dalam merancang perangkat IoT dan aplikasi android untuk sistem keamanan rumah.

### BAB IV PEMBAHASAN

Bab ini akan memberikan hasil implementasi tahapan penelitian dalam merancang perangkat IoT dan aplikasi android untuk sistem keamanan rumah.

### BAB V PENUTUP

Bab ini akan memberikan hasil evaluasi dan kesimpulan penelitian dalam merancang perangkat IoT dan aplikasi Android untuk sistem keamanan rumah.

## <sup>36</sup> BAB 2. TINJAUAN PUSTAKA

### 2.1 Sistem Keamanan Rumah

Dengan perkembangan teknologi, keamanan rumah dapat ditingkatkan dengan memanfaatkan konsep <sup>103</sup> Internet of Things (IoT). IoT memungkinkan perangkat-perangkat elektronik di rumah terhubung secara terus-menerus melalui internet, sehingga dapat dioperasikan dan dimonitor dari jarak jauh. Dengan Sistem Keamanan rumah berbasis IoT memungkinkan pengguna untuk mendapatkan notifikasi secara langsung dengan aplikasi smartphone ketika sensor mendeteksi gerakan atau aktivitas mencurigakan. Data yang dikumpulkan dari sensor, seperti gambar atau video yang diambil oleh kamera, dikirimkan ke server, dan pengguna dapat <sup>141</sup> memantau keadaan rumah secara real-time melalui aplikasi (Kumiasih et al., 2020).

#### <sup>53</sup> 2.1.1 Internet of Things

Internet of Things (IoT) adalah teknologi yang memungkinkan benda-benda di sekitar kita terhubung dengan internet, sehingga memudahkan aktivitas sehari-hari menjadi lebih efisien dan terintegrasi (Susanto et al., 2022). IoT memberikan peluang besar untuk mengoptimalkan berbagai aspek kehidupan dengan memanfaatkan sensor, RFID, dan perangkat pintar lainnya yang terhubung ke jaringan internet. Dengan IoT, pengguna dapat memantau dan mengontrol benda-benda di sekitar mereka secara real-time, kapan saja, dan di mana saja tanpa batasan jarak (Ramdani, 2020).

<sup>156</sup> IoT tidak hanya diterapkan dalam sistem keamanan, tetapi juga di berbagai bidang lain seperti pertanian, kesehatan, dan manajemen energi, yang bertujuan untuk memudahkan manusia dalam memonitor dan mengendalikan perangkat secara otomatis. Teknologi ini merupakan hasil perkembangan dari internet dan komunikasi yang memungkinkan manusia untuk berinteraksi dengan perangkat yang saling terhubung, memberikan kontrol dan akses yang lebih luas dalam kehidupan sehari-hari (Susanto et al., 2022).

#### <sup>100</sup> 2.1.2 Raspberry Pi

Raspberry Pi adalah komputer berukuran kecil seukuran kartu kredit yang dikenal sebagai komputer papan tunggal (SBC). Perangkat ini memiliki semua fungsi dasar layaknya komputer biasa dan menggunakan SoC (System-on-a-chip) berbasis ARM yang dipasang di PCB. Raspberry Pi menjalankan sistem operasi berbasis Linux dan menggunakan kartu SD untuk booting serta penyimpanan data jangka panjang (Samsinar et al., 2020).

#### <sup>111</sup> 2.1.3 Raspberry Pi Camera Module

Raspberry Pi Camera Module atau biasa disebut Raspicam adalah <sup>36</sup> kamera yang berfungsi untuk mengambil foto atau video dengan resolusi sebesar 5 megapixel (Rifandi & Sutarti, 2021). Raspicam juga mendukung night vision memiliki kemampuan untuk melihat dalam lingkungan yang gelap. Teknologi ini digunakan untuk membantu mengatasi masalah pencahayaan rendah yang sering kali menjadi

kendala dalam sistem pengawasan. Dalam sistem keamanan, kamera night vision digunakan untuk mendeteksi wajah, manusia, dan objek lain, serta diaplikasikan dalam berbagai sistem keamanan seperti kamera pengawas atau CCTV. Teknologi ini sangat bermanfaat dalam memastikan pengawasan yang lebih baik pada kondisi cahaya minim, seperti di malam hari atau dalam ruangan gelap (Rachmawati, 2021).

#### 2.1.4 Sensor Passive Infrared (PIR)

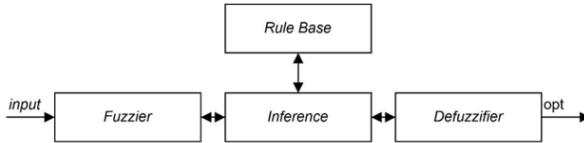
Menurut Juliansyah (2021), sensor PIR (Passive Infrared) adalah perangkat yang berfungsi untuk mendeteksi radiasi inframerah yang dipancarkan oleh objek di sekitarnya. Sensor ini bersifat pasif, yang berarti hanya berperan sebagai penerima radiasi inframerah tanpa memancarkannya. Sensor PIR dirancang khusus untuk mendeteksi radiasi inframerah dengan panjang gelombang antara 8-14 mikrometer, yang sesuai dengan rentang panjang gelombang radiasi yang dihasilkan oleh tubuh manusia, yaitu sekitar 9-10 mikrometer. Dengan adanya sensor PIR, ketika ada pergerakan manusia yang mendekati sensor, sensor tersebut akan mendeteksi keberadaan manusia tersebut. Sensor mampu mendeteksi pergerakan manusia dengan jarak maksimal antara sensor PIR dan objek sekitar  $\pm 5$  meter. Selain menggunakan Arduino, sistem yang memanfaatkan sensor PIR juga dapat dikembangkan dengan Raspberry Pi (Rahayu, 2024).

#### 2.1.5 Sensor Magnetik

Sensor magnetik adalah sensor yang digunakan untuk mendeteksi terbuka atau tertutupnya pintu. Sensor ini merupakan salah satu jenis sensor yang sangat sederhana karena hanya terdiri dari dua plat yang berdekatan (Hermawan, 2024). Magnetik switch dapat merespon medan magnet di area sekitarnya. Magnetic switch mirip seperti sensor limit switch yang diberikan tambahan plat logam sehingga dapat merespon magnet. Magnetik switch ini biasa digunakan dalam pengamanan pintu dan jendela (Gulo et al., 2022).

## 2.2 Fuzzy Logic

Logika fuzzy pertama kali diperkenalkan oleh Prof Lutfi A. Zadeh, seorang peneliti di Universitas California pada tahun 1965. Menurut Professor Zadeh, logika biner yang hanya mengandalkan nilai benar atau salah tidak dapat sepenuhnya mewakili cara berpikir manusia. Oleh sebab itu, dikembangkanlah logika fuzzy yang mampu menggambarkan berbagai keadaan secara lebih fleksibel dan mendekati cara manusia berpikir. Perbedaan utama antara logika fuzzy dan logika tegas terletak pada keanggotaan elemen dalam suatu himpunan. Pada logika tegas, elemen hanya terdiri dari nilai 1 yang berarti benar dan nilai 0 yang berarti salah. Sebaliknya, dalam logika fuzzy, keanggotaan elemen berada dalam rentang nilai antara 0 dan 1 (Supriyadi et al., 2020).



Gambar 2.1 Struktur dasar fuzzy interface system

Pada Gambar 2.1 menunjukkan struktur dasar dari Fuzzy Interface System. Sistem ini bekerja berdasarkan konsep himpunan fuzzy dan logika fuzzy yang digunakan untuk mengambil kesimpulan atau keputusan. Proses ini melibatkan tiga tahap utama: fuzzifikasi, inferensi berdasarkan aturan yang ada, dan defuzzifikasi untuk menghasilkan output yang jelas dari data fuzzy. Menurut Setyawan et al. (2022), Logika fuzzy memiliki beberapa jenis metode seperti Metode Tsukamoto, Metode Mamdani, dan Metode Sugeno. Metode Tsukamoto dalam logika fuzzy menggunakan aturan berbentuk IF-THEN yang diwakili oleh himpunan fuzzy dan fungsi keanggotaan yang bersifat monoton. Sementara itu, metode Mamdani menggunakan himpunan fuzzy baik untuk input maupun output. Metode Sugeno hampir mirip dengan penalaran Mamdani, namun yang membedakannya adalah pada bagian output.

Perbedaan utama antara kedua metode ini terletak pada output yang dihasilkan, serta proses komposisi aturan dan defuzzifikasinya. Pada metode Sugeno, output yang dihasilkan berupa fungsi linear atau konstanta. Sebaliknya, output pada metode fuzzy. Mamdani berupa nilai dalam domain himpunan fuzzy yang diklasifikasikan ke dalam komponen linguistik (Septima, 2023). Pada penelitian ini, metode fuzzy logic yang dipilih adalah Metode Mamdani. Pemilihan metode ini didasarkan pada kebutuhan sistem keamanan rumah berbasis IoT untuk menghasilkan output dalam bentuk linguistik, seperti “aman,” “waspada,” atau “bahaya,” yang mempermudah interpretasi tingkat ancaman oleh pengguna. Dibandingkan dengan metode Sugeno yang menghasilkan output berupa nilai konstan atau fungsi linear, metode Mamdani lebih unggul dalam memberikan representasi linguistik yang intuitif. Selain itu, metode ini juga lebih fleksibel dibandingkan metode Tsukamoto, yang cenderung menghasilkan output crisp tanpa interpretasi linguistik.

### 2.2.1 Fuzzification

Tahap pertama dalam logika fuzzy adalah fuzzification (fuzzifikasi), yaitu proses mengubah nilai crisp dari input sensor menjadi nilai fuzzy berdasarkan fungsi keanggotaan yang telah ditentukan. Pada penelitian ini, input yang digunakan intensitas sensor PIR (Passive Infrared Sensor) dan status pintu. Untuk status pintu nilai akan langsung diubah menjadi “terbuka” dan “tertutup”. Sedangkan dalam intensitas sensor

PIR akan diklasifikasikan ke dalam tiga kategori linguistik yaitu “rendah,” “sedang,” dan “tinggi”. Setiap kategori memiliki fungsi keanggotaan berbentuk segitiga atau trapesium yang diadaptasi dari jurnal Setyawan et al. (2022), dengan penyesuaian rentang nilai berdasarkan kebutuhan sistem. Berikut adalah fungsi keanggotaan yang digunakan:

### 1. Fungsi Keanggotaan Rendah

Fungsi keanggotaan untuk kategori “rendah” menggunakan Menggunakan kurva trapesium menurun dengan rumus (2.1).

$$\mu_{rendah}(x) = \begin{cases} 1, & x \leq A \\ \frac{B-x}{B-A}, & A < x \leq B \\ 0, & x > B \end{cases} \quad (2.1)$$

Dimana:

- 1) A adalah nilai batas awal intensitas PIR yang dianggap rendah.
  - 2) B adalah nilai batas akhir intensitas PIR yang dianggap rendah.
- ### 2. Fungsi Keanggotaan Sedang

Fungsi keanggotaan untuk kategori “sedang” menggunakan kurva segitiga dengan rumus (2.2).

$$\mu_{sedang}(x) = \begin{cases} 0, & x \leq A \text{ atau } x \geq C \\ \frac{x-A}{B-A}, & A \leq x \leq B \\ \frac{C-x}{C-B}, & B \leq x \leq C \\ 1, & x = B \end{cases} \quad (2.2)$$

Dimana:

- 1) A adalah nilai batas awal transisi ke “sedang.”
  - 2) B adalah nilai puncak dari fungsi segitiga
  - 3) C adalah nilai batas akhir transisi dari “sedang.”
- ### 3. Fungsi Keanggotaan Tinggi

Fungsi keanggotaan untuk kategori “tinggi” Menggunakan kurva trapesium menaik dengan rumus (2.3).

$$\mu_{tinggi}(x) = \begin{cases} 0, & x \leq A \\ \frac{x-A}{B-A}, & A \leq x \leq B \\ 1, & x \geq B \end{cases} \quad (2.3)$$

Dimana:

- 1) A adalah nilai batas awal intensitas PIR yang dianggap “tinggi.”
- 2) B adalah nilai batas akhir intensitas PIR yang dianggap “tinggi.”

### 2.2.2 Inference

Proses Selanjutnya dari logika fuzzy yaitu tahap inferensi, yaitu proses mengambil keputusan dengan aturan fuzzy dengan format IF- THEN yang mencakup kombinasi antara intensitas PIR dan status pintu. Proses inferensi dilakukan menggunakan metode Mamdani Max-Min. Nilai Min dan Max diperoleh dari aturan yang digunakan, di mana nilai tertinggi disebut sebagai Max dan nilai terendah disebut sebagai Min. Nilai tersebut dapat dilihat dari hasil output berupa persentase atau aturan yang terbentuk berdasarkan input yang diberikan (Setyawan et al., 2022).

### 2.2.3 Defuzzification

Tahap terakhir dari logika fuzzy yaitu defuzzification, yaitu proses mengubah nilai fuzzy yang dihasilkan dari tahap inferensi menjadi nilai crisp yang dapat digunakan sebagai output akhir. Metode defuzzifikasi akan dilakukan dengan menggunakan metode Centroid. Dalam metode ini, solusi crisp diperoleh dengan menentukan titik pusat dari area fuzzy (Setyawan et al., 2022).

## 2.3 Aplikasi Android

Android merupakan sistem operasi berbasis linux yang pada perangkat mobile yang didalamnya mencakup komponen seperti sistem operasi, middleware, dan aplikasi. Sistem ini menyediakan platform terbuka bagi para pengembang untuk membuat aplikasi ciptaan mereka sendiri. Seiring berjalannya waktu, Android terus mengembangkan versinya demi menawarkan fitur-fitur terbaik (Siswandi & Muhidin, 2022). Berdasarkan penelitian yang dilakukan oleh Aditya et al. (2022), aplikasi android mempunyai keunggulan dikarenakan bersifat mobile sehingga dapat digunakan dalam mobilitas yang tinggi seperti telepon seluler. Android juga mendukung koneksi internet sehingga aplikasi yang memerlukan akses internet dapat berjalan dengan baik. Diharapkan aplikasi tersebut dapat mempercepat proses yang biasanya memakan waktu cukup lama atau lebih dari satu hari.

### 2.3.1 Android Studio

Android Studio merupakan Integrated Development Environment (IDE) yang digunakan untuk pengembangan aplikasi Android. Android Studio dikembangkan oleh Google yang merupakan evolusi dari Eclipse IDE dan dibuat berdasarkan IntelliJ IDEA, sebuah IDE Java yang populer (Siswandi & Muhidin, 2022). Selain berfungsi sebagai editor kode berbasis IntelliJ dan alat pengembangan yang efisien, Android Studio juga menyediakan berbagai fitur tambahan yang dirancang untuk meningkatkan produktivitas dalam pengembangan aplikasi Android (Baco, 2021).

### 2.3.2 Kotlin<sup>46</sup>

Kotlin adalah bahasa pemrograman modern yang dikembangkan oleh perusahaan JetBrains dan berjalan di atas Java Virtual Machine (JVM) (Diantoni et al., 2024). Kotlin merupakan pengembangan dari bahasa Java yang sebelumnya sudah populer. Kotlin menawarkan fitur-fitur bahasa modern yang lebih unggul dibandingkan dengan Java (AryaRafa et al., 2024).

### 2.3.3 Jetpack Compose<sup>7</sup>

*Jetpack Compose* adalah salah satu *library* dari Android *Jetpack* yang digunakan untuk membangun antarmuka aplikasi Android native. *Jetpack Compose* memudahkan pengembangan antarmuka aplikasi Android karena implementasinya lebih sederhana dibandingkan dengan pendekatan berbasis XML (Putra & Tolle, 2023). *Jetpack Compose* memungkinkan pengembang untuk mendefinisikan antarmuka pengguna (UI) secara deklaratif berdasarkan kondisi aplikasi saat ini. *Framework* ini menawarkan manajemen *state* yang sederhana dan menggunakan prinsip reaktivitas untuk memperbarui UI secara otomatis ketika data berubah. *Jetpack Compose* menggunakan prinsip pendekatan *single source of truth*, yang memungkinkan UI untuk mengambil data langsung dari model data atau *state* yang disimpan, sehingga memudahkan pengembangan UI yang sederhana dan terstruktur (Arrizal & Rizal, 2024).

### 2.3.4 MVVM Architecture

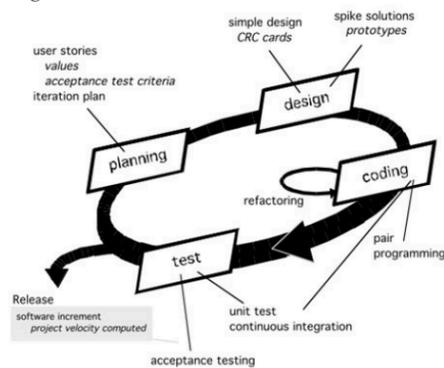
Pola arsitektur *MVVM* adalah salah satu bagian dari *Clean Architecture*. *Clean Architecture* adalah pendekatan desain perangkat lunak yang memisahkan aturan bisnis inti, logika aplikasi, dan detail teknis implementasi (Diantoni et al., 2024). *MVVM* adalah salah satu arsitektur pengembangan aplikasi yang menekankan pemisahan antara kode terkait dengan logika bisnis dan tampilan aplikasi. Arsitektur ini memiliki tiga lapisan utama yaitu *Model*, *View* dan *View Model*. *Model* adalah lapisan yang merepresentasikan data dalam aplikasi. *View* merupakan antarmuka aplikasi yang mengatur bagaimana informasi ditampilkan di layar, dan biasanya terdiri dari kelas *Activity*, *Fragment*, atau *composable function*. Sementara itu, *View Model* berperan dalam berinteraksi dengan *Model* untuk meneruskan data ke *View* sehingga bisa ditampilkan kepada pengguna aplikasi (Putra & Tolle, 2023). Metode *MVVM* dipilih karena memiliki kemampuan untuk memisahkan logika bisnis dari tampilan pengguna. Selain itu, *MVVM* menawarkan keuntungan bagi pengguna, seperti peningkatan performa pada aplikasi berskala besar jika dibandingkan dengan program yang tidak menggunakan arsitektur *MVVM* dalam pengembangannya (Zoromi et al., 2024).

## 2.4 <sup>129</sup> Real Time Streaming Protocol (RTSP)

*RTSP* adalah protokol khusus untuk menghubungkan antara *IP Camera* dengan *client*. *RTSP* dirancang untuk memungkinkan komunikasi antara *server streaming* dengan *media player*. *RTSP* memiliki keunggulan untuk menyediakan koneksi jarak jauh, sehingga dapat mempermudah kinerja ketika ingin akses *IP Camera* walaupun sedang berada di luar jaringan (Utami et al., 2021). Protokol ini akan digunakan untuk mengakses kamera yang terhubung pada *raspberry* secara *realtime*.

## 2.5 <sup>31</sup> Extreme Programming

*Extreme Programming (XP)* adalah metodologi pengembangan perangkat lunak yang bertujuan untuk meningkatkan kualitas perangkat lunak dengan cepat beradaptasi terhadap perubahan dan kebutuhan pelanggan. Metodologi ini dirancang untuk meningkatkan produktivitas serta memungkinkan pengenalan pos pemeriksaan, sehingga persyaratan baru dari pelanggan dapat diadopsi dengan mudah (Ariyanti et al., 2020). Metode *Extreme Programming* digunakan dengan alasan sistem yang akan dikembangkan membutuhkan proses yang cepat, mencakup fase *planning*, fase *design*, fase *coding*, dan fase *testing* (Nurkholis et al., 2021). Berikut ini merupakan gambar 2.2 yang menampilkan kerangka kerja dari *Extreme Programming*.



Gambar 2.2 Alur pada <sup>2</sup>metode extreme programming (Sulistiani et al., 2021)

### 2.5.1 Planning

<sup>10</sup>Tahap perencanaan dimulai dari pengumpulan kebutuhan untuk memahami konteks bisnis dari sebuah aplikasi. Selain itu pada tahap ini juga mendefinisikan *output* yang akan dihasilkan, fitur yang dimiliki oleh aplikasi dan fungsi dari aplikasi

yang dikembangkan (Sulistiani et al., 2021). Pada tahapan ini juga dilakukan identifikasi masalah, menganalisa kebutuhan, dan menentukan jadwal pelaksanaan dalam pembuatan sistem (Coyanda et al., 2022).

### 2.5.2 Design

Selanjutnya yaitu tahap *design* yang dimana dalam tahapan ini dilakukan perancangan kegiatan pemodelan yang bermula dari pemodelan sistem, pemodelan arsitektur hingga pemodelan basis data (Coyanda et al., 2022). Tahap ini menekankan pada desain aplikasi yang sederhana. Untuk merancang aplikasi, dapat digunakan kartu *Class- Responsibility-Collaborator (CRC)*, yang berfungsi untuk mengidentifikasi dan mengatur kelas-kelas dalam aplikasi berbasis *object-oriented* yang akan dikembangkan (Sulistiani et al., 2021).

### 2.5.3 Coding

Tahapan selanjutnya setelah *design* yaitu *coding*. Pada tahapan inilah kegiatan penerapan pemodelan dari hasil *design* sebelumnya dibuat kedalam bentuk *user interface* dengan menggunakan bahasa pemrograman (Coyanda et al., 2022).

### 2.5.4 Testing

Setelah tahap pengkodean selesai, kemudian dilakukan tahapan *testing* sistem yang bertujuan untuk mengetahui kesalahan-kesalahan yang terjadi saat aplikasi sedang dijalankan dan mencari tahu apakah sistem yang telah dibangun sudah sesuai dengan kebutuhan pengguna (Coyanda et al., 2022).

## 2.6 Message Queue Telemetry Transport (MQTT)

*Message Queue Telemetry Transport (MQTT)* adalah protokol pertukaran pesan berbasis model *publish/subscribe* yang dirancang khusus untuk perangkat dengan kemampuan terbatas, jaringan dengan bandwidth kecil, latensi tinggi, atau yang tidak andal. Protokol ini memiliki keunggulan dalam mentransfer data dengan ukuran paket yang kecil, sehingga sangat berguna untuk mengembangkan teknologi *remote* pada perangkat dengan keterbatasan. Prinsip desain utama *MQTT* adalah meminimalkan penggunaan *bandwidth* dan *resource* perangkat, namun tetap memastikan keandalan dan memberikan beberapa tingkat jaminan pengiriman pesan. Inilah yang menjadikan *MQTT* sangat ideal untuk komunikasi *machine-to-machine (M2M)*, *Internet of Things*, serta aplikasi *mobile* yang memiliki keterbatasan *bandwidth* dan daya baterai (Saputra et al., 2020).

## 2.7 Blackbox Testing

*Blackbox Testing* adalah metode pengujian perangkat lunak yang fokus pada analisis fungsi dari aplikasi tanpa memeriksa struktur internal atau kinerja aplikasi tersebut. Metode

ini dapat diterapkan pada berbagai tingkat pengujian perangkat lunak, termasuk pengujian unit, integrasi, sistem, dan penerimaan (Fahrezi et al., 2022).

## 2.8 Whitebox Testing

Pengujian *White Box* adalah metode pengujian perangkat lunak yang memanfaatkan pemahaman terhadap struktur kontrol program pada level komponen untuk merancang skenario uji. Metode ini mencakup berbagai teknik seperti *Data Flow Testing*, *Control Flow Testing*, *Basic Path Testing*, dan *Loop Testing*. Dalam penerapannya, penguji perlu memahami detail kode sumber yang akan diuji. *White Box Testing* bertujuan untuk mengidentifikasi kesalahan logika atau implementasi dalam kode program, dan dapat diterapkan pada tingkat unit, integrasi, maupun sistem secara keseluruhan (Londjo, 2021).

## 2.9 Penelitian Terdahulu

Tabel 2.1 Penelitian Terdahulu

No	Judul	Parameter	Detail Parameter
1	Rancang Bangun Sistem Keamanan Rumah Dengan Metode <i>Fuzzy Logic</i> Menggunakan Mikrokontroler Berbasis <i>Android</i>	Jurnal Publikasi	JEJARING (Jurnal Teknologi dan Manajemen Informatika)
		Tahun	2020
		Penulis	Sugeng Supriyadi, Nunu Nugraha, Erwin Ligar Nugraha
		Metode	<i>RUP (Rational Unified Process)</i>
	Kelebihan	<ol style="list-style-type: none"> <li>1. Peningkatan Keamanan dengan <i>Fuzzy Logic</i>: Penggunaan metode <i>Fuzzy Logic</i> memberikan fleksibilitas dalam mendeteksi berbagai kondisi keamanan rumah, sehingga mampu membuat keputusan lebih baik dibandingkan logika biner sederhana.</li> <li>2. Pengendalian Melalui <i>Android</i>: Dengan kontrol melalui aplikasi <i>Android</i>, pengguna dapat memantau dan mengendalikan sistem keamanan rumah dengan lebih mudah dan real-time.</li> <li>3. Efisiensi Daya: Sistem yang menggunakan mikrokontroler efisien dalam penggunaan daya, sehingga cocok digunakan untuk</li> </ol>	

		perangkat IoT yang bekerja secara terus-menerus.						
	Kekurangan	<ol style="list-style-type: none"> <li>1. Kompleksitas Implementasi Fuzzy Logic: Implementasi Fuzzy Logic memerlukan pengaturan yang kompleks dan bisa sulit untuk di-debug ketika terjadi kesalahan.</li> <li>2. Ketergantungan pada Mikrokontroler: Sistem berbasis mikrokontroler memiliki keterbatasan dalam hal komputasi dan kapasitas memori, yang dapat mempengaruhi kinerja saat menangani data lebih besar atau fitur tambahan.</li> <li>3. Keterbatasan Pengujian di Lingkungan Nyata: Pengujian pada penelitian ini lebih banyak dilakukan di lingkungan simulasi, sehingga efektivitasnya di dunia nyata, khususnya dalam menghadapi ancaman keamanan yang sebenarnya, masih perlu dieksplorasi lebih lanjut.</li> </ol>						
	<sup>37</sup> Hasil dan Kesimpulan	<p>Penelitian ini berhasil mengembangkan sistem keamanan rumah berbasis IoT yang menggunakan Fuzzy Logic untuk pengambilan keputusan yang lebih fleksibel. Sistem ini dapat dioperasikan melalui aplikasi Android dan terbukti efektif dalam mendeteksi kondisi keamanan yang bervariasi. Namun, sistem ini masih memiliki tantangan dalam hal skalabilitas dan implementasi di dunia nyata yang lebih kompleks. Ke depan, sistem perlu disempurnakan dengan pengujian di lingkungan nyata dan peningkatan kemampuan mikrokontroler untuk menangani lebih banyak fitur.</p>						
2	<sup>60</sup> Aplikasi Sistem Keamanan Rumah Berbasis Internet Of Things Dengan Menggunakan Raspbery Pi	<table border="1"> <tr> <td>Jurnal Publikasi</td> <td>SIGMA (Jurnal Teknologi Pelita Bangsa)</td> </tr> <tr> <td>Tahun</td> <td>2021</td> </tr> <tr> <td>Penulis</td> <td>Muhammad Makmun Effendi, Hengki Anthony Juliyanto</td> </tr> </table>	Jurnal Publikasi	SIGMA (Jurnal Teknologi Pelita Bangsa)	Tahun	2021	Penulis	Muhammad Makmun Effendi, Hengki Anthony Juliyanto
Jurnal Publikasi	SIGMA (Jurnal Teknologi Pelita Bangsa)							
Tahun	2021							
Penulis	Muhammad Makmun Effendi, Hengki Anthony Juliyanto							

	Metode	Waterfall
Kelebihan	<p>1. Penggunaan IoT dan Telegram Bot: Sistem memanfaatkan teknologi IoT dan Bot Telegram untuk memberikan notifikasi keamanan secara real-time, sehingga memungkinkan pemantauan rumah dari jarak jauh dengan cepat dan mudah.</p> <p>2. Kontrol yang Komprehensif: Sistem tidak hanya mendeteksi pergerakan dengan sensor PIR, tetapi juga dapat memantau bukaan pintu dengan magnetic switch door, sehingga memberikan cakupan pengamanan yang lebih luas.</p>	
Kekurangan	<p>1. Keterbatasan Perangkat: Raspberry Pi yang digunakan dalam sistem ini memiliki keterbatasan dalam hal komputasi dan kapasitas penyimpanan, yang mungkin tidak optimal jika digunakan untuk menangani banyak sensor atau kamera dalam waktu bersamaan.</p> <p>2. Belum Ada Integrasi dengan Aplikasi Android: Penelitian ini hanya menyediakan akses pemantauan melalui browser dan belum mendukung aplikasi mobile berbasis Android, yang mungkin lebih nyaman dan praktis untuk pengguna.</p>	
Hasil dan Kesimpulan	<p>Penelitian ini berhasil mengembangkan sistem keamanan rumah berbasis IoT yang menggunakan Raspberry Pi, sensor PIR, dan Bot Telegram untuk memberikan notifikasi dan streaming video real-time. Sistem ini memberikan solusi pemantauan rumah jarak jauh yang efektif dan dapat diakses melalui browser. Namun, penelitian ini masih memiliki ruang untuk pengembangan lebih lanjut, terutama dalam integrasi ke aplikasi Android dan penambahan kemampuan pengolahan citra untuk identifikasi wajah.</p>	
3	Rancang Bangun Sistem	Jurnal Publikasi JURASIK (Jurnal Riset

<p>67</p> <p>Keamanan Rumah Berbasis <i>Internet of Things (Iot)</i> Dengan Menggunakan Sensor PIR dan Esp32- Cam</p>		<p>Sistem Informasi Dan Teknik Informatika)</p> <p>Tahun 2020</p> <p>Penulis Stevenius Rumere, Evanita Veronica Manullang</p> <p>Metode <i>Research and development</i></p>
<p>Kelebihan</p>		<p>1. Integrasi ESP32-Cam dan PIR: Sistem mampu mendeteksi pergerakan dengan sensor PIR dan menangkap gambar serta video menggunakan ESP32-Cam. Ini memungkinkan pemilik rumah memantau kondisi secara visual.</p> <p>2. Notifikasi Telegram: Menggunakan bot Telegram untuk mengirim notifikasi real-time, yang membuat sistem</p>
<p>Kekurangan</p>		<p>1. Penumpukan Data Gambar dan Video: Setiap kali ada pergerakan, sistem secara otomatis mengirim gambar dan video, yang berpotensi menyebabkan penumpukan data dan penggunaan <i>bandwidth</i> yang tinggi.</p> <p>2. Ketergantungan pada Telegram: Sistem membutuhkan Telegram untuk pengiriman notifikasi, tanpa aplikasi khusus yang memungkinkan kontrol dan pengaturan lebih fleksibel.</p>
<p>37</p> <p>Hasil dan Kesimpulan</p>		<p>Penelitian ini berhasil merancang sistem keamanan rumah berbasis IoT yang mengintegrasikan sensor PIR dan ESP32-Cam dengan bot Telegram untuk notifikasi <i>real-time</i>. Sistem berfungsi dengan baik untuk mendeteksi pergerakan, mengirimkan gambar dan video, serta memungkinkan pengguna mengontrol alarm dan sengatan listrik dari jarak jauh. Namun, diperlukan pengembangan lebih lanjut untuk mengurangi penumpukan data dan memperbaiki deteksi objek yang lebih efisien, serta menambahkan analisis pergerakan untuk membedakan objek mencurigakan.</p>
<p>4</p> <p><i>Home Surveillance and</i></p>	<p>Jurnal Publikasi</p>	<p><i>Indian Journal of Artificial</i></p>

	<p>77 Alert System using Raspberry Pi Zero W and GSM Modem with MQTT Protocol</p>		<p>Intelligence and Neural Networking (IJAINN)</p>
		Tahun	2023
		Penulis	Banchigize Bazezew Mekecha
		Metode	Research and development
	Kelebihan	<ol style="list-style-type: none"> <li>1. Menggunakan berbagai sensor untuk deteksi ancaman yang lebih komprehensif.</li> <li>2. Sistem mendukung pengiriman data dalam kondisi bandwidth rendah menggunakan MQTT.</li> <li>3. Peringatan real-time melalui berbagai media (email, SMS, telepon).</li> </ol>	
	Kekurangan	<ol style="list-style-type: none"> <li>1. Tidak mendukung fitur kecerdasan buatan untuk analisis data.</li> <li>2. Deteksi bergantung pada sensitivitas sensor, yang mungkin memberikan alarm palsu.</li> </ol>	
	Hasil dan Kesimpulan	<p>Sistem yang dikembangkan berhasil diimplementasikan dan menunjukkan efisiensi dalam mendeteksi berbagai ancaman, seperti intrusi, kebakaran, kebocoran gas, dan kebisingan abnormal. Pemilik rumah dapat menerima notifikasi secara real-time melalui berbagai media, termasuk aplikasi MQTT, email, SMS, atau panggilan telepon, sehingga memberikan respons yang cepat terhadap potensi bahaya. Dengan biaya rendah dan fitur yang efektif, sistem ini dinilai cocok untuk diterapkan di berbagai lokasi, seperti rumah, kantor, atau tempat berharga lainnya. Kesimpulannya, Raspberry Pi terbukti menjadi platform yang andal untuk membangun sistem keamanan rumah berbasis IoT, menawarkan solusi modern yang efisien dan mudah diakses.</p>	
5	<p>102 Sistem Pendeteksi Keamanan Ruangan "Smart Security" Dengan Metode Fuzzy Logic Menggunakan Sensor</p>	Jurnal Publikasi	Jurnal Sistem Komputer dan Informatika (JSON)
		Tahun	2022
		Penulis	Rima Tamara Aldisa, Sechan Alfarisi

153 PIR Berbasis Internet of Things (IoT)	Metode	<i>Research and development</i>
Kelebihan		<ol style="list-style-type: none"> <li>1. Efisien: Sistem mampu mendeteksi keberadaan orang dalam jarak 1 cm hingga 2 meter secara otomatis.</li> <li>2. Notifikasi Cepat: Menggunakan Telegram untuk pengiriman pesan, memungkinkan pemilik mengetahui ancaman secara real-time.</li> <li>3. Fleksibel: Dapat diaplikasikan di berbagai tempat seperti laboratorium atau rumah.</li> </ol>
Kekurangan		<ol style="list-style-type: none"> <li>1. Ketergantungan pada Telegram: Sistem hanya menggunakan Telegram untuk pengiriman notifikasi, tanpa aplikasi khusus yang memungkinkan kontrol dan pengaturan lebih fleksibel.</li> <li>2. Tidak Ada Kamera Pemantau: Sistem tidak dilengkapi dengan kamera untuk mengambil gambar atau livestream, sehingga hanya mengandalkan sensor PIR dan alarm tanpa visualisasi.</li> <li>3. Alarm Tidak Fleksibel: Alarm berbunyi secara otomatis setiap kali ada deteksi, tanpa opsi pengaturan jadwal.</li> </ol>
Hasil dan Kesimpulan		<p>Hasil dari sistem yang telah dirancang dan diuji pada penelitian ini, mampu mendeteksi keberadaan orang dalam jarak 1 cm hingga 2 meter. Kemudian, sensor akan memicu alarm dan mengirimkan notifikasi ke Telegram jika ada orang yang terdeteksi. Sistem ini dinilai sangat membantu dalam meningkatkan keamanan ruangan dengan cara yang efisien, terjangkau, dan mudah dioperasikan. Penerapan logika fuzzy pada penelitian ini digunakan untuk membedakan jarak menjadi tiga kategori: dekat, cukup, dan jauh.</p>

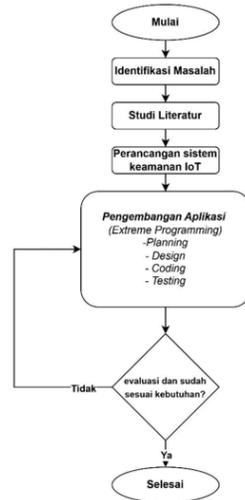
Berdasarkan pada tabel 2.1 penelitian terdahulu, dapat disimpulkan bahwa hasil dari implementasi sistem keamanan berbasis IoT dengan menerapkan logika fuzzy telah

menunjukkan hasil yang positif dalam meningkatkan efektivitas pengawasan rumah. Namun, sebagian besar penelitian memiliki keterbatasan dalam integrasi perangkat keras dan aplikasi yang akan memberikan kemudahan bagi pengguna. Penelitian yang dilakukan saat ini berupaya untuk melengkapi kekurangan dalam penelitian terdahulu dengan mengintegrasikan sensor PIR, sensor magnetik, dan kamera Raspberry Pi ke dalam sistem berbasis fuzzy Mamdani yang menghasilkan output linguistik berupa “aman,” “waspada,” atau “bahaya.” Sistem ini memanfaatkan protokol MQTT dan RTSP untuk <sup>149</sup>memberikan notifikasi real-time serta streaming video langsung, yang belum sepenuhnya diimplementasikan pada penelitian sebelumnya. Penelitian ini diharapkan dapat memberikan solusi yang lebih komprehensif dan fleksibel dalam menjaga keamanan rumah melalui pemantauan jarak jauh secara real-time.

## BAB 3. METODE PENELITIAN

### 3.1 Alur Penelitian

Alur penelitian disusun dengan tujuan agar pada proses penelitian dapat berjalan secara sistematis dan terstruktur. Pada alur tersebut terdapat beberapa langkah yang digambarkan dalam Gambar 3.1 berikut ini.



Gambar 3.1 Alur penelitian

### 3.2 Identifikasi Masalah

Dalam penelitian ini, peneliti berupaya untuk mencari topik masalah yang relevan dengan identifikasi sistem keamanan rumah. Untuk mengidentifikasi masalah tersebut, peneliti mengambil data dari Statistik Kriminal 2023 yang dirilis oleh Badan Pusat Statistik (BPS). Data tersebut mengungkapkan bahwa DKI Jakarta menjadi provinsi dengan angka kejadian pencurian tanpa kekerasan tertinggi, mencapai 57,30% dari total desa/kelurahan. Kemudian, peneliti melakukan wawancara dengan lima warga di daerah Ciracas, Jakarta Timur. Melalui wawancara ini, peneliti menggali informasi mengenai permasalahan keamanan rumah yang terjadi di kawasan tersebut dan bagaimana masyarakat di kawasan tersebut menjaga keamanan. Dari hasil wawancara ini, didapatkan rumusan masalah yang akan dipecahkan yaitu bagaimana Merancang dan membangun sistem keamanan rumah berbasis IoT dengan menerapkan fuzzy logic yang terintegrasi dengan aplikasi Android agar pengguna dapat memantau keamanan rumah dari jarak jauh. Penelitian ini bertujuan untuk membantu para pemilik rumah agar dapat menjaga keamanan rumahnya dari ancaman pencurian.

### 3.3 <sup>16</sup> Studi Literatur

Pada tahap ini, peneliti melakukan pengumpulan referensi terkait teknologi yang akan digunakan dalam pengembangan sistem keamanan rumah meliputi sensor PIR, sensor magnetik, kamera, Raspberry Pi, dan aplikasi Android. Penelitian literatur juga mencakup analisis metode dan solusi yang telah diimplementasikan sebelumnya untuk keamanan rumah berbasis IoT.

### 3.4 Perancangan Sistem Keamanan IoT

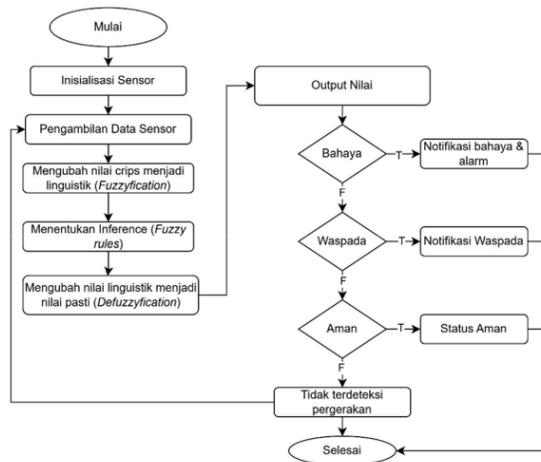
Perancangan sistem keamanan IoT difokuskan pada pengembangan perangkat keras dan alur kerja sistem fuzzy logic yang berfungsi sebagai pengolah data sensor untuk mendeteksi ancaman. Sistem ini mengintegrasikan beberapa sensor dan kamera sebagai komponen utama dalam mendeteksi dan menangkap gambar dari lingkungan sekitar. Perancangan sistem keamanan ini meliputi pemilihan sensor yang sesuai, pengaturan alur fuzzy logic untuk pengambilan keputusan, serta arsitektur keseluruhan sistem yang menghubungkan perangkat keras dengan aplikasi.

#### 3.4.1 Pemilihan Sensor

Pemilihan sensor disesuaikan dengan kebutuhan akan keamanan dan pemantauan lingkungan rumah dengan parameter utama yaitu deteksi keberadaan manusia dan status pintu atau jendela. Berikut adalah sensor-sensor yang digunakan dalam sistem ini:

1. Sensor Passive Infrared (PIR): berfungsi untuk mendeteksi gerakan manusia berdasarkan radiasi inframerah yang dipancarkan oleh tubuh. Ketika ada pergerakan dalam jangkauan deteksi sensor PIR, sistem akan menerima sinyal bahwa ada aktivitas di sekitar area tersebut.
2. Sensor Magnetik: berfungsi untuk mendeteksi kondisi terbuka atau tertutup pada pintu atau jendela. Sensor ini bekerja berdasarkan medan magnet. Jika pintu atau jendela berada dalam posisi tertutup, magnet pada sensor akan aktif. Jika terjadi perubahan kondisi sensor akan mendeteksi perubahan tersebut dan memberikan sinyal kepada sistem.
3. Raspberry Pi Camera Module: berfungsi untuk menangkap gambar dan melakukan streaming video dari area yang diawasi. Kamera ini akan berfungsi secara real-time untuk memantau lingkungan rumah. Kamera ini mendukung fungsi pemantauan visual yang lebih detail dan memungkinkan pengguna untuk melihat lingkungan secara langsung melalui aplikasi.

### 3.4.2 Alur Fuzzy Logic



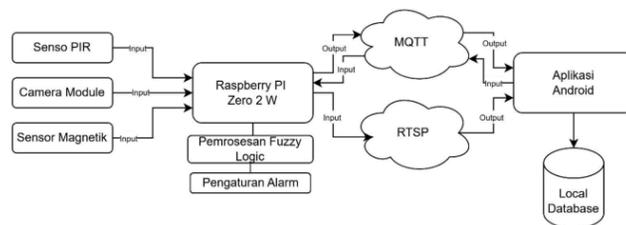
Gambar 3.2 Kerangka kerja fuzzy logic

Sistem fuzzy logic berperan penting dalam pengambilan keputusan berdasarkan data dari sensor. Saat perangkat sedang berjalan sensor mulai menjalankan tugas sesuai fungsinya. Sensor PIR mendeteksi pergerakan manusia dan kamera untuk pemantauan. Hasil awal pada pengukuran sensor berupa nilai awal (nilai crips) yang akan diubah ke nilai linguistik dengan proses *fuzzyfication*. Kemudian, setelah nilai linguistik didapatkan maka akan diteruskan proses inference untuk menentukan aturan nilai fuzzy. Selanjutnya yaitu merupakan tahap akhir yang pada tahap ini nilai linguistik diubah menjadi nilai pasti (*Defuzzyfication*) yang menjadi acuan pengambilan keputusan oleh sistem keamanan. Berikut ini adalah penjelasan pada setiap tahapan fuzzy logic:

1. Mulai: Sistem memulai operasi. Pada tahap ini, Raspberry Pi, sensor PIR, sensor magnetik, dan kamera Raspberry diinisialisasi untuk mendeteksi aktivitas di area yang diawasi.
2. Inisialisasi Sensor: Semua sensor diaktifkan untuk memulai proses pendeteksian. Sensor PIR memantau perubahan gerakan dan Sensor magnetik memantau status pintu atau jendela.
3. Pengambilan Data Sensor: Tahap ini akan mengambil data dari seluruh sensor. Pada sensor PIR akan dihitung berapa kali sensor PIR mendeteksi

- gerakan dalam periode waktu dalam 1 menit. Kemudian, sensor magnetik untuk status pintu atau jendela.
4. Mengubah Nilai Crips menjadi Linguistik (*Fuzzyfication*): Proses fuzzyfication dilakukan untuk mengonversi nilai frekuensi deteksi menjadi kategori linguistik. Intensitas gerakan menjadi “Tinggi”, “Sedang”, “Rendah”. Status pintu/jendela menjadi “Terbuka” atau “Tertutup”.
  5. Menentukan Inference (*Fuzzy Rules*): Berdasarkan nilai linguistik yang didapatkan dari fuzzyfication, sistem akan menentukan aturan inference atau keputusan berdasarkan aturan IF-THEN. Kemudian hasil dari keputusan ini diantaranya “Aman”, “Waspada”, dan “Bahaya”.
  6. Mengubah Nilai Linguistik Menjadi Nilai Pasti (*Defuzzyfication*): Proses defuzzyfication mengonversi nilai linguistik menjadi nilai output pasti yang digunakan untuk menentukan status akhir atau tindakan yang diambil.
  7. Output Nilai: Berdasarkan hasil defuzzyfication, sistem akan mengambil keputusan sistem dalam mengirim notifikasi dan menyalakan alarm. Status “Bahaya” sistem akan mengirim notifikasi bahaya dan alarm. Status “Waspada” sistem akan mengirim notifikasi waspada. Status “Aman” sistem akan memberikan status aman.
  8. Tidak Terdeteksi Pergerakan: Jika tidak ada pergerakan yang terdeteksi, sistem akan mengabaikan dan berhenti melakukan proses fuzzy, serta akan kembali ke tahap awal.

### 3.4.3 Arsitektur Sistem



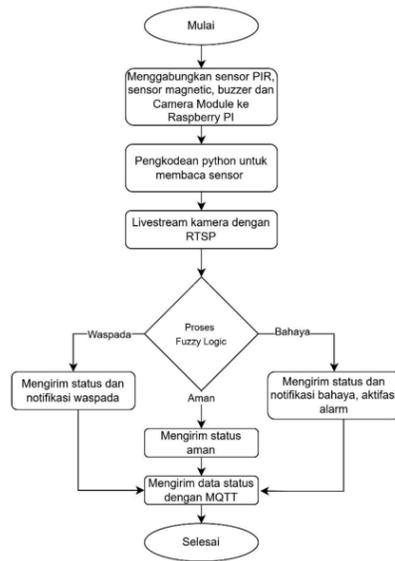
Gambar 3.3 Diagram arsitektur sistem keamanan rumah

Arsitektur sistem keamanan rumah terdiri dari beberapa komponen utama yang saling terhubung. Arsitektur ini bertujuan untuk menghubungkan perangkat keras (sensor, kamera dan raspberry) dengan aplikasi android dengan protokol MQTT dan RTSP, sehingga data pemantauan dapat diakses secara real-time melalui aplikasi android.

90  
1.

- Raspberry Pi Zero 2 W: Raspberry Pi Zero 2 W berperan sebagai pusat pemrosesan dalam sistem ini. Raspberry pi Pi Zero 2 W dipilih karena ukuran yang lebih kecil dan hemat daya, tetapi cukup kuat untuk menangani pemrosesan data dari beberapa (sensor PIR dan sensor magnetik) serta mengelola fuzzy logic. Selain itu, Raspberry Pi ini mampu melakukan tugas tambahan seperti menangkap gambar atau streaming video, yang cukup berat bagi mikrokontroler lain seperti Arduino. Raspberry Pi juga dapat mengelola logika kapan alarm harus dinyalakan, sesuai dengan status yang didapatkan dari aplikasi Android.
2. MQTT: Merupakan protokol untuk mengirim data status ke aplikasi Android. Kemudian, Android dapat menerima pesan status keamanan secara real-time melalui MQTT, yang memungkinkan pengguna mendapatkan informasi seketika saat ada ancaman.
  3. RTSP: Merupakan protokol yang digunakan untuk livestream kamera. Aplikasi Android dapat mengakses livestream ini jika ingin memantau kondisi rumah secara langsung.
  4. Aplikasi Android: Berfungsi sebagai antarmuka pengguna, menerima data status dari MQTT, dan menampilkan status keamanan. Pengguna juga dapat mengatur rentang waktu notifikasi dan mengatur rentang waktu alarm akan berbunyi otomatis ketika bahaya terdeteksi. Pengaturan Alarm diatur di aplikasi Android dan kemudian dikirim ke Raspberry Pi melalui MQTT. Raspberry Pi menyimpan informasi ini dan hanya mengaktifkan alarm dalam rentang waktu tertentu jika kondisi Bahaya terdeteksi. Aplikasi juga dapat menyimpan gambar dari kamera di dalam Local Database ketika ada peringatan waspada atau bahaya.
  5. Local Database: Digunakan untuk menyimpan data yang relevan, seperti status keamanan, waktu deteksi, dan gambar yang diambil saat ada aktivitas mencurigakan. Hal ini dapat memberikan catatan atau log aktivitas kepada pengguna.

### 3.4.4 Alur Perancangan Sistem IoT



Gambar 3.4 Alur perancangan perangkat iot

Perangkat keras yang dipilih dalam perancangan ini adalah Raspberry Pi Zero 2 W dirakit bersama komponen sensor PIR, Raspberry Pi Camera Module, dan buzzer. Perangkat dikendalikan menggunakan bahasa pemrograman Python untuk mengakses dan membaca data dari sensor PIR serta mengontrol kamera dan buzzer. Selain itu, bahasa pemrograman Python juga digunakan untuk mengirim status keamanan dengan protokol MQTT ke perangkat Android dan melakukan *streaming* kamera dengan protokol RTSP.

### 3.5 Pengembangan Aplikasi (Extreme Programming)

Pada tahap ini, dimulai untuk pengembangan aplikasi dengan metode extreme programming. Berikut adalah tahap-tahap pengembangan aplikasi dengan metode ini.

### 3.5.1 Planning (Perencanaan)

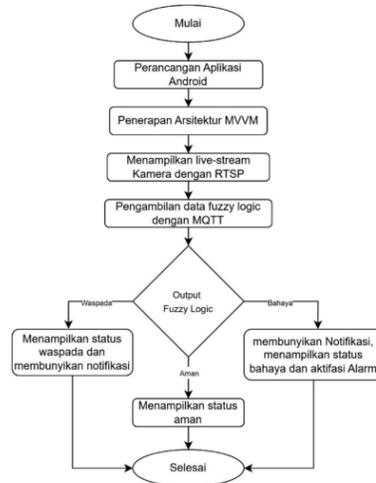
Tahap planning merupakan langkah awal dalam pengembangan aplikasi di mana dilakukan pengumpulan kebutuhan dan spesifikasi sistem. Pada tahap ini, diidentifikasi kebutuhan utama aplikasi, yaitu fitur notifikasi keamanan dan pemantauan kamera secara real-time melalui aplikasi Android serta kontrol alarm. Tujuan dari pengembangan aplikasi ini adalah memberikan solusi yang komprehensif untuk keamanan rumah dengan notifikasi dan pemantauan yang real-time. Hasil dari tahap perencanaan ini adalah daftar fitur yang akan diimplementasikan dalam pengembangan aplikasi.

### 3.5.2 Design (Perancangan)

Pada tahap desain, disusun UML diagram untuk memetakan struktur dan interaksi antar komponen, seperti use case diagram untuk mengidentifikasi fitur utama yaitu streaming kamera, notifikasi keamanan, dan kontrol alarm. Kemudian class diagram dan sequence diagram untuk menggambarkan hubungan antara aplikasi Android dan perangkat IoT. CRC (Class-Responsibility-Collaborator) digunakan untuk mendefinisikan tanggung jawab dan kolaborasi antar kelas. Desain antarmuka dibuat dalam bentuk prototype menggunakan Figma dengan fokus pada kemudahan akses dan pengalaman pengguna yang intuitif, mencakup mockup UI untuk fitur monitoring kamera, status keamanan, notifikasi, serta kontrol alarm.

### 3.5.3 Coding (Pengkodean)

Tahap pengkodean mencakup beberapa langkah penting dalam pengembangan sistem keamanan rumah. Proses ini terdiri dari pengembangan perangkat IoT dan pengembangan aplikasi Android menggunakan Kotlin dengan framework Jetpack Compose. Tujuan dari tahap pengkodean ini adalah untuk menciptakan sistem yang dapat memantau rumah secara real-time dengan menerapkan fuzzy logic untuk mengambil keputusan, serta streaming video dari kamera sebagai pemantauan lingkungan rumah. Berikut ini adalah alur perancangan aplikasi Android.



Gambar 3.5 Alur perancangan aplikasi android

Aplikasi Android dikembangkan menggunakan Kotlin dengan arsitektur MVVM. Aplikasi ini memungkinkan pengguna untuk memantau kondisi rumah secara real-time, mengontrol alarm, dan menerima notifikasi. Dalam pengambilan keputusan pada sistem keamanan ini, data fuzzy logic berperan penting untuk menentukan respons berdasarkan data dari sensor PIR, sensor magnetik dan kamera.

#### 3.5.4 Testing (Pengujian)

Setelah pengkodean selesai, dilakukan testing dengan metode Blackbox <sup>65</sup> Testing untuk memastikan bahwa semua fitur berfungsi dengan baik. Pengujian ini dilakukan tanpa melihat kode sumber, fokus hanya pada hasil yang dihasilkan oleh aplikasi. Setiap fitur utama, seperti notifikasi, streaming video dan kontrol alarm, diuji secara menyeluruh untuk memastikan aplikasi berjalan sesuai dengan kebutuhan. Jika ditemukan masalah, perbaikan dilakukan sebelum melanjutkan ke tahap evaluasi.

### 3.6 Evaluasi

Tahap evaluasi pada penelitian ini dilakukan setelah proses pengujian seluruh komponen sistem selesai dilakukan. Evaluasi bertujuan untuk menilai apakah sistem yang telah dikembangkan sudah berjalan sesuai dengan spesifikasi yang dirancang. Metode

evaluasi yang digunakan adalah dengan melakukan analisis terhadap hasil pengujian pada masing-masing komponen, meliputi sensor PIR, sensor magnetik, perangkat IoT berbasis Raspberry Pi, serta aplikasi Android. Evaluasi dilakukan dengan membandingkan hasil pengujian dengan spesifikasi teknis perangkat yang digunakan, serta dengan melihat kestabilan dan keandalan sistem secara keseluruhan.

### 3.7 Alat Pendukung Penelitian

Berikut adalah alat-alat untuk mendukung penelitian ini yang mencakup perangkat keras dan perangkat lunak:

#### 3.7.1 Perangkat Keras

1. Raspberry Pi Zero 2 W. Digunakan sebagai pusat kontrol IoT untuk menghubungkan dan mengelola semua perangkat sensor dan kamera. Adapun spesifikasinya adalah sebagai berikut:
  - a. Processor: Quad-core ARM Cortex-A53 64-bit SoC @ 1 GHz
  - b. RAM: 512 MB LPDDR2 SDRAM
  - c. Port: 1x mini HDMI, 1x micro USB OTG, 1x micro USB power, 1x CSI camera connector
  - d. Wireless: Wi-Fi 802.11 b/g/n, Bluetooth 4.2
  - e. Fungsi: menjalankan logika fuzzy, mengendalikan sensor, mengirim status melalui MQTT, dan streaming video RTSP.
2. Raspberry Pi Camera OV5647. Digunakan untuk memantau keadaan rumah secara real-time. Kamera ini akan dihubungkan ke Raspberry Pi untuk menangkap gambar atau video. Kamera ini memiliki resolusi 5 megapixel dan memiliki sudut lensa 130 derajat. Kamera dapat melihat cahaya infrared sehingga bisa digunakan untuk penglihatan malam.
3. Sensor Passive Infrared HC-SR501. digunakan untuk mendeteksi pergerakan manusia di area yang dipantau. Memiliki spesifikasi sebagai berikut:
  - a. Voltage: 5V – 20
  - b. Delay time: Adjustable (3 sampai 5 menit)
  - c. Lock time: 0.2 sec
  - d. Sensing range: kurang dari 120 derajat, dengan jarak 7 meters
4. Sensor Magentic Switch. Digunakan untuk mendeteksi pintu terbuka atau tertutup. Memiliki spesifikasi sebagai berikut:
  - a. Size : 30 x15mm
  - b. Line: 28cm
  - c. Voltage : 12V DC
  - d. Open distance : 15mm

5. Buzzer, berfungsi sebagai alarm yang dapat diaktifkan oleh pengguna melalui aplikasi Android.
6. MicroSD Card SanDisk. Diperlukan untuk menyimpan sistem operasi Raspberry Pi dan semua program yang diperlukan untuk menjalankan sistem keamanan rumah berbasis IoT. MicroSD yang digunakan memiliki penyimpanan 128 *gigabyte*.
7. Breadboard dan Kabel Jumper. Digunakan untuk menghubungkan komponen semua komponen IoT.
8. Laptop. Digunakan untuk proses pembangunan perangkat lunak.

### 3.7.2 Perangkat Lunak

1. Raspberry Pi OS, sistem operasi berbasis Linux untuk Raspberry Pi, digunakan untuk menjalankan skrip IoT.
2. Python, bahasa pemrograman utama yang digunakan untuk mengontrol perangkat keras seperti sensor dan kamera.
3. Android Studio, sebagai IDE untuk pengembangan aplikasi Android, digunakan untuk membuat aplikasi yang mengontrol sistem keamanan dan menerima notifikasi dari IoT.
4. Figma, untuk merancang antarmuka aplikasi Android.
5. Windows 11, sistem operasi utama untuk pengembangan perangkat lunak, desain, dan manajemen proyek

### 3.8 Jadwal Penelitian

Tabel 3.1 Jadwal penelitian

Kegiatan	Bulan 1				Bulan 2				Bulan 3				Bulan 4				Bulan 5			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Identifikasi Masalah																				
Studi Literatur																				
Pembangunan Perangkat IoT																				



## BAB 4. PEMBAHASAN

### 4.1 Implementasi Fuzzy Logic

Sistem keamanan berbasis IoT ini menggunakan Alur Fuzzy Logic untuk mengambil keputusan berdasarkan data sensor. Prosesnya dimulai saat perangkat Raspberry Pi, sensor PIR, sensor magnetik, dan kamera dinyalakan dan diinisialisasi. Selama operasi, langkah-langkah berikut terjadi secara berulang:

1. Membaca nilai dari setiap sensor:
  - a. Sensor PIR menghitung frekuensi deteksi gerakan per 10 detik.
  - b. Sensor magnetik mendeteksi apakah pintu/jendela terbuka atau tertutup.
2. Proses fuzzifikasi yaitu sebagai berikut:
  - a. Sensor Pir menjadi frekuensi gerakan dengan hasil “Rendah”, “Sedang”, dan “Tinggi”.
  - b. Status pintu menjadi “Terbuka” atau “Tertutup”.
  - c. Mengambil data lokal waktu untuk menentukan waktu rawan dengan hasil “Siang”, “Malam”, dan “Dini Hari”.
3. Proses inference dilakukan dengan menggabungkan hasil fuzzifikasi dari ketiga input, yaitu PIR, status pintu, dan waktu. Proses inferensi menggunakan metode Mamdani Max-Min.
4. Proses defuzzifikasi dengan metode *Centroid*. Output akhir akan mengklasifikasikan kondisi rumah menjadi tiga status yaitu “Aman”, “Waspada”, dan “Bahaya”.
5. Output sistem selanjutnya diteruskan ke aplikasi Android melalui protokol MQTT, serta sistem akan mengambil gambar dari kamera untuk bukti pendukung apabila status keamanan berada dalam kondisi waspada atau bahaya.

#### 4.1.1 Fuzzifikasi

Pada tahap ini nilai dari sensor PIR dan sensor magnetik akan dilakukan perhitungan untuk diubah dari nilai *crisp* menjadi nilai linguistik. Selain sensor, proses fuzzifikasi juga dilakukan untuk pembagian waktu rawan.

##### 4.1.1.1 Sensor PIR

Sensor PIR mengukur jumlah deteksi gerakan dalam periode waktu 10 detik dengan *delay* 0.5 detik dan diklasifikasikan ke dalam tiga kategori linguistik: rendah, sedang, dan tinggi.

1. Rendah: menggunakan rumus (4.1) fungsi keanggotaan trapesium menurun dengan rentang 0 sampai 10.

$$\mu_{rendah}(x) = \begin{cases} 1, & x \leq 0 \\ \frac{10-x}{10-0}, & 0 \leq x \leq 10 \\ 0, & x \geq 10 \end{cases} \quad (4.1)$$

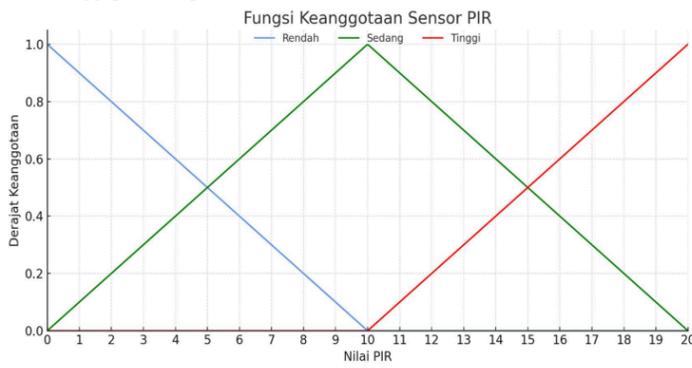
2. **Sedang**: menggunakan rumus (4.2) fungsi keanggotaan segitiga dengan rentang 0-10 menaik dan 0-20 menurun

$$\mu_{\text{sedang}}(x) = \begin{cases} 0, & x \leq 0 \text{ atau } x \geq 20 \\ \frac{x-0}{10-0}, & 0 \leq x \leq 10 \\ \frac{20-x}{20-10}, & 10 \leq x \leq 20 \\ 1, & x = 10 \end{cases} \quad (4.2)$$

3. **Tinggi** menggunakan rumus (4.3) fungsi keanggotaan trapesium menaik dengan rentang 10-20.

$$\mu_{\text{tinggi}}(x) = \begin{cases} 0, & x \leq 10 \\ \frac{x-10}{20-10}, & 10 \leq x \leq 20 \\ 1, & x \geq 20 \end{cases} \quad (4.3)$$

Berikut ini adalah gambar grafik untuk fungsi keanggotaan rendah, sedang dan tinggi pada tahap fuzzifikasi sensor PIR.



Gambar 4.1 Grafik Fungsi Keanggotaan Fuzzy PIR

#### 4.1.1.2 Waktu

Fuzzifikasi waktu menggunakan tiga himpunan linguistik: siang, malam, dan dini hari. Bentuk fungsi keanggotaannya disusun dari kombinasi fungsi

segitiga dan trapesium menurun/menaik. Karena pada waktu akan berjalan selama 24 jam secara berulang, maka nilai  $x$  akan dimoduluskan 24.

1. Siang: Rumus (4.4). Menggunakan fungsi keanggotaan segitiga, dengan rentang 0-12 menaik, 12 puncak segitiga dan 12-18 menurun.

$$\mu_{siang}(x) = \begin{cases} 0, & 18 \leq x_{mod} \leq 23 \\ \frac{x_{mod} - (x_{mod}=0)}{12 - (x_{mod}=0)}, & (x_{mod} = 0) \leq x_{mod} \leq 12 \\ \frac{18 - x_{mod}}{18 - 12}, & 12 \leq x_{mod} \leq 18 \\ 1, & x_{mod} = 12 \end{cases} \quad (4.4)$$

dimana:  $x_{mod} = x \text{ mod } 24$

2. Malam: Rumus (4.5) Fungsi keanggotaan trapesium menaik, dengan rentang jam 12-18 menaik dan 18-23 mendatar.

$$\mu_{malam}(x) = \begin{cases} \frac{x_{mod} - 12}{18 - 12}, & 12 \leq x_{mod} \leq 18 \\ 1, & 18 \leq x_{mod} \leq 23 \\ 0, & (x_{mod} = 0) \leq x_{mod} \leq 12 \end{cases} \quad (4.5)$$

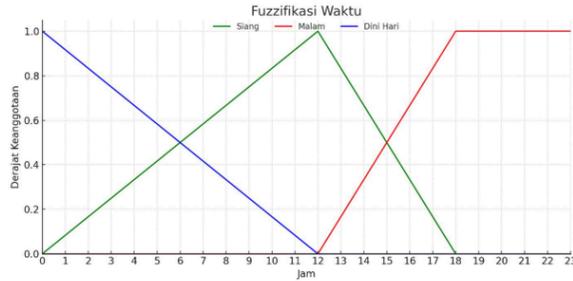
dimana:  $x_{mod} = x \text{ mod } 24$

3. Dini Hari: Rumus (4.6) Menggunakan trapesium menurun, dengan rentang 0-12.

$$\mu_{diniHari}(x) = \begin{cases} 1, & x_{mod} = (x_{mod} = 0) \\ \frac{12 - x_{mod}}{12 - (x_{mod}=0)}, & (x_{mod} = 0) \leq x_{mod} \leq 12 \\ 0, & 12 \leq x_{mod} \leq (x_{mod} = 0) \end{cases} \quad (4.6)$$

dimana:  $x_{mod} = x \text{ mod } 24$

Berikut ini adalah gambar grafik untuk fungsi keanggotaan rendah, sedang dan tinggi pada tahap fuzzifikasi waktu.



Gambar 4.2 Grafik fungsi fuzzifikasi waktu

#### 4.1.1.3 Pintu

Pada tahap fuzzifikasi pintu dikarenakan hasil sensor adalah *proximity* yang menghasilkan output 0 atau 1, maka akan dilakukan perhitungan frekuensi pintu terbuka dan tertutup selama 10 detik dengan *delay* 0.5 detik. Fuzzifikasi ini hanya memiliki dua status linguistik yaitu terbuka dan tertutup, dengan fungsi keanggotaan sebagai berikut:

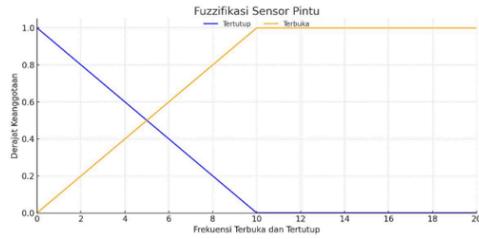
1. Rumus (4.7) Fungsi keanggotaan terbuka:

$$\mu_{\text{terbuka}}(x) = \begin{cases} 0, & x \leq 0 \\ \frac{x-0}{10-0}, & 0 \leq x \leq 10 \\ 1, & x > 10 \end{cases} \quad (4.7)$$

2. Rumus (4.8) Fungsi Keanggotaan tertutup:

$$\mu_{\text{tertutup}}(x) = \begin{cases} 1, & x \leq 0 \\ \frac{10-x}{10-0}, & 0 \leq x \leq 10 \\ 0, & x \geq 10 \end{cases} \quad (4.8)$$

Berikut ini adalah gambar grafik untuk fungsi keanggotaan terbuka dan tertutup pada proses fuzzifikasi pintu.



Gambar 4.3 Grafik fungsi fuzzifikasi sensor pintu

#### 4.1.2 Inference

*Inference* dilakukan dengan menggabungkan hasil fuzzifikasi dari ketiga input, yaitu PIR, status pintu, dan waktu. Proses inferensi menggunakan metode Mamdani Max-Min untuk menentukan aturan *fuzzy*. Berikut ini pada tabel 4.1 merupakan *inference* yang digunakan dalam sistem keamanan rumah.

Tabel 4.1 Tabel Inference

PIR	Pintu	Waktu	Status
Rendah	Tertutup	Siang	Aman
Rendah	Tertutup	Malam	Aman
Rendah	Tertutup	Dini Hari	Waspada
Rendah	Terbuka	Siang	Waspada
Rendah	Terbuka	Malam	Waspada
Rendah	Terbuka	Dini Hari	Bahaya
Sedang	Tertutup	Siang	Aman
Sedang	Tertutup	Malam	Waspada
Sedang	Tertutup	Dini Hari	Bahaya
Sedang	Terbuka	Siang	Waspada
Sedang	Terbuka	Malam	Bahaya
Sedang	Terbuka	Dini Hari	Bahaya
Tinggi	Tertutup	Siang	Waspada
Tinggi	Tertutup	Malam	Waspada
Tinggi	Tertutup	Dini Hari	Bahaya
Tinggi	Terbuka	Siang	Bahaya
Tinggi	Terbuka	Malam	Bahaya
Tinggi	Terbuka	Dini Hari	Bahaya

Dari hasil aturan pada tabel 4.1 akan membentuk sebuah output fuzzy untuk menentukan status keamanan rumah pengguna. Untuk keperluan perhitungan defuzzifikasi, kategori linguistik status ini diberi nilai numerik sebagai berikut:

3. Aman (rendah): 30
4. Waspada (sedang): 60
5. Bahaya (tinggi): 90

Nilai-nilai ini akan digunakan sebagai representasi *output* pada proses perhitungan *centroid* pada tahap defuzzifikasi berikutnya.

#### 4.1.3 Defuzzifikasi

Setelah proses inferensi dilakukan, sistem akan menghasilkan sejumlah aturan fuzzy dengan nilai keanggotaan tertentu ( $\alpha$ ) dan keluaran status keamanan dalam bentuk nilai numerik ( $z$ ). Untuk memperoleh nilai keluaran yang bersifat tegas (crisp output), digunakan metode defuzzifikasi dengan pendekatan *centroid*. Metode *centroid* menghitung nilai keluaran akhir ( $z_{\text{output}}$ ) berdasarkan rumus:

$$z_{\text{output}} = \frac{\sum_{i=1}^n \alpha_i \cdot z_i}{\sum_{i=1}^n \alpha_i} \quad (4.8)$$

Dimana:

3.  $\alpha_i$  adalah nilai derajat keanggotaan hasil dari aturan ke- $i$  (menggunakan operator Min), dan hasil akhir tiap kategori output diperoleh dengan operator Max jika ada lebih dari satu aturan yang menghasilkan kategori tersebut.
4.  $z_i$  adalah nilai output numerik dari aturan ke- $i$
5.  $n$  adalah jumlah aturan

Nilai  $z_{\text{output}}$  yang dihasilkan akan diklasifikasikan ke dalam tiga kategori status keamanan sebagai berikut:

Tabel 4.2 Tabel output defuzzifikasi

Rentang Nilai	Status Keamanan
30-45	Aman
46-75	Waspada
76-100	Bahaya

Rentang nilai pada tabel 4.2 di atas ditentukan berdasarkan titik tengah dari nilai representatif linguistik fuzzy yang digunakan dalam inferensi yaitu aman dengan nilai 30, waspada dengan nilai 60 dan bahaya dengan nilai 90. Dengan pendekatan ini, batas antar kategori ditentukan dari nilai tengah antara masing-masing status:

1. Batas antara Aman dan Waspada adalah 45
2. Batas antara Waspada dan Bahaya adalah 75

Setelah nilai  $z\_output$  diketahui, sistem akan menjalankan tindakan sesuai status keamanan yang diperoleh:

1. Status “Bahaya”: bunyikan alarm jika alarm diaktifkan pengguna dan kirim notifikasi darurat
2. Status “Waspada”: kirim notifikasi peringatan
3. Status “Aman”: tampilkan indikator aman

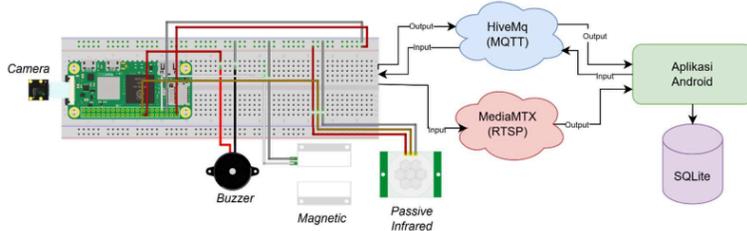
Setelah tindakan diambil, jika tidak ada pergerakan yang terdeteksi sistem akan memberikan status “Aman” dan sistem kembali ke pengambilan data sensor dan proses akan diulangi terus-menerus untuk pemantauan real-time.

#### 4.2 Implementasi Arsitektur Sistem

Pada tahap ini akan dijelaskan implementasi arsitektur sistem yang akan diterapkan pada sistem keamanan rumah yaitu sebagai berikut:

1. Raspberry Pi Zero 2 W yang berperan sebagai pusat pemrosesan dalam sistem ini. Raspberry akan memproses hasil inputan yang diambil dari sensor PIR dan sensor magnetik untuk digunakan dalam memproses algoritma fuzzy logic. Hasil dari fuzzy ini akan dikirimkan ke jaringan mqtt untuk nantinya akan diterima di perangkat Android. Raspberry juga menjalankan RTSP untuk melakukan streaming kamera. Jaringan RTSP ini akan diteruskan ke server publik untuk dapat diakses di luar jaringan lokal.
2. MQTT berperan sebagai jalur komunikasi antara raspberry dan perangkat Android. MQTT yang digunakan dalam proyek ini adalah HiveMQ Cloud, sebuah layanan managed MQTT broker berbasis cloud yang menawarkan berbagai kemudahan. Dengan menggunakan HiveMQ, peneliti tidak perlu lagi menyediakan dan mengelola server MQTT sendiri secara manual, yang biasanya membutuhkan setup server publik, instalasi broker, konfigurasi TLS, dan pengelolaan keamanan koneksi.
3. RTSP berperan untuk jaringan livestream pemantauan kamera. Pada penelitian ini agar RTSP dapat berjalan di jaringan publik dari luar jaringan lokal, alur RTSP tidak langsung dikirimkan ke aplikasi Android dari Raspberry Pi, melainkan diteruskan terlebih dahulu ke sebuah server publik. Raspberry Pi mengirimkan RTSP ke server tersebut, kemudian pada server akan menjalankan MediaMTX sebagai perantara. Dengan pendekatan ini, kamera dapat diakses secara real-time melalui internet.

4. Aplikasi Android berperan dalam memberikan informasi ke pengguna terkait keamanan rumah. Aplikasi Android memberikan informasi ke pengguna dengan mengambil hasil output yang dikirim oleh Raspberry Pi melalui MQTT. Kemudian, aplikasi Android juga akan menampilkan livestream video real-time menggunakan hasil output dari RTSP.
5. *Local Database* berperan untuk menyimpan riwayat informasi yang dikirimkan oleh MQTT dan data pengguna di Aplikasi Android. Selain itu, *local database* juga digunakan untuk menyimpan data pengguna di Raspberry Pi. *Local database* pada penelitian ini dalam implementasiannya menggunakan database SQLite.



Gambar 4.4 Implementasi arsitektur sistem keamanan rumah

### 4.3 Implementasi Sistem IoT

Implementasi sistem IoT dilakukan dengan mengikuti alur perancangan sistem IoT yang terdiri dari menggabungkan komponen sensor, pengkodean python untuk autentikasi, livestream kamera dengan RTSP dan pengkodean python untuk sistem logika fuzzy:

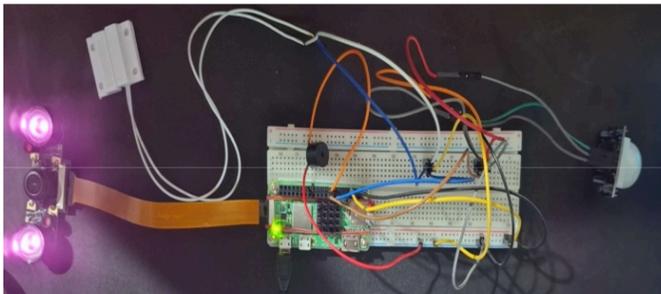
#### 4.3.1 Pemasangan Komponen Sensor Sistem Keamanan

Tahap implementasi yang pertama, dilakukan pemasangan komponen sensor untuk sistem keamanan rumah sebagai berikut :

1. *Raspberry Pi Zero 2 W* dihubungkan ke sumber listrik melalui kabel USB yang terpasang pada Raspberry Pi.
2. *Raspberry Pi Camera Module* tipe OV5647 5 MP dipasang pada *port Camera Serial Interface (CSI)* raspberry pi. untuk di sambungkan ke *Raspberry Pi Zero 2 W* menggunakan kabel HBV FFC. Kamera ini dilengkapi iluminator inframerah untuk penglihatan malam.
3. Sebelum pemasangan sensor – sensor Breadboard disambungkan ke header

GPIO Raspberry Pi sebagai tempat penyaluran tegangan dan ground.

- a. PIN 5V dihubungkan ke “+” pada BreadBoard
  - b. PIN GND dihubungkan ke “-“ pada BreadBoard
4. Pemasangan sensor pertama yaitu *Sensor Magentic Switch*, yang digunakan untuk mendeteksi pintu terbuka atau tertutup.
- a. PIN SIGNAL dihubungkan pada GPIO4
  - b. PIN GND dihubungkan pada GND BreadBoard
5. Pemasangan sensor kedua yaitu *Sensor Passive Infrared (PIR)*, yang digunakan untuk mendeteksi pergerakan manusia di area yang dipantau.
- a. PIN VCC dihubungkan ke 5 V pada BreadBoard
  - b. PIN OUT dihubungkan ke GPIO22
  - c. PIN GND dihubungkan ke GND pada BreadBoard
6. Terakhir yaitu *Buzzer*, yang berfungsi sebagai alarm yang dapat diaktifkan oleh pengguna melalui aplikasi Android.
- a. PIN + dihubungkan ke GPIO di raspberry Pi
  - b. PIN – dihubungkan ke GND di BreadBoard



Gambar 4.5 Hasil pengimplementasian perancangan iot

### 4.3.2 Konfigurasi Jaringan

Sebelum Raspberry Pi menjalankan sistem keamanan rumah, terdapat konfigurasi jaringan terlebih dahulu untuk implementasinya. Konfigurasi jaringan ini bertujuan untuk mengkoneksikan perangkat Raspberry Pi ke jaringan wifi yang tersedia di sekitarnya. Konfigurasi jaringan ini bekerja dengan cara raspberry pi akan memancarkan jaringan *hotspot* yang kemudian nantinya di aplikasi Android, pengguna dapat mengkonfigurasi jaringan wifi apa yang ingin disambungkan pada Raspberry Pi ketika dilakukan tahap registrasi. Konfigurasi jaringan ini memanfaatkan *framework* Flask untuk membuat Application Programming Interface (API) yang digunakan sebagai komunikasi antara Raspberry Pi dengan aplikasi android ketika perangkat Android terhubung ke *hotspot* yang dihasilkan Raspberry Pi. Konfigurasi jaringan ini memiliki beberapa fungsi yang diantaranya adalah sebagai berikut:

1. `is_connected()` digunakan untuk mengecek apakah Raspberry Pi sudah terhubung ke internet dengan cara melakukan ping ke DNS Google (8.8.8.8).

```
def is_connected():
    """
    21 Memeriksa koneksi internet dengan melakukan ping ke
    8.8.8.8.
    """
    try:
        subprocess.check_output(["ping", "-c", "1",
                                "8.8.8.8"])
        return True
    except subprocess.CalledProcessError:
        return False
```

2. `enable_ap_mode()` Jika Raspberry Pi belum terhubung ke internet, fungsi ini akan mengaktifkan mode hotspot (Access Point) dengan menjalankan service `hostapd` dan `dnsmasq`, sehingga perangkat Android dapat terhubung ke jaringan yang dipancarkan Raspberry Pi

```
def enable_ap_mode():
    """
    Mengaktifkan mode AP (hotspot) dengan menjalankan
    hostapd dan dnsmasq.
    """
    print("No internet. Switching to AP mode...")
    subprocess.run(["sudo", "systemctl", "enable",
                   "hostapd", "dnsmasq"])
    subprocess.run(["sudo", "systemctl", "restart",
                   "hostapd", "dnsmasq"])
```

3. `connect_to_wifi(ssid, password)` digunakan untuk menghubungkan Raspberry Pi ke jaringan WiFi yang dikonfigurasi oleh pengguna. Data SSID dan password diterima dari aplikasi Android, lalu disambungkan menggunakan perintah `nmcli`.

```
def connect_to_wifi(ssid, password):
    """
    Menghubungkan ke jaringan WiFi menggunakan nmcli.
    """
    try:
        print(f"Connecting to WiFi SSID: {ssid}")
        result = subprocess.run(
            ["sudo", "nmcli", "dev", "wifi", "connect",
            ssid, "password", password],
            stdout=subprocess.PIPE,
            stderr=subprocess.PIPE,
            text=True,
            check=True
        )
        print(result.stdout)
        return True, "Connected successfully."
    except subprocess.CalledProcessError as e:
        print(e.stderr)
        return False, e.stderr
```

4. `start_config_server()` menjalankan *service* berbasis Flask yang menyediakan dua *endpoint* API, yaitu:
- `/test-connection` untuk menguji koneksi antara Android dan Raspberry Pi. *Endpoint* dimanfaatkan aplikasi Android untuk memeriksa apakah pengguna sudah benar-benar menghubungkan perangkat Android ke *hotspot* Raspberry Pi
  - `/set-wifi` untuk menerima konfigurasi jaringan dari aplikasi Android.

```
def start_config_server():
    """
    Menjalankan server konfigurasi menggunakan Flask.
    Endpoint:
    - GET /test-connection : Untuk menguji koneksi dari
    perangkat Android.
    - POST /set-wifi : Untuk menerima konfigurasi
    WiFi (SSID dan password).
    """
```

```

113 app = Flask(__name__)

@app.route("/test-connection", methods=["GET"])
def test_connection():
    return "OK", 200
42 @app.route("/set-wifi", methods=["POST"])
def set_wifi():
    data = request.get_json()
    ssid = data.get("ssid")
    password = data.get("password")
    if not ssid or not password:
        return jsonify({"success": False, "message":
"SSID or password missing"}), 400

    success, message = connect_to_wifi(ssid, password)
96 if success:
    return jsonify({"success": True, "message":
message}), 200
    else:
    return jsonify({"success": False, "message":
message}), 500

107 def run_app():
    app.run(host='0.0.0.0', port=80)

71 server_thread = threading.Thread(target=run_app)
server_thread.daemon = True
server_thread.start()
print("Configuration server started on port 80.")

```

5. `setup_network_configuration()` Fungsi utama yang mengatur seluruh proses konfigurasi jaringan. Jika tidak ada koneksi internet, maka Raspberry Pi akan otomatis masuk ke mode hotspot, menjalankan server Flask, dan menunggu koneksi WiFi berhasil dilakukan sebelum melanjutkan proses berikutnya.

```

def setup_network_configuration():
    """
    Jika koneksi internet tidak tersedia:
    1. Aktifkan mode AP.
    2. Jalankan server Flask konfigurasi.
    3. Tunggu hingga koneksi internet tersedia.
    """

```

```

if not is_connected():
    print("No internet connection detected.")
    enable_ap_mode()
    start_config_server()
    print("Hotspot mode enabled. Connect your phone to
the Pi AP and POST to http://192.168.4.1/set-wifi")
    while not is_connected():
        print("Waiting for WiFi connection...")
        time.sleep(10)
    print("WiFi connected. You can now disable AP mode
or switch to station mode.")

```

Dengan sistem ini, pengguna tidak perlu melakukan konfigurasi jaringan secara manual menggunakan terminal atau kabel LAN. Cukup dengan aplikasi Android, Raspberry Pi dapat disambungkan ke WiFi rumah secara praktis dan efisien.

#### 4.3.3 Pengodean Autentikasi

Tahap selanjutnya dari implementasi sistem IoT adalah pengkodean untuk fitur autentikasi. Fitur ini dibuat untuk mencegah orang yang tidak memiliki hak untuk mengakses sistem keamanan rumah ini. Dalam pengkodean autentikasi ini, terdapat beberapa fungsi, yaitu:

1. `is_authenticated()` berfungsi untuk mengetahui apakah pengguna sudah login atau belum.

```

def is_authenticated():
    global is_logged_in
    return is_logged_in

```

2. `on_message_register()` Fungsi ini menangani proses pendaftaran pengguna baru yang datanya dikirim melalui MQTT, lalu disimpan ke dalam database lokal.

```

24
def on_message_register(client, userdata, msg):
    global topic_register, is_logged_in
    payload = json.loads(msg.payload.decode("utf-8"))
    username = payload.get("username")
    password = payload.get("password")
    email = payload.get("email")

    conn = sqlite3.connect('security.db')
    cursor = conn.cursor()
    cursor.execute('''CREATE TABLE IF NOT EXISTS users

```

```

122         (username TEXT, password TEXT,
email TEXT)'''
        cursor.execute("SELECT * FROM users WHERE email=?",
(email,))

        if cursor.fetchone():
            client.publish(f"{topic_register}/response",
                json.dumps({"success": False,
"message": "User already registered"}))
            conn.close()
            return

94         cursor.execute("INSERT INTO users (username,
password, email) VALUES (?, ?, ?)",
                (username, password, email))
        conn.commit()
        conn.close()

        print("Created Data: ", get_data_user())

        is_logged_in = True

        client.publish(f"{topic_register}/response",
            json.dumps(
                {"success": True, "message": f"User {username}
registered successfully"}))

```

3. `on_message_login()` Fungsi ini digunakan untuk proses login. Sistem akan mencocokkan email dan password dengan data yang ada di database, lalu memberikan respon melalui MQTT.

```

24 def on_message_login(client, userdata, msg):
    global topic_login, code, is_logged_in
143     payload = json.loads(msg.payload.decode("utf-8"))
    email = payload.get("email")
    password = payload.get("password")

43     conn = sqlite3.connect('security.db')
    cursor = conn.cursor()
    cursor.execute(
        "SELECT * FROM users WHERE email=? AND
password=?", (email, password))
    user = cursor.fetchone()
    conn.close()

```

```

if user:
    print("UserData: ", user)
    is_logged_in = True
    userData = {"username": user[0], "email":
user[2], "password": user[1], "code_device": code}
    client.publish(f"{topic_login}/response",
        json.dumps({"success": True,
"message": "Login successful", "data": userData}))
    else:
        client.publish(f"{topic_login}/response",
            json.dumps({"success": False,
"message": "Invalid credentials"}))

```

4. `get_data_user()` Fungsi ini digunakan untuk mengambil data pengguna yang tersimpan di database, terutama digunakan saat sistem dijalankan ulang. Fungsi ini juga berfungsi untuk memeriksa device sudah diregistrasi atau belum.

```

def get_data_user():
    conn = sqlite3.connect('security.db')
    cursor = conn.cursor()
    cursor.execute("SELECT username, password FROM
user")
    user_data = cursor.fetchone()
    conn.close()

    if user_data:
        return {"username": user_data[0], "password":
user_data[1]}
    return None

```

5. `on_message_check_username()` berguna untuk mengecek apakah suatu username sudah pernah didaftarkan di sistem. Hal ini dilakukan untuk mencegah terjadinya duplikat username.

```

def on_message_check_username(client, userdata, msg):
    username = msg.topic.split("/") [0]
    conn = sqlite3.connect('security.db')
    cursor = conn.cursor()
    cursor.execute("SELECT * FROM users WHERE
username=?", (username,))
    exists = cursor.fetchone() is not None

```

```
conn.close()
client.publish(f"{username}/check/response",
              json.dumps({"exists": exists}))
```

6. `on_message_check_device()` digunakan untuk mengecek apakah perangkat sudah memiliki data pengguna yang terdaftar atau belum.

```
def on_message_check_device(client, userdata, msg):
    conn = sqlite3.connect('security.db')
    cursor = conn.cursor()
    cursor.execute("SELECT * FROM users")
    registered = cursor.fetchone() is not None
    conn.close()
    client.publish(f"{DEVICE_CODE}/registered/response",
                  json.dumps({"registered":
                              registered}))
```

7. `auth()` mengatur semua topik MQTT yang digunakan untuk autentikasi dan menghubungkan fungsi-fungsi di atas ke sistem komunikasi MQTT.

```
def auth(client, device_code):
    global topic_register, topic_login, code

    topic_register = f"{device_code}/auth/register"
    topic_login = "auth/login"
    code = device_code

    client.message_callback_add(topic_register,
                                on_message_register)
    client.message_callback_add(topic_login,
                                on_message_login)
    client.message_callback_add(f"+/check",
                                on_message_check_username)
    #
    client.message_callback_add(f"{DEVICE_CODE}/registered",
                                on_message_check_device)

    client.subscribe(topic_register)
    client.subscribe(topic_login)
    # client.subscribe(f"{DEVICE_CODE}/registered")
    client.subscribe("+/check")
```

Seluruh fungsi ini bekerja sama untuk memastikan bahwa hanya pengguna yang terdaftar dan telah berhasil login yang dapat mengakses sistem keamanan rumah berbasis IoT.

#### 4.3.4 Livestream Kamera

Tahap selanjutnya yaitu melakukan livestream kamera. Livestream dilakukan dengan menggunakan perintah libcamera-vid untuk menangkap video dari kamera. Kemudian, output video yang didapatkan dari kamera tersebut langsung dilakukan streaming dengan program ffmpeg. Pada bagian bawah kamera hasil video akan ditambahkan timestamp untuk memberikan info tanggal dan waktu secara real-time.

```
def start_camera_stream(topic_user):
    stream_url = f"rtsp://52.77.241.213:8554/mystream/{topic_user}"
    command = f"""
libcamera-vid --awbgains 1.3,1.0 -t 0 --inline --framerate 15 --width
480 --height 270 --codec h264 -o - | \
ffmpeg -probesize 5M -analyzeduration 10M -f h264 -i - \
-vf "drawtext=fontfile=/usr/share/fonts/truetype/dejavu/DejaVuSans-
Bold.ttf:text='%{{localtime}}':x=w-tw-10:y=h-th-
10:fontsize=14:fontcolor=white@0.8:box=1:boxcolor=black@0.4" \
-c:v h264_v4l2m2m -pix_fmt yuv420p \
-rtsp_transport tcp -f rtsp {stream_url}
"""

    while True:
        print("Starting camera stream...")
        proc = subprocess.Popen(
            command,
            shell=True,
            executable="/bin/bash",
            stdout=subprocess.DEVNULL,
            stderr=subprocess.DEVNULL
        )
        proc.wait()
        print("Stream
```

Hasil dari proses tersebut dikonversi dan dikirimkan sebagai RTSP stream ke alamat server tertentu dengan format:

```
rtsp://[alamat_server]/mystream/[user_topic]
```

Dengan pendekatan ini, video dari kamera dapat dipantau secara real-time melalui jaringan oleh aplikasi client, tanpa perlu menyimpan video secara lokal di perangkat.

#### 4.3.5 Pengodean Sistem Logika Fuzzy

Pada tahap ini algoritma logika fuzzy diterapkan dalam bentuk kode python untuk membuat sistem IoT ini berjalan. Kode untuk logika fuzzy memiliki banyak fungsi diantaranya adalah sebagai berikut:

1. `fuzzy_membership_low()`, `fuzzy_membership_medium()`, dan `fuzzy_membership_high()`. Ketiga fungsi ini digunakan untuk menentukan tingkat keanggotaan dari input sensor gerak (PIR) berdasarkan nilai yang diterima. Nilai input ini kemudian diklasifikasikan menjadi rendah, sedang, atau tinggi sesuai dengan logika fuzzy.

```
def fuzzy_membership_low(x, A, B):
    if x <= A:
        return 1
    elif A <= x <= B:
        return (B - x) / (B - A)
    else:
        return 0

def fuzzy_membership_medium(x, A, B, C):
    if x == B:
        return 1
    elif A <= x <= B:
        return (x - A) / (B - A)
    elif B <= x <= C:
        return (C - x) / (C - B)
    else:
        return 0

def fuzzy_membership_high(x, A, B):
    if x <= A:
        return 0
    elif A <= x <= B:
        return (x - A) / (B - A)
    else:
        return 1
```

2. `fuzzy_siang()`, `fuzzy_malam()`, dan `fuzzy_dini_hari()`. Ketiga fungsi ini digunakan untuk menentukan tingkat keanggotaan kerawanan waktu dalam 24 jam. Nilai input ini kemudian diklasifikasikan menjadi siang, malam, atau dini hari sesuai perhitungan logika *fuzzy*.

```
def fuzzy_dini_hari(x):
    x = x % 24
    if x == 0:
        return 1
    elif 0 < x <= 12:
        return (12 - x) / 12
    elif 12 <= x <= 23:
        return 0
    else:
        return 0

def fuzzy_siang(x):
    x = x % 24
    if x == 12:
        return 1
    elif 0 < x <= 12:
        return x / 12
    elif 12 <= x <= 18:
        return (18 - x) / 6
    elif 18 <= x <= 23:
        return 0
    else:
        return 0

def fuzzy_malam(x):
    x = x % 24
    if 18 <= x <= 23:
        return 1
    elif 12 <= x <= 18:
        return (x - 12) / 6
    elif 0 < x < 12:
        return 0
    else:
        return 0
```

3. `fuzzy_inference()` digunakan untuk menyimpulkan status keamanan berdasarkan hasil dari fungsi keanggotaan fuzzy dan status pintu. Hasil akhirnya dapat berupa “aman”, “waspada”, atau “bahaya”.

```
def fuzzy_inference(pir, door):
    global pir_status, door_status
```

```
now = datetime.datetime.now()
hour = now.hour

diniHari = fuzzy_dini_hari(hour)
siang = fuzzy_siang(hour)
malam = fuzzy_malam(hour)

pir_low = fuzzy_membership_low(pir, 0, 10)
pir_medium = fuzzy_membership_medium(pir, 0, 10, 20)
pir_high = fuzzy_membership_high(pir, 10, 20)

pintu_low = fuzzy_membership_low(door, 0, 10)
pintu_high = fuzzy_membership_high(door, 0, 10)

# Label status sensor
pir_values = [pir_low, pir_medium, pir_high]
pir_labels = ['Rendah', 'Sedang', 'Tinggi']
pir_status = get_max_label(pir_values, pir_labels)

door_values = [pintu_low, pintu_high]
door_labels = ['Tertutup', 'Terbuka']
door_status = get_max_label(door_values, door_labels)

rules = []
rules.append((min(pir_low, pintu_low, siang), 30))
rules.append((min(pir_low, pintu_low, malam), 30))
rules.append((min(pir_low, pintu_low, diniHari), 60))
rules.append((min(pir_low, pintu_high, siang), 60))
rules.append((min(pir_low, pintu_high, malam), 60))
rules.append((min(pir_low, pintu_high, diniHari),
90))
rules.append((min(pir_medium, pintu_low, siang), 30))
rules.append((min(pir_medium, pintu_low, malam), 60))
rules.append((min(pir_medium, pintu_low, diniHari),
90))
rules.append((min(pir_medium, pintu_high, siang),
60))
rules.append((min(pir_medium, pintu_high, malam),
90))
rules.append((min(pir_medium, pintu_high, diniHari),
90))
rules.append((min(pir_high, pintu_low, siang), 60))
rules.append((min(pir_high, pintu_low, malam), 60))
```

```

rules.append((min(pir_high, pintu_low, diniHari),
90))
rules.append((min(pir_high, pintu_high, siang), 90))
rules.append((min(pir_high, pintu_high, malam), 90))
rules.append((min(pir_high, pintu_high, diniHari),
90))

z_output = defuzzifikasi_max(rules)

if z_output <= 45:
    status = "aman"
elif z_output <= 75:
    status = "waspada"
else:
    status = "bahaya"

return status

```

4. `defuzzifikasi_max(rules)` digunakan untuk melakukan proses defuzzifikasi pada perhitungan *fuzzy* dengan menerapkan rumus *centroid*.

```

def defuzzifikasi_max(rules):
    grouped = defaultdict(list)
    for alpha, z in rules:
        grouped[z].append(alpha)
    max_rules = [(max(alphas), z) for z, alphas in
grouped.items()]
    numerator = sum(alpha * z for alpha, z in max_rules)
    denominator = sum(alpha for alpha, _ in max_rules)
    return numerator / denominator if denominator != 0 else 0

```

5. `capture_image()` digunakan untuk menangkap gambar dari kamera saat status rumah berada pada kondisi "waspada" atau "bahaya". Gambar dikirim dalam bentuk base64 melalui MQTT ke client.

```

def capture_image(user):
    filename = "image.jpg"
    try:
        subprocess.run([
            "ffmpeg", "-y",
            "-rtsp_transport", "tcp",
            "-i",
            f"rtsp://52.77.241.213:8554/mystream/{user}",
            "-vf", "select='eq(pict_type\\,I)'"

```

```

        "-frames:v", "1",
        filename
    ], check=True)
    return filename
except subprocess.CalledProcessError as e:
    print("Gagal capture dari stream via ffmpeg")
    print(e)
    return None

```

6. `send_to_mqtt()` fungsi ini bertanggung jawab untuk mengirimkan data status rumah. Gambar akan dikirim jika status dalam kondisi “waspada” atau “bahaya” ke server MQTT, serta menangani proses retry jika belum menerima balasan (ACK) dari client. Apabila belum menerima ack dari client maka status “waspada” atau “bahaya” akan disimpan di dalam array terlebih dahulu sampai client mengirimkan ACK.

```

def send_to_mqtt(client, status, topic_status, topic_ack,
image_path=None):
    global message_received, pir_status, door_status

    data = {
        "status": status,
        "time": str(datetime.datetime.now()),
        "image": None if status == "Aman" else None,
        "pir": pir_status,
        "door": door_status
    }

    if status in ["waspada", "bahaya"] and image_path:
        with open(image_path, "rb") as img_file:
            base64_image =
base64.b64encode(img_file.read()).decode("utf-8")
            data["image"] = base64_image

    client.message_callback_add(topic_ack, on_message)
    client.subscribe(topic_ack)

    unsent_messages.append(data)

    while unsent_messages:
        # print("Pesan tertahan:\n", unsent_messages)
        message_received = False
        json_data = json.dumps(unsent_messages[0])

```

```

client.publish(topic_status, json_data)

start_time = time.time()
timeout_duration = 60

while True:
    if message_received:
        print("ACK diterima, buang pesan:\n",
unsent_messages)
        unsent_messages.pop(0)
        break

    if time.time() - start_time >=
timeout_duration:
        print("No ACK received, message stored
for retry.")
        break
    time.sleep(0.1)

```

7. `on_message()` fungsi ini menangani pesan masuk dari topik MQTT yang berkaitan dengan aktivasi buzzer dan penerimaan ACK.

```

24 def on_message(client, userdata, msg):
    148 global message_received, alarm_activated, topic_buzzer
    payload = msg.payload.decode("utf-8")
    print(payload)
    if payload == "ACK":
        message_received = True
    if payload == "buzzer_activated":
        alarm_activated = True
        client.publish(f"{topic_buzzer}/ack", "ACK")
    if payload == "buzzer_deactivated":
        alarm_activated = False
        client.publish(f"{topic_buzzer}/ack", "ACK")

```

8. `buzzer_status_init()` digunakan untuk mengatur *subscribe* topik MQTT yang berkaitan dengan status buzzer agar dapat merespons jadwal alarm berbunyi jika kondisi "bahaya".

```

def buzzer_status_init(client, user):
    global alarm_activated
    topik_buzzer = f"{user}/status_buzzer"
    client.subscribe(topik_buzzer)
    client.message_callback_add(topik_buzzer, on_message)

```

9. `change_wifi()` dan `on_message_wifi()` merupakan fitur tambahan yang memungkinkan perangkat berpindah jaringan Wi-Fi secara remote melalui perintah dari client.

```
def change_wifi(ssid, password):
    try:
        print(f"Connecting to SSID: {ssid}")
        if password:
            subprocess.run(
                ["sudo", "nmcli", "dev", "wifi",
                "connect", ssid, "password", password],
                check=True
            )
        else:
            subprocess.run(
                ["sudo", "nmcli", "dev", "wifi", "connect",
                ssid],
                check=True
            )
        print("Connected successfully!")
        return True
    except subprocess.CalledProcessError as e:
        print(f"Connection failed: {e}")
        return False

def on_message_wifi(client, userdata, msg):
    global topic_wifi
    TOPIC_ACK_WIFI = f"{topic_wifi}/ack"
    print(f"Message received on {msg.topic}")
    try:
        data = json.loads(msg.payload.decode())
        ssid = data.get("ssid")
        password = data.get("password", "")

        if not ssid:
            print("SSID is missing!")
            client.publish(TOPIC_ACK_WIFI, "FAIL")
            return

        success = change_wifi(ssid, password)

        if success:
```

```

        print("hasil: ", success)
        print("Send To: ", TOPIC_ACK_WIFI)
        time.sleep(5)
        client.publish(TOPIC_ACK_WIFI, "ACK")
    else:
        print("hasil: ", success)
        print("Send To: ", TOPIC_ACK_WIFI)
        client.publish(TOPIC_ACK_WIFI, "FAIL")

128 except Exception as e:
    print(f"Error: {e}")
    client.publish(TOPIC_ACK_WIFI, "FAIL")

def change_wifi_init(client, user):
    global topic_wifi
    topic_wifi = f"{user}/wifi"
    client.subscribe(topic_wifi)
    client.message_callback_add(topic_wifi, on_message_wifi)
    print("Wifi Initialized")

```

10. `start_fuzzy_security_system()` merupakan fungsi utama yang mengatur seluruh proses pemantauan, mulai dari membaca sensor PIR dan pintu magnetik, melakukan inferensi fuzzy, menangani alarm, menangkap gambar, serta mengirimkan status ke MQTT secara terus-menerus dalam loop.

```

def start_fuzzy_security_system(client, user):
    global unsent_messages, topic_buzzer

    GPIO.setmode(GPIO.BCM)
    34 GPIO.setup(PIR_PIN, GPIO.IN)
    GPIO.setup(MAGNETIC_PIN, GPIO.IN, pull_up_down=GPIO.PUD_UP)
    GPIO.setup(BUZZER_PIN, GPIO.OUT)

    topic_ack = f"{user}/status_rumah/ack"
    topic_status = f"{user}/status_rumah"
    topic_buzzer = f"{user}/status_buzzer"

    print(topic_ack)
    print(topic_status)

    try:
        buzzer_status_init(client, user)

```

```

change_wifi_init(client, user)
while True:
    pir_count = 0
    door_count = 0
    sampling_duration = 10
    interval = 0.5
    total_samples = int(sampling_duration / interval)

    for _ in range(total_samples):
        pir_count += GPIO.input(PIR_PIN)
        if GPIO.input(MAGNETIC_PIN) == 1:
            door_count += 1
        time.sleep(interval)

    print("PIR COUNT = " + str(pir_count))
    print("Door COUNT = " + str(door_count))

    status = fuzzy_inference(pir_count, door_count)

    image_path = capture_image(user) if status in [
        "waspada", "bahaya"] else None
    send_to_mqtt(client, status, topic_status,
                topic_ack, image_path)

    if (status == "bahaya" and alarm_activated):
        GPIO.output(BUZZER_PIN, GPIO.HIGH)
        time.sleep(1)
        GPIO.output(BUZZER_PIN, GPIO.LOW)
except KeyboardInterrupt:
    GPIO.cleanup()

```

Melalui kombinasi fungsi-fungsi ini, sistem keamanan rumah dapat mendeteksi gerakan dan kondisi pintu secara real-time, menilai tingkat ancaman menggunakan logika fuzzy, serta merespons dengan mengaktifkan alarm jika status “bahaya”.

#### 4.3.6 Penamaan Node Kode IoT

Penamaan node dilakukan untuk mempermudah pelacakan fungsi dalam proses pengujian *whitebox*. Setiap fungsi penting pada perangkat IoT diberi kode seperti 1A, 2A, dan seterusnya, sesuai dengan urutan alur program. Penandaan ini memudahkan identifikasi bagian kode yang dijalankan saat fitur tertentu diuji.

Tabel 4.3 Tabel Penamaan node kode iot

No	Fungsi	Node
1	is_authenticated()	1A
2	on_message_register(client, userdata, msg)	2A
3	on_message_login()	3A
4	get_data_user()	4A
5	on_message_check_username()	5A
6	on_message_check_device(client, userdata, msg)	6A
7	auth(client, device_code)	7A
8	is_connected()	1J
9	enable_ap_mode()	2J
10	connect_to_wifi(ssid, password)	3J
11	start_config_server()	4J
12	setup_network_configuration()	5J
13	fuzzy_membership_low(), fuzzy_membership_medium(), fuzzy_membership_high()	1F
14	fuzzy_siang(), fuzzy_malam(), fuzzy_dini_hari()	2F
15	fuzzy_inference()	3F
16	defuzzifikasi_max(rules)	4F
17	capture_image()	5F
18	send_to_mqtt()	6F
19	on_message()	7F
20	buzzer_status_init()	8F
21	change_wifi(), on_message_wifi(), change_wifi_init()	9F
22	start_fuzzy_security_system()	10F
23	start_camera_stream(topic_user):	1L

#### 4.4 Implementasi Pengembangan Aplikasi

Pengembangan aplikasi android ini menerapkan metodologi yaitu Extreme Programming (XP) yang terdiri dari tahap perancangan, desain, pengodean dan testing. Berikut ini akan dijelaskan implementasi dan hasil dari pengembangan aplikasi.

##### 4.4.1 Perancangan

Tahap awal dari pengembangan aplikasi sistem keamanan rumah adalah tahap perancangan. Untuk memenuhi kebutuhan sistem keamanan maka diperlukan beberapa fitur utama dalam aplikasi yaitu:

1. Menampilkan status keamanan rumah pengguna secara real-time
2. Memberikan notifikasi apabila status waspada atau bahaya
3. Mengatur jadwal alarm akan berbunyi otomatis jika status bahaya sesuai waktu yang ditentukan pengguna.

Selain itu dalam merancang sistem keaman ini terdapat juga beberapa fitur pendukung yaitu:

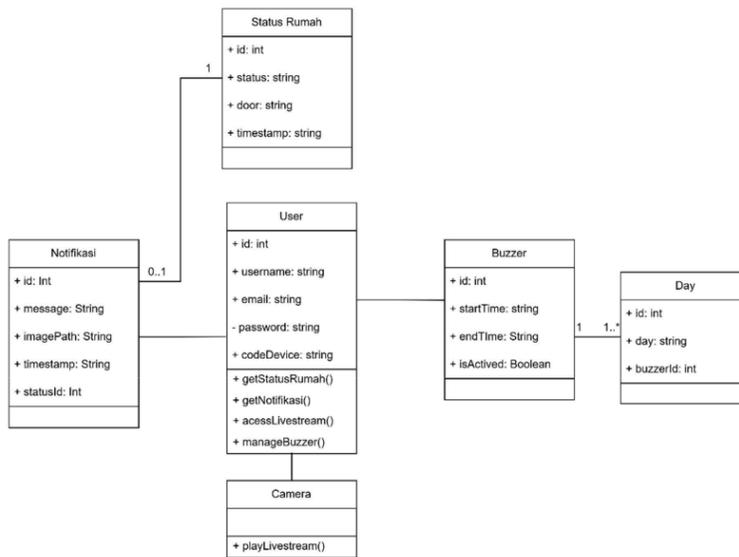
1. Autentikasi pengguna, untuk mencegah seseorang yang tidak mempunyai hak untuk mengakses sistem ini
2. Pengaturan wifi untuk perangkat IoT agar perangkat terus terhubung ke jaringan internet.

Dalam penelitian ini, sistem aplikasi dirancang menggunakan pendekatan satu pengguna untuk satu perangkat yang berarti setiap akun hanya dapat dikaitkan dengan satu perangkat IoT.

##### 4.4.2 Design

Tahap desain bertujuan untuk menggambarkan struktur dan interaksi antar komponen aplikasi dibangun sebelum nanti dilanjutkan ke dalam tahap pengodean. Desain sistem dalam penelitian ini menggunakan beberapa jenis diagram UML untuk memodelkan struktur dan perilaku aplikasi. Diagram yang digunakan antara lain *Class Diagram*, *Use Case Diagram*, *Activity Diagram*, serta *Class Responsibility Collaborator Card* sebagai pelengkap untuk menggambarkan tanggung jawab setiap kelas. Selain itu, rancangan antarmuka pengguna disusun untuk memvisualisasikan tampilan dan alur penggunaan aplikasi.

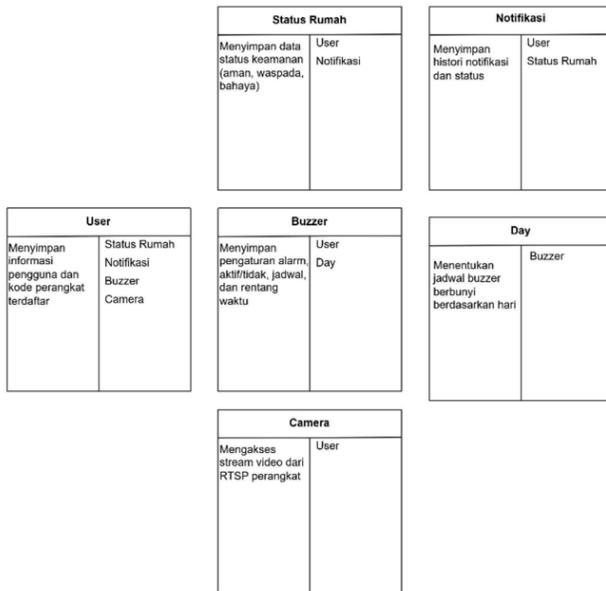
#### 4.4.2.1 <sup>61</sup> Class Diagram



Gambar 4.6 Class diagram sistem aplikasi keamanan rumah

<sup>13</sup> Class diagram diatas menggambarkan struktur data dan relasi antar entitas utama dalam sistem, seperti User, Status Rumah, Notifikasi, Buzzer, Day, dan Camera. Masing-masing entitas memiliki atribut serta relasi satu-sama-banyak yang direpresentasikan melalui garis penghubung. Dalam class diagram ini entitas user direlasikan dengan beberapa entitas lain seperti Status Rumah, Notifikasi, dan Buzzer bukan berdasarkan hubungan relasional database, melainkan berdasarkan logika interaksi sistem. User sebagai aktor utama memiliki akses terhadap informasi dari entitas tersebut, sehingga secara fungsional dianggap memiliki keterkaitan langsung. Diagram ini menjadi dasar dalam membangun struktur database serta pemetaan objek dalam kode aplikasi Android.

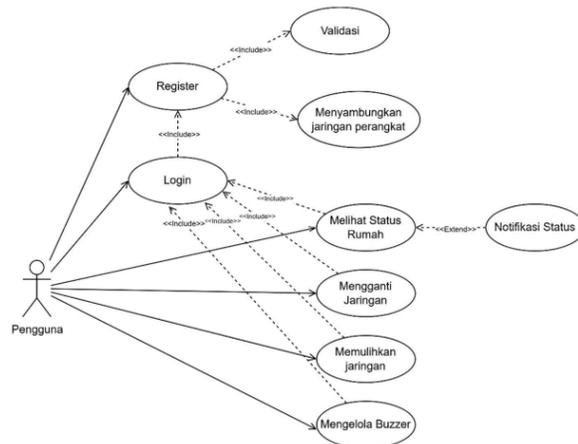
#### 4.4.2.2 Class-Responsibility-Collaborator Card



Gambar 4.7 CRC card sistem aplikasi keamanan rumah

CRC Card digunakan untuk menggambarkan tanggung jawab dan kolaborasi antar kelas. Setiap kotak dalam diagram menunjukkan satu kelas, dengan daftar tanggung jawab utama serta kelas lain yang berkolaborasi dengannya. CRC ini membantu dalam memahami struktur logika program, dan mempermudah perencanaan alur komunikasi antar class selama pengembangan.

#### 88 4.4.2.3 Use Case Diagram



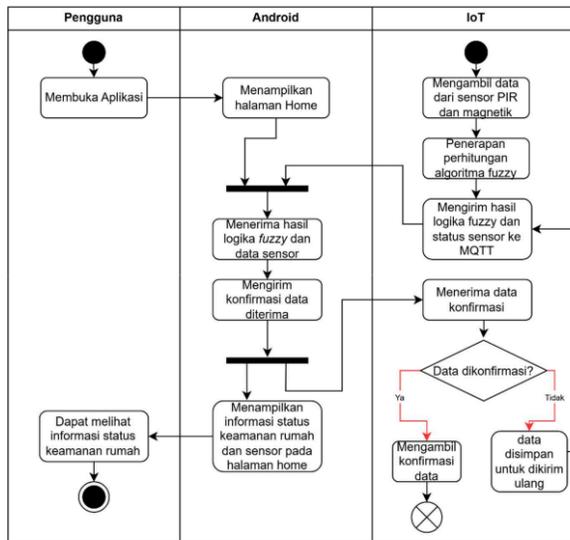
Gambar 4.8 Use case diagram sistem aplikasi keamanan rumah

127  
Use case diagram menggambarkan fungsionalitas sistem dari perspektif pengguna. Diagram ini menunjukkan interaksi antara aktor User dengan berbagai fitur utama, seperti Login, Register, Melihat Status Rumah, Mengelola Buzzer, Mengganti Jaringan, dan Memulihkan Jaringan. Beberapa use case menggunakan relasi <<include>> untuk menunjukkan fungsi yang selalu dipanggil, seperti Validasi dan Menyambungkan Jaringan, serta relasi <<extend>> untuk aksi tambahan seperti Notifikasi Status yang hanya muncul saat status rumah dalam kondisi tidak aman.

#### 4.4.2.4 Activity Diagram

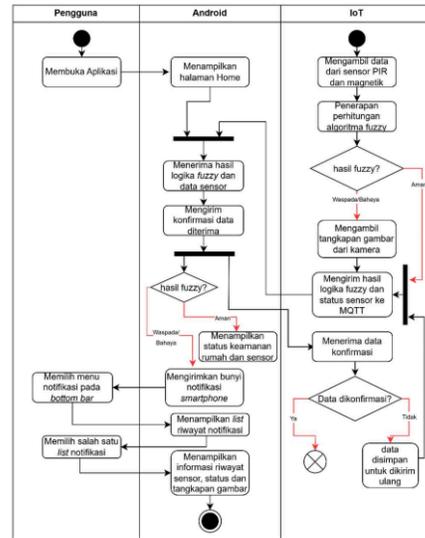
Pada activity diagram akan digambarkan tahapan proses yang dilakukan oleh sistem ketika sedang menjalankan fitur. Fitur yang digambarkan antara lain proses autentikasi, melihat status rumah, notifikasi, alarm, *manage* Wifi, dan *recovery* Wifi.





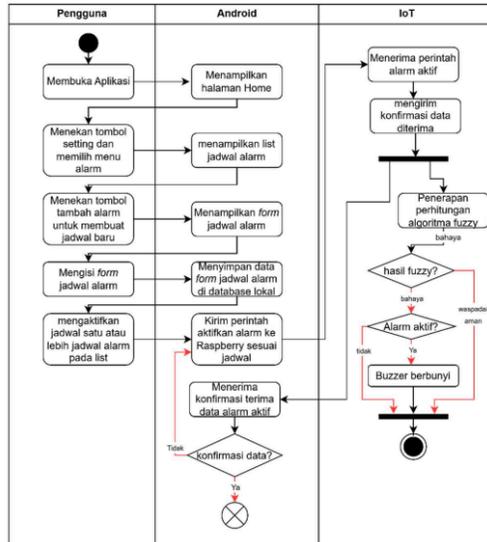
Gambar 4.10 Activity diagram status keamanan rumah

Pada gambar 4.10 menggambarkan alur proses sistem ketika pengguna melihat status keamanan rumah pada aplikasi. Pada proses ini, pengguna cukup membuka aplikasi untuk melihat status keamanan rumah pada halaman *home*. Proses pengambilan status keamanan rumah dilakukan dengan perangkat IoT yang mengirim hasil perhitungan logika *fuzzy* ke MQTT dan perangkat Android akan menerima informasi dengan mengambil data dari MQTT.



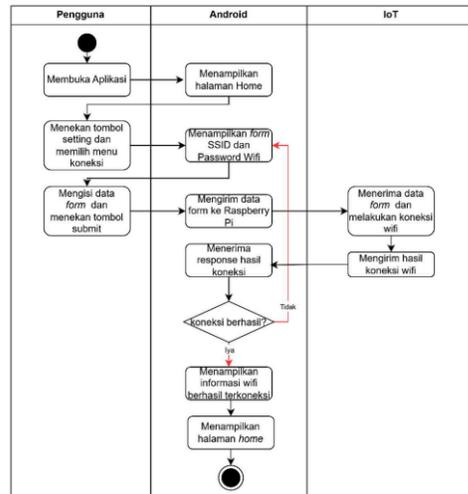
**Gambar 4.11** Activity diagram notifikasi

Pada gambar 4.11 menggambarkan alur proses notifikasi pada aplikasi. Pada proses ini, notifikasi dijalankan ketika aplikasi menerima status waspada atau bahaya pada hasil perhitungan *fuzzy* yang dijalankan oleh perangkat IoT. Hasil dari notifikasi ini akan disimpan dalam *database* lokal aplikasi. Pengguna dapat melihat riwayat notifikasi yang menampilkan status rumah, waktu, dan gambar yang diambil dari kamera.



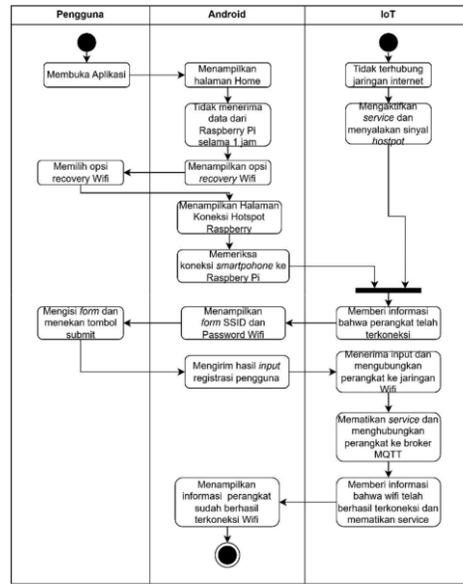
48 Gambar 4.12 Activity diagram alarm

Pada gambar 4.12 menggambarkan alur proses sistem menjalankan alarm. Pada proses ini, pengguna terlebih dahulu membuat jadwal alarm berbunyi yang diinginkan berdasarkan hari, jam dan menit. Kemudian, sistem aplikasi akan mengirim informasi pada perangkat *IoT* jika alarm harus diaktifkan. Pada perangkat *IoT* alarm berbunyi jika hasil status rumah dalam kondisi bahaya dan mendapatkan perintah dari perangkat android untuk menyalakan alarm.



48  
Gambar 4.13 Activity diagram *manage wifi*

Pada gambar 4.13 menggambarkan alur proses sistem *manage Wifi*. Pada fitur ini, pengguna dapat menambahkan jaringan Wifi yang ingin disimpan dalam perangkat IoT. Pengguna cukup mengisi *form* SSID dan Password Wifi yang nantinya perangkat IoT akan menyimpan informasi jaringan tersebut. Fitur ini berjalan jika perangkat IoT masih terhubung ke jaringan internet karena komunikasi antar perangkat masih menggunakan MQTT.

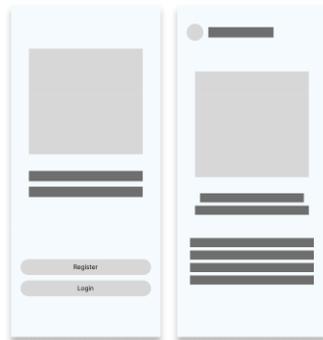


**1** Gambar 4.14 Activity diagram *recovery* wifi

Pada gambar 4.14 menggambarkan alur proses sistem *recovery* Wifi. Fitur ini tersedia jika aplikasi tidak menerima informasi status rumah yang dijalankan oleh perangkat IoT. Kemudian, jika perangkat IoT kehilangan jaringan internet maka akan mengaktifkan *service* dan memberikan sinyal *hotspot*. Pengguna dapat memulihkan jaringan internet pada IoT dengan menyambungkan *smartphone* ke jaringan IoT dan memasukkan form sesuai dengan Wifi yang ingin digunakan.

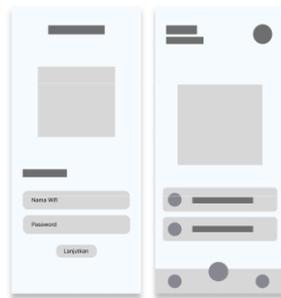
#### 4.4.2.5 Mockup User Interface

Mockup UI merupakan rancangan visual dari tampilan aplikasi yang dirancang langsung dalam bentuk final. Desain dibuat menggunakan Jetpack Compose dan/atau Figma, dengan fokus pada kemudahan navigasi dan pengalaman pengguna. Tampilan yang dirancang mencakup login, dashboard status rumah, tampilan kamera, notifikasi, pengaturan WiFi, serta pengelolaan buzzer. Berikut ini akan ditampilkan gambar yang *mockup user interface* yang digunakan dalam sistem keamanan rumah.



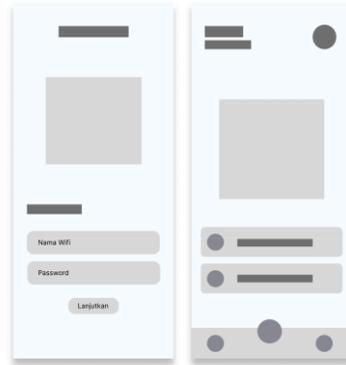
Gambar 4.15 Halaman *mockup* UI landing dan koneksi raspberry

Pada gambar 4.15 adalah tampilan dari halaman *landing* dan koneksi raspberry. Pada halaman *landing* terdapat dua tombol yaitu untuk register dan login. Ketika pengguna menekan register maka akan diarahkan skenario registrasi pengguna dan aplikasi akan mengganti ke tampilan koneksi Raspberry. Pada halaman koneksi Raspberry jika pengguna sudah menyambungkan perangkat *smartphone* ke jaringan wifi, maka aplikasi akan langsung mengarahkan ke *form* untuk mengisi SSID dan password wifi yang ingin digunakan.



Gambar 4.16 Halaman *mockup* UI form koneksi Wifi dan *home*

Pada gambar 4.16 adalah tampilan dari halaman *form* Wifi dan halaman *home*. Pada halaman *form* wifi terdapat tempat input untuk memasukkan data dan tombol untuk submit data wifi. Kemudian, pada halaman *home* menampilkan status keamanan rumah dan status deteksi sensor.



Gambar 4.17 Halaman *Mockup* UI notifikasi dan schedule alarm

Pada gambar 4.17 adalah tampilan dari halaman notifikasi dan jadwal alarm. Pada halaman notifikasi menampilkan *list* histori notifikasi yang jika ditekan salah satunya akan menampilkan detail dari histori notifikasi tersebut. Kemudian, pada halaman jadwal alarm, pengguna dapat menambahkan dan mengaktifkan jadwal alarm berbunyi jika sistem mendeteksi bahaya.



Gambar 4.18 Halaman *mockup* UI detail notifikasi

Pada gambar 4.18 adalah tampilan detail halaman notifikasi yang menampilkan informasi dari status bahaya, waktu, status sensor, dan gambar tangkapan kamera.

#### 4.4.3 Pengodean

Pada tahap pengodean ini, hasil rancangan sistem yang telah disusun dalam tahap desain akan dilanjutkan untuk dilakukan pengodean. Pengodean sistem ini menggunakan bahasa kotlin dengan arsitektur *Model-ViewModel-View* (MVVM) untuk memisahkan logika tampilan, data, dan kontrol secara terstruktur. *Model* memiliki tanggung jawab terhadap manajemen entitas data. *ViewModel* untuk mengelola logika dan *state* untuk masing-masing fitur. *View* untuk memerepresentasikan tampilan antarmuka pengguna yang dibuat menggunakan Jetpack Compose. Selain itu, dalam sistem ini juga dilengkapi beberapa pacakage tambahan untuk mendukung arsitektur MVVM yaitu dao, database, service, repository, dan di.

##### 4.4.3.1 View Model

Peran ViewModel yaitu menangani logika dan state yang ada pada view aplikasi. Berikut ini adalah ViewModel yang digunakan dalam sistem keamanan rumah ini:

1. `LoginViewModel` digunakan untuk mengatur logika login dengan mengirim data login ke MQTT. Jika data valid maka pengguna dapat mengakses sistem keamanan rumah.

2. `RegisterViewModel` digunakan untuk mengatur registrasi pengguna dan perangkat. Data pengguna akan disimpan di dalam entitas `User` dan hasil registrasi juga akan dikirim ke mqtt untuk dilakukan registrasi perangkat pada sistem IoT.
3. `HomeViewModel` digunakan untuk menangani logika dalam mengambil data status rumah secara realtime dari MQTT.
4. `CameraViewModel` digunakan untuk menangani URL kamera RTSP.
5. `NotificationViewModel` menangani logika mengambil data notifikasi untuk daftar notifikasi yang dikirimkan oleh perangkat IoT.
6. `DetailNotificationViewModel` menangani logika mengambil data notifikasi lebih detail untuk notifikasi yang dipilih dari daftar notifikasi.
7. `BuzzerListViewModel` menangani logika untuk mengambil data buzzer dan menampilkan daftar jadwal buzzer yang ada di perangkat. `ViewModel` ini juga menangani logika dalam menambahkan, mengedit, dan status aktif buzzer.
8. `InfoConnectionViewModel` digunakan untuk memeriksa apakah pengguna sudah terhubung ke hotspot Raspberry Pi sebelum dilakukan pengaturan koneksi wifi.
9. `WifiConnectionViewModel` menangani logika dalam pengaturan koneksi wifi untuk perangkat IoT. `ViewModel` ini akan mengirimkan data ssid dan password wifi ke perangkat IoT melalui service lokal. Selain itu, `ViewModel` ini juga menangani logika dalam mengganti jaringan wifi.

#### 4.4.3.2 View

Peran View Pada sistem ini adalah untuk pengodean tampilan sistem aplikasi keamanan rumah ini. View menangani logika dalam mengambil data dari inputan pengguna yang nantinya data akan diteruskan ke `ViewModel`. Selain itu, View juga menangani logika dalam menampilkan data yang didapatkan dari `ViewModel` dengan tampilan yang interaktif dan responsif menggunakan Jetpack Compose.

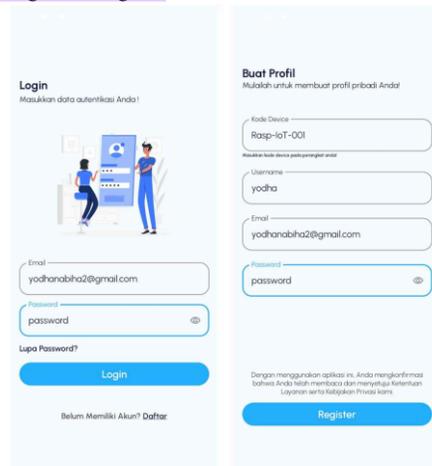
Jetpack Compose memungkinkan View untuk langsung beradaptasi terhadap perubahan data yang disediakan oleh `ViewModel` secara *real-time* melalui observasi `StateFlow` sehingga dapat menangani data yang perlu diubah secara *real-time* seperti halaman status rumah. Setiap tampilan, seperti halaman login, dashboard status rumah, daftar notifikasi, kontrol *buzzer*, serta pengaturan jaringan, dibangun menggunakan komponen `Composable`. Pada aplikasi ini, berikut ini adalah hasil tampilan-tampilan yang telah dibuat dalam pengodean view yaitu antara lain:

## 1. Halaman Landing



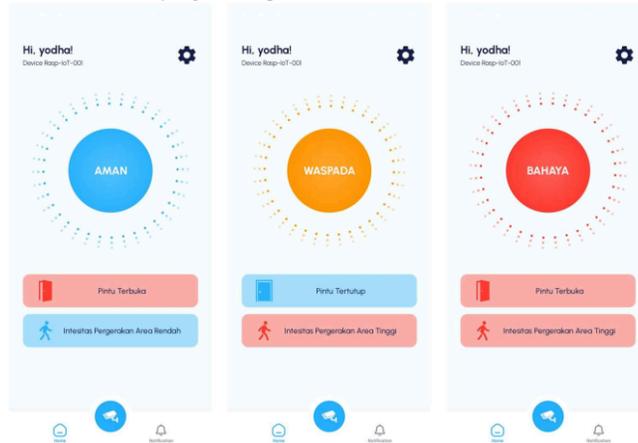
Gambar 4.19 Halaman Landing page

## 2. Halaman Login dan Register.



Gambar 4.20 Halaman login dan register

3. Dashboard Home yang menampilkan status keamanan rumah.



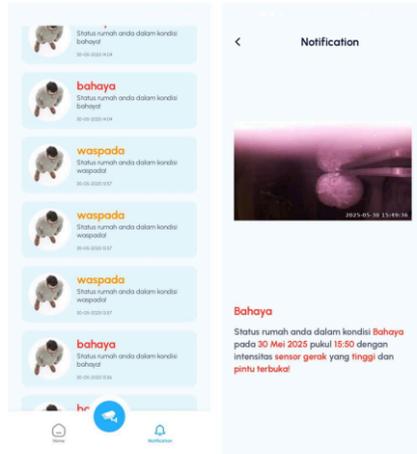
Gambar 4.21 Halaman Status keamanan rumah.

4. Tampilan kamera realtime (streaming).



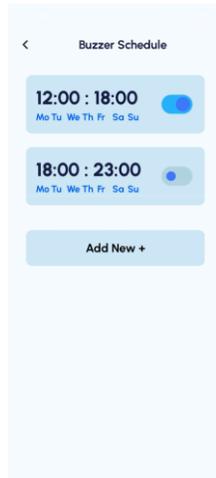
Gambar 4.22 Tampilan kamera realtime

5. Notifikasi keamanan.



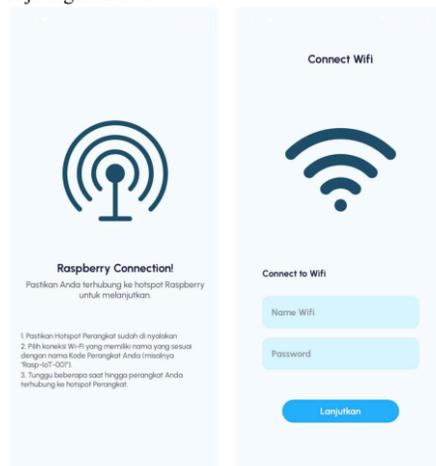
Gambar 4.23 Notifikasi dan detail notifikasi

6. Halaman penjadwalan buzzer.



Gambar 4.24 Penjadwalan buzzer

## 7. Pengaturan jaringan WiFi.



Gambar 4.25 Peringatan koneksi raspberry dan jaringan wifi

### 4.4.3.3 Model

Dalam arsitektur MVVM, layer Model tidak hanya mencakup entitas data, tetapi juga seluruh komponen yang bertugas menyediakan data, baik dari sumber lokal maupun eksternal. Model juga bertugas sebagai penyedia data yang kemudian dikelola oleh Repository dan diteruskan ke ViewModel. Pada sistem ini entitas data yang digunakan antara lain:

1. `UserEntity` digunakan untuk menyimpan data pengguna.
2. `StatusRumahEntity` digunakan untuk menyimpan status rumah (aman, waspada, bahaya), serta data dari status sensor PIR dan pintu.
3. `NotifEntity` digunakan untuk menyimpan riwayat notifikasi yang didapatkan dari perangkat Android.
4. `BuzzerEntity` digunakan untuk menyimpan pengaturan jadwal alarm berbunyi.
5. `DayEntity` digunakan untuk menyimpan hari-hari aktif dari masing-masing buzzer.

Semua entitas ini diatur dengan anotasi `@Entity` dari Room dan digunakan sebagai bagian dari database lokal.

Kemudian, untuk pengelolaan data lokal dari setiap entitas, sistem ini menggunakan pendekatan DAO (*Data Access Object*). DAO berfungsi sebagai antarmuka antara aplikasi dan database Room. Masing-masing entitas memiliki DAO-nya tersendiri, antara lain:

1. `UserDao` digunakan untuk menyimpan dan mengambil data pengguna.
2. `StatusRumahDao` mengelola penyimpanan status keamanan rumah dari perangkat IoT.
3. `NotifDao` menangani data notifikasi yang diterima dan ditampilkan kepada pengguna.
4. `BuzzerDao` mengelola data pengaturan jadwal buzzer (aktif atau tidak).
5. `DayDao` mengatur penyimpanan hari-hari yang dipilih untuk pengaktifan buzzer.

Seluruh DAO ini digunakan oleh repository untuk melakukan operasi *insert*, *update*, *delete*, dan *query* terhadap database lokal.

Selain penyimpanan lokal, aplikasi juga melakukan komunikasi dengan perangkat *IoT* menggunakan protokol MQTT. Untuk itu, terdapat dua service utama dalam sistem ini, yaitu:

1. `MqttService` yang bertanggung jawab dalam pengaturan koneksi MQTT, langganan topik (*subscribe*), serta pengiriman dan penerimaan data status rumah.
2. `MqttForegroundService` digunakan untuk menjalankan komunikasi MQTT secara berkelanjutan meskipun aplikasi sedang berada di latar belakang (*background*), dengan memanfaatkan fitur service dari Android.

Keduanya bertindak sebagai data source eksternal, karena mengambil dan mengelola data secara *real-time* dari perangkat berbasis Raspberry Pi melalui broker MQTT.

Untuk menjaga arsitektur tetap bersih dan modular, sistem ini menggunakan *pattern Repository*. `SecurityHomeRepository` berperan sebagai perantara antara `ViewModel` dan sumber data. `Repository` menggabungkan akses ke DAO untuk data lokal dan Service untuk data eksternal dalam satu lapisan abstraksi.

#### 4.4.3.4 Database

Database lokal pada aplikasi ini dikelola menggunakan library *Room*, yang merupakan bagian dari *Android Jetpack*. *Room* menyediakan layer abstraksi di atas SQLite agar interaksi dengan database.

```
@Database(
    entities = [UserEntity::class, StatusRumahEntity::class,
```

```

NotifEntity::class, BuzzerEntity::class, DayEntity::class],
    version = 1,
    exportSchema = false
)
abstract class AppDatabase : RoomDatabase() {
    abstract fun userDao(): UserDao
    abstract fun statusRumahDao(): StatusRumahDao
    abstract fun notifDao(): NotifDao
    abstract fun dayDao(): DayDao
    abstract fun buzzerDao(): BuzzerDao
    companion object {
        @Volatile
        private var INSTANCE: AppDatabase? = null

        fun getDatabase(context: Context): AppDatabase {
            return INSTANCE ?: synchronized(this) {
                val instance = Room.databaseBuilder(
                    context.applicationContext,
                    AppDatabase::class.java,
                    "security_home_db"
                ).build()
                INSTANCE = instance
                instance
            }
        }
    }
}

```

Konfigurasi **database** didefinisikan dalam *class* `AppDatabase.kt`, yang merupakan *abstract class* dan mewarisi dari *RoomDatabase*. Di dalamnya terdapat deklarasi seluruh DAO yang digunakan sistem, seperti:

1. `abstract fun userDao(): UserDao`
2. `abstract fun statusRumahDao(): StatusRumahDao`
3. `abstract fun notifDao(): NotifDao`
4. `abstract fun buzzerDao(): BuzzerDao`
5. `abstract fun dayDao(): DayDao`

`AppDatabase` diakses secara *singleton* agar tidak terjadi konflik instansiasi saat *runtime*. *Class* ini juga dihubungkan ke modul *dependency injection* agar bisa di-*inject* ke komponen lain seperti *Repository*.

*Room* secara otomatis menghasilkan kode untuk *query* berdasarkan definisi pada DAO, sehingga seluruh proses akses data dilakukan dengan aman, konsisten, dan sesuai dengan arsitektur MVVM.

#### 4.4.3.5 Di

Untuk mengatur dependensi antar komponen secara efisien dan menghindari kode *boilerplate*, sistem ini menggunakan pendekatan *Dependency Injection* (DI) dengan *library Hilt* dari *Android Jetpack*.

```

80
@InstallIn(SingletonComponent::class)
@Module
class AppModule {
    @Provides
    @Singleton
    fun provideMqttClient(@ApplicationContext context:
Context): MqttAndroidClient {
        val brokerUri = BuildConfig.MQTT_BROKER
        return MqttAndroidClient(context, brokerUri,
MqttClient.generateClientId())
    }
    23
    @Provides
    @Singleton
    fun provideDatabase(@ApplicationContext context: Context):
AppDatabase {
        return Room.databaseBuilder(
            context,
            AppDatabase::class.java,
            "security_home_db"
        ).build()
    }
}

```

Konfigurasi DI dilakukan di dalam `AppModule.kt`, yang ditandai dengan anotasi `@Module` dan `@InstallIn(SingletonComponent::class)`. Di dalam modul ini, seluruh dependensi penting didefinisikan dan disediakan secara global menggunakan anotasi `@Provides`, seperti:

1. `AppDatabase` untuk menyediakan instance database Room
2. Seluruh Dao dari database
3. `SecurityHomeRepository`
4. `MqttService`

Setiap `ViewModel` kemudian akan menerima dependensi ini melalui constructor yang diberi anotasi `@HiltViewModel`, sehingga tidak perlu membuat *instance* manual secara langsung di dalam UI.

#### 4.5 Testing

Pada tahap <sup>134</sup> *testing* pada penelitian ini akan dilakukan dengan pengujian terhadap sensor yang digunakan terlebih dahulu yaitu sensor PIR dan sensor magnetik. Kemudian, dilanjutkan dengan pengujian perangkat IoT dengan menggunakan penggabungan metode pengujian *blackbox* dan pengujian *whitebox*. Selanjutnya yaitu pengujian aplikasi android yang dilakukan dengan pengujian *blackbox*. Terakhir yaitu pengujian simulasi untuk menguji beberapa skenario dalam penerapan sistem keamanan rumah dengan menggunakan miniatur rumah untuk menggambarkan kondisi menyerupai peristiwa sebenarnya.

#### 4.5.1 Hasil Pengujian <sup>6</sup> Sensor PIR

<sup>6</sup> Pengujian dilakukan dengan menempatkan objek (manusia) pada jarak tertentu dari sensor PIR. Pengujian dilakukan dengan melihat hasil yang terdeteksi selama 5 detik dengan interval 0,5 detik untuk pembacaan sensor. Selain berdasarkan jarak, pengujian juga dilakukan dengan memvariasikan sudut kedatangan objek terhadap sensor pada sudut 0°, 40°, 80°, dan 120° dalam jarak 1 meter dari sensor PIR untuk mengevaluasi cakupan area deteksi dari sensor PIR.

Tabel 4.4 Pengujian sensor pir

No	Jarak (Cm)	Status	
		Detect	UnDetect
1	100	Motion Detected !	
2	100	Motion Detected !	
3	100	Motion Detected !	
4	100	Motion Detected !	
5	100	Motion Detected !	
6	200	Motion Detected !	
7	200	Motion Detected !	
8	200	Motion Detected !	
9	200	Motion Detected !	
10	200	Motion Detected !	
11	300	Motion Detected !	
12	300	Motion Detected !	
13	300	Motion Detected !	
14	300		No Motion Detected !
15	300		No Motion Detected !
16	400		No Motion Detected !
17	400	Motion Detected !	
18	400		No Motion Detected !
19	400	Motion Detected !	
20	400		No Motion Detected !
21	500		No Motion Detected !
22	500		No Motion Detected !

23	500		No Motion Detected !
24	500	Motion Detected !	
25	500		No Motion Detected !

Hasil pengujian memperlihatkan bahwa sensor PIR dapat mendeteksi gerakan secara akurat hingga jarak  $\pm 300$  cm. Pada jarak di atas 300–400 cm, deteksi mulai bervariasi, sesuai dengan karakteristik fisik sensor.

Tabel 4.5 Tabel respons sensor pir terhadap variasi sudut deteksi

No	Sudut	Status	
		Detect	UnDetect
1	0°	Motion Detected !	
2	0°	Motion Detected !	
3	0°	Motion Detected !	
4	0°	Motion Detected !	
5	0°	Motion Detected !	
6	40°	Motion Detected !	
7	40°	Motion Detected !	
8	40°	Motion Detected !	
9	40°	Motion Detected !	
10	40°	Motion Detected !	
11	80°	Motion Detected !	
12	80°	Motion Detected !	
13	80°	Motion Detected !	
14	80°	Motion Detected !	
15	80°	Motion Detected !	
16	120°	Motion Detected !	
17	120°	Motion Detected !	
18	120°	Motion Detected !	
19	120°	Motion Detected !	

20	120°	Motion Detected !	
----	------	-------------------	--

Berdasarkan hasil pengujian, sensor PIR mampu mendeteksi gerakan secara konsisten pada seluruh variasi sudut yang diuji, yaitu dari 0° hingga 120°. Hal ini menunjukkan bahwa sensor memiliki cakupan deteksi yang luas dan responsif terhadap pergerakan dari berbagai arah, baik dari depan maupun samping. Tidak ditemukan kegagalan deteksi pada seluruh pengujian sudut yang dilakukan.

#### 4.5.2 Hasil Pengujian Sensor Magnetik

Pengujian dilakukan dengan cara mengatur jarak antara sensor magnetik dengan magnet yang dipasang pada daun pintu, kemudian mengamati apakah perubahan kondisi terbuka atau tertutup dapat terdeteksi dan dikirimkan ke sistem.

Tabel 4.6 Pengujian sensor magnetik

No	Jarak Sensor ke Magnet (CM)	Status	
		Open	Close
1	0		Close
2	0		Close
3	0.5		Close
4	0.5		Close
5	1		Close
6	1		Close
7	1.5	Open	
8	1.5	Open	

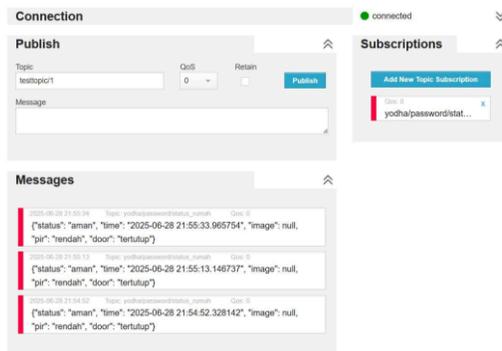
Berdasarkan hasil pengujian pada Tabel 4.6 dapat disimpulkan bahwa sensor magnet dapat mendeteksi magnet sejauh 1cm. Hasil dari tabel ini akan membantu untuk mengetahui jarak pemasangan sensor magnetik akan bekerja.

#### 4.5.3 Hasil Pengujian Perangkat IoT

Pengujian perangkat IoT dilakukan untuk memastikan bahwa seluruh proses utama pada Raspberry Pi berjalan dengan baik dan dapat terintegrasi dengan aplikasi Android. Pengujian mencakup koneksi jaringan, otentikasi pengguna, streaming video, dan pengiriman status keamanan melalui logika *fuzzy*. Pengujian dilakukan dengan

mengamati respons perangkat menggunakan MQTT *client*, VLC, dan Postman. Adapun skenario pengujiannya adalah sebagai berikut:

1. Koneksi Device ke Wi-Fi (Mode AP): Pengujian dilakukan dengan mengakses API lokal melalui Postman saat Raspberry Pi dalam mode Access Point. Data SSID dan password dikirim, lalu diamati apakah perangkat berhasil beralih ke jaringan Wi-Fi.
2. Manajemen Koneksi Wi-Fi (via MQTT): Dilakukan pengiriman konfigurasi jaringan menggunakan MQTT pada topik <username>/<password>/wifi, kemudian dicek apakah perangkat membalas ACK pada topik <username>/<password>/wifi/ack.
3. Register: Data pengguna dikirim melalui MQTT ke topik <device\_code>/auth/register, lalu diverifikasi apakah pesan balasan diterima di topik <device\_code>/auth/register/response dan data tersimpan di database SQLite.
4. Login: Data login dikirim ke topik auth/login dan diuji apakah perangkat mengirim respons valid di topik auth/login/response.
5. Live Stream Kamera: Pengujian dilakukan dengan membuka VLC dan memasukkan alamat RTSP yang sesuai. Uji berhasil jika video real-time tampil dengan lancar.
6. Fuzzy Logic dan Alarm: Sistem membaca sensor PIR dan pintu, mengirim status melalui MQTT. Jika “bahaya,” kamera mengambil gambar dan alarm menyala bila diaktifkan dari aplikasi.



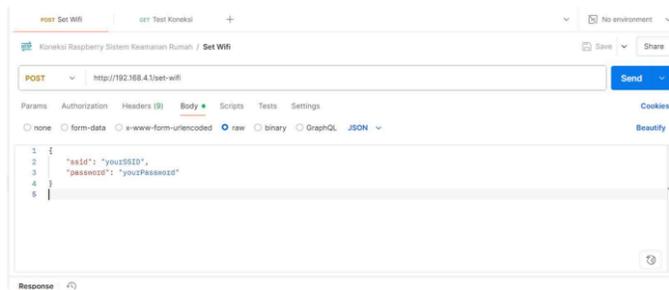
Gambar 4.26 Contoh pengujian data yang dikirim IoT ke MQTT

63 Pada gambar 4.26 merupakan contoh pengujian dengan cara melakukan *subscribe* pada topik MQTT untuk mengamati data yang dikirim oleh perangkat IoT, seperti status keamanan, status pintu, dan intensitas sensor PIR. Pengujian ini dilakukan menggunakan MQTT *client*, dengan melakukan *subscribe* atau *publish* ke topik tertentu dan mengamati pesan JSON yang dikirim oleh perangkat.



Gambar 4.27 Hasil pengujian livestreaming

Selain pengujian melalui MQTT, pengujian livestream kamera dilakukan menggunakan aplikasi VLC Media Player dengan memasukkan alamat RTSP. Pada gambar 4.27 merupakan hasil pengujian yang dilakukan dengan memasukkan alamat RTSP yang digunakan sistem untuk melakukan livestreaming.



Gambar 4.28 Pengujian api local dengan postman

Untuk pengujian konfigurasi jaringan saat perangkat dalam mode AP, digunakan aplikasi Postman. Dalam hal ini, pengujian dilakukan dengan mengakses API lokal yang disediakan oleh Raspberry Pi.

Dengan kombinasi ketiga metode ini MQTT untuk komunikasi utama, RTSP untuk pemantauan video, dan API lokal untuk konfigurasi awal pengujian sistem IoT dapat dilakukan secara menyeluruh dan mendalam sesuai dengan fungsionalitas yang dirancang. Berikut ini pada tabel 4.7 merupakan hasil dari pengujian yang telah dilakukan pada keseluruhan skenario perangkat IoT

Tabel 4.7 Pengujian alat iot

No	Proses yang diuji	Fungsi		Fungsi yang diuji
		Berfungsi	Tidak Berfungsi	
1	Manajemen Koneksi Wi-Fi	Berfungsi		10F – 9F
2	Koneksi Device ke Wifi	Berfungsi		5J – 1J – 2J – 4J – 3J
3	Register	Berfungsi		7A – 5A – 6A – 2A
4	Login	Berfungsi		7A – 1A – 3A – 4A
5	Live Stream	Berfungsi		4A – 7L
6	Fuzzy Logic dan Alarm	Berfungsi		4A – 10F – 8F – 3F – 1F – 2F – 4F – 5F – 6F – 7F

Berdasarkan hasil pengujian pada Tabel 4.7, perangkat IoT berfungsi dengan baik dalam menjalankan proses logika fuzzy serta mengirim dan menerima data. Pengujian dilakukan melalui tiga jalur utama, yaitu MQTT untuk pertukaran data status, RTSP untuk livestream kamera, dan API lokal untuk konfigurasi jaringan saat mode AP aktif. Ketiganya menunjukkan respons yang sesuai, sehingga perangkat IoT dinyatakan siap untuk diintegrasikan dengan aplikasi *Android* dalam mendukung pemantauan dan pengendalian sistem keamanan rumah secara *real-time*. Selain itu, hasil dari pengujian *whitebox* untuk memastikan bahwa fungsi-fungsi utama dalam perangkat IoT berjalan sesuai logika program yang dirancang. Setiap proses pengujian dipetakan ke fungsi-fungsi kode tertentu yang dijalankan, ditandai dengan kode *node*. Dengan begitu, selain memastikan hasil keluaran sistem, pengujian ini juga memverifikasi jalannya setiap bagian kode secara internal.

#### 4.5.4 Hasil Pengujian Aplikasi Keamanan Rumah

Pengujian dilakukan terhadap fitur pemantauan dan pengendalian keamanan rumah yang terintegrasi dalam aplikasi *Android* dengan metode pengujian *blackbox*.

Pengujian pertama dilakukan untuk memantau halaman awal aplikasi ketika baru digunakan. Pengujian dilakukan dengan melakukan alur registrasi perangkat *IoT* dengan akun yang akan digunakan dalam aplikasi. Alur ini mencakup pengaturan wifi yang akan digunakan pada perangkat, registrasi akun dan login akun yang telah dibuat.

Tabel 4.8 Pengujian halaman awal aplikasi

No	Fitur	Fungsi	
		Berfungsi	Tidak Berfungsi
1.	Koneksi Hotspot Raspberry	Terhubung Dengan Raspberry.	Tidak berfungsi ketika perangkat <i>handphone</i> tidak terhubung ke <i>hotspot</i> Raspberry Pi atau data internet tidak dimatikan terlebih dahulu.
2.	Koneksi Device ke Wifi	Terhubung dengan wifi	Tidak berfungsi ketika user salah memasukan SSID dan password wifi. Mengeluarkan <i>counter Wifi Connection Failed.</i>
3.	Register	Menyimpan data <i>form input</i>	Tidak berfungsi ketika device sudah didaftarkan dengan aplikasi sebelumnya dan mengeluarkan <i>counter username sudah dipakai!</i> . Tidak berfungsi ketika device sudah didaftarkan dan akan mengeluarkan

			counter <b>device sudah terdaftar!</b>
4.	Login	Data berhasil di verifikasi dan melakukan navigasi ke halaman <i>home</i>	Tidak berfungsi ketika user salah memasukkan email dan password akun.

Berdasarkan pengujian yang telah dilakukan pada halaman awal aplikasi (Tabel 4.8), secara keseluruhan dapat berjalan dengan baik sesuai dengan fungsinya. Fitur koneksi berhasil menghubungkan perangkat dengan *hotspot Raspberry* maupun jaringan *Wifi* selama data yang dimasukkan sudah benar. Proses *Register* dan *Login* juga berfungsi dengan baik dalam kondisi normal. Adapun beberapa kendala yang ditemukan umumnya disebabkan oleh kesalahan input pengguna, seperti salah memasukkan SSID atau *password Wifi* dan menggunakan email yang sudah terdaftar sebelumnya. Sistem secara responsif memberikan *feedback* ketika adanya kesalahan.

Pengujian selanjutnya dilakukan untuk menguji fitur halaman *home* aplikasi yang didalam halaman ini menampilkan informasi status keamanan rumah, intensitas sensor PIR, dan status pintu rumah yang diambil dari jaringan MQTT.

Tabel 4.9 Pengujian status keamanan rumah

No	Status Keamanan Rumah	Pir Status	Door Status	Time Detected	Fungsi	
					Ya	Tidak
1.	Aman	Rendah	Tertutup	30-05-2025 13:45	Berfungsi	
2.	Aman	Rendah	Tertutup	30-05-2025 13:46	Berfungsi	
3.	Aman	Rendah	Tertutup	30-05-2025 13:47	Berfungsi	
4.	Waspada	Sedang	Terbuka	30-05-2025 13:57	Berfungsi	
5.	Waspada	Sedang	Terbuka	30-05-2025	Berfungsi	

				13:57		
6.	Waspada	Sedang	Terbuka	30-05-2025 13:57	Berfungsi	
7.	Bahaya	Sedang	Terbuka	30-05-2025 14:04	Berfungsi	
8.	Bahaya	Sedang	Terbuka	30-05-2025 14:04	Berfungsi	
9.	Bahaya	Sedang	Terbuka	30-05-2025 14:04	Berfungsi	

Berdasarkan tabel pengujian 4.9 dapat disimpulkan bahwa pengujian halaman *home* aplikasi untuk mendapatkan data status rumah dan sensor dari dari MQTT telah sepenuhnya berfungsi. Sistem aplikasi sudah berhasil menampilkan informasi kepada pengguna terkait status rumah dan status sensor di halaman *home* aplikasi.

Kemudian, pengujian selanjutnya dilakukan untuk menguji fitur notifikasi. Pengujian dilakukan dengan memperhatikan notifikasi yang di terima pada perangkat *smartphone* pengguna ketika status keamanan rumah waspada atau bahaya.

Tabel 4.10 Pengujian notifikasi keamanan rumah

No	Status	Time Detected	Fungsi	
			Ya	Tidak
1.	Waspada	30-05-2025 13:57	Berfungsi	
2.	Waspada	30-05-2025 13:57	Berfungsi	
3.	Waspada	30-05-2025 13:57	Berfungsi	
4.	Bahaya	30-05-2025 14:04	Berfungsi	
5.	Bahaya	30-05-2025 14:04	Berfungsi	
6.	Bahaya	30-05-2025 14:04	Berfungsi	

Dari hasil pengujian pada tabel 4.10 dapat disimpulkan bahwa notifikasi yang diterima di perangkat *smartphone* pengguna telah sepenuhnya berfungsi. Hasil dari notifikasi akan menampilkan status keamanan pengguna jika terjadi waspada ataupun bahaya. Hasil notifikasi juga dapat disimpan di *list* notifikasi pada aplikasi untuk melihat riwayat notifikasi pengguna.

Pengujian selanjutnya yaitu untuk menguji alarm ketika kondisi rumah dalam bahaya. Pengujian dilakukan dengan menentukan jadwal alarm di aplikasi dan jadwal yang telah dibuat kemudian diaktifkan agar buzzer yang ada pada perangkat *IoT* dapat berbunyi.

Tabel 4.11 Pengujian alarm keamanan rumah

No	Jadwal		Status		Fungsi	
	Jam	Day	Aktif	Tidak Aktif	Ya	Tidak
1.	12:00 – 18:00	Senin, Selasa, Rabu, Kamis, Jumat, Sabtu, Minggu	Aktif		Berfungsi	
2.	21:00 – 04:00	Senin, Selasa, Rabu, Kamis, Jumat, Sabtu, Minggu	Aktif		Berfungsi	

Dari hasil pengujian alarm pada tabel 4.11 dapat disimpulkan bahwa fitur alarm sudah sepenuhnya berfungsi sesuai jadwal yang telah ditentukan oleh pengguna. Alarm juga tidak akan berfungsi jika memang pengguna tidak mengaktifkan atau sedang berada di luar jadwal alarm yang telah ditentukan.

#### 4.5.5 Testing Simulasi

Pada tahap pengujian sistem keamanan rumah berbasis IoT ini, dilakukan simulasi menggunakan sebuah miniatur rumah pintar yang ditampilkan pada Gambar 4.29. Miniatur rumah ini dirancang khusus untuk menguji fungsi sistem dalam skala kecil namun tetap menggambarkan kondisi nyata sebuah rumah.



Gambar 4.29 Miniatur rumah pintar

Miniatur rumah pintar dibuat menggunakan bahan karton *board*. Bahan karton *board* berfungsi sebagai fondasi utama dan rangka untuk menopang seluruh struktur miniatur. Bentuk miniatur ini sederhana tanpa pembagian ruangan, hanya berupa bangunan polos menyerupai rumah, yang memudahkan penempatan sensor-sensor dan komponen pendukung di dalamnya.

Pada gambar 4.29 dipasang beberapa perangkat utama sistem keamanan yang berfungsi untuk mendukung proses pengujian. Sensor Passive Infrared (PIR) dipasang di atas pintu bagian depan rumah untuk mendeteksi pergerakan di area akses utama. Selain itu, sensor magnetik dipasang pada bagian pintu miniatur untuk memantau kondisi pintu dalam keadaan tertutup atau terbuka. Untuk mendukung pengawasan visual, digunakan Kamera Raspberry Pi ditempatkan di atas pintu pada bagian luar, berfungsi sebagai kamera pengawas yang mengamati area depan miniature rumah secara real-time. Sementara itu, buzzer dipasang langsung pada breadboard sebagai perangkat alarm yang akan memberikan peringatan suara apabila sistem mendeteksi kondisi berbahaya.

Pengujian dilakukan pada miniatur rumah memiliki dimensi panjang 30 cm, lebar 30 cm, dan tinggi 15 cm. Model ini merepresentasikan ukuran rumah sebenarnya dengan dimensi panjang 9 meter, lebar 9 meter, dan tinggi sekitar 4,5 meter. Berdasarkan perbandingan ini, skala miniatur terhadap model nyata adalah sekitar 1:30, di mana setiap 1 cm pada miniatur mewakili 30 cm pada ukuran sebenarnya. Durasi deteksi pada setiap skenario ditetapkan selama 10 detik untuk memberikan waktu sensor PIR dalam membaca intensitas gerakan secara akurat. Setiap kondisi diuji dengan variasi intensitas gerakan yang diimplementasi menjadi rendah, sedang, dan tinggi, serta status pintu yang berbeda (terbuka atau tertutup) pada tiga periode waktu yang berbeda, yaitu siang, malam, dan dini hari. Hasil pengujian dicatat pada Tabel 4.12 berikut.

Tabel 4.12 Tabel testing simulasi

No	PIR	Pintu	Jarak (cm)	Waktu	Status	Alarm
1	Rendah	Tertutup	30	Siang	Aman	Tidak
2	Rendah	Terbuka	30	Malam	Waspada	Tidak
3	Rendah	Terbuka	30	Dini Hari	Bahaya	Aktif
4	Sedang	Tertutup	30	Siang	Waspada	Tidak
5	Sedang	Terbuka	20	Malam	Bahaya	Aktif
6	Sedang	Terbuka	20	Dini Hari	Bahaya	Aktif
7	Tinggi	Tertutup	10	Siang	Waspada	Tidak

8	Tinggi	Terbuka	10	Malam	Bahaya	Aktif
9	Tinggi	Terbuka	10	Dini Hari	Bahaya	Aktif

Berdasarkan Tabel 4.9, dapat disimpulkan bahwa sistem keamanan rumah berbasis IoT yang diuji menggunakan miniatur rumah mampu merespon berbagai skenario kondisi dengan baik. Terlihat bahwa kombinasi antara intensitas gerakan (PIR) dan status pintu berhasil diproses oleh sistem fuzzy logic untuk menghasilkan status keamanan yang sesuai, yaitu *Aman*, *Waspada*, dan *Bahaya*. Selain itu, pada kondisi *Bahaya*, sistem juga secara otomatis mengaktifkan alarm (buzzer), menunjukkan bahwa integrasi antara sensor, kamera berjalan sesuai dengan rancangan. Variasi waktu pengujian (siang, malam, dini hari) juga tidak menghambat kinerja sistem dalam mendeteksi perubahan kondisi keamanan. Dengan demikian, hasil pengujian simulasi ini menunjukkan bahwa sistem telah berfungsi secara efektif sesuai dengan skenario yang diharapkan.

#### 4.6 Evaluasi

Berdasarkan hasil pengujian yang telah dilakukan pada seluruh komponen sistem, mulai dari sensor PIR, sensor magnetik, perangkat IoT berbasis Raspberry Pi, serta aplikasi Android yang dikembangkan, dapat disimpulkan bahwa sistem keamanan rumah berbasis IoT ini telah berjalan sesuai dengan rancangan dan tujuan penelitian.

##### 4.6.1 Evaluasi Sensor PIR

Pengujian sensor PIR dilakukan dengan menempatkan objek manusia pada jarak bertahap dari sensor, dengan interval 0,5 detik dan objek bergerak selama 5 detik pada setiap jarak. Berdasarkan datasheet dari MPJA (n.d.), sensor PIR yang digunakan memiliki jarak deteksi hingga 7 meter dalam kondisi ideal. Namun, hasil pengujian menunjukkan bahwa:

1. Jarak 1-3 meter: tingkat keberhasilan deteksi mencapai 100%.
2. Jarak 3-4 meter: tingkat keberhasilan mencapai 60%.
3. Jarak 4-5 meter: deteksi turun menjadi 30%.

Berdasarkan hasil pengujian sensor PIR dapat disimpulkan bahwa pada jarak 1-3 meter sensor dapat mendeteksi dengan stabil. Kemudian, pada jarak 3-4 meter sensor pir mulai tidak konsisten. Selanjutnya pada jarak 4-5 meter tingkat pendeteksi sensor semakin menurun dengan banyaknya objek yang tidak terdeteksi. Perbedaan ini kemungkinan dipengaruhi oleh beberapa faktor, antara lain pantulan sinyal inframerah yang berbeda dan kecepatan sampling yang diterapkan dalam sistem (sampling tiap 0.7 detik, akumulasi 5 detik).

Selain pengujian jarak, sensor PIR juga diuji berdasarkan sudut kedatangan objek dengan variasi sudut 0°, 40°, 80°, dan 120°. Hasil menunjukkan bahwa pada seluruh sudut yang diuji, sensor berhasil mendeteksi gerakan dengan tingkat keberhasilan 100%. Hal ini menunjukkan bahwa sensor memiliki cakupan deteksi sudut yang luas dan tetap responsif terhadap pergerakan dari berbagai arah, baik dari depan maupun samping. Cakupan ini sangat mendukung efektivitas sistem keamanan rumah karena tidak hanya mendeteksi objek yang datang dari arah depan sensor, tetapi juga dari arah menyamping.

Dikarenakan pengujian sensor PIR dilakukan pada miniatur rumah dengan skala 1:30, maka jika sistem ini diterapkan pada skala nyata (1:1), diperlukan konfigurasi ulang terhadap pengaturan sensor. Konfigurasi ulang ini meliputi penyesuaian posisi dan sudut pemasangan sensor, serta penyesuaian parameter jarak agar sesuai dengan dimensi ruang yang sebenarnya. Hal ini dilakukan agar jangkauan deteksi sensor tetap optimal dan tidak terjadi kesalahan dalam pembacaan gerakan pada lingkungan rumah yang sesungguhnya.

#### 4.6.2 Evaluasi Sensor Magnetik

Sensor magnetik menunjukkan performa yang sangat baik dan konsisten dalam mendeteksi kondisi pintu. Jarak optimal pemasangan (berdasarkan pengujian Tabel 4.3) adalah  $\leq 1$  cm, sesuai dengan karakteristik sensor. Pada jarak di atas 1.5 cm, status menjadi open. Hal ini sesuai ekspektasi dan sesuai dengan datasheet (MPJA, n.d.) sensor magnetik.

#### 4.6.3 Evaluasi Perangkat IoT dan Proses Fuzzy

Perangkat IoT yang dikembangkan menggunakan Raspberry Pi telah diuji terhadap enam aspek utama yang menjadi inti dari fungsionalitas sistem. Hasil pengujian sebagaimana ditunjukkan pada Tabel 4.7 menunjukkan bahwa seluruh fungsi penting dari perangkat berjalan dengan baik. Raspberry Pi berhasil terhubung ke jaringan WiFi dan dapat menerima konfigurasi dari aplikasi Android. Proses registrasi dan login juga berfungsi sebagaimana mestinya, serta data pengguna berhasil dikirim melalui MQTT. Fitur live streaming dari kamera berjalan lancar di jaringan lokal, meskipun terdapat delay sekitar 5–15 detik saat diakses melalui jaringan publik. Sistem juga berhasil menjalankan logika fuzzy Mamdani dalam mengklasifikasikan status keamanan rumah, dan hasilnya dapat dikirim ke aplikasi Android secara real-time.

Namun, pada kondisi jaringan lemah, pengiriman balasan (“ACK”) dari aplikasi *Android* menjadi terhambat, sehingga perangkat melakukan pengiriman ulang pesan. Hal ini sudah ditangani dengan baik oleh sistem melalui mekanisme `unsent_messages`. Secara keseluruhan, perangkat *IoT* mampu menjalankan seluruh

proses yang dirancang, mulai dari koneksi, komunikasi, pengolahan data, hingga pengiriman informasi status keamanan ke pengguna.

#### 4.6.4 Evaluasi Aplikasi Android

Aplikasi Android yang dikembangkan dalam penelitian ini telah berhasil menjalankan berbagai fungsi utama sesuai dengan rancangan sistem. Berdasarkan hasil pengujian, aplikasi mampu menampilkan status keamanan rumah secara real-time pada halaman *home*, dengan update status yang diterima melalui protokol MQTT dari perangkat Raspberry Pi. Status keamanan ditampilkan dengan indikator yang jelas, sehingga memudahkan pengguna untuk mengetahui kondisi rumah setiap saat.

Selain itu, aplikasi berhasil memberikan notifikasi secara otomatis ketika status rumah berada dalam kondisi Waspada atau Bahaya. Notifikasi muncul dengan tepat waktu di perangkat Android saat status dikirimkan oleh perangkat IoT. Pada fitur pemantauan visual, aplikasi mampu menampilkan livestream kamera berbasis RTSP secara stabil ketika perangkat terhubung ke jaringan WiFi lokal. Tampilan video berjalan dengan lancar, dan tidak ditemukan gangguan signifikan selama pengujian di jaringan lokal.

Fitur pengaturan jaringan WiFi untuk perangkat IoT juga berjalan dengan baik. Melalui aplikasi, pengguna dapat memilih dan mengatur jaringan WiFi yang akan digunakan oleh Raspberry Pi. Proses pengaturan berlangsung dengan respons yang cepat, dan perangkat dapat tersambung ke jaringan yang baru sesuai dengan input pengguna. Selain itu, fitur pengaturan dan pengontrolan alarm sesuai jadwal yang telah ditentukan oleh pengguna juga berfungsi sebagaimana mestinya. Alarm dapat diaktifkan atau dinonaktifkan berdasarkan jadwal yang sudah diatur melalui aplikasi, dan perangkat IoT merespon pengaturan tersebut dengan tepat.

Pada pengujian *livestream* RTSP yang diakses melalui jaringan publik (menggunkan server MediaMTX), ditemukan adanya *latency* tambahan antara 5 hingga 15 detik, bergantung pada kondisi kualitas jaringan internet yang digunakan saat pengujian. Meskipun terdapat *delay*, *livestream* tetap dapat diakses dengan lancar dan cukup stabil untuk kebutuhan pemantauan keamanan rumah secara jarak jauh. Secara keseluruhan, aplikasi Android telah memenuhi fungsi yang dirancang, baik dalam aspek tampilan status, notifikasi keamanan, pengaturan jaringan, pengontrolan alarm, maupun pemantauan visual melalui *livestream* kamera.

#### 4.6.5 Evaluasi Simulasi Sistem

Berdasarkan hasil simulasi menggunakan miniatur rumah pintar, sistem menunjukkan performa yang sesuai dengan skenario yang dirancang. Kombinasi antara intensitas gerakan dari sensor PIR dan status pintu berhasil diproses dengan baik oleh logika fuzzy, sehingga mampu menghasilkan klasifikasi status keamanan yang sesuai,

yaitu Aman, Waspada, dan Bahaya. Sistem juga dapat mengaktifkan buzzer secara otomatis saat kondisi diklasifikasikan sebagai Bahaya. Respon sistem terhadap berbagai waktu pengujian (siang, malam, dan dini hari) tidak mengalami penurunan kinerja yang signifikan. Selain itu, integrasi antar perangkat seperti sensor, kamera, dan alarm juga berjalan lancar selama pengujian.

Namun demikian, terdapat beberapa keterbatasan pada sistem. Salah satunya adalah ketidakmampuan sistem dalam membedakan jenis gerakan yang terjadi di sekitar rumah. Sistem hanya mengandalkan intensitas gerakan dan status pintu untuk menentukan kondisi keamanan, tanpa memperhitungkan konteks. Misalnya, jika ada orang yang hanya lewat di depan rumah tanpa berniat masuk, sistem tetap menganggapnya sebagai ancaman dan akan mengirimkan notifikasi ke pengguna. Hal ini berpotensi menimbulkan notifikasi berlebih (*spam*), terutama pada rumah yang berada di lingkungan ramai. Meskipun begitu, secara keseluruhan, sistem telah berfungsi dengan baik dalam mendeteksi kondisi yang dirancang dan dapat dijadikan dasar untuk pengembangan lebih lanjut dengan kemampuan klasifikasi objek atau integrasi teknologi pengenalan wajah untuk mengurangi kesalahan deteksi.

## <sup>87</sup> BAB 5. KESIMPULAN

### 5.1 Kesimpulan

Berdasarkan hasil implementasi, pengujian, dan evaluasi terhadap sistem keamanan rumah berbasis IoT dengan algoritma Fuzzy Logic Mamdani dan pemantauan real-time melalui aplikasi Android, maka diperoleh kesimpulan sebagai berikut:

1. Sistem berhasil dirancang dan diimplementasikan dengan mengintegrasikan sensor Passive Infrared (PIR), sensor magnetik, Raspberry Pi, kamera Raspberry Pi, dan *buzzer*. Algoritma Fuzzy Logic Mamdani diterapkan untuk menentukan status keamanan rumah dalam tiga kategori, yaitu Aman, Waspada, dan Bahaya. Proses inferensi fuzzy berjalan secara real-time dan hasilnya dapat dikirim ke aplikasi Android menggunakan protokol MQTT.
2. Aplikasi Android yang dikembangkan menggunakan arsitektur MVVM dan Jetpack Compose mampu menampilkan status keamanan rumah secara real-time, memberikan notifikasi peringatan kepada pengguna, serta menyediakan fitur livestream kamera berbasis RTSP. Selain itu, pengguna juga dapat mengatur jadwal aktifasi alarm melalui aplikasi.
3. Berdasarkan pengujian *Blackbox* yang dilakukan, baik pada sensor PIR, sensor magnetik, perangkat *IoT*, maupun aplikasi *Android*, seluruh fitur sistem berjalan dengan baik dan sesuai dengan rancangan. Pada pengujian simulasi dengan miniatur rumah, sistem mampu memberikan respon yang akurat terhadap berbagai kombinasi skenario yang diuji, menunjukkan bahwa sistem efektif dalam meningkatkan keamanan rumah.

Sistem ini telah berhasil memmbagun perangkat *IoT* yang mampu melakukan deteksi kondisi keadaan rumah dan mengambil keputusan dengan menggunakan algoritma *fuzzy logic* mamdani. Sistem ini juga telah terintegrasi dalam perangkat *Android* dan mampu memberikan notifikasi serta pemantauan *real-time*. Namun demikian, sistem ini masih memiliki beberapa keterbatasan, seperti belum mampu membedakan jenis objek yang terdeteksi (manusia, hewan, atau benda lain), serta potensi munculnya notifikasi berlebih (*spam*) pada area rumah yang ramai. Hal ini menjadi peluang untuk pengembangan lebih lanjut dengan menambahkan fitur klasifikasi objek atau integrasi teknologi pengenalan wajah guna meningkatkan akurasi deteksi ancaman.

### 5.2 Saran

Meskipun sistem telah berjalan dengan baik, <sup>16</sup>terdapat beberapa hal yang dapat dijadikan saran untuk pengembangan lebih lanjut, yaitu:

1. Optimalisasi Performa Sistem, dapat dilakukan optimasi pada efisiensi pemrosesan Raspberry Pi, terutama saat menangani proses inferensi fuzzy dan pengiriman data secara bersamaan agar sistem dapat merespon lebih cepat.
2. Peningkatan Keamanan Jaringan, keamanan komunikasi data antara perangkat IoT dan aplikasi Android melalui protokol MQTT dapat ditingkatkan dengan implementasi enkripsi TLS yang lebih kuat serta autentikasi yang lebih ketat.
3. Pengembangan Fitur Tambahan, sistem dapat dikembangkan lebih lanjut dengan menambahkan fitur pengenalan wajah berbasis kamera untuk meningkatkan tingkat kecerdasan sistem dalam mendeteksi ancaman yang lebih spesifik.
4. Pengujian di Lingkungan Nyata, pengujian di lingkungan rumah sebenarnya pada beberapa lokasi berbeda dengan kondisi jaringan dan bangunan yang beragam, agar sistem dapat lebih siap diimplementasikan secara luas di masyarakat.

## DAFTAR PUSTAKA

- Manullang, E. V., & Rumere, S. (2024). Rancang Bangun Sistem Keamanan Rumah Berbasis Internet Of Things (Iot) Dengan Menggunakan Sensor Pir Dan Esp32-Cam. *Jurnal Teknologi Informasi*, 12(1), 9-15.
- Effendi, M. M., & Juliyanto, H. A. (2021). Aplikasi Sistem Keamanan Rumah Berbasis Internet Of Things Dengan Menggunakan Raspberry Pi. *Jurnal SIGMA*, 12(3), 133-138.
- Supriyadi, S., Nugraha, N., & Nugraha, E. L. (2020). Rancang Bangun Sistem Keamanan Rumah dengan Metode Fuzzy Logic menggunakan Mikrokontroler berbasis Android. *JEJARING: Jurnal Teknologi dan Manajemen Informatika*, 5(2), 18-25.
- Kurniasih, W., Rakhman, A., & Salamah, I. (2020). Sistem Keamanan Pintu dan Jendela Rumah Berbasis IoT. *Jurasik (Jurnal Riset Sistem Informasi dan Teknik Informatika)*, 5(2), 266-274.
- Putra, I. P. R. P., & Tolle, H. (2023). Pengembangan Aplikasi Pembelajaran Bahasa Bali berbasis Android menggunakan MVVM Architecture dan Jetpack Compose. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 7(5), 2205-2214.
- Samsinar, R., Septian, R., & Fadliandi, F. (2020). Alat Monitoring Suhu Kelembapan dan Kecepatan Angin dengan Akuisisi Database Berbasis Raspberry Pi. *RESISTOR (Elektronika Kendali Telekomunikasi Tenaga Listrik Komputer)*, 3(1), 29-36.
- Libertin, D. (2020). Sistem Pemantauan Ruang Laboratorium Dengan Raspberry Pi Camera. *Electricres: Journal of Electrical Automation and Renewable Energy*, 2(1), 11-16.
- Rachmawati, M. D., Rakhman, A., & Handayani, A. S. (2021). Perancangan Sistem Keamanan Pintar Kamera Night Vision Auto Color berbasis Raspberry Pi. *Jurnal Qua Teknika*, 11(2), 81-89.
- Juliansyah, A., Ramlah, R., & Nadiani, D. (2021). Sistem Pendeteksi Gerak Menggunakan Sensor PIR dan Raspberry Pi. *JTIM: Jurnal Teknologi Informasi Dan Multimedia*, 2(4), 199-205.
- Rahayu, D. S. (2024). Pemantauan Real-Time Dan Sistem Alarm Otomatis Pada Proteksi Transformer Menggunakan Arduino Nano Dan Sensor Pir. *PROSISKO: Jurnal Pengembangan Riset dan Observasi Sistem Komputer*, 11(2), 301-309.
- Setyawan, H. P., Ahmad, E., Muharror, A., Aryanto, V., Mufti, D., Laksono, B. Y., & Pambudi, W. S. (2022, August). Implementasi Fuzzy Logic Pada Produksi Resep Pembagian Kopi. In *Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi)* (Vol. 6, No. 1, pp. 031-035).
- Septima, R. (2023). *Buku Referensi Sistem Inferensi Fuzzy dengan Metode Mamdani*.
- Aditya, F., Putra, A. D., & Surahman, A. (2022). Rancang Bangun Aplikasi Penjualan Berbasis

Android (Studi Kasus: Pada Toko Murah Jaya Alumunium). *Jurnal Informatika dan Rekayasa Perangkat Lunak*, 3(3), 316-329.

Siswandi, A., & Muhidin, A. (2022). Sistem Informasi Aplikasi Sewa Gedung Wilayah Karawang Berbasis Android. *Jurnal SIGMA*, 13(4), 199-206.

Diantoni, C., Komarudin, O., & Rizal, A. (2024). Arsitektur Mvvm Dan Framework Jetpack Compose Pada Pengembangan Aplikasi Android. *Jati (Jurnal Mahasiswa Teknik Informatika)*, 8(3), 3216-3224.

Zoromi, F., Rio, U., & Nasution, T. (2024). Aplikasi Penjualan Produk It Dan Jasa Service Berbasis Android Dengan Metode Mvvm. *Jurnal Tekinkom (Teknik Informasi dan Komputer)*, 7(1), 62-71.

Arrizal, M., & Rizal, A. (2024). Aplikasi Berbagi Makanan Berbasis Android Menggunakan Clean Architecture Dan Jetpack Compose. *Jurnal Informatika dan Teknik Elektro Terapan*, 12(3).

AryaRafa, D., Wahyuni, E. D., & Arifiyanti, A. A. (2024). Rancang Bangun Aplikasi Donor Darah Darurat Donora Berbasis Android Dengan Konsep Gamifikasi Menggunakan Kotlin. *Jurnal Informatika dan Teknik Elektro Terapan*, 12(3).

Nugraha, T. S., Kusnadi, K., & Hardian, R. (2021). Rancang Bangun Sistem Informasi Company Profile dengan Menggunakan Metode Scrum pada PT. Hasna Satya Negara Berbasis Web. *Jurnal Ilmiah Intech: Information Technology Journal of UMUS*, 3(02), 171-179.

Amarta, A. A. F., & Anugrah, I. G. (2021). Implementasi Agile Scrum Dengan Menggunakan Trello Sebagai Manajemen Proyek Di PT Andromedia. *Jurnal Nasional Komputasi dan Teknologi Informasi*, 4(6), 528-34.

Fahrezi, A., Salam, F. N., Ibrahim, G. M., Syaiful, R. R., & Saifudin, A. (2022). Pengujian Black Box Testing pada Aplikasi Inventori Barang Berbasis Web di PT. AINO Indonesia. *LOGIC: Jurnal Ilmu Komputer dan Pendidikan*, 1(1), 1-5.

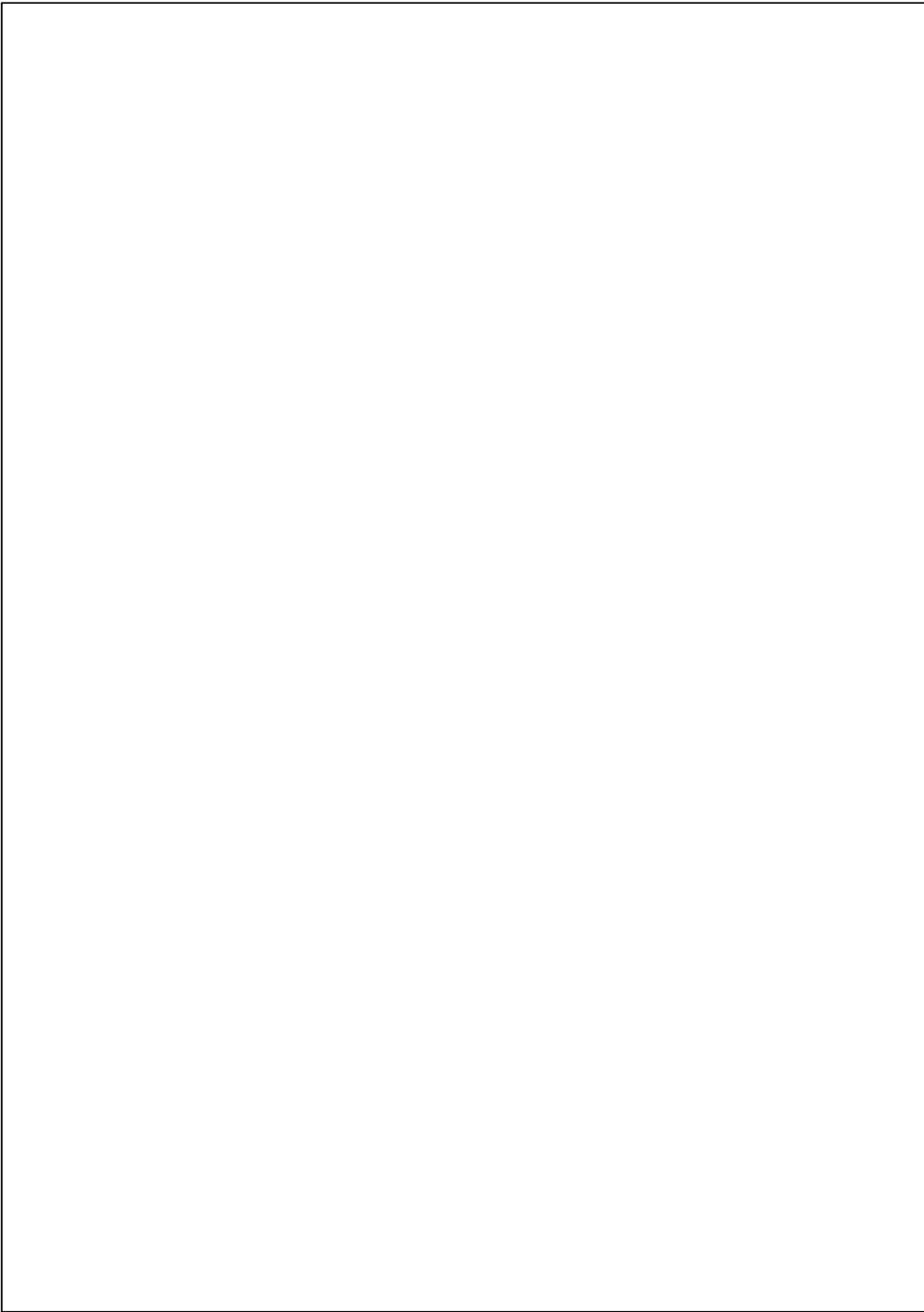
Saputra, D. I., Karmel, G. M., & Zainal, Y. B. (2020). Perancangan Dan Implementasi Rapid Temperature Screening Contactless Dan Jumlah Orang Berbasis Iot Dengan Protokol Mqtt. *Journal of Energy and Electrical Engineering*, 2(1).

Rifandi, R., & Sutarti, S. (2021). Rancang bangun kamera pengawas menggunakan raspberry dengan aplikasi telegram berbasis internet of things. *PROSISKO: Jurnal Pengembangan Riset dan Observasi Sistem Komputer*, 8(1), 18-32.

Coyanda, J. R., Ariati, N., Sunardi, H., & Ghazali, K. G. K. (2022). Sistem Informasi Ekspedisi Barang Pada PT. New Power Global Energy Dengan Menggunakan Metode Extreme Programming. *Jurnal Ilmiah Informatika Global*, 13(2).

Ariyanti, L., Satria, M. N. D., & Alita, D. (2020). Sistem Informasi Akademik Dan Administrasi Dengan Metode Extreme Programming Pada Lembaga Kursus Dan Pelatihan. *Jurnal Teknologi dan Sistem Informasi*, 1(1), 90-96.

- Nurkholis, A., Susanto, E. R., & Wijaya, S. (2021). Penerapan Extreme Programming dalam Pengembangan Sistem Informasi Manajemen Pelayanan Publik. *J-SAKTI (Jurnal Sains Komputer Dan Informatika)*, 5(1), 124-134.
- Utami, F., Suhendri, S., & Mujib, M. A. (2021). Implementasi Algoritma Haar Cascade Pada Aplikasi Pengenalan Wajah Personel. *Journal of Information Technology*, 3(1), 33-38.
- Hermawan, R. (2024). Implementasi Keamanan Rumah Cerdas Menggunakan Sensor Magnetic Door. *Cyber, Education and Research*, 3(1).
- Gulo, S., Suherdi, D., & Yetri, M. (2022). Rancang Bangun Sistem Keamanan Rumah Menggunakan Telegram Berbasis Nodemcu. *Jurnal Sistem Komputer Triguna Dharma (JURSIK TGD)*, 1(4), 137-141.
- Bazewew, M. B. (2023). Home Surveillance and Alert System using Raspberry Pi Zero W and GSM Modem with MQTT Protocol. *Indian Journal of Artificial Intelligence and Neural Networking (IJAINN)*, 3(2), 1-7.
- Badan Pusat Statistik. (2023). *Statistik kriminal 2023* (Vol. 14). Badan Pusat Statistik.
- Gultom, H. E. P. (2024, Juni 7). Jakarta Timur ranking 3 rawan kejahatan, patroli besar digelar di kawasan industri Pulogadung. *Tribunnews*.  
[https://www.tribunnews.com/metropolitan/2024/06/07/jakarta-timur-ranking-3-rawan-kejahatan-patroli-besar-digelar-di-kawasan-industri-pulogadung#google\\_vignette](https://www.tribunnews.com/metropolitan/2024/06/07/jakarta-timur-ranking-3-rawan-kejahatan-patroli-besar-digelar-di-kawasan-industri-pulogadung#google_vignette).
- Kompas. (2024, Maret 20). Detik-detik Pencurian Brankas di Rumah Ciracas: Pelaku Panjat Pagar dan Bobol Pintu, lalu Tersenyum Usai Beraksi. *Kompas*  
<https://megapolitan.kompas.com/read/2024/03/20/10521861/detik-detik-pencurian-brankas-di-rumah-ciracas-pelaku-panjat-pagar-dan>.
- BelChip. (n.d.). MC-38 wired door window sensor magnetic switch [Datasheet].  
<https://belchip.by/sitedocs/31153.pdf>.
- MPJA. (n.d.). HC-SR501 PIR motion detector sensor module [Datasheet].  
<https://www.mpja.com/download/31227sc>
- Londjo, M. F. (2021). Implementasi White Box Testing Dengan Teknik Basis Path Pada Pengujian Form Login. *Jurnal Siliwangi Seri Sains dan Teknologi*, 7(2).



# RANCANG BANGUN SISTEM KEAMANAN RUMAH BERBASIS IOT MENGGUNAKAN ALGORITMA FUZZY LOGIC MAMDANI DENGAN PEMANTAUAN REAL-TIME MELALUI APLIKASI ANDROID

ORIGINALITY REPORT



14%

SIMILARITY INDEX

12%

INTERNET SOURCES

6%

PUBLICATIONS

6%

STUDENT PAPERS

PRIMARY SOURCES

1	<a href="https://repository.uin-suska.ac.id">repository.uin-suska.ac.id</a> Internet Source	1%
2	<a href="https://repository.teknokrat.ac.id">repository.teknokrat.ac.id</a> Internet Source	<1%
3	<a href="https://docplayer.info">docplayer.info</a> Internet Source	<1%
4	Mohamad Arrizal, Adhi Rizal. "APLIKASI BERBAGI MAKANAN BERBASIS ANDROID MENGGUNAKAN CLEAN ARCHITECTURE DAN JETPACK COMPOSE", Jurnal Informatika dan Teknik Elektro Terapan, 2024 Publication	<1%
5	<a href="https://repository.ub.ac.id">repository.ub.ac.id</a> Internet Source	<1%
6	<a href="https://123dok.com">123dok.com</a> Internet Source	<1%
7	Submitted to Universitas Brawijaya Student Paper	<1%
8	<a href="https://jurnal.unsil.ac.id">jurnal.unsil.ac.id</a> Internet Source	<1%
9	<a href="https://text-id.123dok.com">text-id.123dok.com</a> Internet Source	<1%

[ojs.stmikdharmapalariau.ac.id](https://ojs.stmikdharmapalariau.ac.id)

10	Internet Source	<1 %
11	forum.raspberry-pi.fr Internet Source	<1 %
12	repository.untag-sby.ac.id Internet Source	<1 %
13	journal.lembagakita.org Internet Source	<1 %
14	Submitted to Hankuk University of Foreign Studies Student Paper	<1 %
15	e-jurnal.lppmunsera.org Internet Source	<1 %
16	id.123dok.com Internet Source	<1 %
17	daily.dev Internet Source	<1 %
18	eprints.uny.ac.id Internet Source	<1 %
19	Submitted to University College London Student Paper	<1 %
20	jurnal.murnisadar.ac.id Internet Source	<1 %
21	dspace.vut.cz Internet Source	<1 %
22	Submitted to Universitas Muslim Indonesia Student Paper	<1 %
23	medium.com Internet Source	<1 %
24	hdl.handle.net Internet Source	<1 %

<1 %

25

[repository.poliupg.ac.id](https://repository.poliupg.ac.id)

Internet Source

<1 %

26

Moh. Anshori Aris Widya, Yuda Adi Wijaya.  
"Aplikasi Monitoring Unsur Hara Berbasis  
Android", Exact Papers in Compilation (EPiC),  
2023

Publication

<1 %

27

[etheses.uin-malang.ac.id](https://etheses.uin-malang.ac.id)

Internet Source

<1 %

28

[repository.its.ac.id](https://repository.its.ac.id)

Internet Source

<1 %

29

Submitted to LL Dikti IX Turnitin Consortium

Student Paper

<1 %

30

[ejournal.itn.ac.id](https://ejournal.itn.ac.id)

Internet Source

<1 %

31

[jim.teknokrat.ac.id](https://jim.teknokrat.ac.id)

Internet Source

<1 %

32

Daud Elia Leander, Muhammad Fikry  
Ashidiqie, Kodar Udoyono. "PERANCANGAN  
SISTEM MONITORING JARAK JAUH PINTU  
PINTAR RUMAH INDEKOS BERBASIS IOT  
(INTERNET OF THINGS) MENGGUNAKAN  
PLATFORM BLYNK", Jurnal Teknologi  
Informasi dan Komunikasi, 2024

Publication

<1 %

33

[dspace.uui.ac.id](https://dspace.uui.ac.id)

Internet Source

<1 %

34

[www.classace.io](https://www.classace.io)

Internet Source

<1 %

Submitted to algebra

35	Student Paper	<1 %
36	<a href="http://asfiaafga193035.blogspot.com">asfiaafga193035.blogspot.com</a> Internet Source	<1 %
37	<a href="http://e-journal.hamzanwadi.ac.id">e-journal.hamzanwadi.ac.id</a> Internet Source	<1 %
38	<a href="http://eprints.itenas.ac.id">eprints.itenas.ac.id</a> Internet Source	<1 %
39	<a href="http://eprints.walisongo.ac.id">eprints.walisongo.ac.id</a> Internet Source	<1 %
40	<a href="http://jurnal.umj.ac.id">jurnal.umj.ac.id</a> Internet Source	<1 %
41	Submitted to Forum Perpustakaan Perguruan Tinggi Indonesia Jawa Tengah Student Paper	<1 %
42	Submitted to Multiversity Student Paper	<1 %
43	<a href="http://academy.hsoub.com">academy.hsoub.com</a> Internet Source	<1 %
44	<a href="http://bif.telkomuniversity.ac.id">bif.telkomuniversity.ac.id</a> Internet Source	<1 %
45	<a href="http://journal.uniku.ac.id">journal.uniku.ac.id</a> Internet Source	<1 %
46	Submitted to Institut Bisnis dan Teknologi Indonesia (INSTIKI) Student Paper	<1 %
47	Submitted to Universitas Siliwangi Student Paper	<1 %
48	<a href="http://begawe.unram.ac.id">begawe.unram.ac.id</a> Internet Source	<1 %

49	<a href="http://eprints.uty.ac.id">eprints.uty.ac.id</a> Internet Source	<1 %
50	<a href="http://journals.ukitoraja.ac.id">journals.ukitoraja.ac.id</a> Internet Source	<1 %
51	<a href="http://jurnal.ardenjaya.com">jurnal.ardenjaya.com</a> Internet Source	<1 %
52	<a href="http://repository.radenfatah.ac.id">repository.radenfatah.ac.id</a> Internet Source	<1 %
53	<a href="http://www.contohblog.com">www.contohblog.com</a> Internet Source	<1 %
54	Submitted to New College of the Humanities Student Paper	<1 %
55	<a href="http://afcgn.org">afcgn.org</a> Internet Source	<1 %
56	<a href="http://eprints.untirta.ac.id">eprints.untirta.ac.id</a> Internet Source	<1 %
57	<a href="https://github.com">github.com</a> Internet Source	<1 %
58	<a href="http://prosiding.pnj.ac.id">prosiding.pnj.ac.id</a> Internet Source	<1 %
59	<a href="http://www.neliti.com">www.neliti.com</a> Internet Source	<1 %
60	<a href="http://jurnal.ranahresearch.com">jurnal.ranahresearch.com</a> Internet Source	<1 %
61	<a href="http://pt.scribd.com">pt.scribd.com</a> Internet Source	<1 %
62	<a href="http://rakyatmerdekanews.com">rakyatmerdekanews.com</a> Internet Source	<1 %
63	<a href="http://repository.dinamika.ac.id">repository.dinamika.ac.id</a> Internet Source	<1 %

64 Ibnu Alfitra Salam, Kamal Prihandani, Intan Purnamasari. "RANCANG BANGUN APLIKASI PROFIT PENJUALAN MOTOR BERBASIS DESKTOP KONSEP ARSITEKTUR MODEL VIEW CONTROLLER (MVC)", Jurnal Informatika dan Teknik Elektro Terapan, 2023

Publication

<1 %

65 Muhammad Naufal Hady Anshari Jaelani, Yuli Asriningtias. "Optimalisasi Aplikasi Financial Tracker berbasis Mobile dengan Penerapan Design Pattern MVVM untuk Mengelola Keuangan", Edumatic: Jurnal Pendidikan Informatika, 2024

Publication

<1 %

66 Submitted to University of Essex

Student Paper

<1 %

67 jih.akamigas.ac.id

Internet Source

<1 %

68 Submitted to Universitas Gadjah Mada

Student Paper

<1 %

69 Submitted to University of Carthage

Student Paper

<1 %

70 jptam.org

Internet Source

<1 %

71 Submitted to Kingston University

Student Paper

<1 %

72 Mohammad Hanif Yuhdi, Anggi Indah Yuliana, Sujono Sujono. "Rancang Bangun Alat Deteksi dan Pengendali Hama Burung Pipit Pada Pertanaman Padi Sawah Berbasis Arduino WeMos ESP8266", Exact Papers in Compilation (EPiC), 2023

Publication

<1 %

73	Submitted to UIN Syarif Hidayatullah Jakarta Student Paper	<1 %
74	Submitted to Universitas Pendidikan Ganesha Student Paper	<1 %
75	<a href="http://digilib.uin-suka.ac.id">digilib.uin-suka.ac.id</a> Internet Source	<1 %
76	<a href="http://etd.iain-padangsidempuan.ac.id">etd.iain-padangsidempuan.ac.id</a> Internet Source	<1 %
77	<a href="http://www.ijeat.org">www.ijeat.org</a> Internet Source	<1 %
78	Submitted to Universitas Dian Nuswantoro Student Paper	<1 %
79	Submitted to Universiti Teknologi Petronas Student Paper	<1 %
80	Submitted to University of Northampton Student Paper	<1 %
81	<a href="http://journal.ittelkom-sby.ac.id">journal.ittelkom-sby.ac.id</a> Internet Source	<1 %
82	<a href="http://ojs.unud.ac.id">ojs.unud.ac.id</a> Internet Source	<1 %
83	<a href="http://repositorio.utc.edu.ec">repositorio.utc.edu.ec</a> Internet Source	<1 %
84	<a href="http://www.tribunnews.com">www.tribunnews.com</a> Internet Source	<1 %
85	<a href="http://en.wikipedia.org">en.wikipedia.org</a> Internet Source	<1 %
86	<a href="http://hendardjisoepandji.net">hendardjisoepandji.net</a> Internet Source	<1 %
87	<a href="http://katalog.ukdw.ac.id">katalog.ukdw.ac.id</a> Internet Source	<1 %

88	<a href="http://repota.jti.polinema.ac.id">repota.jti.polinema.ac.id</a> Internet Source	<1 %
89	<a href="http://robodk.com">robodk.com</a> Internet Source	<1 %
90	<a href="http://www.autobotic.com.my">www.autobotic.com.my</a> Internet Source	<1 %
91	<a href="http://www.ingentaconnect.com">www.ingentaconnect.com</a> Internet Source	<1 %
92	<a href="http://www.misterpangalayo.com">www.misterpangalayo.com</a> Internet Source	<1 %
93	Huda, Nurul. "Analisa dan Desain Turbin Sumbu Vertical Savionus Untuk Penerangan di Industri Cerobong Asap Desa Wonosari Kab. Demak.", Universitas Islam Sultan Agung (Indonesia) Publication	<1 %
94	Submitted to St Benedict's Ealing Student Paper	<1 %
95	Submitted to UIN Maulana Malik Ibrahim Malang Student Paper	<1 %
96	Submitted to University of Westminster Student Paper	<1 %
97	<a href="http://apprili.com">apprili.com</a> Internet Source	<1 %
98	<a href="http://data.goodstats.id">data.goodstats.id</a> Internet Source	<1 %
99	<a href="http://ejurnal.its.ac.id">ejurnal.its.ac.id</a> Internet Source	<1 %
100	<a href="http://hasanasariputragaza.blogspot.com">hasanasariputragaza.blogspot.com</a> Internet Source	<1 %

101	<a href="https://issuu.com">issuu.com</a> Internet Source	<1 %
102	<a href="https://jurnal.politeknikaceh.ac.id">jurnal.politeknikaceh.ac.id</a> Internet Source	<1 %
103	<a href="https://media.neliti.com">media.neliti.com</a> Internet Source	<1 %
104	<a href="https://mitradiani.co.id">mitradiani.co.id</a> Internet Source	<1 %
105	<a href="https://openlibrarypublications.telkomuniversity.ac.id">openlibrarypublications.telkomuniversity.ac.id</a> Internet Source	<1 %
106	<a href="https://pdxscholar.library.pdx.edu">pdxscholar.library.pdx.edu</a> Internet Source	<1 %
107	<a href="https://pypi.org">pypi.org</a> Internet Source	<1 %
108	<a href="https://ratu.ai">ratu.ai</a> Internet Source	<1 %
109	<a href="https://repository.uib.ac.id">repository.uib.ac.id</a> Internet Source	<1 %
110	<a href="https://repository.upnjatim.ac.id">repository.upnjatim.ac.id</a> Internet Source	<1 %
111	<a href="https://s3.amazonaws.com">s3.amazonaws.com</a> Internet Source	<1 %
112	<a href="https://www.jurnal.unsyiah.ac.id">www.jurnal.unsyiah.ac.id</a> Internet Source	<1 %
113	Submitted to Birla Institute of Technology and Science Pilani Student Paper	<1 %
114	M. Ali Hanafiah, Witman Rasyid, Agus Purwoko. "HUBUNGAN KARAKTERISTIK, MOTIVASI DAN KOMPETENSI TERHADAP	<1 %

PRODUKTIVITAS KERJA PENYULUH  
PERTANIAN DI KOTA BENGKULU", Jurnal  
AGRISEP, 2013

Publication

---

115 Mohammad Taufan Asri Zaen, Asep Yunandy. <1 %  
"PENGENDALI LAMPU MONITORING RUMAH  
DENGAN SHORT MESSAGE SERVICE (SMS)  
BERBASIS ARDUINO UNO", Jurnal Informatika  
dan Rekayasa Elektronik, 2018  
Publication

---

116 Muh. Irwan Sudirman, Rinto Suppa, Muhlis <1 %  
Muhallim. "PEMBUATAN MESIN PENETAS  
TELUR AYAM BERBASIS ARDUINO", Jurnal  
Informatika dan Teknik Elektro Terapan, 2025  
Publication

---

117 Pambudi, Arief Prasetyo. "Monitoring Ruang <1 %  
Operasi Berbasis Fuzzy Logic Dengan  
Menggunakan IoT", Universitas Islam Sultan  
Agung (Indonesia), 2024  
Publication

---

118 Yoga Randi Prasetio, Budy Satria. "Sistem <1 %  
Keamanan Rumah Menggunakan Sensor  
Passive Infrared Receiver dan SMS Gateway  
Berbasis Arduino", The Indonesian Journal of  
Computer Science, 2022  
Publication

---

119 dcckotabumi.ac.id <1 %  
Internet Source

---

120 jurnalelectron.org <1 %  
Internet Source

---

121 jurnalsyntaxadmiration.com <1 %  
Internet Source

---

122 pastebin.com <1 %  
Internet Source

123	<a href="http://prosiding.unipma.ac.id">prosiding.unipma.ac.id</a> Internet Source	<1 %
124	<a href="http://refanhando.blogspot.com">refanhando.blogspot.com</a> Internet Source	<1 %
125	<a href="http://repository.ipb.ac.id">repository.ipb.ac.id</a> Internet Source	<1 %
126	<a href="http://repository.polman-babel.ac.id">repository.polman-babel.ac.id</a> Internet Source	<1 %
127	<a href="http://repository.poltekapp.ac.id">repository.poltekapp.ac.id</a> Internet Source	<1 %
128	<a href="http://runebook.dev">runebook.dev</a> Internet Source	<1 %
129	<a href="http://sugito.staff.telkomuniversity.ac.id">sugito.staff.telkomuniversity.ac.id</a> Internet Source	<1 %
130	<a href="http://teknoif.itp.ac.id">teknoif.itp.ac.id</a> Internet Source	<1 %
131	Bernadeta Kuswarini Wardaninggar. "Sebaran Potensi Budaya Prasejarah di Enrekang, Sulawesi Selatan", Kapata Arkeologi, 2016 Publication	<1 %
132	Borketey, Humphrey B.. "Investigating Differencing Rules for Java to Enhance the Comprehension of Code Changes.", Bowling Green State University Publication	<1 %
133	Imam Ahmad, Rohmat Indra Borman, Jafar Fakhrurozi, Gavan Gorbi Caksana. "Software Development Dengan Extreme Programming (XP) Pada Aplikasi Deteksi Kemiripan Judul Skripsi Berbasis Android", INOVTEK Polbeng - Seri Informatika, 2020 Publication	<1 %

134 Nazmah Wulan Rhomadhona, Meizano Ardhi Muhammad, Puput Budi Wintoro, Yessi Mulyani. "PENERAPAN METODE RAPID APPLICATION DEVELOPMENT UNTUK SISTEM INFORMASI EVENT BERBASIS WEB PADA UNIVERSITAS LAMPUNG", Jurnal Informatika dan Teknik Elektro Terapan, 2025  
Publication

---

135 cometscomets.blogspot.com  
Internet Source

---

136 danangpriyamboro.wordpress.com  
Internet Source

---

137 digilib.unila.ac.id  
Internet Source

---

138 garuda.kemdikbud.go.id  
Internet Source

---

139 journal.uniga.ac.id  
Internet Source

---

140 kc.umn.ac.id  
Internet Source

---

141 sipora.polije.ac.id  
Internet Source

---

142 www.scribd.com  
Internet Source

---

143 www.thethingsnetwork.org  
Internet Source

---

144 www.umm.ac.id  
Internet Source

---

145 Kodar Udoyono, Muhammad Alif Rizky. "IMPLEMENTASI SISTEM PENGENDALI WASTAFEL CUCI TANGAN BERBASIS IoT

MENGGUNAKAN PLATFORM BLNYK", Jurnal  
Teknologi dan Komunikasi STMIK Subang,  
2023

Publication

---

146 Muhamad Galuh Sutisna, Muchtar Ali Setyo  
Yudono, Marina Artiyasa, Panji Narputo, Any  
Elvia Jakfar. "Sistem Pendukung Keputusan  
Tingkat Stres Mahasiswa dengan Fuzzy  
Mamdani", RIGGS: Journal of Artificial  
Intelligence and Digital Business, 2025  
Publication

<1 %

---

147 Muhamad Sadali, Mahpuz, Muhammad Arifin,  
Muhammad Wasil. "Implementasi Teknologi  
Internet of Things (IoT) pada Sistem  
Keamanan Loker dengan Verifikasi  
Biometrik", Jurnal PRINTER: Jurnal  
Pengembangan Rekayasa Informatika dan  
Komputer, 2025  
Publication

<1 %

---

148 [gitlab.stud.idi.ntnu.no](https://gitlab.stud.idi.ntnu.no)  
Internet Source

<1 %

---

149 [jurnal.um-palembang.ac.id](https://jurnal.um-palembang.ac.id)  
Internet Source

<1 %

---

150 [ojs.ummetro.ac.id](https://ojs.ummetro.ac.id)  
Internet Source

<1 %

---

151 Herdi Muhammad Syaban, Teuku Mufizar,  
Ruuhan Ruuwan. "RANCANG BANGUN  
ALAT KEAMANAN RUMAH MENGGUNAKAN  
SENSOR PIR DENGAN NOTIFIKASI TELEGRAM  
BERBASIS IOT DAN CATU DAYA PLTS", Jurnal  
Informatika dan Teknik Elektro Terapan, 2024  
Publication

<1 %

---

152 Jenifer Theresia Sirait, Ahmad Imam Santoso.  
"Perancangan Sistem Keamanan Rumah

<1 %

berbasis IoT dan Aplikasi Telegram", Jurnal  
Minfo Polgan, 2025

Publication

153

Taufik Kurnialensya, Puput Candra Saputra.  
"Absensi SISTEM MONITORING KEHADIRAN  
SISWA MENGGUNAKAN MIKROKONTROLLER  
BERBASIS WEB", Rabit : Jurnal Teknologi dan  
Sistem Informasi Univrab, 2023

Publication

<1 %

154

[ejournal.unesa.ac.id](http://ejournal.unesa.ac.id)

Internet Source

<1 %

155

[huggingface.co](https://huggingface.co)

Internet Source

<1 %

156

[repository.iainpurwokerto.ac.id](http://repository.iainpurwokerto.ac.id)

Internet Source

<1 %

Exclude quotes On

Exclude matches Off

Exclude bibliography On