

BAB III

METODE PENELITIAN

3.1 Lokasi Penelitian

Penelitian akan dilaksanakan di rumah penulis, yang bertempat di RT 04 RW 01 Desa Danasari Kecamatan Cisaga Ciamis, 46386 dan kampus Universitas Siliwangi Jl. Siliwangi No.24, Kahuripan, Kec. Tawang, Kab. Tasikmalaya, Jawa Barat 46115.

3.2 Metode Pengumpulan Data

Metode-metode yang digunakan dalam pengumpulan data pada tugas akhir ini adalah :

a) Metode Studi Literatur

Merupakan metode untuk mengumpulkan kajian-kajian teori yang dapat menunjang dalam tugas akhir sehingga dapat menjadi dasar dalam pembuatan tugas akhir ini. Merupakan metode untuk mengumpulkan kajian-kajian teori yang dapat menunjang dalam tugas akhir sehingga dapat menjadi dasar dalam pembuatan tugas akhir ini.

b) Metode Observasi

Metode ini adalah melakukan pengamatan langsung terhadap objek penelitian atau percobaan. Adapun tujuan penggunaan metode ini adalah untuk membuktikan studi literatur dengan melihat hasil dari suatu pengujian atau percobaan.

3.3 Pengolahan Data

Data yang diperoleh dari survei dan pengamatan perlu melalui tahap pengolahan sebelum dapat dianalisis. Proses pengolahan data ini bertujuan agar data tersebut dapat menghasilkan informasi yang berguna untuk merancang dan mengoptimalkan Sistem yang lebih efisien.

3.4 Analisis Data

Dalam pembuatan Analisis Data, akan didapatkan antara perbandingan dan kajian teori yang akan menghasilkan pengujian atau percobaan yang optimal. Jika terdapat perbedaan antara keduanya, maka akan didapat data yang nantinya dari data tersebut akan dapat kita pelajari untuk menentukan penyebab terjadinya perbedaan tersebut. Apabila terjadi kesamaan berarti hasil pengujian yang kita buat sesuai dengan kajian teori.

3.5 Subjek dan Objek Penelitian

Subjek penelitian ini adalah "Sistem Pembatas Area Kerja untuk Perangkat Bergerak Berbasis *Internet of Things*." Subjek penelitian merujuk pada topik atau masalah yang menjadi fokus utama dari penelitian ini, yaitu pengembangan dan implementasi sistem pembatas area kerja berbasis IoT menggunakan perangkat ESP32, mikrotik, dan *Firebase*.

Objek penelitian ini adalah "Perangkat Bergerak Berbasis *Internet of Things*" dan komponen-komponen yang terlibat dalam sistem, seperti ESP32, mikrotik, router, Access Point (AP), antarmuka pengguna berbasis React.js, serta koneksi jaringan menggunakan protokol TCP/IP dan *Firebase* sebagai server penyimpanan data.

Secara lebih spesifik, objek penelitian mencakup studi mengenai teknologi dan mekanisme kerja perangkat ESP32, mikrotik, dan *Firebase*, serta bagaimana perangkat-perangkat tersebut berinteraksi dan terintegrasi dalam sistem pembatas area kerja berbasis IoT. Penelitian juga akan melibatkan pengujian dan analisis performa sistem, keamanan data, serta pengaruh perubahan lingkungan dan kondisi pada kinerja sistem tersebut.

3.6 Metode Pembangunan Perangkat

Metode pembangunan perangkat yang sekaligus bertujuan untuk langkah *prototyping* adalah metode *Rapid Prototyping*. Metode ini fokus pada pembuatan prototipe perangkat secara cepat dan iteratif untuk menguji dan mengidentifikasi desain, fungsi, dan kinerja perangkat sebelum mengembangkan produk *final*. *Rapid Prototyping* memungkinkan para pengembang untuk dengan cepat merancang, membangun, dan menguji model perangkat, sehingga mempercepat proses pengembangan dan meminimalkan risiko kesalahan pada tahap awal. Beberapa langkah dalam metode *Rapid Prototyping* meliputi:

1. Pengidentifikasian Kebutuhan: Mengidentifikasi dan memahami kebutuhan dan tujuan dari perangkat yang akan dikembangkan.
2. Perencanaan Desain: Merencanakan dan merancang prototipe perangkat, termasuk spesifikasi teknis, antarmuka pengguna, dan fitur-fitur utama.
3. Pembuatan Prototipe: Membuat prototipe perangkat berdasarkan desain yang telah direncanakan. Dalam tahap ini, seringkali digunakan teknologi cetak 3D, *breadboard*, atau platform pengembangan seperti Arduino atau ESP32.

4. Pengujian dan Evaluasi: Menguji prototipe perangkat untuk mengidentifikasi kelemahan, kesalahan, dan potensi perbaikan. Pengujian ini bisa mencakup pengujian fungsionalitas, performa, dan keamanan perangkat.
5. Refining: Berdasarkan hasil pengujian, dilakukan perbaikan dan penyempurnaan pada desain dan fungsi perangkat. Iterasi ini dilakukan secara berulang hingga mencapai hasil yang diinginkan.
6. Pengembangan Produk *Final*: Setelah prototipe dianggap memadai, hasilnya digunakan sebagai dasar untuk mengembangkan produk *final* dengan fitur dan spesifikasi yang telah dioptimalkan.

Keunggulan metode *Rapid Prototyping* adalah memungkinkan para pengembang untuk dengan cepat menguji dan menyempurnakan ide, serta mengidentifikasi masalah sebelum memulai produksi massal. Dengan pendekatan ini, risiko pembuatan produk yang tidak sesuai dengan kebutuhan dan harapan pengguna dapat dikurangi, sehingga mempercepat dan efisien dalam menghasilkan perangkat yang berkualitas.

3.7 Tujuan Prototyping

Dibuatnya sebuah Prototyping bagi pengembang sistem bertujuan untuk mengumpulkan informasi dari pengguna sehingga pengguna dapat berinteraksi dengan model prototype yang dikembangkan, sebab prototype menggambarkan versi awal dari sistem untuk kelanjutan sistem sesungguhnya yang lebih besar.

Telah ditemukan bahwa dalam analisis dan desain sistem, terutama untuk proses transaksi, di mana dialog yang ditampilkan lebih mudah difahami. Semakin besar interaksi antara komputer dan pengguna, besar pula manfaat yang diperoleh

ketika proses pengembangan sistem informasi akan lebih cepat dan membuat pengguna akan lebih interaktif dalam proses pengembangannya.

Prototyping dapat diterapkan pada pengembangan sistem kecil maupun besar dengan harapan agar proses pengembangan dapat berjalan dengan baik, tertata serta dapat selesai tepat waktu. Keterlibatan pengguna secara penuh ketika *prototype* terbentuk akan menguntungkan seluruh pihak yang terlibat, bagi pimpinan, pengguna sendiri serta pengembang sistem.

Manfaat lainnya dari penggunaan *prototyping* adalah :

1. Mewujudkan sistem sesungguhnya dalam sebuah replika sistem yang akan berjalan, menampung masukan dari pengguna untuk kesempurnaan sistem.
2. Pengguna akan lebih siap menerima setiap perubahan sistem yang berkembang sesuai dengan berjalannya *prototype* sampai dengan hasil akhir pengembangan yang akan berjalan nantinya.
3. *Prototype* dapat ditambah maupun dikurangi sesuai berjalannya proses pengembangan. Kemajuan tahap demi tahap dapat diikuti langsung oleh pengguna.
4. Penghematan sumberdaya dan waktu dalam menghasilkan produk yang lebih baik dan tepat guna bagi pengguna.

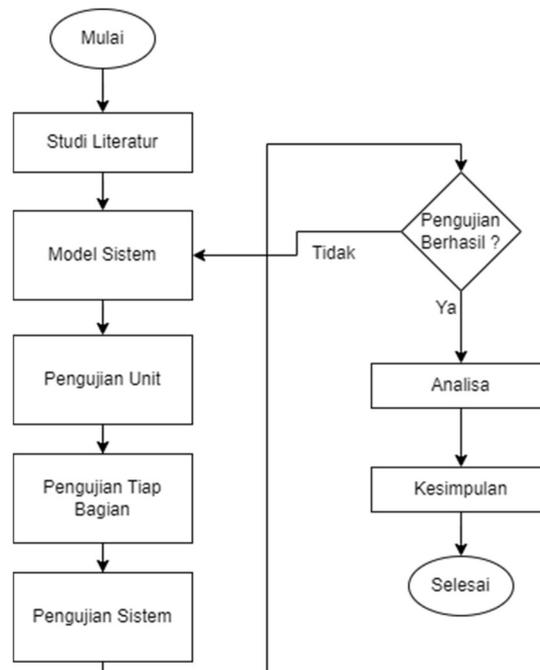
Prototyping dimulai dengan pengumpulan kebutuhan, melibatkan pengembang dan pengguna sistem untuk menentukan tujuan, fungsi dan kebutuhan operasional sistem.

Langkah-langkah dalam *prototyping* adalah sebagai berikut :

1. Pengumpulan Kebutuhan.
2. Proses desain yang cepat.

3. Membangun prototipe.
4. Evaluasi dan perbaikan.

Mengumpulkan kebutuhan melibatkan pertemuan antara pengembang dan pelanggan untuk menentukan keseluruhan tujuan dibuatnya perangkat lunak; mengidentifikasi kebutuhan berupa garis besar kebutuhan dasar dari sistem yang akan dibuat. Desain berfokus pada representasi dari aspek perangkat lunak dari sudut pengguna ini mencakup *input*, proses dan *format output*. Desain cepat mengarah ke pembangunan prototipe, prototipe dievaluasi oleh pengguna dan bagian analis desain dan digunakan untuk menyesuaikan kebutuhan perangkat lunak yang akan dikembangkan. *prototype* diatur untuk memenuhi kebutuhan pengguna, dan pada saat itu pula pengembang memahami secara lebih jelas dan detil apa yang perlu dilakukannya. Setelah keempat langkah *prototyping* dijalankan, maka langkah selanjutnya adalah pembuatan atau perancangan produk yang sesungguhnya tertuju pada gambar 3.1 di bawah.



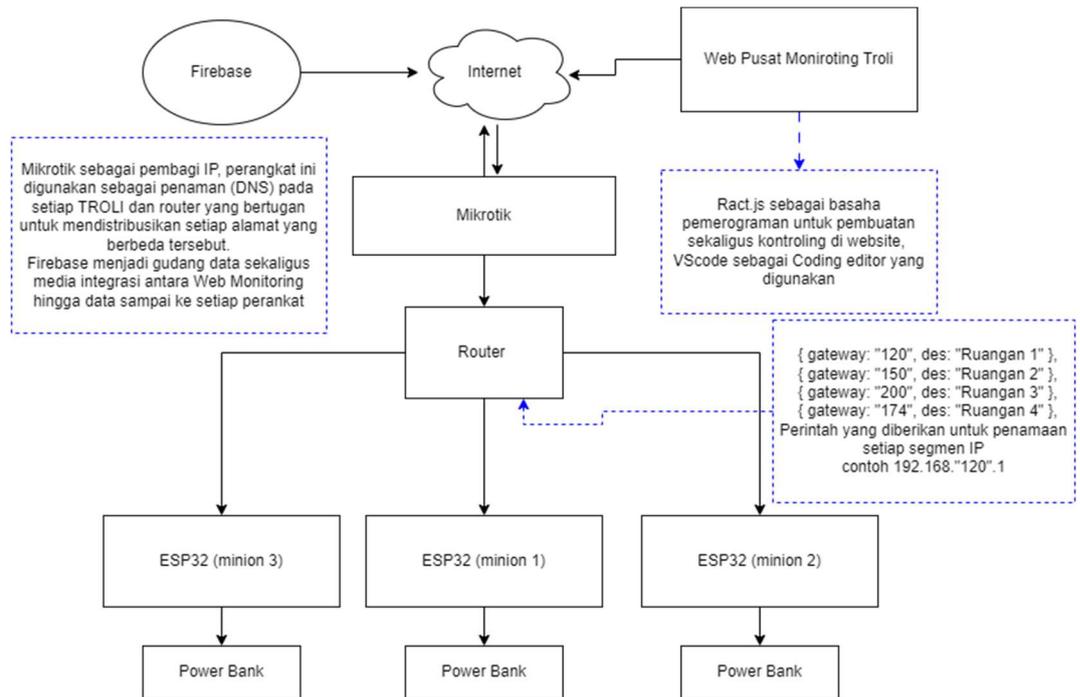
Gambar 3. 1 Flowchart Penelitian

3.8 Model Sistem

Untuk mempermudah dalam memahami model sistem maka dibuat arsitektur, blok diagram, dan flowchart sistem.

3.8.1 Analisis Gambaran Umum Alat

Analisis gambaran umum alat adalah proses menganalisis dan memahami secara keseluruhan tentang alat yang akan dikembangkan. Pada tahap ini, dilakukan pengumpulan informasi terkait dengan kebutuhan, spesifikasi, dan fitur yang diinginkan dari alat tersebut. Selanjutnya, dilakukan pemetaan atau pemahaman terhadap fungsi-fungsi utama yang harus ada dalam alat tersebut. Analisis gambaran umum alat juga melibatkan pemahaman tentang komponen-komponen yang akan digunakan, interaksi antara komponen-komponen tersebut, serta kemungkinan masalah yang mungkin muncul dalam implementasinya. Tujuan dari analisis gambaran umum alat adalah untuk menghasilkan pemahaman yang jelas dan komprehensif tentang alat yang akan dikembangkan, sehingga dapat memberikan dasar yang kuat dalam proses pengembangan selanjutnya.



Gambar 3. 2 Arsitektur Perangkat

Pada Gambar 3.2 Arsitektur perangkat diperlukan agar struktur, fungsi, dan komponen dari suatu sistem atau perangkat dapat dirancang, diatur, dan diimplementasikan secara efisien dan terarah. Arsitektur perangkat adalah rencana atau panduan yang menggambarkan bagaimana semua komponen perangkat akan berinteraksi, bekerja sama, dan berfungsi bersama-sama untuk mencapai tujuan yang diinginkan.

3.8.2 Analisis Arsitektur Perancangan Sistem

Sistem yang dirancang dalam penelitian ini mencakup beberapa komponen utama, termasuk ESP32, MikroTik, *Firestore*, dan aplikasi monitoring pusat. Arsitektur sistem ini dirancang dengan tujuan membatasi area kerja untuk troli barang berbasis *Internet of Things* (IoT) secara efektif dan efisien. Berikut adalah analisis detail dari setiap komponen dan hubungannya dalam sistem.

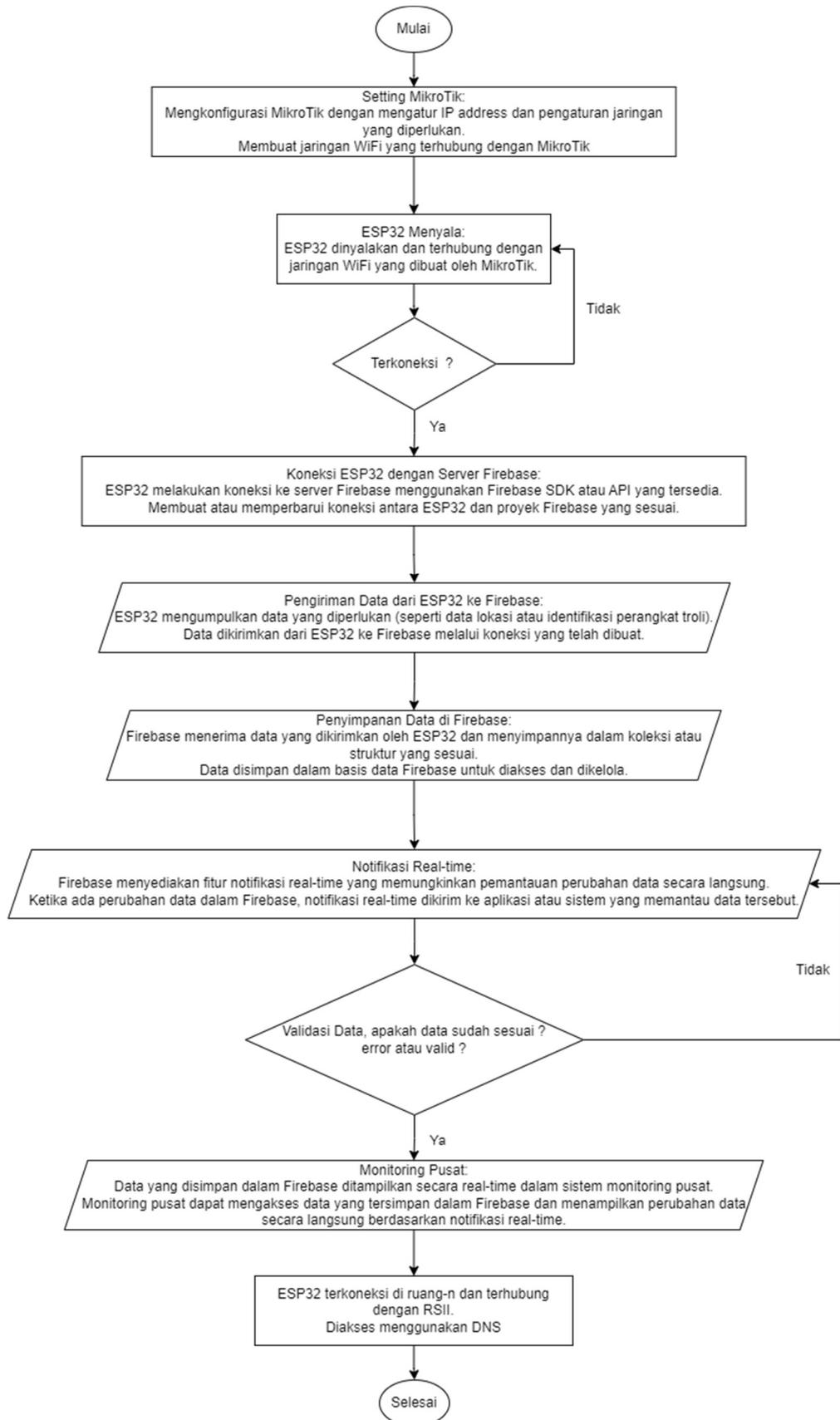
ESP32 berperan sebagai perangkat bergerak yang melacak dan mengirimkan data troli barang. Ia mengumpulkan data seperti lokasi troli dan identifikasi perangkat. ESP32 juga bertanggung jawab untuk mengirimkan data tersebut ke server Firebase menggunakan konektivitas WiFi yang dimilikinya. MikroTik berfungsi sebagai perangkat router yang bertanggung jawab untuk mengatur jaringan WiFi yang terhubung dengan ESP32.

MikroTik melakukan konfigurasi jaringan dan menyediakan konektivitas yang diperlukan agar ESP32 dapat terhubung ke server *Firebase* dan aplikasi monitoring pusat. Selain itu, MikroTik juga berperan dalam mengatur pembagian IP untuk identifikasi ruangan dan perangkat dalam sistem.

Firebase digunakan sebagai database untuk menyimpan data yang dikirimkan oleh ESP32. Data seperti lokasi troli, identifikasi perangkat, dan informasi lainnya disimpan dalam *Firebase*. *Firebase* juga menyediakan fitur notifikasi *real-time* yang memungkinkan pemantauan perubahan data secara langsung. Hal ini memungkinkan aplikasi monitoring pusat untuk menerima pembaruan data secara *real-time* dan menampilkan informasi terbaru kepada pengguna.

Aplikasi monitoring pusat merupakan antarmuka yang digunakan untuk memantau data troli barang secara *real-time*. Aplikasi ini terhubung ke server Firebase untuk mengambil data yang disimpan dan menampilkannya kepada pengguna dalam bentuk yang mudah dipahami. Aplikasi monitoring pusat dapat menampilkan informasi seperti posisi troli, status perangkat, dan laporan kejadian penting lainnya.

Arsitektur sistem ini memanfaatkan ESP32 sebagai perangkat bergerak untuk melacak dan mengirimkan data, MikroTik sebagai perangkat router untuk mengatur jaringan dan konektivitas WiFi, *Firestore* sebagai database untuk menyimpan data troli barang, dan aplikasi monitoring pusat sebagai antarmuka untuk memantau data secara *real-time*. Integrasi antara komponen-komponen ini memungkinkan sistem pembatas area kerja untuk troli barang berbasis IoT menjadi efektif dan efisien. Maka dari pada itu dibutuhkan Flowchart Perancangan Sistem Pembatas Area Kerja pada gambar 3.3 di bawah.



Gambar 3. 3 Flowchart Perancangan Sistem Pembatas Area Kerja

Berikut adalah pembaruan flowchart untuk menyertakan alur *Firestore* yang digunakan sebagai database dan hubungannya dengan notifikasi *real-time*:

1. Memulai: Inisialisasi penelitian dan pengumpulan semua persyaratan dan kebutuhan penelitian.
2. Setting MikroTik: Bagian ini mencakup konfigurasi awal MikroTik yang melibatkan pengaturan jaringan, termasuk pembagian IP dan pengaturan jaringan WiFi. Tujuannya adalah untuk menyediakan konektivitas yang diperlukan agar ESP32 dapat terhubung ke jaringan WiFi.
3. ESP32 Menyala: Bagian ini menunjukkan langkah-langkah yang dilakukan ketika ESP32 dinyalakan. Ini mencakup inisialisasi ESP32, mengaktifkan modul WiFi dan Bluetooth, serta mendapatkan koneksi ke jaringan WiFi yang telah dikonfigurasi sebelumnya.
4. Pengiriman Data dari ESP32 ke *Firestore*: Bagian ini mencakup proses pengumpulan data dari ESP32, seperti lokasi troli atau identifikasi perangkat, dan mengirimkannya ke server *Firestore*. Data ini dikirim melalui koneksi WiFi yang telah dibuat sebelumnya.
5. Penyimpanan Data di *Firestore*: Bagian ini menunjukkan langkah-langkah untuk menyimpan data yang diterima dari ESP32 ke dalam database *Firestore*. Data yang dikirim dari ESP32 akan disimpan dalam *Firestore* untuk dianalisis atau ditampilkan di aplikasi monitoring pusat.
6. Monitoring Pusat: Bagian ini melibatkan aplikasi monitoring pusat yang terhubung ke server *Firestore*. Aplikasi ini akan memantau data yang tersimpan dalam *Firestore* dan menampilkan informasi yang relevan, seperti posisi troli, status perangkat, atau laporan peristiwa penting lainnya. Data ini akan diambil

secara *real-time* dari Firebase untuk memastikan informasi yang ditampilkan adalah yang terbaru.

Melalui flowchart ini, dapat terlihat bagaimana alur kerja sistem dari setting awal MikroTik, ESP32 yang menyala, koneksi dengan server *Firestore*, pengiriman dan penyimpanan data, hingga pemantauan data melalui aplikasi monitoring pusat.

3.8.3 Analisis Kebutuhan Fungsional

Berdasarkan data yang telah diperoleh sepanjang percakapan ini, berikut adalah analisis kebutuhan fungsional untuk penelitian ini:

1. Pelacakan dan Identifikasi Troli: Sistem harus mampu melacak posisi dan identifikasi troli secara akurat. Hal ini memungkinkan pengguna untuk mengetahui lokasi dan status troli barang dalam waktu nyata.
2. Pembatasan Area Kerja: Sistem harus dapat membatasi area kerja troli barang berdasarkan aturan dan batasan yang ditentukan. Ini akan membantu mencegah kemacetan, tabrakan, atau masalah keamanan lainnya dalam lingkungan kerja.
3. Integrasi dengan MikroTik: Sistem harus terintegrasi dengan perangkat MikroTik untuk mengatur jaringan WiFi yang diperlukan oleh troli barang. Ini memastikan konektivitas yang stabil antara ESP32 dan server *Firestore*.
4. Koneksi ke Server Firebase: Sistem harus dapat terhubung ke server *Firestore* untuk menyimpan dan mengambil data troli barang secara efisien. Koneksi yang andal dan cepat diperlukan untuk memastikan data yang akurat dan *up-to-date*.
5. Notifikasi *Real-Time*: Sistem harus mampu memberikan notifikasi *secara real-time* kepada pengguna melalui aplikasi monitoring pusat ketika terjadi

perubahan atau kejadian penting terkait troli barang. Notifikasi ini memungkinkan tindakan cepat dalam mengatasi situasi yang memerlukan perhatian.

6. Analisis Data dan Laporan: Sistem harus menyediakan kemampuan untuk menganalisis data troli barang yang terkumpul dan menghasilkan laporan yang bermanfaat. Analisis ini dapat membantu pengambilan keputusan yang lebih baik dalam pengelolaan troli barang dan perbaikan proses operasional.

Dengan memenuhi kebutuhan fungsional ini, sistem dapat memberikan solusi yang efektif dan efisien dalam membatasi area kerja troli barang berbasis IoT. Hal ini akan meningkatkan keamanan, efisiensi, dan pengelolaan aliran barang dalam berbagai sektor, seperti perdagangan, industri, atau logistik.

3.8.4 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional dilakukan untuk mengetahui spesifikasi kebutuhan untuk sistem. Spesifikasi kebutuhan melibatkan analisis perangkat keras atau *hardware*, analisis perangkat lunak atau *software*, analisis pengguna atau user.

3.8.5 Analisis Perangkat Keras

Perangkat keras merupakan salah satu hal yang sangat penting dalam pengembangan sistem keamanan ini, karena tanpa perangkat keras yang memenuhi syarat, aplikasi yang akan dibangun tidak akan berjalan secara optimal. Berikut spesifikasi perangkat keras yang dibutuhkan agar sistem berjalan secara optimal:

- a. *Smartphone* Android
- b. Laptop / PC

- c. ESP32
- d. Router (sebagai repeater, menggunakan RBwsAP-5Hac2nD)
- e. Mikrotik
- f. Power Bank
- g. USB
- h. Kabel LAN

3.8.6 Analisis Perangkat Lunak

Perangkat lunak dalam sebuah sistem merupakan intruksi yang digunakan untuk memproses seluruh informasi pada perangkat keras agar bisa saling berinteraksi diantara keduanya. Berikut perangkat lunak yang dibutuhkan untuk membangun sistem keamanan :

- a. Arduino IDE: Arduino IDE digunakan untuk memprogram dan mengunggah kode ke modul ESP32. Perangkat lunak ini menyediakan lingkungan pengembangan yang *user-friendly* untuk menulis dan menguji kode mikrokontroler.
- b. MikroTik RouterOS: MikroTik RouterOS adalah sistem operasi yang digunakan oleh perangkat MikroTik Router. Perangkat lunak ini menyediakan berbagai fitur dan fungsi untuk mengatur jaringan WiFi dan melakukan pembagian IP.
- c. Firebase: *Firebase* adalah platform pengembangan aplikasi yang menyediakan layanan *cloud* untuk penyimpanan dan pengelolaan data. Dalam penelitian ini, *Firebase* digunakan sebagai server basis data untuk

menyimpan informasi troli barang dan memfasilitasi pertukaran data antara perangkat keras dan perangkat lunak.

- d. *React.js*: *React.js* adalah perpustakaan *JavaScript* yang digunakan untuk membangun antarmuka pengguna pada aplikasi web. Dalam penelitian ini, *React.js* digunakan untuk mengembangkan antarmuka pengguna pada aplikasi monitoring pusat.
- e. *Visual Studio Code* (*VS Code*): *VS Code* adalah editor kode sumber yang populer dan kuat. Digunakan untuk mengembangkan kode, mengedit, dan memeriksa sintaksis dalam proses pengembangan perangkat lunak.

Perangkat lunak ini digunakan untuk mengembangkan, menghubungkan, dan mengelola berbagai aspek sistem pembatas area kerja untuk troli barang berbasis IoT. Masing-masing perangkat lunak memiliki peran dan fungsi tertentu dalam menjalankan sistem secara keseluruhan.

3.8.7 Perancangan Sistem dan Arsitektur Perancangan Program

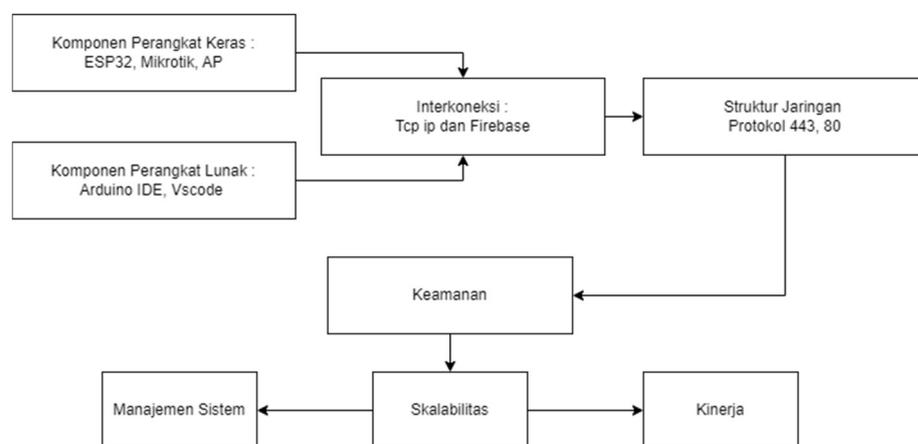
Perancangan Sistem dan Arsitektur Perancangan Program adalah tahap penting dalam pengembangan penelitian ini. Pertama, dilakukan analisis kebutuhan sistem berdasarkan tujuan penelitian dan persyaratan yang telah ditetapkan sebelumnya. Kebutuhan fungsional dan *non-fungsional* sistem akan diidentifikasi dan dianalisis secara menyeluruh.

Selanjutnya, dilakukan perancangan sistem secara keseluruhan. Ini melibatkan perencanaan arsitektur sistem, pemilihan teknologi yang tepat, dan desain antarmuka pengguna yang sesuai. Aspek-aspek penting seperti skala, kinerja, keamanan, dan ketersediaan sistem akan dipertimbangkan dengan cermat.

Setelah itu, dilakukan perancangan arsitektur program yang akan digunakan dalam implementasi sistem. Arsitektur program mencakup struktur komponen program dan hubungan antara mereka. Pemilihan bahasa pemrograman, *framework* atau *library*, serta desain modul dan integrasi komponen akan diperhatikan dengan teliti.

Terakhir, dilakukan penentuan alur program yang rinci. Alur program menentukan urutan langkah-langkah dalam implementasi sistem, termasuk identifikasi fungsi utama, logika bisnis, dan algoritma yang akan digunakan. Pemodelan data juga dilakukan untuk menggambarkan struktur data yang digunakan dalam sistem, seperti desain basis data, tabel, relasi antara entitas, dan skema database yang diperlukan.

Dengan melakukan perancangan sistem dan arsitektur perancangan program yang matang, penelitian ini dapat menghasilkan solusi yang efektif dan efisien dalam membatasi area kerja untuk troli barang berbasis IoT. Gambar 3.4 merupakan Perancangan yang baik memastikan sistem berjalan dengan baik, sesuai dengan kebutuhan, dan mudah untuk dikembangkan dan dipelihara di masa mendatang.



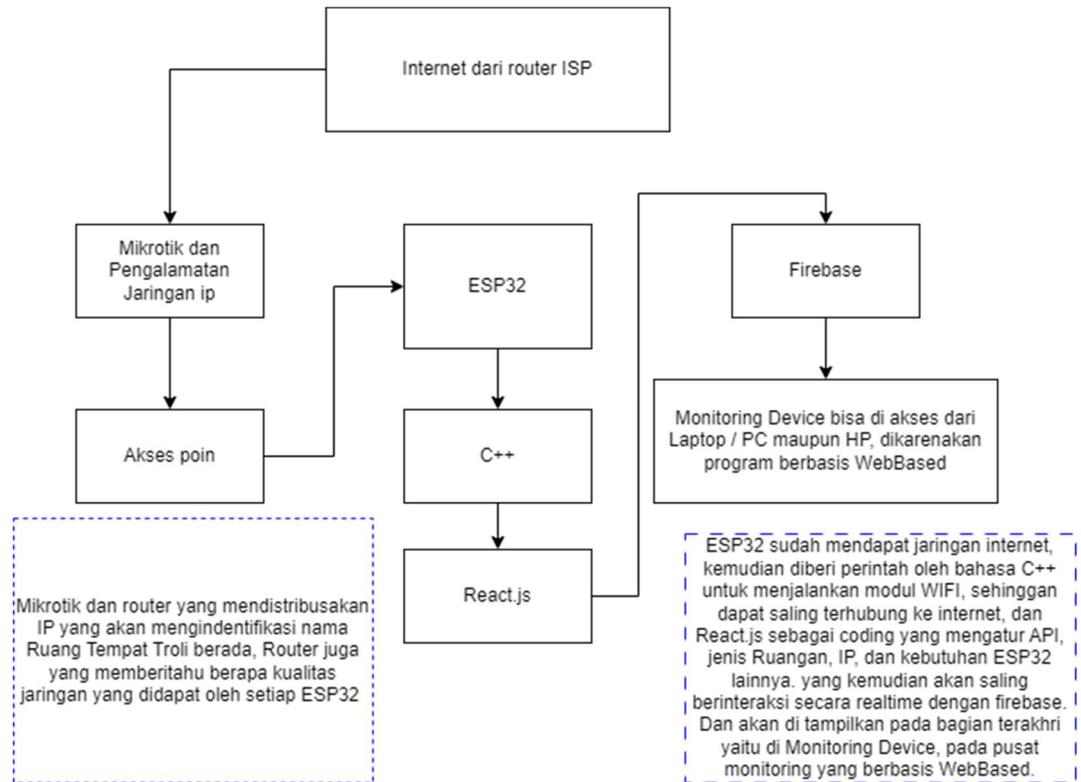
Gambar 3. 4 Flowchart Arsitektur Program

Dalam penelitian ini, struktur sistem yang digunakan mencakup:

1. **Komponen Perangkat Keras:** Sistem ini menggunakan ESP32 sebagai perangkat pembatas area kerja. ESP32 memiliki prosesor dan memori yang cukup untuk menjalankan program yang diperlukan dalam mengimplementasikan pembatasan area kerja.
2. **Komponen Perangkat Lunak:** Sistem ini menggunakan program aplikasi yang dikembangkan menggunakan *Visual Studio Code* (VSCoDe). Program ini berfungsi untuk mengontrol ESP32, mendeteksi dan memantau perangkat yang berada dalam area kerja yang telah ditentukan, serta mengirimkan data ke *Firebase* untuk penyimpanan dan analisis lebih lanjut. Sistem ini juga menggunakan program aplikasi yang dikembangkan dengan menggunakan *Arduino IDE*. Program ini berfungsi untuk mengontrol ESP32, mengumpulkan data dari sensor, dan mentransmisikan data ke *Firebase* untuk dipantau dan dikelola.
3. **Interkoneksi:** Sistem ini menggunakan protokol TCP/IP untuk menghubungkan ESP32 dengan perangkat lain dalam jaringan. Selain itu, sistem juga menggunakan protokol *Firebase API* untuk mentransmisikan data antara ESP32 dan database *Firebase*.
4. **Struktur Jaringan:** Dalam struktur jaringan, sistem menggunakan jaringan WiFi untuk menghubungkan ESP32 dengan router, dan kemudian data dikirimkan melalui internet menggunakan protokol TCP/IP dan protokol *Firebase API*.

5. Keamanan: Keamanan sistem dijaga dengan menggunakan pengamanan jaringan dan enkripsi data saat mentransmisikan data antara ESP32 dan *Firestore*.
6. Skalabilitas: Sistem ini dirancang dengan kemampuan skalabilitas untuk mengatasi peningkatan jumlah perangkat dan area kerja yang harus dipantau.
7. Kinerja: Sistem ini berusaha untuk memberikan kinerja yang optimal dengan meminimalkan latensi dan memastikan koneksi yang stabil.
8. Manajemen Sistem: Manajemen sistem dilakukan melalui VSCode, dimana program aplikasi dapat dipantau dan dikelola, serta pembaruan perangkat lunak dapat dilakukan secara efisien.

Pada gambar 3.5 dengan struktur sistem yang terencana dengan baik, penelitian ini dapat menghasilkan solusi yang efisien dan andal dalam membatasi area kerja menggunakan ESP32 sebagai perangkat pembatas berbasis *Internet of Things* (IoT), dengan dukungan dari *Visual Studio Code*, TCP/IP, dan *Firestore* untuk pengolahan data dan monitoring secara *real-time*.



Gambar 3. 5 Arsitektur Program

Arsitektur program untuk pembatas area kerja berbasis IoT dengan melibatkan komponen-komponen yang disebutkan dapat dijelaskan sebagai berikut:

1. Sumber Internet dari ISP: Sumber internet dari penyedia layanan internet (ISP) merupakan input utama untuk menghubungkan seluruh sistem ke internet. Internet ini menjadi jalur komunikasi utama antara perangkat-perangkat dalam sistem dan *Firebase* sebagai penyimpanan data.
2. Mikrotik: Mikrotik berfungsi sebagai router yang mengatur lalu lintas data dalam jaringan lokal. Mikrotik akan mengatur aliran data dari perangkat ESP32 ke jaringan internet dan sebaliknya. Mikrotik juga dapat memberikan fitur keamanan untuk melindungi jaringan dari ancaman luar.

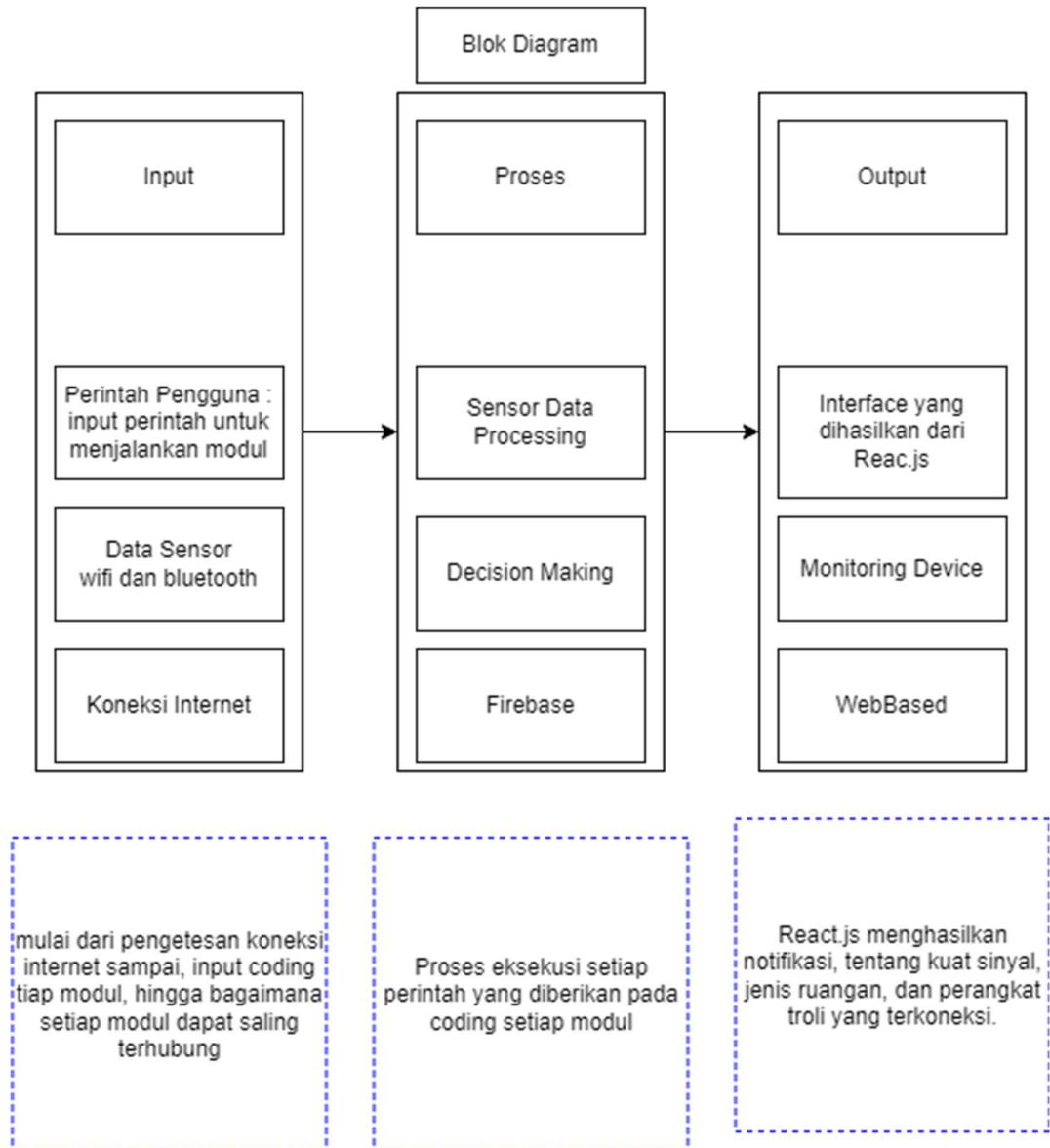
3. AP (*Access Point*): *Access Point* digunakan sebagai titik akses ke jaringan WiFi yang akan diakses oleh perangkat ESP32. AP akan memberikan jaringan WiFi yang dibutuhkan ESP32 untuk berkomunikasi dengan perangkat lain dalam sistem.
4. ESP32: ESP32 merupakan perangkat mikrokontroler yang menjadi otak sistem. Program di ESP32 akan membaca data dari sensor-sensor yang terpasang, menjalankan logika pengendalian untuk membatasi area kerja, dan mengirimkan data ke *Firebase* untuk penyimpanan dan analisis lebih lanjut.
5. C++: Bahasa pemrograman C++ digunakan untuk mengembangkan program di ESP32. C++ merupakan bahasa pemrograman yang efisien dan mendukung pemrograman berorientasi objek, sehingga memungkinkan pengembangan program yang kompleks dan canggih untuk sistem pembatas area kerja.
6. React.js: React.js digunakan untuk mengembangkan antarmuka pengguna (*user interface*) pada aplikasi web yang akan digunakan sebagai *monitoring device*. React.js adalah *framework* JavaScript yang populer untuk membangun antarmuka pengguna yang interaktif dan responsif.
7. Firebase: *Firebase* digunakan sebagai penyimpanan data dalam sistem. Data dari ESP32 akan dikirimkan ke *Firebase* untuk disimpan dan diakses dari aplikasi *monitoring device*. *Firebase* juga menyediakan fitur-fitur yang memudahkan manajemen data, otentikasi pengguna, dan notifikasi *real-time*.
9. WebView: *WebView* digunakan sebagai komponen pada aplikasi *monitoring device* yang menampilkan antarmuka web dari *Firebase*. Melalui *WebView*,

pengguna dapat melihat data IP, ruangan, update, kuat sinyal RSSI, dan penomeran troli atau perangkat ESP32 secara *real-time*.

Dengan arsitektur program yang terintegrasi dengan baik, sistem pembatas area kerja berbasis IoT dapat berfungsi secara efisien dan memberikan pengawasan dan pengendalian yang akurat dan *real-time* dalam area kerja yang ditentukan. Antarmuka pengguna yang dikembangkan menggunakan React.js memudahkan pengaturan dan pemantauan sistem secara interaktif dan responsif, sementara *Firebase* menyediakan penyimpanan data yang handal dan fitur notifikasi *real-time* untuk pemantauan yang lebih efisien.

3.8.8 Blok Diagram

Block diagram memiliki peran yang penting dalam penelitian ini karena menyajikan gambaran *visual* yang jelas tentang alur kerja dan hubungan antara komponen-komponen yang ada dalam sistem. *Block diagram* membantu untuk memahami input yang masuk, proses yang terjadi, dan *output* yang dihasilkan. Dengan melihat *block diagram*, kita dapat melihat dengan jelas bagaimana data sensor diolah, keputusan diambil, dan informasi ditampilkan kepada pengguna. *Block diagram* juga membantu dalam mengidentifikasi bagian-bagian yang kritis, menentukan hubungan antara komponen, dan mempermudah dalam perancangan, implementasi, serta evaluasi sistem secara keseluruhan. Pada gambar 3.6 di bawah dengan memahami pentingnya *block diagram*, dapat memastikan bahwa keseluruhan sistem berjalan dengan efisien, sesuai dengan kebutuhan, dan mampu mencapai tujuan penelitian dengan baik.



Gambar 3. 6 Blok Diagram

1. Input

Data Sensor: Data sensor yang menjadi input utama program adalah data lokasi, data jarak, dan data identifikasi troli barang. Data lokasi mengacu pada informasi tentang posisi troli dalam ruangan atau area kerja. Data jarak mengukur jarak antara troli dan titik referensi tertentu, seperti batasan area kerja. Data identifikasi troli barang mencakup informasi tentang troli spesifik yang sedang dipantau, yang dapat berupa kode unik atau identifikasi lainnya.

Perintah Pengguna: Perintah pengguna menjadi input tambahan yang memungkinkan interaksi dengan sistem. Pengguna dapat memberikan perintah, seperti mengubah batasan area kerja, memperoleh informasi troli barang tertentu, atau mengakses riwayat data troli.

2. Proses

Sensor Data Processing: Data sensor yang diperoleh dari perangkat ESP32 akan diproses dalam sistem. Proses ini melibatkan analisis dan pengolahan data sensor, termasuk penggunaan algoritma untuk mengidentifikasi lokasi troli berdasarkan data lokasi dan jarak yang diterima. Pemrosesan data sensor juga melibatkan penggunaan metode pengenalan dan identifikasi troli untuk memastikan bahwa setiap troli barang dapat dikenali secara unik.

Decision Making: Berdasarkan data sensor yang telah diproses, sistem akan mengambil keputusan berdasarkan aturan dan logika yang telah ditentukan sebelumnya. Misalnya, jika troli melebihi batasan area kerja yang telah ditentukan, sistem akan menghasilkan peringatan atau tindakan untuk mencegah pelanggaran. Keputusan juga dapat melibatkan pengiriman notifikasi kepada pengguna atau pihak terkait jika terjadi kejadian penting, seperti pencurian troli atau masalah keamanan lainnya.

3. Output

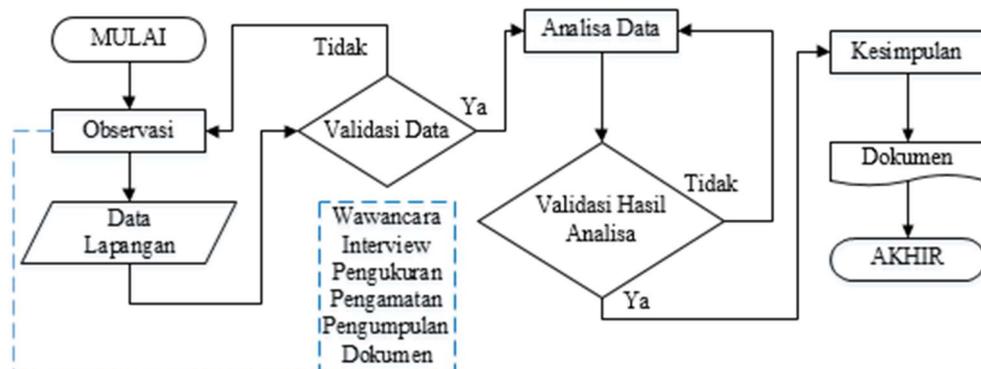
Antarmuka Pengguna: Informasi yang dihasilkan dari proses akan ditampilkan pada antarmuka pengguna yang dikembangkan menggunakan React.js. Antarmuka pengguna akan menampilkan status troli, peringatan, riwayat data troli, dan fitur-fitur lain yang relevan. Pengguna dapat memantau dan mengakses informasi ini melalui antarmuka yang intuitif dan mudah digunakan.

Monitoring Pusat: Data yang relevan akan dikirim ke server *Firestore* untuk disimpan dan diperbarui secara *real-time*. Monitoring pusat memungkinkan pemantauan keseluruhan dari perangkat lain yang terhubung, seperti pusat kontrol keamanan atau manajemen operasional. Data ini dapat memberikan wawasan yang berguna dalam pengelolaan dan pengawasan troli barang secara efisien.

Dengan memahami input, proses, dan output dalam penelitian ini, dapat dikembangkan sistem yang dapat menerima data sensor, memprosesnya dengan menggunakan algoritma dan aturan yang tepat, dan menghasilkan output yang relevan dan berguna bagi pengguna dan pemangku kepentingan terkait.

3.9 Observasi lapangan

Observasi lapangan merupakan kegiatan pengamatan langsung untuk melakukan studi mekanisme kerja alat dan untuk mengetahui unit yang diperlukan oleh sistem dalam bekerja, kapasitas dan jumlah operator yang terlibat. Adapun langkah-langkah dalam Observasi Lapangan sebagai berikut :



Gambar 3. 7 Flowchart Observasi Lapangan

Pada Gambar 3.7 langkah-langkah observasi yaitu mendatangi pengembang sebelumnya dan melakukan beberapa pencarian referensi dari internet, lalu data-

data yang diperoleh diperiksa kembali, jika sudah mencukupi maka dibuatkan dokumen yang menjadi dasar perencanaan penelitian.

Penjelasan Flowchart Observasi Lapangan meliputi :

1. Wawancara dan *interview Moderator Sistem Smart Security*.
2. Pengukuran dan pengamatan dilakukan untuk menentukan letak dari komponen-komponen yang akan digunakan.
3. Pengumpulan Dokumen merupakan proses pencarian jurnal.
4. Proses Validasi Data merupakan kegiatan pengkajian kembali data-data yang diperoleh dari hasil observasi.
5. Analisis data dilakukan ketika proses validasi data dirasa cukup, pada tahap ini dilakukan perencanaan perancangan Perangkat.
6. Validasi hasil analisis dilakukan untuk mengkaji kembali perencanaan perancangan perangkat.
7. Setelah perencanaan perancangan perangkat dirasa cukup, maka ditarik kesimpulan mengenai alat dan bahan yang akan digunakan.
8. Lalu dibuatkan sebuah dokumen, dimana dokumen ini berisikan hasil dari awal sampai akhir proses Observasi Lapangan.

3.10 Validasi Data

Proses analisis dari hasil yang didapat dari Observasi. Apabila masih ada data yang diperlukan maka bisa dilakukan Observasi kembali, ketika data yang diperlukan dirasa cukup, maka selanjutnya mempersiapkan alat dan bahan.

3.11 Pengujian Tiap Unit

Pengujian Tiap Unit dilakukan untuk mengetahui kondisi Komponen-komponen yang digunakan dengan urutan pengujian sebagai berikut :

1. Pengujian Sensor ESP32:

- Pengujian akurasi sensor lokasi: Melakukan pengujian untuk memastikan sensor ESP32 mampu mendeteksi dan melaporkan lokasi troli dengan akurasi yang tinggi.
- Pengujian sensor jarak: Menguji sensor jarak pada ESP32 untuk memverifikasi ketepatan pengukuran jarak antara troli dan batasan area kerja.
- Pengujian sensor identifikasi troli: Melakukan pengujian untuk memastikan bahwa sensor ESP32 dapat mengidentifikasi troli secara unik dan akurat.

2. Pengujian Integrasi Komponen:

- Pengujian integrasi ESP32 dengan Mikrotik: Memastikan koneksi dan komunikasi yang stabil antara ESP32 dan perangkat Mikrotik dalam membatasi area kerja troli.
- Pengujian integrasi ESP32 dengan *Firebase*: Menguji koneksi dan pengiriman data dari ESP32 ke *server Firebase* untuk pemantauan pusat dan penyimpanan data secara *real-time*.
- Pengujian integrasi ESP32 dengan antarmuka pengguna: Memastikan antarmuka pengguna yang dikembangkan menggunakan React.js dapat menerima dan menampilkan data yang diterima dari ESP32 dengan benar.

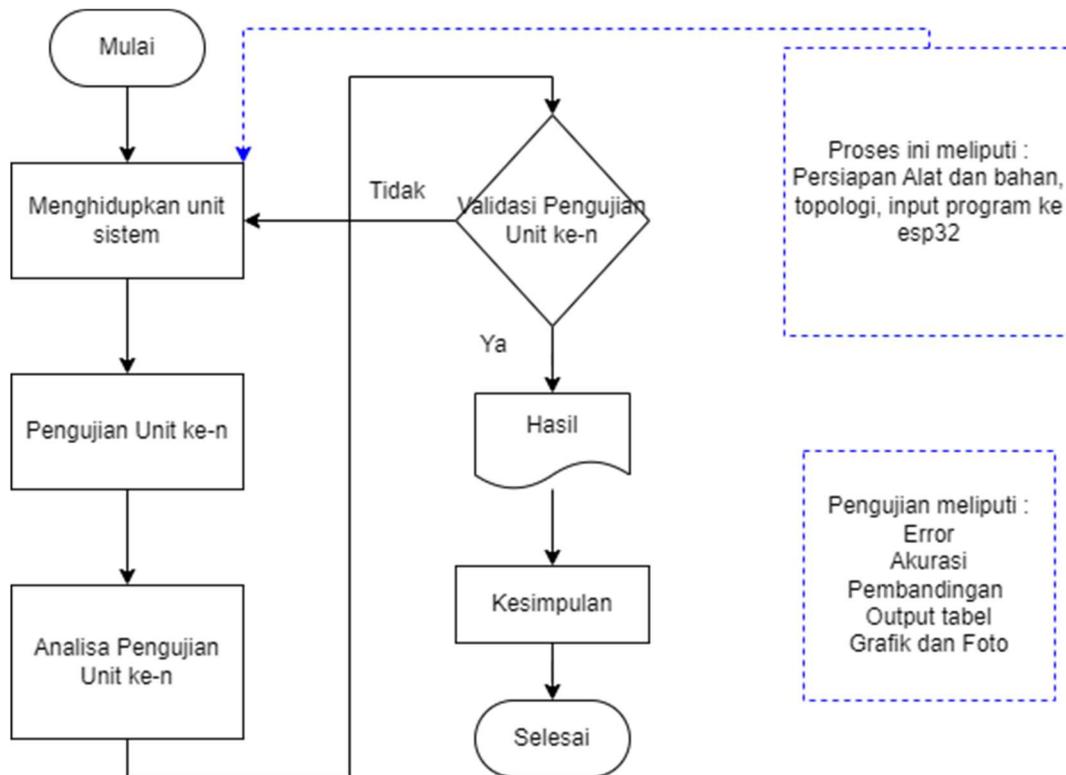
3. Pengujian Fungsionalitas Sistem:

- Pengujian batasan area kerja: Memverifikasi bahwa sistem dapat membatasi pergerakan troli sesuai dengan batasan yang ditentukan dan memberikan peringatan jika terjadi pelanggaran.
- Pengujian notifikasi *realtime*: Menguji kemampuan sistem dalam mengirimkan notifikasi secara *real-time* kepada pengguna atau pihak terkait jika terjadi kejadian penting, seperti pelanggaran batasan area kerja atau kejadian keamanan.
- Pengujian respons antarmuka pengguna: Memverifikasi responsivitas dan kinerja antarmuka pengguna dalam menampilkan informasi troli secara *real-time* dan mengakomodasi perintah pengguna dengan baik.

4. Pengujian Keamanan:

- Pengujian keamanan jaringan: Menguji keamanan koneksi antara ESP32 dan router Mikrotik untuk memastikan bahwa komunikasi tidak rentan terhadap serangan atau akses yang tidak sah.
- Pengujian keamanan data: Memverifikasi bahwa data yang dikirim dari ESP32 ke server *Firebase* diamankan dengan enkripsi dan mekanisme keamanan yang tepat. Pengujian Integrasi Operator server terhadap sistem *smart security*.

Pengujian ini bertujuan untuk memastikan bahwa setiap unit dalam sistem berfungsi dengan baik, komponen terintegrasi dengan lancar, dan sistem dapat beroperasi sesuai dengan kebutuhan dan tujuan penelitian.



Gambar 3. 8 Flowchart Pengujian Tiap Unit

Gambar 3.8 menjelaskan Proses Pengujian Unit diawali dengan Perancangan dan persiapan tiap-tiap komponen yang akan diuji, setelah itu dilakukan pengujian dan hasilnya akan dianalisis dan disesuaikan dengan Spesifikasi pabrikan dari tiap-tiap komponen.

3.12 Pengujian Tiap Bagian

Bagian ini merupakan sebuah sistem yang terintegrasi terkecil pada Perangkat *Smart Security* berbasis IoT yang diterapkan pada troli. Setiap bagian mempunyai langkah kerja masing-masing. Pada Gambar 3.9 di bawah merupakan flowchart pengujian bagian.

1. Sensor ESP32:

- Pengujian keakuratan sensor lokasi: Memverifikasi akurasi pengukuran lokasi troli dengan membandingkan data sensor dengan referensi yang diketahui.
- Pengujian sensor jarak: Mengukur keakuratan pengukuran jarak antara troli dan batasan area kerja dengan menggunakan referensi yang diketahui.
- Pengujian identifikasi troli: Memastikan bahwa sensor dapat mengidentifikasi dan membedakan troli secara unik.

2. Integrasi Komponen:

- Pengujian koneksi ESP32 dengan Mikrotik: Memastikan koneksi jaringan yang stabil dan komunikasi yang baik antara ESP32 dan router Mikrotik.
- Pengujian koneksi ESP32 dengan *Firebase*: Menguji kemampuan ESP32 untuk terhubung dan mengirim data secara *real-time* ke server *Firebase*.
- Pengujian koneksi ESP32 dengan antarmuka pengguna: Memverifikasi bahwa data yang diterima dari ESP32 dapat ditampilkan dengan benar pada antarmuka pengguna.

3. Fungsionalitas Sistem:

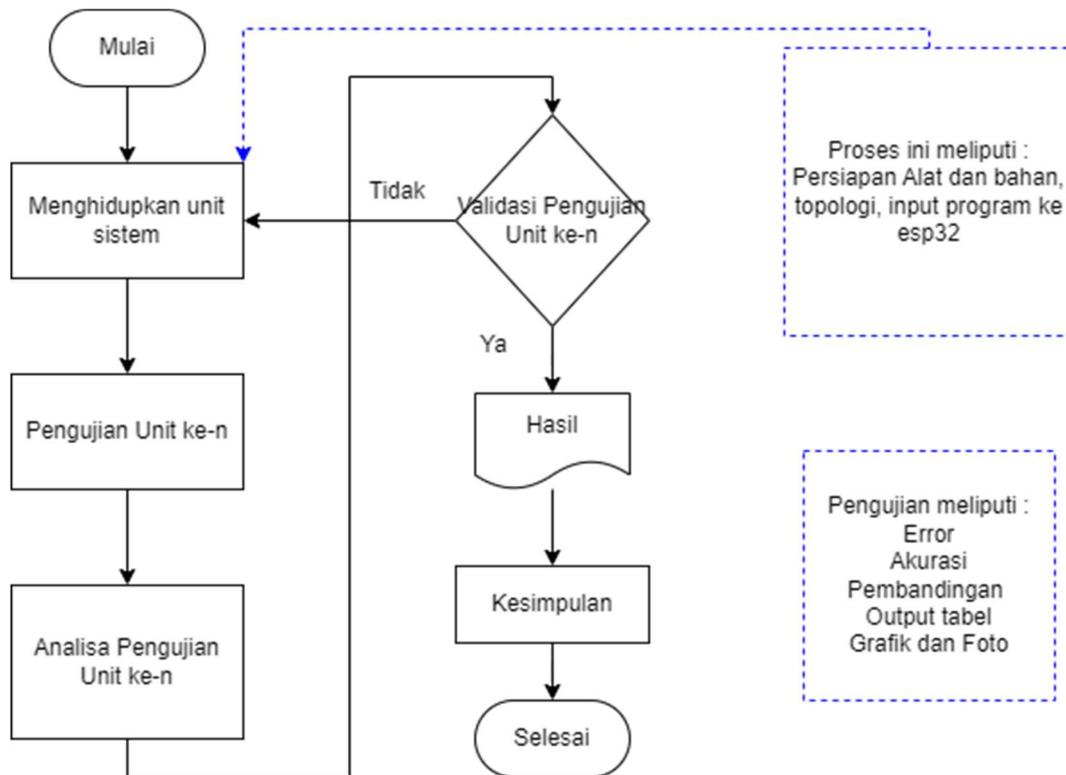
- Pengujian batasan area kerja: Memastikan bahwa sistem dapat membatasi pergerakan troli sesuai dengan batasan yang ditetapkan dan memberikan peringatan jika ada pelanggaran.
- Pengujian notifikasi *real-time*: Menguji kemampuan sistem dalam mengirimkan notifikasi secara *real-time* kepada pengguna atau pihak terkait.

- Pengujian respons antarmuka pengguna: Memastikan bahwa antarmuka pengguna merespons *input* pengguna dengan baik dan menampilkan informasi troli secara *real-time*.

4. Keamanan:

- Pengujian keamanan jaringan: Memverifikasi keamanan jaringan antara ESP32, router Mikrotik, dan server Firebase untuk melindungi data dari serangan atau akses yang tidak sah.
- Pengujian keamanan data: Menguji enkripsi data yang dikirimkan antara ESP32 dan server Firebase untuk melindungi kerahasiaan dan integritas data.

Pengujian ini bertujuan untuk memastikan bahwa setiap bagian sistem berfungsi dengan baik, terintegrasi dengan lancar, dan sesuai dengan kebutuhan dan tujuan penelitian.



Gambar 3. 9 Flowchart Pengujian Bagian

Pengujian perangkat dilakukan untuk menguji kinerja dan keandalan sistem dalam berbagai konfigurasi jaringan. Berikut adalah metode pengujian yang dilakukan dengan menggunakan jumlah router yang berbeda:

9. Pengujian Menggunakan 1 Router:

Pada pengujian ini, sistem akan diuji dalam konfigurasi yang melibatkan satu router sebagai titik akses. Tujuannya adalah untuk memeriksa kinerja sistem dalam skenario jaringan yang sederhana. Pengujian akan melibatkan pengujian kecepatan transfer data, responsivitas sistem, dan kemampuan sistem dalam membatasi area kerja serta melacak pergerakan troli.

10. Pengujian Menggunakan 2 Router:

Pada pengujian ini, sistem akan diuji dengan melibatkan dua router yang saling terhubung. Tujuan pengujian ini adalah untuk menguji kinerja

sistem dalam skenario jaringan yang lebih kompleks, di mana perangkat bergerak akan berpindah antara dua router. Pengujian akan mencakup pemantauan pergerakan troli, pemindahan koneksi antara router, dan pengujian kestabilan jaringan.

11. Pengujian Menggunakan 3 Router:

Pengujian ini akan melibatkan tiga router yang membentuk jaringan yang lebih luas. Tujuan pengujian ini adalah untuk menguji kemampuan sistem dalam beroperasi di lingkungan jaringan yang lebih kompleks, di mana perangkat bergerak akan berpindah antara tiga router yang berbeda. Pengujian akan mencakup pelacakan pergerakan troli, responsivitas sistem, dan kestabilan jaringan.

12. Pengujian Menggunakan 4 Router:

Pada pengujian ini, sistem akan diuji dalam konfigurasi jaringan yang melibatkan empat router. Tujuannya adalah untuk menguji kinerja sistem dalam skenario jaringan yang lebih besar dan kompleks. Pengujian akan melibatkan pengujian responsivitas sistem, pemantauan pergerakan troli, kehandalan jaringan, dan kemampuan sistem dalam mengatur dan membatasi area kerja.

Pengujian akan dilakukan dengan menggunakan sejumlah router yang berbeda untuk mengevaluasi kinerja sistem dalam berbagai skenario jaringan. Hasil pengujian akan memberikan pemahaman yang lebih baik tentang kapabilitas sistem dalam menghadapi lingkungan jaringan yang berbeda.

Berikut adalah skema pengujian untuk mengukur sinyal dengan tingkat kekuatan yang bervariasi dari terendah ke tertinggi, serta pengujian kehilangan sinyal dalam dB:

1. Pengujian Tingkat Sinyal Terendah:

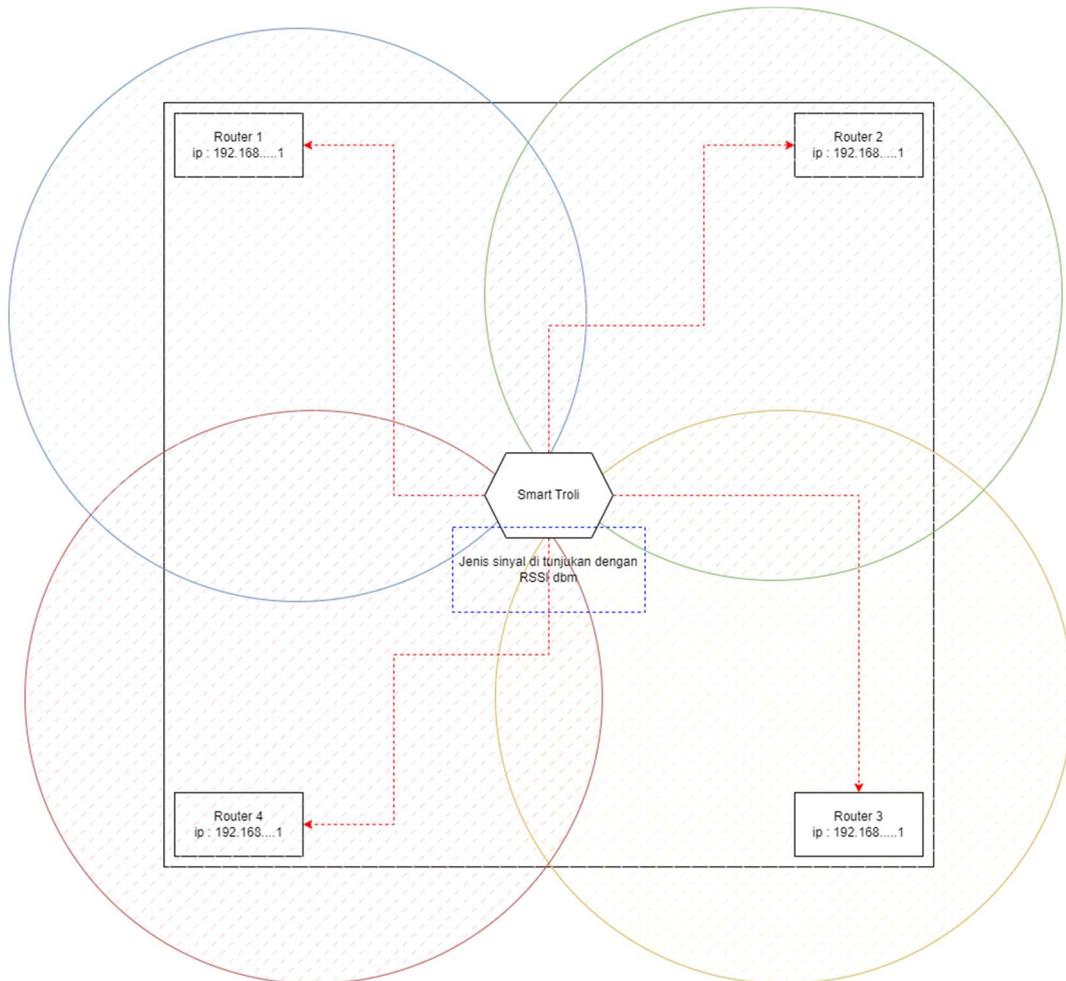
Pada pengujian ini, sinyal dengan kekuatan terendah akan digunakan. Tujuan dari pengujian ini adalah untuk mengevaluasi kinerja sistem dalam kondisi jaringan yang memiliki tingkat sinyal yang sangat lemah. Pengujian dilakukan dengan mengukur dan mencatat kekuatan sinyal yang diterima oleh perangkat dengan tingkat sensitivitas yang tinggi.

2. Pengujian Tingkat Sinyal Tertinggi:

Pada pengujian ini, sinyal dengan kekuatan tertinggi akan digunakan. Pengujian ini bertujuan untuk menguji kinerja sistem dalam kondisi jaringan yang memiliki tingkat sinyal yang sangat kuat. Pengujian dilakukan dengan mengukur dan mencatat kekuatan sinyal yang diterima oleh perangkat dengan tingkat sensitivitas yang tinggi.

3. Pengujian Kehilangan Sinyal (Loss):

Pengujian ini bertujuan untuk mengukur kehilangan sinyal (loss) yang terjadi dalam jaringan. Kehilangan sinyal dapat terjadi karena faktor-faktor seperti jarak, hambatan fisik, atau interferensi. Pengujian dilakukan dengan mengirimkan sinyal dari sumber ke perangkat yang menerima, dan kemudian mengukur perbedaan antara kekuatan sinyal yang dikirimkan dan diterima. Hasil pengujian akan memberikan informasi tentang keandalan dan kestabilan jaringan.



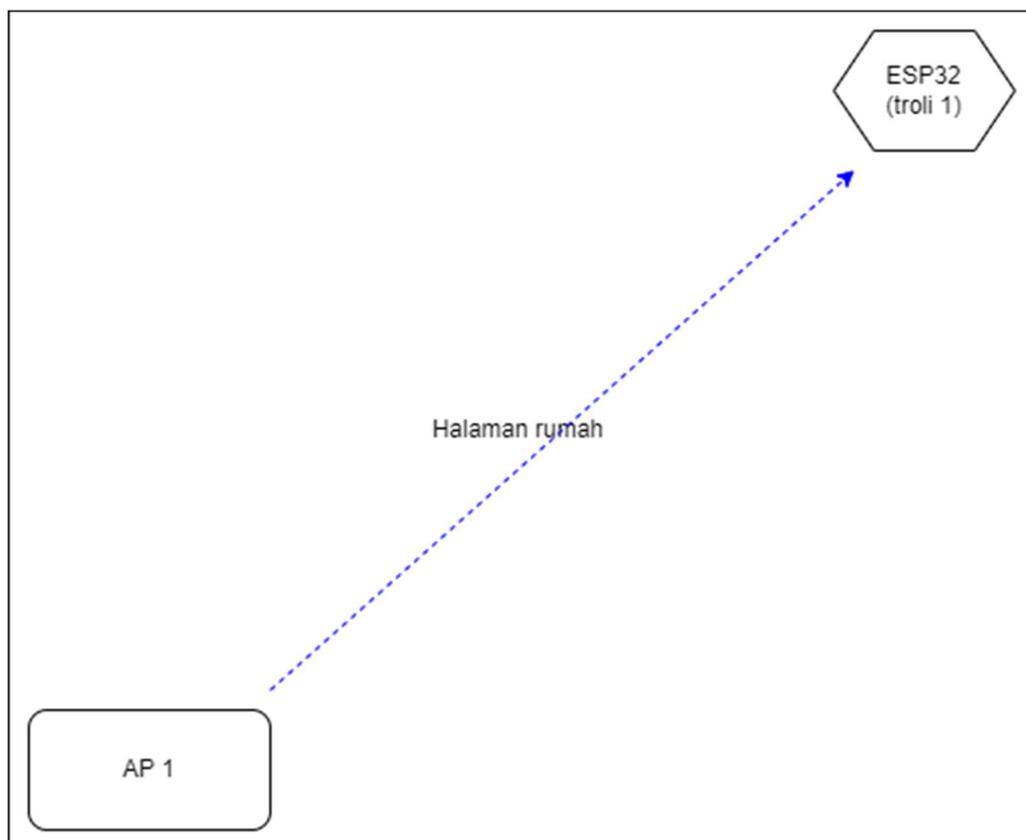
Gambar 3. 10 Denah Pengujian

Pada gambar 3.10 pengujian ini dilakukan dengan menggunakan peralatan pengukuran yang sesuai, seperti *spectrum analyzer* atau *power meter*, untuk mendapatkan data yang akurat. Hasil pengujian akan memberikan pemahaman yang lebih baik tentang performa sistem dalam kondisi sinyal yang berbeda-beda, serta kemampuan sistem dalam mengatasi kehilangan sinyal dalam jaringan.

3.12.1 Denah pengujian LOS (*Line-of-Sight*)

Line-of-Sight (LOS) merupakan kondisi dimana antenna pada perangkat ESP32 memiliki jalur langsung dan tidak terhalang untuk berkomunikasi dengan perangkat lain atau *Access Point* (AP) dalam jaringan. Pada kondisi

LOS, tidak ada hambatan fisik seperti dinding, pohon, atau objek lain yang menghalangi jalur transmisi sinyal. Kondisi LOS biasanya menghasilkan sinyal yang kuat dan stabil, sehingga memungkinkan perangkat ESP32 untuk beroperasi dengan efisiensi tinggi dan akurat dalam membatasi area kerja.

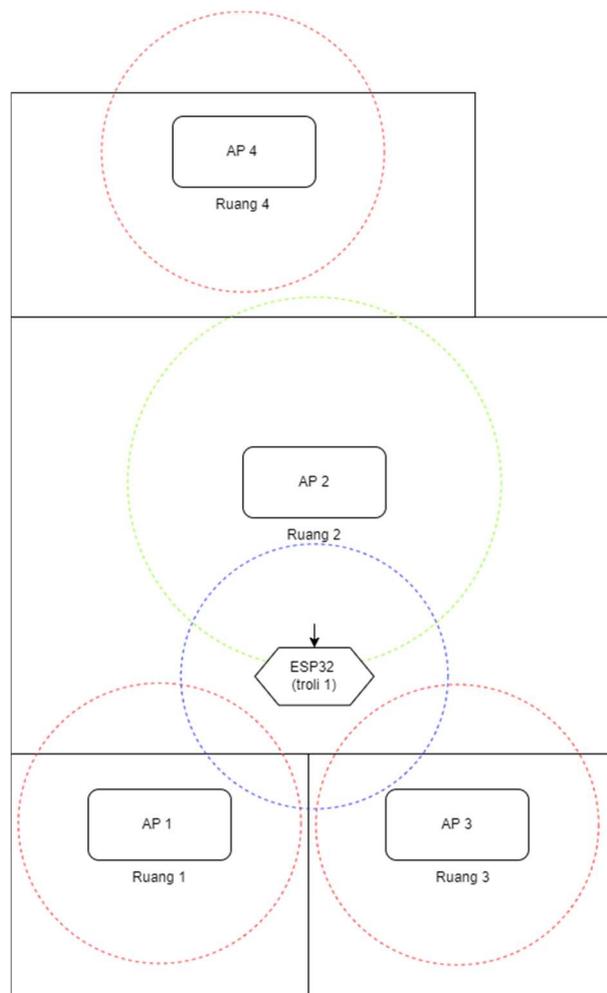


Gambar 3. 11 Denah Pengujian Los

Terlihat pada gambar 3.11 pengujian LOS di halaman rumah akan memastikan bahwa sistem dapat beroperasi dengan optimal dalam kondisi tanpa hambatan fisik yang signifikan. Pada kondisi LOS, antena pada ESP32 akan memiliki jalur langsung ke *Access Point* (AP) atau perangkat lainnya tanpa adanya halangan seperti dinding atau benda lain yang menghalangi sinyal. Pengujian ini akan membantu mengidentifikasi sejauh mana jangkauan dan kekuatan sinyal pada kondisi ideal, yang akan menjadi acuan untuk mengukur performa sistem dalam kondisi non-LOS di dalam rumah.

3.12.2 Denah Pengujian non-LOS (*Non Line of Sight*)

Non-LOS merupakan kondisi dimana antena pada perangkat ESP32 menghadapi hambatan fisik atau objek-objek tertentu yang menghalangi jalur transmisi sinyal ke perangkat lain atau AP dalam jaringan. Kondisi non-LOS dapat terjadi ketika perangkat ESP32 berada di area yang memiliki banyak dinding, gangguan interferensi, atau objek-objek lain yang menghalangi jalur sinyal. Kondisi non-LOS menyebabkan sinyal menjadi lemah, fluktuasi *latency* terjadi, dan keakuratan dalam pembatasan area kerja menjadi berkurang. Berikut denah pengujian pada gambar 3.12.



Gambar 3. 12 Denah Pengujian *non-Los*

Pengujian *non-LOS* di dalam rumah dengan menggunakan 4 ruangan berbeda akan mensimulasikan kondisi lingkungan yang lebih realistis, di mana terdapat hambatan fisik seperti dinding dan ruangan yang berbeda. Pengujian ini akan membantu mengidentifikasi bagaimana perangkat ESP32 berkinerja dalam kondisi yang lebih kompleks, di mana sinyal harus melewati dinding dan benda lainnya. Hasil dari pengujian ini akan memberikan informasi tentang fluktuasi sinyal, tingkat kekuatan sinyal yang terjadi, dan apakah ada pengaruh gangguan interferensi di dalam ruangan.

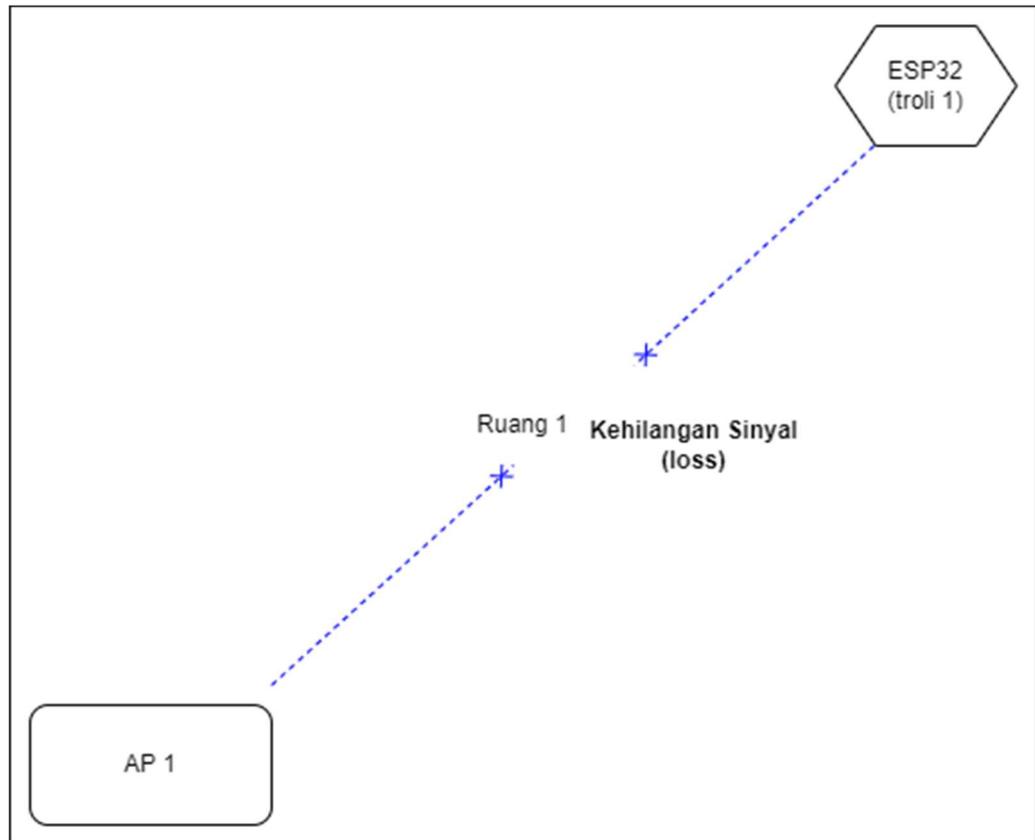
3.12.3 Denah Pengujian *LOSS*

Ketika ESP32 berada dalam keadaan *loss*, berarti sinyal yang diterima oleh perangkat tersebut sangat lemah atau bahkan tidak dapat terdeteksi sama sekali. Hal ini dapat terjadi ketika perangkat berada dalam jarak yang sangat jauh dari *Access Point* (AP) atau terdapat banyak hambatan yang menghalangi jalur sinyal antara ESP32 dan AP.

Dalam keadaan *loss*, ESP32 tidak dapat berkomunikasi dengan AP atau akses jaringan lainnya, sehingga mengakibatkan terputusnya koneksi dan akses internet. Penggunaan ESP32 dalam kondisi *loss* mungkin dapat mengganggu fungsi utama dari sistem pembatasan area kerja berbasis IoT yang menggunakan perangkat ini.

Meskipun keadaan *loss* mungkin tidak diinginkan, namun informasi tentang kehilangan sinyal (*loss*) dapat memberikan informasi penting dalam pengoperasian sistem. Data mengenai kehilangan sinyal dapat digunakan untuk melakukan pemantauan dan analisis kualitas jaringan. Dengan mengetahui daerah atau titik-

titik yang mengalami *loss*, pengaturan posisi AP dapat diperbaiki atau ditingkatkan untuk meningkatkan kualitas jaringan dan meminimalkan area dengan sinyal lemah.



Gambar 3. 13 Denah Pengujian Ketika Terjadi Loss

Dapat kita lihat pada gambar 3.13 keadaan *loss* juga bisa menjadi indikator adanya masalah dalam infrastruktur jaringan atau lingkungan yang mempengaruhi sinyal. Hal ini memungkinkan tim teknis untuk melakukan pemecahan masalah dan peningkatan untuk mencapai koneksi yang lebih stabil dan handal.

Dalam konteks sistem pembatasan area kerja berbasis IoT, deteksi *loss* pada ESP32 memainkan peran penting dalam memastikan sistem berfungsi dengan baik dan memenuhi tujuan keamanan serta efisiensi yang diinginkan. Dengan memahami kondisi *loss* dan mengambil langkah-langkah perbaikan yang tepat,

sistem dapat mencapai kinerja yang lebih optimal dan meningkatkan pengawasan dan keamanan area kerja secara efektif.

3.13 Pengujian Sistem

Pengujian Sistem ditujukan untuk menilai apakah perangkat *smart security* lebih efektif atau tidak dibandingkan penggunaan keamanan menggunakan satpam atau human energy. Parameter yang diuji meliputi :

1. Pengujian Kinerja Sistem:

- Pengujian stabilitas sistem: Memastikan bahwa sistem dapat berjalan dengan stabil tanpa adanya kegagalan atau crash yang sering.
- Pengujian respons sistem: Memeriksa respons sistem terhadap perintah pengguna atau *input* data sensor dalam waktu yang sesuai.
- Pengujian waktu respons: Mengukur waktu respons sistem dalam menanggapi perintah atau perubahan status troli.

2. Pengujian Integrasi Keseluruhan:

- Pengujian komunikasi antara komponen: Memastikan bahwa semua komponen dalam sistem dapat berkomunikasi dan berinteraksi dengan lancar.
- Pengujian keselarasan data: Memverifikasi bahwa data yang dikirim dan diterima oleh sistem sesuai dengan format yang diharapkan dan tidak ada kesalahan atau kehilangan data.

3. Pengujian Kegunaan dan Keandalan:

- Pengujian kegunaan sistem: Melibatkan pengguna atau pihak terkait untuk menggunakan sistem dan memberikan umpan balik mengenai

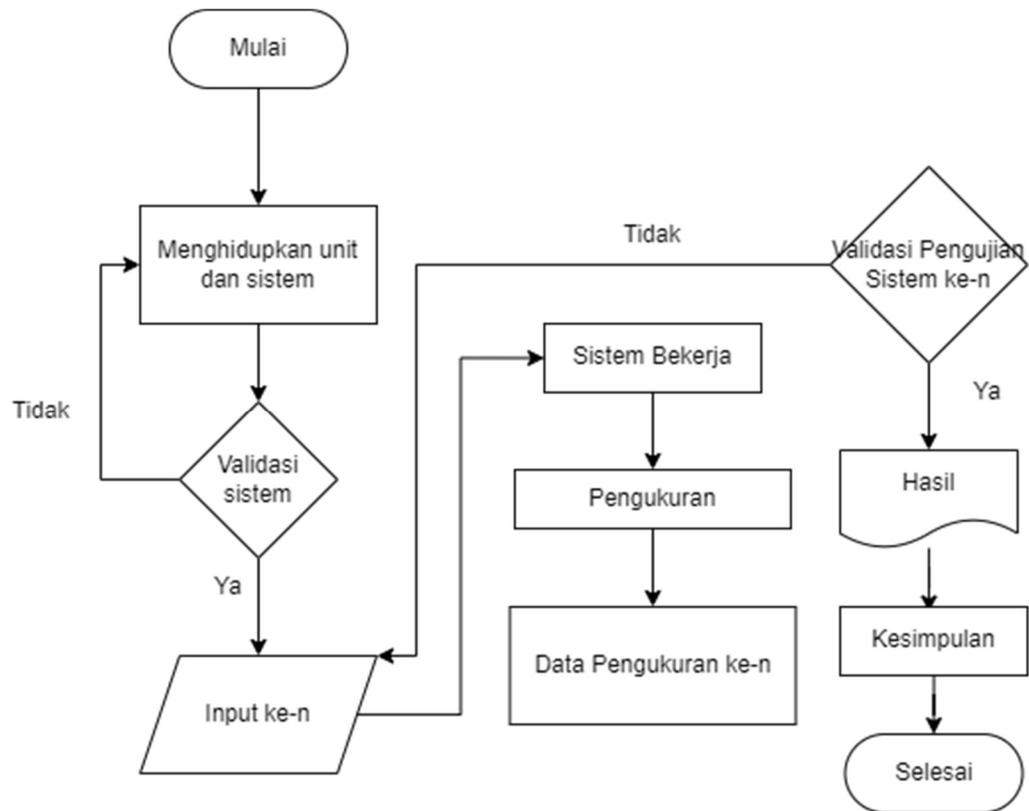
pengalaman penggunaan, kemudahan penggunaan, dan kegunaan sistem secara keseluruhan.

- Pengujian keandalan sistem: Melakukan pengujian berkelanjutan atau simulasi penggunaan sistem dalam jangka waktu yang cukup lama untuk memastikan bahwa sistem dapat beroperasi secara konsisten dan dapat diandalkan.

4. Pengujian Skalabilitas:

- Pengujian skalabilitas sistem: Memeriksa kemampuan sistem untuk menangani beban kerja yang tinggi dan sejumlah besar data tanpa mengalami penurunan kinerja yang signifikan.
- Pengujian skalabilitas jaringan: Menguji kemampuan jaringan untuk menangani jumlah perangkat yang lebih besar dan komunikasi yang intensif.

Gambar 31.4 di bawah merupakan pengujian sistem ini yang bertujuan untuk memastikan bahwa sistem yang dikembangkan berkinerja baik, dapat diandalkan, dan memenuhi kebutuhan pengguna serta tujuan penelitian secara keseluruhan.



Gambar 3. 14 Flowchart Pengujian Sistem

3.13.1 Hasil Pengujian Menggunakan Bluetooth

Bluetooth pada sistem pembatas area kerja berbasis *Internet of Things* (IoT) dengan ESP32 merupakan salah satu fitur yang dimiliki oleh *board* ESP32 dan menjadi bagian penting dalam membandingkan performa komunikasi nirkabel dengan fitur lain yang ada dalam sistem.

Dengan menguji fitur Bluetooth pada ESP32, kita dapat membandingkan performa komunikasi Bluetooth dengan fitur lain yang ada dalam sistem, seperti WiFi atau protokol kustom. Hal ini akan membantu kita menilai keunggulan dan kelemahan dari masing-masing fitur komunikasi dan menentukan fitur mana yang paling cocok untuk digunakan dalam skenario tertentu.

ESP32 telah dilengkapi dengan modul Bluetooth yang sudah terintegrasi, sehingga memudahkan penggunaan fitur ini tanpa memerlukan tambahan perangkat keras atau konfigurasi kompleks. Dengan menganalisis fitur Bluetooth yang ada dalam ESP32, kita dapat mengevaluasi kemudahan penggunaan dan ketersediaan fitur ini dalam implementasi sistem.

Bluetooth adalah protokol komunikasi nirkabel yang banyak digunakan dalam perangkat konsumen, seperti headphone nirkabel, keyboard, dan perangkat wearable. Dengan menguji fitur Bluetooth pada ESP32, kita dapat menilai fleksibilitas dan kompatibilitasnya dengan perangkat lain yang menggunakan Bluetooth, serta potensi untuk terintegrasi dengan berbagai aplikasi dan ekosistem.

Bluetooth Low Energy (BLE) adalah varian dari Bluetooth yang dirancang khusus untuk penggunaan dalam IoT. Dengan menganalisis fitur Bluetooth pada ESP32, kita dapat mengevaluasi kemampuan ESP32 dalam mendukung komunikasi BLE dan memperkirakan daya tahan baterai saat menggunakan fitur Bluetooth dalam mode BLE.

Dengan menganalisis dan membandingkan fitur Bluetooth pada ESP32 dengan fitur lain dalam sistem, kita dapat mengidentifikasi kekuatan dan kelemahan masing-masing fitur, memahami bagaimana fitur-fitur ini berinteraksi dan berdampak pada kinerja keseluruhan sistem, serta memastikan bahwa sistem pembatas area kerja berbasis IoT dengan ESP32 dapat beroperasi dengan efisien dan efektif dalam memenuhi kebutuhan dan tujuan penelitian. (*ESP32-S3 Series Hardware Design Guidelines Introduction*, 2023)

3.13.2 Hasil Pengujian Menggunakan Beberapa *Access Point* (AP)

Pengujian menggunakan beberapa *access point* (AP) bertujuan untuk mengevaluasi bagaimana perangkat ESP32 berinteraksi dengan jaringan yang terdiri dari beberapa titik akses. Dalam konteks penelitian sistem pembatas area kerja berbasis *Internet of Things* (IoT), pengujian ini menjadi penting karena memungkinkan peneliti untuk mengidentifikasi dan memahami bagaimana perangkat ESP32 dapat beradaptasi dengan lingkungan jaringan yang kompleks. Tujuan dari pengujian menggunakan beberapa *access point* adalah:

1. **Evaluasi Jangkauan Sinyal:** Pengujian ini akan membantu menilai sejauh mana jangkauan sinyal WiFi dari masing-masing *access point*. Dengan demikian, peneliti dapat menentukan bagian mana dari area kerja yang cakupan sinyalnya masih kuat dan di mana mungkin terdapat "*blind spots*" atau area dengan sinyal lemah.
2. **Handover dan Roaming:** Pengujian ini akan memastikan bahwa perangkat ESP32 dapat melakukan *handover* atau *roaming* dengan mulus saat berpindah dari satu *access point* ke *access point* lain. Hal ini penting untuk memastikan konektivitas yang terus-menerus saat perangkat bergerak di sekitar area kerja.
3. **Peningkatan Kinerja Jaringan:** Dengan menggunakan beberapa *access point*, penelitian dapat mengevaluasi kinerja jaringan secara keseluruhan dan mengidentifikasi area-area di mana *throughput* atau kecepatan transfer data mungkin berkurang karena terlalu banyak perangkat yang terhubung ke satu *access point*.

4. Redundansi dan Keandalan: Dengan mengaktifkan beberapa *access point*, pengujian ini akan membantu menilai keandalan jaringan dalam skenario jika salah satu *access point* mengalami gangguan atau kegagalan. *access point* lain akan memastikan bahwa jaringan tetap berfungsi dengan baik dan penggunaan perangkat ESP32 tidak terganggu.

Melalui pengujian ini, penelitian akan memperoleh wawasan yang lebih mendalam tentang kinerja dan kehandalan jaringan dalam lingkungan yang lebih kompleks dengan beberapa *access point*. Hal ini akan menjadi landasan penting dalam merancang solusi yang optimal untuk sistem pembatas area kerja berbasis IoT dengan memastikan kualitas sinyal dan konektivitas yang baik di seluruh area kerja. (ZoneFlex TM 7372 DUAL-BAND 802.11N SMART WI-FI ACCESS POINTS High Performance, 802.11n Mid-Range Smart Wi-Fi Access Points with Adaptive Antenna Technology, no date)

3.14 Analisis hasil Penelitian

Data yang diperoleh dari semua Pengujian dihimpun dalam sebuah tabel untuk melihat kinerja dari masing-masing komponen yang digunakan. Dalam analisis hasil penelitian ini, kita akan berfokus pada empat aspek utama untuk mendapatkan pemahaman yang jelas. Evaluasi kinerja sistem dari segi responsivitas, stabilitas, dan keandalan. Kami akan menganalisis waktu respons sistem, kecepatan transfer data, serta tingkat kegagalan sistem untuk mengukur sejauh mana sistem beroperasi dengan baik dan dapat diandalkan.

Mengevaluasi efektivitas fungsionalitas sistem, seperti kemampuan sistem dalam membatasi area kerja, melakukan pelacakan trolley, dan memberikan notifikasi secara *real-time*. Analisis ini akan membantu kami memahami sejauh mana sistem

dapat memenuhi tujuan dan kebutuhan penelitian. *Usability dan User Experience*, pengumpulan umpan balik dari pengguna atau pihak terkait yang akan membantu kami mengevaluasi kegunaan sistem, kepuasan pengguna, dan pengalaman pengguna secara keseluruhan.

Proses analisis aspek keamanan sistem, termasuk keamanan jaringan dan perlindungan data. Selain itu, kami juga akan mengevaluasi kemampuan sistem dalam menangani pertumbuhan dan peningkatan beban kerja untuk memastikan sistem dapat berskala dengan baik seiring dengan perkembangan kebutuhan.

Dengan melakukan analisis mendalam terhadap kinerja sistem, efektivitas fungsionalitas, *usability* dan *user experience*, serta keamanan dan skalabilitas, kami dapat mendapatkan wawasan yang jelas tentang keberhasilan penelitian ini dalam mencapai tujuan yang telah ditetapkan.