

**PERANCANGAN SISTEM PEMANTAUAN DAN  
PENGENDALIAN ASAP ROKOK BERBASIS *INTERNET OF  
THINGS (IOT)***

**Studi Kasus: Aula KAPMI D. I. Yogyakarta**

**LAPORAN TUGAS AKHIR**



**KHAIRIL HAMDI**

**5211011024**

**PROGRAM STUDI TEKNIK KOMPUTER  
FAKULTAS SAINS & TEKNOLOGI  
UNIVERSITAS TEKNOLOGI YOGYAKARTA  
YOGYAKARTA  
2025**

**PERANCANGAN SISTEM PEMANTAUAN DAN  
PENGENDALIAN ASAP ROKOK BERBASIS *INTERNET OF  
THINGS (IOT)***

Disusun oleh:  
**KHAIRIL HAMDI**  
**5211011024**

Telah dipertahankan di depan Dewan Penguji  
pada tanggal 20 Juni 2025

Nama	Jabatan	Tanda tangan	Tanggal
<u>Ari Sugiharto, S.Si., M.Eng.</u> NIK. 1109816089	Ketua Penguji	1. 	25/06/2025
<u>Anna Dina Kalifia, S.Kom., M.Cs.</u> NIK. 110221183	Penguji I	2. 	25/06/2025
<u>Ir. Agus Suhendar, S.T., M.Eng.</u> NIK. 110919178	Penguji II (Dosen Pembimbing)	3. 	26/06/2025



## **PERNYATAAN**

Yang bertanda tangan di bawah ini, saya

Nama : Khairil Hamdi  
NPM : 5211011024  
Program Studi : Teknik Komputer  
Program : Sarjana  
Fakultas : Sains & Teknologi

menyatakan bahwa laporan Tugas Akhir dengan Judul “Perancangan Sistem Pemantauan Dan Pengendalian Asap Berbasis *Internet of things (Iot)*” adalah hasil pekerjaan saya sendiri dan tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi. Sepanjang pengetahuan saya, laporan Tugas Akhir ini tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dengan mengikuti tata cara dan etika penulisan karya ilmiah yang lazim.

Apabila ternyata terbukti bahwa pernyataan ini tidak benar, sepenuhnya menjadi tanggung jawab saya.

Yogyakarta, 26 Juni 2025



Khairil Hamdi

## KATA PENGANTAR

Puji syukur saya panjatkan ke hadirat Tuhan Yang Maha Esa atas limpahan rahmat dan karunia-Nya, sehingga saya dapat menyelesaikan laporan *Tugas Akhir* yang berjudul “Perancangan Sistem Pemantauan dan Pengendalian Asap Rokok Berbasis Internet of Things (IoT)” dengan baik. Laporan ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana pada Program Studi Teknik Komputer, Fakultas Sains dan Teknologi, Universitas Teknologi Yogyakarta.

Saya menyadari bahwa penyusunan laporan Tugas Akhir ini tidak akan berhasil tanpa dukungan dari berbagai pihak. Oleh karena itu, dengan tulus saya mengucapkan terima kasih kepada:

1. Allah SWT, Tuhan Yang Maha Esa, atas segala rahmat dan karunia-Nya.
2. Kedua orang tua (Bapak Farodi dan Ibu Fatkhiyah) dan keluarga tercinta yang selalu memberikan doa, motivasi, dan dukungan moral maupun material.
3. Bapak Dr. Bambang Moertono Setiawan, M.M., CA., Akt., selaku Rektor Universitas Teknologi Yogyakarta.
4. Ibu Dr. Endy Marlina, S.T., M.T., selaku Dekan Fakultas Sains dan Teknologi Universitas Teknologi Yogyakarta.
5. Dr. Eng. Arif Pramudwiatmoko, S. T., M.Eng., selaku Ketua Program Studi Teknik Komputer.
6. Ir. Agus Suhendar, S.T., M.Eng., selaku Dosen Pembimbing Tugas Akhir yang telah memberikan bimbingan, pengetahuan, dan arahan yang sangat berarti dalam proses penyusunan laporan ini.
7. Pihak KAPMI D.I. Yogyakarta yang telah memberikan kesempatan dan fasilitas dalam proses pengujian sistem.
8. Seluruh teman-teman Teknik Komputer Angkatan 2021 atas semangat, kerja sama, dan dukungan yang tidak ternilai.
9. Serta semua pihak lain yang tidak dapat saya sebutkan satu per satu, yang telah membantu secara langsung maupun tidak langsung.

Saya menyadari bahwa laporan ini masih jauh dari kata sempurna. Oleh karena itu, kritik dan saran yang membangun sangat saya harapkan demi perbaikan di masa mendatang. Semoga laporan ini dapat memberikan manfaat dan menjadi referensi yang berguna bagi pengembangan sistem serupa di masa mendatang.

Yogyakarta,.....

Khairil Hamdi

## DAFTAR ISI

HALAMAN PENGESAHAN LAPORAN TUGAS AKHIR.....	i
PERNYATAAN.....	ii
KATA PENGANTAR .....	iii
DAFTAR ISI.....	v
DAFTAR TABEL.....	vii
DAFTAR GAMBAR .....	viii
DAFTAR LAMPIRAN.....	ix
ABSTRAK .....	x
<i>ABSTRACT</i> .....	xi
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Sistematika Penulisan.....	3
BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI .....	6
2.1 Tinjauan Pustaka .....	6
2.2 Landasan Teori.....	10
2.2.1 Asap Rokok .....	10
2.2.2 Kategori Indeks Standar Pencemaran Udara .....	10
2.2.3 <i>Internet of things</i> .....	11
2.2.4 <i>Firebase Realtime Database</i> .....	12
2.2.5 Mikrokontroler ESP32 .....	12
2.2.6 Sensor MQ135 .....	13
2.2.7 MIT App Inventor.....	14
2.2.8 <i>Flowchart</i> .....	14
2.2.9 <i>Blackbox</i> .....	16
BAB III METODE PENELITIAN.....	21
3.1 Metode Penelitian.....	21
3.2 Alat.....	23

3.3	Bahan .....	24
BAB IV ANALISIS DAN PERANCANGAN .....		26
4.1	Perancangan dan Analisis Sistem.....	26
4.1.1	Analisis Kebutuhan Fungsional .....	26
4.1.2	Analisis Kebutuhan Non-Fungsional.....	27
4.1.3	Struktur Sistem .....	28
4.1.4	Perancangan Rangkaian.....	29
4.2	Perancangan Perangkat Lunak .....	31
4.2.1	<i>Flowchart</i> Sistem.....	31
4.2.2	Desain Antar Muka Aplikasi .....	33
BAB V HASIL DAN PEMBAHASAN.....		40
5.1	Implementasi Sistem .....	40
5.1.1	Implementasi Perangkat Keras .....	40
5.1.2	Implementasi Perangkat Lunak .....	45
5.2	Pengujian Hardware .....	51
5.2.1	Pengujian Akurasi Sensor MQ-135 .....	51
5.2.2	Pengujian Buzzer .....	52
5.2.3	Pengujian <i>Relay</i> dan <i>Fan</i> .....	53
5.3	Pengujian Alat Secara Keseluruhan .....	54
5.3.1	Pengujian Dengan Asap Pembakaran Kertas.....	54
5.3.2	Pengujian Dengan Asap Rokok .....	56
5.3.3	Pengujian Dengan Asap <i>Vape</i> (Rokok Elektrik) .....	57
5.4	Analisis Hasil Pengujian .....	59
5.4.1	Analisis Pengujian Asap Pembakaran Kertas.....	59
5.4.2	Analisis Pengujian Asap Rokok .....	59
5.4.3	Analisis Pengujian Asap <i>Vape</i> .....	60
5.4.4	Analisis Umum Sistem .....	60
5.5	Kesesuaian Dengan Tujuan.....	60
BAB VI PENUTUP .....		60
6.1	Kesimpulan .....	60
6.2	Saran.....	61
DAFTAR PUSTAKA .....		62
LAMPIRAN.....		65

## DAFTAR TABEL

Tabel 2. 1 Tinjauan pustaka .....	8
Tabel 2. 2 Kategori Indeks Standar Udara .....	11
Tabel 3. 1 Alat Penelitian.....	23
Tabel 3. 2 Daftar bahan.....	24
Tabel 4. 1 Pin ESP 32 dan Mq-135.....	30
Tabel 4. 2 Pin ESP 32 dan <i>Relay</i> .....	30
Tabel 4. 3 Pin ESP 32 dan <i>Buzzer</i> .....	30
Tabel 4. 4 Pin <i>Relay</i> , Adapter, dan <i>Fan</i> .....	30
Tabel 5. 1 Akurasi Sensor MQ135 dan Alat Ukur Standar .....	52
Tabel 5. 2 Pengujian Sensor Mq-135 dan Buzzer.....	53
Tabel 5. 3 Pengujian <i>Relay</i> dan <i>Fan</i> .....	53
Tabel 5. 4 Pengujian dengan asap pembakaran kertas .....	55
Tabel 5. 5 Pengujian dengan Asap Rokok .....	57
Tabel 5. 6 Pengujian dengan Asap Vape(Rokok Elektrik) .....	58

## DAFTAR GAMBAR

Gambar 2. 1 Mikrokontroler ESP32 (Muliadi , , 2024) .....	13
Gambar 2. 2 Sensor MQ-135(Nur Hakim, 2023) .....	13
Gambar 2. 3 Arduino Ide .....	<b>Error! Bookmark not defined.</b>
Gambar 2. 4 MIT App Inventor (Top & Gökbüllüt, 2022)...	<b>Error! Bookmark not defined.</b>
Gambar 3. 1 <i>Flowchart</i> Penelitian .....	21
Gambar 4. 1 Diagram Blok Perencanaan Sistem .....	28
Gambar 4. 2 Schematic rangkaian .....	29
Gambar 4. 3 <i>Flowchart</i> Sistem Alat .....	31
Gambar 4. 4 <i>Flowchart</i> Sistem <i>Firebase</i> dan Aplikasi.....	32
Gambar 4. 12 Desain Antarmuka Aplikasi .....	33
Gambar 5. 1 Implementasi Rangkaian .....	40
Gambar 5. 2 Inisialisasi Koneksi WiFi dan <i>Firebase</i> .....	41
Gambar 5. 3 Deklarasi Pin dan Variabel Sistem.....	42
Gambar 5. 4 Fungsi Pembacaan Sensor.....	42
Gambar 5. 5 Setup Koneksi dan Konfigurasi Sistem.....	43
Gambar 5. 6 Proses Utama dan Logika Pengambilan Keputusan.....	44
Gambar 5. 7 Tampilan Antarmuka Aplikasi.....	46
Gambar 5. 8 Tampilan data <i>real-time</i> pada <i>Firebase Realtime Database</i> .....	47
Gambar 5. 9 Pengaturan rules <i>Firebase Realtime Database</i> .....	48
Gambar 5. 10 Tampilan Desain Aplikasi pada MIT App Inventor.....	49
Gambar 5. 11 Konfigurasi Komponen <i>Firebase</i> di MIT App Inventor.....	49
Gambar 5. 12 Blok Diagram Aplikasi.....	50
Gambar 5. 13 Kondisi Normal .....	54
Gambar 5. 14 Pengujian alat dengan Asap Pembakaran Kertas .....	55
Gambar 5. 15 Kondisi Sebelum pengujian Asap Rokok.....	56
Gambar 5. 16 Pengujian dengan Asap Rokok .....	56
Gambar 5. 17 Kondisi sebelum pengujian dengan <i>Vape</i> .....	58
Gambar 5. 18 Pengujian dengan Asap <i>Vape</i> (Rokok Elektrik).....	58

## **DAFTAR LAMPIRAN**

Lampiran 1 Dokumentasi Pembuatan Alat .....	65
Lampiran 2 Tempat Penelitian .....	65

## ABSTRAK

Penelitian ini merancang dan mengembangkan sistem pemantauan dan pengendalian asap rokok berbasis *Internet of things* (IoT) dengan studi kasus di Aula KAPMI D.I. Yogyakarta, yang memiliki tingkat aktivitas merokok cukup tinggi dan belum tersedia sistem pemantauan kualitas udara otomatis. Sistem ini bertujuan untuk mendeteksi dan merespons keberadaan asap rokok secara *real-time* melalui aplikasi *mobile*. Perancangan mencakup integrasi sensor MQ-135 untuk deteksi asap, mikrokontroler ESP32 sebagai pusat kendali, serta *Firebase Realtime Database* untuk penyimpanan data secara *cloud*. Aplikasi berbasis MIT App Inventor memungkinkan pengguna memantau kadar asap dalam PPM serta status kipas (ON/OFF). Ketika konsentrasi asap melebihi ambang batas 100 PPM, sistem secara otomatis mengaktifkan *buzzer* sebagai alarm dan kipas sebagai sistem ventilasi, serta mengirimkan notifikasi ke aplikasi. Pengujian menunjukkan bahwa sensor MQ-135 efektif dalam mendeteksi berbagai jenis asap seperti asap rokok, vape, dan pembakaran kertas, dengan respons cepat dan akurat. Hasil tersebut menunjukkan bahwa arsitektur sistem IoT yang digunakan mampu memberikan solusi pemantauan kualitas udara yang responsif dan aplikatif.

**Kata Kunci:** *Internet of things* (IoT), Asap Rokok, Sensor MQ-135, ESP32, *Firebase Realtime Database*, MIT App Inventor, Kualitas Udara.

## ***ABSTRACT***

*This research designs and develops an Internet of things (IoT)-based cigarette smoke monitoring and control system with a case study at KAPMI Hall, Yogyakarta, which has a high level of smoking activity and no automatic air quality monitoring system. This system aims to detect and respond to the presence of cigarette smoke in real-time through a mobile application. The design includes the integration of MQ-135 sensor for smoke detection, ESP32 microcontroller as the control center, and Firebase Realtime Database for cloud data storage. The MIT App Inventor-based application allows users to monitor smoke levels in PPM as well as fan status (ON/OFF). When the smoke concentration exceeds the 100 PPM threshold, the system automatically activates the buzzer as an alarm and the fan as a ventilation system, and sends a notification to the app. Tests show that the MQ-135 sensor is effective in detecting various types of smoke such as cigarette smoke, vape, and paper burning, with a fast and accurate response. These results show that the IoT system architecture used is able to provide a responsive and applicable air quality monitoring solution.*

**Keywords:** *Internet of things (IoT), Cigarette Smoke, MQ-135 Sensor, ESP32, Firebase Realtime Database, MIT App Inventor, Air Quality.*

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Indonesia adalah salah satu negara yang paling banyak menghasilkan dan mengonsumsi tembakau di dunia. Menurut data dari Survei Demografi Kesehatan Indonesia tahun 2017 dan Riset Kesehatan Dasar tahun 2018, proporsi orang Indonesia yang mengonsumsi tembakau pada umur 15 tahun ke atas sebesar 62,9% dan 4,8% di antaranya merupakan perokok pasif. Angka kematian tahunan akibat konsumsi tembakau mencapai 225.700 orang (Septian & Djamaruddin, 2022). Asap tembakau tidak hanya berbahaya bagi perokok aktif, tetapi juga bagi perokok pasif. Asap yang dihirup langsung oleh perokok disebut asap utama, sedangkan asap yang dihasilkan dari pembakaran ujung rokok disebut asap sampingan. Menariknya, asap sampingan mengandung konsentrasi zat berbahaya yang lebih tinggi dibandingkan asap utama (Nur Hakim, 2023).

Salah satu sumber utama polusi udara dalam ruangan adalah asap rokok, yang berdampak signifikan terhadap kesehatan penghuni rumah. Studi oleh Nuryati et al. (2023) menunjukkan bahwa merokok di dalam atau di luar rumah dapat meningkatkan konsentrasi partikel halus seperti PM2.5 dan PM10 di ruang tamu melebihi ambang batas aman. Bahkan, merokok di teras dengan pintu tertutup tetap meningkatkan konsentrasi PM2.5 dari 27  $\mu\text{g}/\text{m}^3$  menjadi 63  $\mu\text{g}/\text{m}^3$  dan PM10 dari 68  $\mu\text{g}/\text{m}^3$  menjadi 127  $\mu\text{g}/\text{m}^3$  di dalam ruangan (Nuryati., 2023). Kondisi ini dapat memicu berbagai masalah kesehatan, seperti asma, infeksi saluran pernapasan akut (ISPA), dan kanker paru-paru. Oleh karena itu, penting untuk meningkatkan kesadaran akan bahaya merokok di sekitar rumah dan mengambil langkah-langkah konkret untuk menjaga kualitas udara dalam ruangan.

Sekretariat KAPMI (Keluarga Pelajar dan Mahasiswa Indramayu) D.I. Yogyakarta merupakan tempat tinggal bagi mahasiswa dan pelajar asal

Indramayu yang sedang menempuh pendidikan di Yogyakarta. Mayoritas penghuni, yaitu sebesar 85%, merupakan perokok aktif. Kondisi tersebut berpotensi menurunkan kualitas udara dalam ruangan serta membahayakan kesehatan, khususnya bagi perokok pasif. Belum tersedia sistem yang mampu memantau kualitas udara secara otomatis di sekretariat ini. Akibatnya, penghuni tidak dapat mengetahui kondisi kualitas udara secara real-time.

Masalah tersebut memerlukan solusi berupa sistem pemantauan kualitas udara berbasis Internet of Things (IoT) yang mampu mendeteksi konsentrasi asap rokok secara real-time. Teknologi IoT memungkinkan perangkat fisik seperti sensor MQ-2 atau MQ-135 terhubung dan bertukar data melalui jaringan internet. Data yang diperoleh dari sensor dapat dikirim ke sistem pusat dan ditampilkan melalui aplikasi atau platform pemantauan. Penerapan sistem ini akan memungkinkan penghuni Sekretariat KAPMI D.I. Yogyakarta memantau kualitas udara secara langsung serta mengambil tindakan preventif guna menjaga kesehatan lingkungan.

## 1.2 Rumusan Masalah

Berdasarkan identifikasi dari permasalahan latar belakang maka rumusan masalahnya yaitu Bagaimana merancang dan mengembangkan sistem pemantauan asap rokok berbasis IoT?

## 1.3 Batasan Masalah

Adapun Batasan masalah yang didapat dari penelitian ini sebagai berikut:

- a. Sistem hanya mendeteksi asap rokok menggunakan sensor MQ-135 tanpa mengukur parameter kualitas udara lain seperti PM2.5 dan PM10.
- b. Pengujian dilakukan di Sekretariat KAPMI D.I. Yogyakarta, sehingga hasilnya tidak dapat digeneralisasi ke lokasi lain.
- c. Tindakan otomatis terbatas pada notifikasi aplikasi, aktivasi kipas ventilasi, dan alarm suara tanpa fitur pemurni udara.

- d. Fitur dan tampilan masih terbatas dibandingkan platform pengembangan lainnya.

#### **1.4 Tujuan Penelitian**

Berdasarkan rumusan masalah yang diangkat maka penelitian ini bertujuan: Merancang sistem monitoring asap rokok berbasis IoT menggunakan sensor MQ-135 yang dapat mendeteksi dan merespons keberadaan asap rokok secara *real-time* melalui aplikasi.

#### **1.5 Manfaat Penelitian**

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

- a. Memungkinkan pemantauan otomatis kualitas udara dalam ruangan dan meningkatkan kesadaran perokok pasif akan bahaya asap rokok.
- b. Memberikan data kualitas udara secara *real-time* dan memungkinkan tindakan otomatis untuk menjaga lingkungan lebih sehat, seperti menghidupkan kipas ventilasi.
- c. Memberikan wawasan tentang penerapan sensor MQ-135 dan ESP32 dalam pemantauan asap rokok.

#### **1.6 Sistematika Penulisan**

Berikut sistematika penulisan dalam laporan tugas akhir yang berjudul "**Perancangan Sistem Pemantauan dan Pengendalian Asap Rokok Berbasis *Internet of things* (IoT)**":

### **BAB I PENDAHULUAN**

Bab ini memberikan gambaran umum mengenai penelitian yang dilakukan terkait sistem pemantauan dan pengendalian asap rokok berbasis *Internet of things* (IoT). Bab ini mencakup latar belakang penelitian, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, serta sistematika penulisan.

## **BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI**

Bab ini berisi kajian pustaka yang mendukung penelitian, mencakup konsep dan teori yang relevan dengan sistem pemantauan asap rokok berbasis IoT. Beberapa aspek yang dibahas meliputi prinsip kerja sensor MQ-135, peran mikrokontroler ESP32, komunikasi data berbasis IoT, serta penggunaan MIT App Inventor sebagai platform pengembangan aplikasi monitoring. Selain itu, dibahas pula penelitian-penelitian terdahulu yang berkaitan dengan topik ini.

## **BAB III METODE PENELITIAN**

Bab ini membahas tentang metode dalam penelitian ini yaitu penerapan penelitian dari langkah-langkah sistematis mulai perencanaan, perancangan sistem, hingga pengujian. Selain itu, Dalam bab ini juga memberi daftar alat dan bahan yang digunakan sistem pemantauan dan pengendalian asap rokok, alat dan bahan ini terdiri dari perangkat keras dan perangkat lunak.

## **BAB IV ANALISIS DAN PERANCANGAN**

Bab ini menjelaskan proses perancangan dan penerapan sistem pemantauan asap rokok berbasis IoT. Pembahasan meliputi desain sistem, blok diagram yang menggambarkan hubungan antar komponen seperti sensor MQ-135, ESP32, aktuator (kipas atau alarm), serta integrasi dengan aplikasi MIT App Inventor. Selain itu, dijelaskan pula tahapan implementasi perangkat keras dan perangkat lunak hingga sistem dapat berfungsi sesuai tujuan.

## **BAB V HASIL DAN PEMBAHASAN**

Bab ini berisi hasil yang diperoleh dari implementasi sistem pemantauan dan pengendalian asap rokok. Analisis dilakukan terhadap data yang dikumpulkan dari sensor MQ-135, efektivitas sistem notifikasi dan tindakan otomatis, serta respons pengguna terhadap sistem yang diterapkan.

Selain itu, hasil pengujian dibandingkan dengan teori yang telah dibahas pada bab sebelumnya guna mengevaluasi keakuratan dan efisiensi sistem.

## **BAB VI PENUTUP**

Bab ini berisi kesimpulan yang diperoleh berdasarkan hasil penelitian dan pembahasan pada bab sebelumnya. Kesimpulan yang disajikan menjawab rumusan masalah yang telah dirumuskan di awal. Selain itu, diberikan pula rekomendasi untuk pengembangan lebih lanjut agar sistem menjadi lebih optimal dan dapat diterapkan secara lebih luas.

## **BAB II**

### **TINJAUAN PUSTAKA DAN LANDASAN TEORI**

#### **2.1 Tinjauan Pustaka**

Beberapa penelitian sebelumnya telah mengembangkan sistem pemantauan dan pengendalian asap rokok berbasis IoT. Salah satu penelitian yang relevan adalah penelitian yang dilakukan oleh (Nur Hakim, 2023) yang membahas prototipe sistem pemantauan asap rokok di ruangan yang menggunakan Wemos D1 R1 ESP8266. Melalui sensor MQ-135, sistem ini memiliki kemampuan untuk mendeteksi asap rokok dan memberikan peringatan terhadapnya, serta mengontrol kipas angin dan *buzzer* saat tingkat asap meningkat. Selain itu, platform IoT seperti Thingspeak dan Blynk menerima data sensor untuk pemantauan jarak jauh . Meskipun penelitian ini telah menghasilkan sistem pemantauan berbasis IoT, penggunaan Wemos D1 R1 ESP8266 memiliki keterbatasan dalam kecepatan pemrosesan dan jumlah pin I/O. Akibatnya, ini tidak ideal untuk pengolahan data yang lebih rumit. Selain itu, tanpa pengembangan aplikasi pemantauan yang lebih fleksibel, sistem ini terus bergantung pada platform *Internet of things* eksternal.

Selanjutnya, penelitian yang dilakukan oleh (Basuki & Efendi, 2022). membahas tentang sistem IoT yang menggunakan NodeMCU ESP8266 sebagai pusat kontrol untuk memantau dan mengurangi jumlah asap rokok yang dihasilkan. Selain itu, sistem ini memanfaatkan sensor DHT-11 untuk mengukur suhu dan kelembaban serta sensor MQ-2 untuk mendeteksi asap. Sementara kipas DC digunakan untuk ventilasi otomatis ruangan, hasil pemantauan ditampilkan melalui dot matrix P10 dan aplikasi Blynk. Jika dibandingkan dengan sensor MQ-135, sensor MQ-2 lebih sensitif terhadap zat kimia berbahaya yang ditemukan dalam asap rokok, meskipun sistem ini dapat mendeteksi asap dengan cukup baik. Selain itu, sistem peringatan yang digunakan hanya terbatas pada kipas dan tidak memiliki *buzzer* atau fitur peringatan dini lainnya. Akibatnya, untuk meningkatkan efektivitas

pemantauan, penelitian ini masih dapat dilakukan dengan menggunakan sensor MQ-135 yang lebih akurat serta menambahkan sistem peringatan *buzzer* dan aplikasi monitoring khusus.

Penelitian lain yang relevan dilakukan oleh (Hardika & Nurfiana, 2019), yang mengembangkan sistem monitoring asap rokok berbasis IoT untuk mendukung Gerakan Disiplin Kampus (GDK). Dengan menggunakan sensor MQ-135 dan Arduino Uno, sistem ini mendeteksi asap. Selain itu, melalui shield Arduino Ethernet, data dikirim ke server Thingspeak, sehingga smartphone dapat memantau asap secara *real-time*. Namun, Arduino Uno memiliki keterbatasan dalam kecepatan pemrosesan dan tidak memiliki konektivitas Wi-Fi bawaan, sehingga penggunaan modul tambahan seperti Ethernet Shield menjadi tidak praktis dibandingkan dengan ESP32, yang memiliki konektivitas Wi-Fi bawaan. Keuntungan dari penelitian ini adalah menggunakan koneksi internet untuk memperluas jangkauan pemantauan dibandingkan dengan sistem berbasis Wi-Fi yang memiliki jangkauan terbatas. Selain itu, sistem ini tidak memiliki fitur kontrol otomatis seperti kipas atau alarm. Fitur-fitur ini dapat meningkatkan kinerja sistem dan mengurangi efek asap rokok.

Selain itu, penelitian oleh (Rombang ., 2022). membahas tentang pengembangan prototipe detektor asap rokok yang dilengkapi dengan sistem purifier yang dirancang untuk menyaring polutan udara. Penelitian ini menggunakan sensor yang terdiri dari kombinasi MQ-135 dan MQ-2 untuk mendeteksi asap, mengukur kualitas udara, dan menguji kinerja alat dalam berbagai jenis polusi asap rokok. Meskipun sistem ini dapat mendeteksi dan menyaring asap rokok di dalam ruangan dengan baik, pengguna tidak dapat memantauanya karena penelitian ini tidak memiliki fitur IoT untuk memantau jarak jauh. Selain itu, dibandingkan dengan sistem notifikasi dan kontrol otomatis yang dapat mempercepat respons terhadap peningkatan kadar asap, penelitian ini lebih berfokus pada penyaringan asap. Akibatnya, sistem berbasis *Internet of things* dengan

pemantauan *real-time* mungkin merupakan solusi yang lebih efisien untuk memperluas jangkauan pemantauan kualitas udara di dalam ruangan.

Berikut adalah tabel perbandingan penjelasan hasil dari beberapa peneliti terdahulu:

Tabel 2. 1 Tinjauan pustaka

No	Peneliti (Tahun)	Judul	Metode	Hasil
1	Lukman Nur Hakim (2023)	Prototipe Sistem Pemantauan Asap Rokok di Ruangan Berbasis IoT dan Wemos D1 R1 ESP8266	Menggunakan sensor MQ135, DHT11, Flame, Wemos D1 R1 ESP8266, dengan platform Thingspeak dan Blynk. Data dikirimkan ke IoT, diintegrasikan dengan kipas angin dan <i>buzzer</i> sebagai respons terhadap asap rokok.	Prototipe berhasil mendeteksi asap rokok, memberikan peringatan, dan mengurangi asap di ruangan melalui kipas angin dan <i>buzzer</i> .
2	Arif Basuki, (2022)	Model Sistem IoT untuk Memantau dan Mengurangi Asap Rokok	Mikrokontroler NodeMCU ESP8266, sensor MQ-2, DHT-11, kipas DC, dot matrix P10, speaker, dan aplikasi Blynk digunakan untuk memonitor serta memberikan alarm dan ventilasi asap rokok.	Sistem mampu mendeteksi dan mengurangi asap rokok di area tertentu dengan mengaktifkan kipas dan memberikan informasi melalui alarm dan display dot matrix.
3	Deka Hardika & Nurfiana (2019)	Pengembangan Sistem Monitoring Asap Rokok Menggunakan	Menggunakan sensor MQ-135, Arduino Uno, dan Arduino Ethernet Shield untuk mengirim data ke	Sistem berhasil meningkatkan jangkauan monitoring melalui koneksi internet, sehingga

		Smartphone Berbasis IoT	Thingspeak. Data dapat diakses melalui aplikasi smartphone selama terhubung ke internet.	memungkinkan petugas untuk memantau secara lebih efektif di berbagai lokasi.
4	Rombang, I. A., Setyawan, L. B., & Dewantoro, G. (2022)	Perancangan Prototipe Alat Deteksi Asap Rokok dengan Sistem Purifier Menggunakan Sensor MQ-135 dan MQ-2	Menggunakan sensor MQ-135 dan MQ-2 untuk mendeteksi asap. Prototipe mencakup alat penyaring udara yang diuji pada berbagai kategori polusi asap untuk menganalisis kualitas udara yang disaring.	Alat berhasil mendeteksi dan menyaring polutan asap rokok dengan berbagai waktu penyaringan, meningkatkan kualitas udara di lingkungan dalam ruangan.

Penelitian ini mengusulkan beberapa perbaikan dan inovasi untuk sistem IoT untuk memantau dan mengontrol asap rokok. ESP32 akan berfungsi sebagai mikrokontroler utama untuk sistem ini, dan dibandingkan dengan ESP8266 dan Arduino Uno, ia lebih cepat dalam pemrosesan dan berkoneksi. Selain itu, penggunaan detektor MQ-135 akan menjadi lebih efisien untuk mendeteksi zat berbahaya dalam asap rokok. Selain itu, sistem ini akan memiliki *buzzer* untuk memberikan peringatan dini dan kipas otomatis untuk mengontrol kadar asap yang meningkat. Penelitian ini akan mengembangkan aplikasi pemantauan berbasis MIT App Inventor yang akan memungkinkan pengawasan data kualitas udara secara *real-time* melalui perangkat *mobile* untuk meningkatkan aksesibilitas pengguna. Penelitian ini diharapkan dapat memberikan solusi yang lebih efektif dan responsif untuk memantau dan mengendalikan asap rokok di lingkungan dalam ruangan dengan menggabungkan fitur-fitur ini.

## 2.2 Landasan Teori

### 2.2.1 Asap Rokok

Asap rokok merupakan hasil pembakaran tembakau yang mengandung lebih dari 7000 zat kimia, termasuk berbagai senyawa toksik yang berbahaya bagi kesehatan manusia dan lingkungan. Asap rokok terbagi menjadi beberapa jenis, yaitu *mainstream smoke* (MS) atau asap utama yang dihirup langsung oleh perokok, *sidestream smoke* (SS) atau asap sampingan yang berasal dari ujung rokok yang menyala, *secondhand smoke* (SHS) atau asap rokok pasif yang terpapar pada orang di sekitar perokok, serta *thirdhand smoke* (THS) yang merupakan residu dari asap rokok yang menempel pada permukaan benda dan dapat bertahan dalam waktu lama (Soleimani., 2022).

Tidak hanya perokok aktif yang terkena dampaknya, tetapi juga perokok pasif yang menghirup asap sampingan dan asap tangan kedua. Selain itu, residu asap tangan ketiga yang menempel pada permukaan benda dapat terus melepaskan zat berbahaya ke udara selama waktu yang lama. Hal ini menunjukkan bahwa asap rokok tidak hanya mengganggu kesehatan manusia tetapi juga menimbulkan ancaman bagi lingkungan karena sisa pembakaran rokok mencemari ekosistem. Oleh karena itu, pemantauan dan pengendalian asap rokok sangat penting untuk menjaga kesehatan masyarakat dan kualitas lingkungan.

### 2.2.2 Kategori Indeks Standar Pencemaran Udara

Berdasarkan peraturan Menteri Lingkungan Hidup dan Kehutanan Republik Indonesia NOMOR P.14/MENLHK/SETJEN/KUM.1/7/2020 tentang Indeks Standar Pencemar Udara(ISPU) pencemaran udara dapat dikategorikan sebagai berikut (Mentri Lingkungan Hidup, t.t.)

Tabel 2. 2 Kategori Indeks Standar Udara

Kategori	Angka Rentang	Keterangan
Baik	1-50	Tingkat kualitas udara yang sangat baik tidak berdampak negatif pada manusia, tumbuhan, atau hewan.
Sedang	51-100	Kesehatan manusia, tumbuhan, dan hewan masih aman dengan tingkat kualitas udara ini.
Tidak Sehat	101-100	Tingkat kualitas udara berbahaya bagi manusia, tumbuhan, dan hewan.
Sangat Tidak Sehat	201-300	Kualitas udara yang buruk dapat meningkatkan risiko kesehatan bagi beberapa populasi.
Berbahaya	301>	Kualitas udara yang buruk membahayakan kesehatan populasi dan memerlukan penanganan segera.

Sumber : (Mentri Lingkungan Hidup, t.t.)

### 2.2.3 *Internet of things*

*IoT* didefinisikan sebagai jaringan objek fisik yang dilengkapi dengan sensor, perangkat lunak, dan teknologi lainnya untuk menghubungkan dan bertukar data dengan perangkat dan sistem lain melalui internet. Konsep ini memungkinkan integrasi dunia fisik dengan sistem komputasi, menciptakan peluang baru untuk otomatisasi, efisiensi, dan pengambilan keputusan berdasarkan data (Kopetz & Steiner, 2022). Arsitektur *IoT* umumnya terdiri dari tiga lapisan utama yaitu:

1. *Device Layer*

Perangkat fisik seperti sensor, aktuator, dan perangkat embedded terletak di lapisan ini dan berfungsi untuk mengumpulkan data dari lingkungan sekitarnya.

2. *Network Layer*

Lapisan ini mengirimkan data yang dikumpulkan oleh perangkat ke sistem pengolahan data. Di bagian ini, teknologi komunikasi seperti Wi-Fi, Bluetooth, LoRa, dan 5G sering digunakan.

### 3. *Application Layer*

Lapisan ini menangani data dan menggunakannya untuk berbagai aplikasi, seperti rumah pintar, kota pintar, dan industri 4.0.

#### **2.2.4 *Firebase Realtime Database***

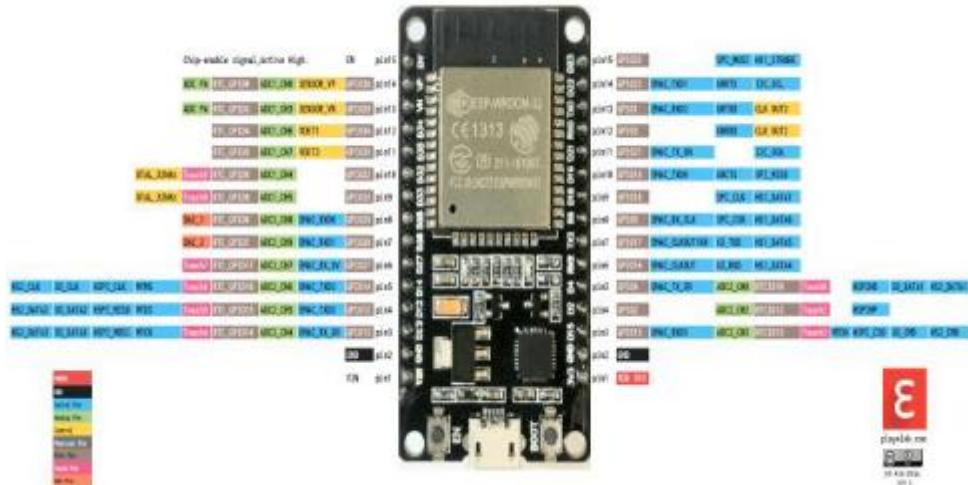
*Firebase Realtime Database* adalah sistem manajemen basis data berbasis *cloud* yang menggunakan model NoSQL untuk menyinkronkan dan menyimpan data secara *real-time*. Ini memungkinkan data diperbarui dan diakses oleh banyak pengguna sekaligus dengan latensi rendah. Ini sangat cocok untuk aplikasi yang membutuhkan sinkronisasi data yang cepat dan efisien(Trimbakrao Gaikwad Bharati Vidyapeeth. 2022). *Firebase Realtime Database* memiliki beberapa karakteristik utama, yaitu:

1. Berbasis *Cloud*: Data disimpan di server *Firebase* dan dapat diakses dari berbagai platform (Android, iOS, dan Web).
2. Sinkronisasi *Real-time*: Data diperbarui secara otomatis di semua klien tanpa perlu melakukan polling atau permintaan ulang ke server.
3. Mode Offline: *Firebase* menyediakan cache lokal sehingga aplikasi tetap bisa berfungsi meskipun tidak ada koneksi internet. Data akan disinkronkan kembali setelah koneksi tersedia.
4. Struktur Data JSON: *Firebase* menggunakan format JSON untuk menyimpan data dalam bentuk hierarki, yang memudahkan akses dan manipulasi data.

#### **2.2.5 Mikrokontroler ESP32**

ESP32 bekerja dengan mengatur mode operasionalnya, seperti mode download atau SPI Flash, saat memulai proses boot. Prosesor Xtensa 32-bit LX6 berinti tunggal atau ganda menjalankan instruksi dan memproses data dengan memanfaatkan memori internal dan eksternal. Di sisi lain, berbagai periferal, seperti UART, SPI, I2C, ADC, dan DAC, menangani komunikasi dan interaksi perangkat. Dengan kalibrasi internal, modul Wi-Fi dan Bluetooth mendukung konektivitas nirkabel untuk kinerja optimal. Selain itu, chip ini secara dinamis mengelola daya dengan beralih antara mode tidur aktif, mode tidur modem, mode tidur terang, dan mode tidur dalam untuk meningkatkan

efisiensi energi. Pada mode tidur dalam, chip ini dapat mengkonsumsi daya hingga 10  $\mu\text{A}$ . Koprosesor ULP memungkinkan operasi daya rendah untuk pemantauan sensor dan tugas RTC tanpa membangunkan prosesor utama, dan keamanan dijamin melalui boot aman dan enkripsi data menggunakan akselerator perangkat keras (Espressif, t.t.).



Gambar 2. 1 Mikrokontroler ESP32 (Muliadi dkk., 2024)

### 2.2.6 Sensor MQ135

Sensor MQ-135 berfungsi untuk mendeteksi kualitas udara dengan mendeteksi berbagai gas seperti amonia ( $\text{NH}_3$ ), nitrogen oksida ( $\text{NO}_x$ ), alkohol, benzena, asap, dan karbon dioksida ( $\text{CO}_2$ ). Alat ini bekerja dengan memanfaatkan lapisan sensitif berbahan timah dioksida ( $\text{SnO}_2$ ), yang resistansinya berubah sesuai dengan konsentrasi gas di sekitarnya. Ketika gas target muncul, senyawa gas bereaksi dengan oksigen di permukaan lapisan  $\text{SnO}_2$ , yang menyebabkan perubahan Untuk mengukur perubahan ini, rangkaian elektronik digunakan untuk menghitung konsentrasi gas yang ditemukan(olimex.com, t.t.).



Gambar 2. 2 Sensor MQ-135(Nur Hakim, 2023)

Sensor MQ-135 memiliki 4 pin, yang terdiri dari:

- Pin 1 = VCC(+5Volt)
- Pin 2 = Ground
- Pin 3 = Digital Out, dan
- Pin 4 = Analog out

### **2.2.7 MIT App Inventor**

MIT App Inventor adalah sebuah platform berbasis web yang dikembangkan oleh *Massachusetts Institute of Technology* (MIT) untuk memfasilitasi pembuatan aplikasi *mobile* dengan pendekatan visual berbasis blok. MIT App Inventor menggunakan sistem pemrograman berbasis blok yang mirip dengan Scratch, memungkinkan pengguna untuk menyusun logika pemrograman tanpa harus menulis kode secara manual (Kong & Abelson, 2019). MIT App Inventor memiliki dua bagian utama:

1. *Designer*: Editor visual yang memungkinkan pengguna untuk merancang tampilan aplikasi dengan menyeret dan melepaskan elemen UI (*User Interface*).
2. *Blocks Editor*: Editor berbasis blok yang digunakan untuk memprogram perilaku aplikasi. Blok ini terdiri dari *event handlers*, kondisi, variabel, dan operasi logis.

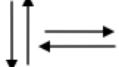
### **2.2.8 Flowchart**

*Flowchart* berfungsi sebagai alat penting yang digunakan saat merancang algoritma dan dalam dunia pemrograman. *Flowchart* itu sejenis gambar yang menunjukkan langkah-langkah untuk menyelesaikan masalah, dan biasanya pakai simbol-simbol yang sudah ditentukan. Menggunakan *Flowchart* terutama bertujuan untuk menunjukkan langkah-langkah program dengan cara yang jelas dan teratur, sehingga lebih mudah dipahami dan diimplementasikan(Khesya, 2021).

Berikut tabel simbol dan keterangan fungsi *Flowchart*:

Tabel 2. 3 Simbol dan Keterangan *Flowchart*

No.	Simbol	Nama	Keterangan
1.		“Terminal”	Awal atau akhir proses.
2.		“Data”	Proses <i>input</i> atau <i>output</i> .
3.		“Process”	Proses kerja komputer.
4.		“Decision”	Menunjukkan suatu kondisi tertentu yang mengarah pada dua kemungkinan ya/tidak.
5.		“Connector”	Penghubung antar proses lain di halaman yang sama.
6.		“Off page connector”	Penghubung antar proses lain di halaman yang berbeda.
7.		“Preparation”	Mewakili ketentuan penyimpanan untuk diproses untuk memberikan awal harga.
8.		“Card”	<i>Input</i> dan <i>output</i> berasal dari kartu atau ditulis ke kartu.
9.		“Punched Tape”	-
10.		“Document”	Mencetak <i>output</i> dalam format dokumen

No.	Simbol	Nama	Keterangan
11.		“Flow”	Menentukan jalannya alur Proses.

Sumber: (Khesya, 2021)

### 2.2.9 Blackbox

Pengujian *black-box* merupakan metode evaluasi perangkat lunak yang berorientasi pada hasil. Caranya adalah dengan memberikan masukan tertentu (data uji) dan memverifikasi apakah keluaran yang dihasilkan sesuai dengan fungsionalitas yang diharapkan dan telah ditetapkan dalam persyaratan. Pendekatan ini menitikberatkan pada pengujian antarmuka pengguna (tampilan) dan aspek fungsional aplikasi, termasuk memastikan alur kerja setiap fungsi telah selaras dengan kebutuhan pengguna. Penting untuk dicatat bahwa metode ini tidak melibatkan pemeriksaan kode sumber program (Priyaungga dkk., 2020).

Pelaksanaan pengujian *black-box* melibatkan beberapa tahapan. Tahap awal adalah perancangan kasus uji (*test case*) yang bertujuan untuk menguji setiap fungsi yang ada dalam aplikasi. Selanjutnya, kasus uji juga dibuat untuk memvalidasi kesesuaian alur kerja atau urutan operasional suatu fungsi program dengan kebutuhan dan permintaan spesifik dari pengguna. Tahap terakhir adalah identifikasi *bug* atau *error* yang terdeteksi melalui pengamatan pada antarmuka (tampilan) aplikasi (Priyaungga dkk., 2020).

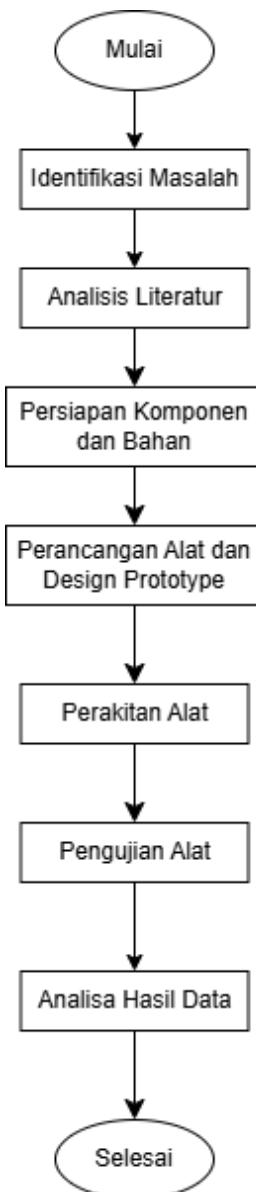
Dalam proses pengujian, pemilihan teknik yang tepat sangat krusial. Teknik yang efektif akan mampu menemukan kesalahan-kesalahan yang sebelumnya tidak terdeteksi, sehingga berkontribusi pada peningkatan kualitas perangkat lunak secara keseluruhan. Penelitian ini secara spesifik menerapkan metode *Equivalence Partitions*. Metode ini merupakan teknik pengujian yang didasarkan pada analisis masukan data pada setiap formulir. Prinsip kerja *Equivalence Partitions* adalah dengan membagi domain masukan program ke dalam beberapa kelas data, yang kemudian dikelompokkan menjadi kategori data yang valid dan tidak valid untuk pengujian (Priyaungga dkk., 2020).

## BAB III

### METODE PENELITIAN

#### 3.1 Metode Penelitian

Diagram alir berikut menunjukkan proses metode penelitian yang dilakukan dalam penelitian ini:



Gambar 3. 1 Langkah Penelitian

Berdasarkan gambar 3.1 Langkah Penelitian, berikut penjelasan tentang langkah-langkah jalan penelitian:

## 1. Mulai

Penelitian diawali dengan maksud mengembangkan Rancang Bangun Sistem Pemantauan Asap Berbasis *Internet of things* (IoT) di Aula KAPMI Yogyakarta memerlukan solusi untuk menjaga kualitas udara di ruang umum.

## 2. Identifikasi masalah

Pada tahap ini, masalah utama yang dihadapi di Aula KAPMI D.I. Yogyakarta telah diidentifikasi. Salah satunya adalah asap rokok yang mencemari udara ruangan tertutup, yang mengganggu kenyamanan dan kesehatan orang yang menggunakan aula. Masalah ini menjadi dasar dari solusi teknologi.

## 3. Analisis literatur

Setelah masalah ditemukan, penelitian pustaka dilakukan untuk mengeksplorasi teknologi pendukung. Fokus penelitian adalah mendapatkan pemahaman yang lebih baik tentang cara kerja sensor MQ135 dan modul ESP32, serta aplikasi *Internet of things* (IoT) yang digunakan untuk pemantauan dan kontrol otomatis. Selain itu, penelitian sebelumnya yang relevan digunakan sebagai referensi.

## 4. Persiapan komponen dan bahan

Sensor MQ135, ESP32, *buzzer*, dan modul pendukung lainnya telah disiapkan. Ini adalah langkah pertama untuk memastikan bahwa semua komponen yang diperlukan untuk sistem di Aula KAPMI D.I. Yogyakarta tersedia dan siap untuk dipasang.

## 5. Perancangan alat dan design prototipe

Di Aula KAPMI D.I. Yogyakarta, sistem pemantauan berbasis IoT dirancang untuk memenuhi persyaratan. Prototipe ini dimaksudkan untuk mendeteksi asap rokok secara *real-time* dan mengeluarkan peringatan melalui *buzzer* jika tingkat asap melewati ambang batas.

## 6. Perakitan alat

Semua komponen digabungkan menjadi satu sistem yang lengkap, yang dimaksudkan untuk melacak udara di aula dan terhubung ke jaringan *Internet of things* untuk mengakses data secara online.

## 7. Pengujian alat

Untuk memastikan bahwa fungsi sistem sesuai dengan tujuan penelitian, sistem diuji di Aula KAPMI D.I. Yogyakarta, menggunakan metode pengujian Blackbox.

## 8. Analisa hasil data

Tingkat sensitivitas deteksi asap dan keefektifan kontrol *buzzer* adalah beberapa contoh data yang dianalisis dari pengujian. Hasilnya digunakan untuk menilai sistem pemantauan asap rokok di aula.

## 9. Selesai

Penelitian ini diakhiri dengan membuat laporan hasil penelitian.

## 3.2 Alat

Daftar alat yang digunakan pada penelitian ditunjukan pada tabel 3.1

Tabel 3. 1 Alat Penelitian

No	Nama Alat	Spesifikasi
1	Laptop	Aspire A314-36M Intel(R) Core(TM) i3-N305 (8 CPUs) 8GB LPDDR5 512GB PCIe NVMe SSD Windows 11 Home Single Language 64-bit
2	Smartphone	Redmi Note 8 Ram 6/128 Chipset Qualcomm Snapdragon 665 Baterai 4000 mAh

### 3.3 Bahan

Daftar bahan yang digunakan pada penelitian ditunjukan pada tabel 3.2

Tabel 3. 2 Daftar bahan

No	Nama	Spesifikasi
1.	ESP32	Prosesor: Dual-core Xtensa LX6, hingga 240 MHz. Komunikasi: Wi-Fi (802.11 b/g/n, 2.4 GHz) dan Bluetooth v4.2 (BLE dan BR/EDR). GPIO: Hingga 34 pin. ADC: Resolusi 12-bit, hingga 18 saluran. DAC: Resolusi 8-bit, 2 saluran. Protokol I/O: Mendukung UART, SPI, I2C, I2S, SDIO, PWM, dan CAN. Kapasitas Memori: 520 KB SRAM dan Flash 4 MB. Hemat Daya: Mode deep sleep (<10 $\mu$ A).
2.	MQ-135	Tipe MQ-135 Rentang Pengukuran 10-1000 ppm Ketepatan $\pm 5\%$ Tegangan Operasi 5V Arus Operasi 150-100 mA Suhu Operasi -20°C hingga 50°C Kelembaban perasi 30-80% RH Dimensi: sekitar 35 x 35 x 22 mm
3.	Buzzer	Tegangan Operasi: 3-12V (DC) Arus Operasi: 10-50 mA Frekuensi Suara: 1-4 kHz Kekuatan Suara: 70-90 dB Waktu Respons: Instan
4.	Kabel Jumper	Tegangan Maksimum: 300V Arus Maksimum: 2A Resistansi: Rendah ( $< 1\Omega$ ) Temperatur Operasi: -20°C hingga 80°C Ketahanan: Tahan terhadap korosi dan oksidasi Panjang: Variatif (10-20 Cm)
5.	Breadboard	Jumlah Titik: 830, 400, 170, atau lainnya Jarak Antartitik: 2,54 mm (1 inci) Temperatur Operasi: -20°C hingga 80°C Tegangan Maksimum: 300V
6.	Relay	Tegangan Coil: 5V DC Tipe Kontak: SPDT (Single Pole Double Throw)

No	Nama	Spesifikasi
		<p>Kapasitas Kontak: 10A 250VAC / 10A 30VDC</p> <p>Tegangan Aktifasi: 4.5V (typical), maksimal 5V</p> <p>Resistansi Coil: Sekitar 70 ohm</p> <p>Waktu Aktifasi: &lt;10 ms</p> <p>Waktu Pelepasan: &lt;5 ms</p> <p>Ukuran Modul: sekitar 5 x 2 x 1.8 cm</p> <p>Indikator LED: Ya</p> <p>Isolasi Optocoupler: Tersedia pada beberapa modul</p>
7.	DC Fan	<p>Tegangan Operasi: 12V DC Arus: <math>\pm 0.1A - 0.25A</math> (tergantung ukuran dan merk)</p> <p>Ukuran Umum: 40x40 mm, 60x60 mm, 80x80 mm, 120x120 mm</p> <p>Jenis Konektor: 2-pin (umum)</p> <p>Kecepatan: 1000 – 4000 RPM (tergantung ukuran)</p> <p>Aliran Udara: 10 – 60 CFM</p> <p>Tingkat Kebisingan: 20 – 35 dBA</p>
8.	DC Barrel Jack Female	<p>Tipe: DC Barrel Jack Female</p> <p>Diameter Lubang Dalam: 2.1 mm</p> <p>Diameter Luar: 5.5 mm</p> <p>Tegangan Maksimum: hingga 30V</p> <p>Arus Maksimum: 2A – 5A (tergantung material dan build quality)</p> <p>Jenis Pemasangan: PCB Mount atau dengan kabel solder</p>
9.	Adapter 12V	<p>Tegangan Output: 12V DC</p> <p>Arus Output: Tersedia dalam berbagai varian (1A, 2A, 5A)</p> <p>Tegangan Input: 100 – 240V AC, 50/60Hz</p> <p>Jenis Plug: Barrel Jack 2.1 mm</p> <p>Proteksi: Over-voltage, over-current, short circuit</p> <p>Efisiensi: &gt;80%</p> <p>Panjang Kabel: <math>\pm 1</math> meter</p>

## **BAB IV**

### **ANALISIS DAN PERANCANGAN**

#### **4.1 Perancangan dan Analisis Sistem**

Pada tahap ini dilakukan proses perancangan alat sebagai dasar dalam pengembangan sistem yang akan dibuat. Perancangan dilakukan untuk menentukan kebutuhan perangkat keras (hardware) dan hubungan antar komponen agar sistem dapat bekerja sesuai dengan tujuan. Proses ini mencakup identifikasi komponen, penyusunan skema rangkaian, serta perencanaan integrasi antara sensor, mikrokontroler, aktuator, dan media komunikasi data.

##### **4.1.1 Analisis Kebutuhan Fungsional**

Fungsi-fungsi esensial yang perlu dimiliki sistem agar beroperasi sesuai tujuan yang direncanakan dijelaskan dalam kebutuhan fungsional. Sistem deteksi asap rokok ini memiliki tiga komponen penting:

###### *1. Input*

Sensor tersebut bertugas mengidentifikasi asap rokok. Mikrokontroler ESP32 kemudian membaca data dari sensor ini dan mengubahnya ke dalam bentuk digital. Masukan sensor inilah yang menjadi acuan utama sistem dalam menilai kondisi kualitas udara ruangan secara berkelanjutan. Proses

###### *2. Proses*

Proses inti sistem dijalankan oleh mikrokontroler ESP32, yang secara teratur membaca data dari sensor MQ-135. Data ini lalu dibandingkan dengan standar kualitas udara yang telah ditetapkan. Jika pembacaan sensor melampaui standar tersebut, sistem akan secara otomatis mengaktifkan *buzzer* sebagai alarm suara dan menyalaikan kipas untuk membantu sirkulasi udara. Bersamaan dengan itu, data sensor juga dikirim langsung ke *Firebase* untuk disimpan dan dapat dipantau melalui aplikasi.

### 3. Output

Keluaran sistem meliputi notifikasi kepada pengguna melalui aplikasi monitoring berbasis MIT App Inventor, terutama jika terdeteksi kadar gas berbahaya yang memerlukan peringatan segera. Selain pemberitahuan, aktivasi otomatis kipas dan *buzzer* juga merupakan output responsif sistem terhadap kondisi lingkungan. Seluruh data yang terkumpul ditampilkan secara *real-time* dalam aplikasi dan juga tersimpan di *Firebase* sebagai data historis untuk pemantauan berkelanjutan.

#### 4.1.2 Analisis Kebutuhan Non-Fungsional

Kebutuhan non-fungsional adalah istilah yang digunakan untuk menggambarkan spesifikasi teknis dari software dan hardware yang digunakan dalam proses pengembangan sistem. Kebutuhan ini tidak mempengaruhi proses utama secara langsung, tetapi sangat penting untuk menjaga sistem berjalan dengan baik dan stabil.

Kebutuhan non-fungsional yang digunakan pada penilitian ini yaitu:

a. Hardware

1). Laptop

Laptop minimal memiliki prosesor Intel Core i3 generasi ke-8 atau AMD Ryzen 3, RAM sebesar 4 GB, dan penyimpanan internal minimal 128 GB, baik SSD maupun HDD, untuk menjalankan tugas tersebut dengan baik. Windows 10 atau Linux Ubuntu 20.04 adalah sistem operasi yang disarankan. Selain itu, laptop harus memiliki setidaknya satu port USB untuk dapat terhubung ke ESP32, serta konektivitas Wi-Fi untuk mendukung komunikasi online dengan *Firebase*.

b. Software

1). Arduino IDE

Software pemrograman yang digunakan untuk menulis, menguji, dan mengunggah kode ke mikrokontroler ESP32.

## 2). MIT App Inventor

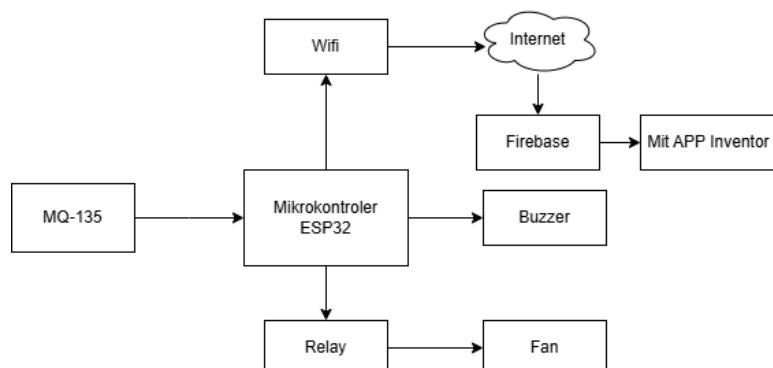
Platform berbasis web untuk membuat aplikasi Android secara visual menggunakan metode drag-and-drop. Digunakan untuk membangun antarmuka pengguna (UI) aplikasi monitoring asap rokok yang terhubung ke *Firebase*.

## 3). *Firebase*

Layanan *cloud* dari Google yang digunakan sebagai *database realtime*. *Firebase* berfungsi untuk menyimpan dan menampilkan data pembacaan sensor dari ESP32 ke aplikasi Android secara langsung dan terintegrasi.

### 4.1.3 Struktur Sistem

Blok diagram berikut menunjukkan arsitektur sistem dan alur data yang digunakan:



Gambar 4. 1 Diagram Blok Perencanaan Sistem

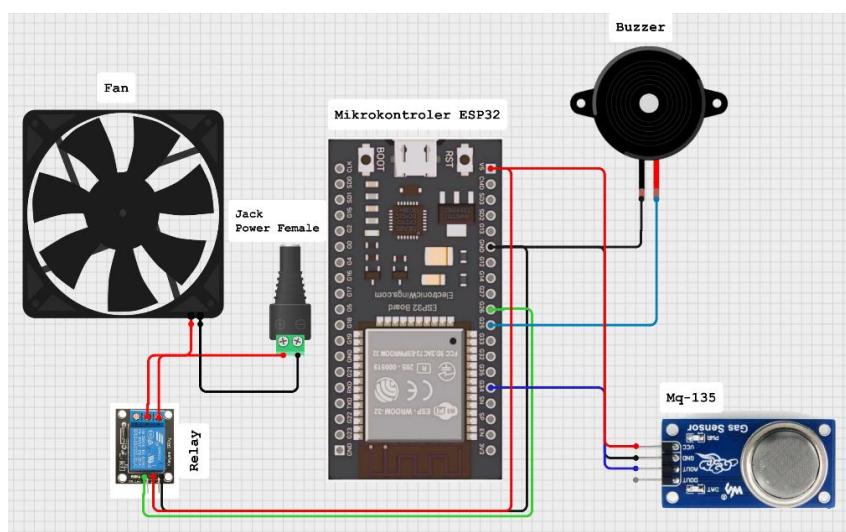
Sistem pemantauan dan pengendalian kualitas udara yang terintegrasi berbasis *Internet of things* (IoT) digambarkan di blok diagram ini. Ini dimulai dengan sensor MQ-135, yang berfungsi untuk mengidentifikasi asap, CO<sub>2</sub>, dan NH<sub>3</sub>, antara lain, dalam lingkungan. Selanjutnya, sensor mengirimkan data ke mikrokontroler ESP32, yang berfungsi sebagai pusat pengolah data. ESP32 mengamati batasan dan mengambil keputusan berdasarkan data tersebut. ESP32 akan mengaktifkan dua fungsi utama jika kadar polutan melebihi ambang aman. Pertama, itu akan menyalakan *Buzzer* sebagai peringatan suara,

dan kedua, akan mengontrol *Relay* untuk menghidupkan *fan* (kipas) untuk meningkatkan sirkulasi udara dan mengurangi konsentrasi polutan.

Selain itu, modul *Wi-Fi* memungkinkan ESP32 menghubungkan ke internet untuk mengirimkan data sensor ke platform *Firebase* secara langsung. *Firebase* berfungsi sebagai penyimpanan *cloud* yang memungkinkan aplikasi *mobile* yang dikembangkan dengan MIT App Inventor mengakses data. Aplikasi ini memungkinkan pengguna untuk melihat riwayat data dan melihat kondisi udara secara langsung. Oleh karena itu, sistem ini adalah solusi lengkap untuk manajemen kualitas udara di lingkungan rumah, industri, atau publik. Ini memiliki otomatisasi berbasis sensor, respons fisik melalui aktuator, dan pemantauan jarak jauh melalui aplikasi.

#### 4.1.4 Perancangan Rangkaian

Sebelum sistem ini direalisasikan secara fisik, terlebih dahulu dilakukan tahap perancangan alat guna menggambarkan susunan komponen dan hubungan antar perangkat yang digunakan. Perancangan ini bertujuan untuk memberikan gambaran menyeluruh mengenai alur kerja sistem serta memastikan integrasi komponen berjalan dengan baik. Adapun rancangan alat yang telah disusun dapat dilihat pada gambar berikut.



Gambar 4. 2 Schematic rangkaian

Tabel berikut menunjukkan hubungan pin yang digunakan pada mikrokontroler ESP32 dengan masing-masing komponen:

Tabel 4. 1 Pin ESP 32 dan Mq-135

ESP 32	Mq-135
34	Data
5V	VCC
GND	GND

Tabel 4. 2 Pin ESP 32 dan *Relay*

ESP 32	<i>Relay</i>
25	Data
5V	VCC
GND	GND

Tabel 4. 3 Pin ESP 32 dan *Buzzer*

ESP 32	<i>Buzzer</i>
GND	-
26	+

Tabel 4. 4 Pin *Relay*, Adapter, dan *Fan*

<i>Relay</i>	Adapter( <i>Jack Power Female</i> )	<i>Fan</i>
NO	+12v	-
COM	-	+
NC	Tidak digunakan	Tidak digunakan

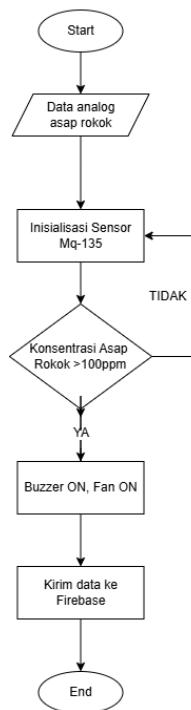
Berdasarkan Gambar 4.2 dan Tabel 4.1 sampai 4.4, terlihat bahwa setiap komponen telah terhubung sesuai dengan fungsi dan kebutuhan sistem. Sensor MQ-135 dihubungkan ke ESP32 untuk mendeteksi keberadaan gas atau asap rokok, sedangkan modul *relay* digunakan sebagai saklar elektronik untuk mengontrol kipas DC secara otomatis. *Buzzer* juga terhubung ke ESP32 sebagai indikator suara ketika asap terdeteksi. Jalur koneksi dari *relay* ke adaptori dan kipas angin diatur sedemikian rupa sehingga kipas angin hanya aktif ketika sensor mendeteksi ambang batas asap tertentu dan sinyal logika diberikan oleh mikrokontroler. Semua koneksi ini bertujuan untuk memastikan bahwa sistem dapat bekerja secara terintegrasi, mulai dari pendekstian, peringatan, hingga penanganan asap melalui kipas.

## 4.2 Perancangan Perangkat Lunak

Setelah tahap perancangan perangkat keras selesai, langkah selanjutnya adalah merancang perangkat lunak yang berfungsi sebagai pengendali utama dalam sistem. Perangkat lunak dirancang untuk membaca data dari sensor MQ-135, mengolah data, dan mengaktifkan aktuator seperti *buzzer* dan kipas melalui *relay* berdasarkan kondisi yang terdeteksi. Selain itu, sistem juga dilengkapi dengan fitur monitoring berbasis *Internet of things* (IoT) menggunakan platform *Firebase* yang terintegrasi dengan aplikasi *mobile* yang dirancang melalui MIT App Inventor. Perancangan perangkat lunak ini meliputi pembuatan diagram alir logika sistem, antarmuka aplikasi, dan penulisan kode program pada mikrokontroler ESP32.

### 4.2.1 Flowchart Sistem

Berikut ditampilkan dua *Flowchart* yang menggambarkan alur kerja sistem. *Flowchart* pertama menunjukkan proses kerja alat secara keseluruhan, sedangkan *Flowchart* kedua menjelaskan alur komunikasi data antara sistem, *Firebase*, dan aplikasi *mobile*.



Gambar 4. 3 *Flowchart* Sistem Alat

*Flowchart* pada Gambar 4.3 menunjukkan alur kerja sistem perangkat keras secara keseluruhan. Proses diawali dengan bagaimana mikrokontroler ESP32 menginisialisasi terhadap sensor MQ-135. Sistem tersebut akan melakukan pembacaan konsentrasi asap rokok di lingkungan sekitar setelah sensor tersebut siap. Sistem tersebut akan memberikan sebuah respon dengan mengaktifkan *buzzer* sebagai alarm peringatan dan juga menyalakan kipas melalui modul *relay* untuk membantu mengurangi kadar asap tersebut, apabila konsentrasi asap yang terdeteksi itu melebihi ambang batas 100 ppm. Selanjutnya, data akan dikirim guna disimpan di *Firebase* serta ditampilkan melalui aplikasi monitoring. Namun, sistem itu akan terus menerus melakukan pembacaan ulang sensor secara berulang untuk memantau kondisi tersebut secara *real-time*. Konsentrasi asap yang terdeteksi tersebut harus di atas ambang batas tertentu agar sistem berhenti.

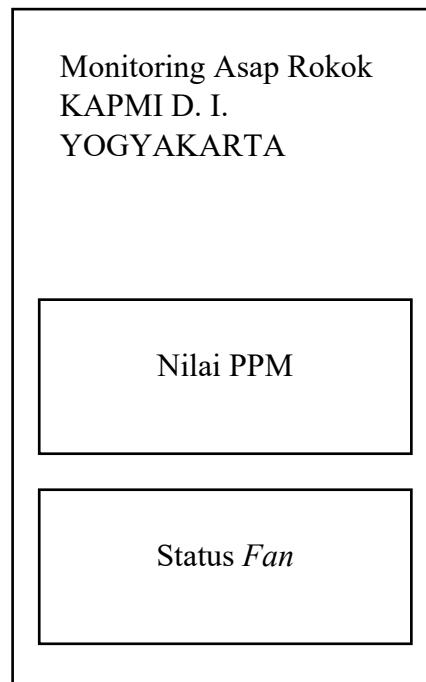


Gambar 4. 4 *Flowchart* Sistem *Firebase* dan Aplikasi

*Flowchart* pada Gambar 4.4 menggambarkan alur komunikasi data antara mikrokontroler serta aplikasi *mobile database Firebase*. Proses dimulai dengan mikrokontroler ESP32 membaca data dari sensor dan mengirimkannya ke *Firebase* secara *real-time*. Setelah data tersimpan di *Firebase*, layanan *cloud* akan mengirimkan informasi ke aplikasi *mobile* yang telah terhubung. Selanjutnya, aplikasi yang dibuat menggunakan MIT App Inventor akan menunjukkan kepada pengguna data sensor dan status kipas (*fan*). Mekanisme ini memungkinkan pengguna untuk memantau aktivitas sistem dan kondisi lingkungan secara jarak jauh melalui perangkat *mobile*.

#### 4.2.2 Desain Antar Muka Aplikasi

Sebuah antarmuka aplikasi berbasis MIT App Inventor dibuat untuk membantu pengguna memantau kondisi kualitas udara dan status perangkat. Antarmuka ini dirancang untuk menjadi mudah digunakan dan memungkinkan untuk menampilkan data sensor secara *real-time* yang terhubung ke platform *Firebase*.



Gambar 4. 5 Desain Antarmuka Aplikasi

Desain antarmuka aplikasi ditampilkan pada Gambar 4.5. Pada bagian atas antarmuka terdapat judul aplikasi “*Monitoring Asap Rokok*” beserta informasi identitas pengguna, yaitu KAPMI D. I. Yogyakarta.

Aplikasi menampilkan dua panel utama yang berfungsi sebagai tampilan data utama. Panel pertama yang menunjukkan nilai kadar asap rokok dalam satuan PPM (Part Per Million), dimana data ini diperoleh secara *real-time* dari sensor MQ-135 melalui koneksi *Firebase*. Panel kedua adalah yang menampilkan kondisi kipas (*fan*) apakah dalam keadaan menyala (ON) atau mati (OFF), tergantung dari hasil pengukuran konsentrasi asap yang diterima sistem.

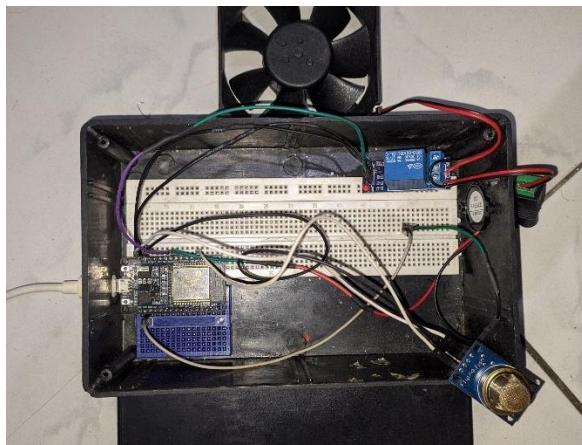
## BAB V

### HASIL DAN PEMBAHASAN

#### 5.1 Implementasi Sistem

##### 5.1.1 Implementasi Perangkat Keras

Visualisasi dari implementasi rangkaian perangkat keras sistem monitoring asap rokok ditunjukkan pada Gambar 5.1.



Gambar 5. 1 Implementasi Rangkaian

Implementasi hardware sistem monitoring asap rokok ini adalah bagian dari perancangan keseluruhan perangkat keras yang digunakan untuk mendukung operasi sistem. Gambar 5.1 menunjukkan cara perangkat keras yang digunakan dalam sistem yang dirancang untuk memantau keberadaan asap rokok di lingkungan sekitar. Sensor MQ-135 disimpan di dalam kotak media berwarna hitam, yang melindungi seluruh perangkat.

Mikrokontroler ESP32 berfungsi sebagai pusat sistem dan membaca data dari sensor MQ-135 melalui kabel jumper yang terletak di atas breadboard. Sistem juga memiliki modul *relay* 1 channel yang dikendalikan oleh ESP32, yang mengatur apakah kipas DC sebagai alat pengendali asap aktif atau tidak.

Rangkaian ini memiliki desain yang sederhana namun efektif, dengan sensor yang ditempatkan secara strategis untuk mendeteksi asap. Ketika konsentrasi asap melebihi ambang batas tertentu, ESP32 akan mengaktifkan *relay* untuk secara otomatis menyalaikan kipas DC, memungkinkan pengurangan asap segera.

Sebelum data dikirimkan ke *Firebase* untuk ditampilkan pada aplikasi Android berbasis MIT App Inventor, perangkat keras ini berfungsi sebagai unit pengendali dan pendekripsi utama. Integrasi ini memungkinkan pengguna untuk memantau kondisi kualitas udara secara *real-time* melalui perangkat seluler mereka. Implementasi ini menunjukkan bahwa desain sistem dan kebutuhan fungsional untuk pemantauan dan pengendalian asap rokok secara otomatis telah dipenuhi dengan baik.

#### 5.1.1.1 Kode Program Sistem

Pada bagian awal program, dilakukan inisialisasi terhadap library yang digunakan, yaitu *FirebaseESP32.h* untuk koneksi *Firebase* dan *WiFi.h* untuk koneksi internet. Selanjutnya, informasi jaringan WiFi dan kredensial *Firebase* didefinisikan agar ESP32 dapat mengirim data sensor ke *database* secara *real-time*.

```
#incloude <FirebaseESP32.h>
#incloude <WiFi.h>

const char* ssid = "eril1234";
const char* password = "eril1234";

#define FIREBASE_HOST "https://monitoringasaprokok-default-
rtdb.firebaseio.com/"

#define FIREBASE_AUTH "sa0RC8H0GKilpkh2n4S8slg2ulmjh8dpXMUfoszD"

FirebaseData firebaseData;
FirebaseAuth auth;
FirebaseConfig config;
```

Gambar 5. 2 Inisialisasi Koneksi WiFi dan *Firebase*

Bagian ini merupakan fondasi dari komunikasi data sistem. Dengan informasi tersebut, perangkat akan terhubung ke *Firebase Realtime Database* untuk mengirim dan menampilkan data sensor ke aplikasi.

Deklarasi pin digunakan untuk mendefinisikan koneksi fisik antara ESP32 dengan sensor maupun aktuator seperti *buzzer* dan *relay*. Selain itu, ditentukan juga variabel-variabel penting seperti sensitivitas dan ambang batas asap.

```
const int mq135_pin = 34; // Pin ADC untuk MQ135
const int buzzerPin = 25; // Pin untuk buzzer
const int relayPin = 26; // Pin untuk relay

// Variabel sensor dan ambang batas
int sensitivity = 500;
int air_quality = 0;
const int mqThreshold = 100;
```

Gambar 5. 3 Deklarasi Pin dan Variabel Sistem

Variabel *air\_quality* menyimpan hasil pembacaan sensor yang dikalibrasi menggunakan faktor sensitivitas, sedangkan *mqThreshold* merupakan batas ambang untuk menentukan apakah kualitas udara tergolong berbahaya.

Fungsi *readMQ()* digunakan untuk membaca nilai dari sensor MQ-135 yang kemudian diolah menjadi nilai kualitas udara dalam satuan PPM (Parts Per Million). Nilai ini akan menjadi acuan sistem untuk mengambil keputusan.

```
void readMQ() {
    int sensorVal = analogRead(mq135_pin);
    air_quality = sensorVal * sensitivity / 4096;
}
```

Gambar 5. 4 Fungsi Pembacaan Sensor

Perhitungan menggunakan faktor *sensitivity* dan skala ADC (0–4095) dari ESP32 untuk mendapatkan estimasi konsentrasi gas di udara.

Bagian *setup()* dijalankan sekali saat sistem pertama kali menyala. Pada bagian ini, ESP32 menghubungkan diri ke jaringan WiFi dan menginisialisasi

koneksi dengan *Firebase*. Selain itu, dilakukan juga pengaturan mode pin dan kondisi awal aktuator.

```
void setup_wifi() {  
    Serial.print("Menghubungkan ke WiFi: ");  
    Serial.println(ssid);  
    WiFi.begin(ssid, password);  
    while (WiFi.status() != WL_CONNECTED) {  
        delay(500);  
        Serial.print(".");  
    }  
    Serial.println("\nWiFi terhubung!");  
    Serial.print("IP Address: ");  
    Serial.println(WiFi.localIP());  
}  
  
void setup() {  
    Serial.begin(115100);  
    setup_wifi();  
  
    config.host = FIREBASE_HOST;  
    config.signer.tokens.legacy_token = FIREBASE_AUTH;  
    Firebase.begin(&config, &auth);  
    Firebase.reconnectWiFi(true);  
  
    pinMode(mq135_pin, INPUT);  
    pinMode(buzzerPin, OUTPUT);  
    pinMode(relayPin, OUTPUT);  
}
```

Gambar 5. 5 Setup Koneksi dan Konfigurasi Sistem

Fungsi `setup_wifi()` bertugas menghubungkan ESP32 ke jaringan. Setelah berhasil terhubung, konfigurasi *Firebase* dilakukan menggunakan objek `config` dan `auth`.

Pada fungsi `loop()`, sistem secara terus-menerus membaca data dari sensor, menampilkannya di serial monitor, mengirimkan data ke *Firebase*, dan mengambil keputusan berdasarkan nilai ambang batas. Jika kualitas udara buruk, maka *relay* diaktifkan untuk menyalakan kipas, dan *buzzer* akan berbunyi sebagai peringatan.

```
void loop() {
    readMQ0();

    Serial.print("Air Quality (MQ135): ");
    Serial.print(air_quality);
    Serial.println(" PPM");

    if (Firebase.setInt(firebaseData, "/monitoring/air_quality", air_quality)) {
        Serial.println("Data terkirim ke Firebase.");
    } else {
        Serial.print("Gagal mengirim data: ");
        Serial.println(firebaseData.errorReason());
    }

    if (air_quality > mqThreshold) {
        Serial.println("Peringatan: Kadar Asap Berbahaya (> 100 PPM)!");
        digitalWrite(relayPin, HIGH); // Nyalakan relay
        Firebase.setString(firebaseData, "/monitoring/relay_status", "ON");
    }
}
```

Gambar 5. 6 Proses Utama dan Logika Pengambilan Keputusan

```
for (int i = 0; i < 5; i++) {  
    digitalWrite(buzzerPin, HIGH);  
    delay(100);  
    digitalWrite(buzzerPin, LOW);  
    delay(100);  
}  
} else {  
    Serial.println("Kadar Asap Normal.");  
    digitalWrite(relayPin, LOW); // Matikan relay  
    Firebase.setString(firebaseData, "/monitoring/relay_status", "OFF");  
  
    digitalWrite(buzzerPin, LOW);  
}  
delay(1000);  
}
```

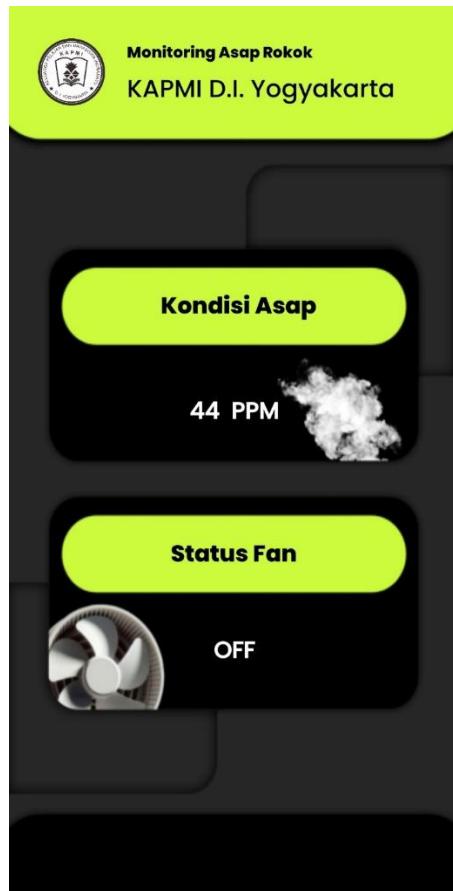
Gambar 5. 7 Lanjutan Kode Loop

Data dikirim ke dua path dalam *Firebase*, yaitu *air\_quality* untuk nilai sensor, dan *relay\_status* untuk status kipas. Selain itu, sistem memberikan peringatan melalui *buzzer* jika mendeteksi kadar asap berbahaya.

### 5.1.2 Implementasi Perangkat Lunak

Implementasi perangkat lunak merupakan tahap penerapan dari rancangan sistem yang telah dibuat sebelumnya. Bagian ini mencakup tiga komponen utama, yaitu tampilan antarmuka aplikasi, layanan *Firebase* untuk penyimpanan data, dan penggunaan MIT App Inventor sebagai platform pengembangan aplikasi Android.

### 5.1.2.1 Tampilan Antarmuka Aplikasi



Gambar 5. 8 Tampilan Antarmuka Aplikasi

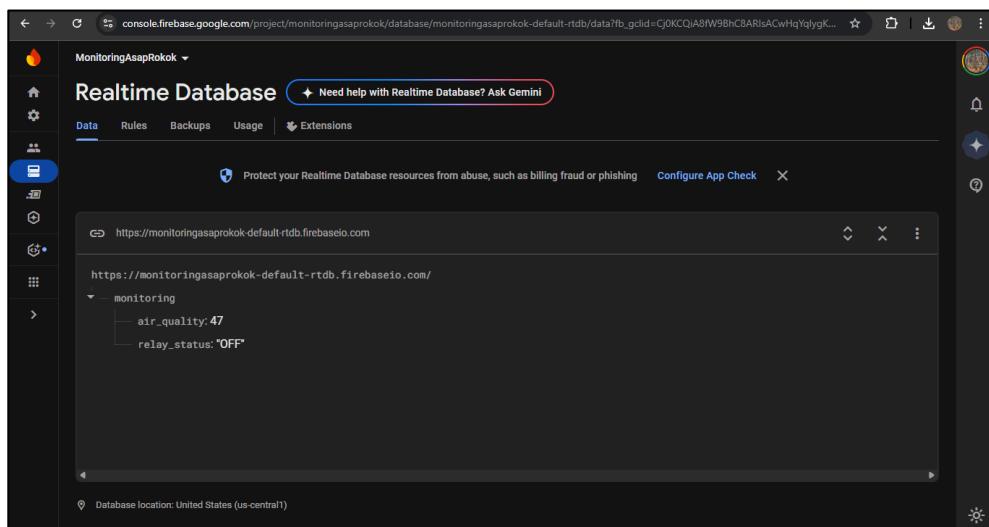
Antarmuka aplikasi yang ditampilkan pada gambar merupakan hasil implementasi dari rancangan sistem monitoring asap rokok berbasis IoT. Tampilan antarmuka ini dirancang dengan pendekatan yang sederhana dan intuitif agar mudah dipahami oleh pengguna. Pada bagian atas terdapat header yang memuat judul aplikasi "Monitoring Asap Rokok" beserta identitas organisasi "KAPMI D.I. Yogyakarta" untuk memperkuat konteks dan sumber sistem.

Selanjutnya, informasi utama ditampilkan dalam dua bagian utama: Kondisi Asap dan Status *Fan*. Bagian Kondisi Asap menampilkan nilai kualitas udara yang dibaca oleh sensor dalam satuan PPM (Part Per Million), dan disertai dengan ikon visual asap untuk meningkatkan pemahaman visual

tentang kondisi. Di bawahnya, ada komponen Status *Fan* yang menampilkan status *relay* atau kipas yang dikendalikan secara otomatis, dengan ikon kipas. Label dan ikon pendukung memiliki warna hijau terang yang dikontraskan dengan latar hitam sehingga informasi dapat dibaca dengan jelas. Antarmuka ini tidak hanya menampilkan data tetapi juga memberikan pengalaman pengguna yang menarik dan responsif untuk memantau kualitas udara secara *real-time*.

### 5.1.2.2 Firebase

Pada Gambar 5.9 terlihat struktur data yang disimpan pada *Firebase*, dengan dua *node* utama yaitu *air\_quality* dan *relay\_status* di bawah jalur monitoring. *Node* *air\_quality* berisi nilai hasil pembacaan sensor MQ-135, sedangkan *node* *relay\_status* menunjukkan status ON/OFF dari kipas yang dikendalikan oleh *relay*. Data ini diperbarui secara berkala dan dapat diakses oleh aplikasi Android melalui integrasi dengan *Firebase*, sehingga memungkinkan monitoring kualitas udara secara *real-time*.

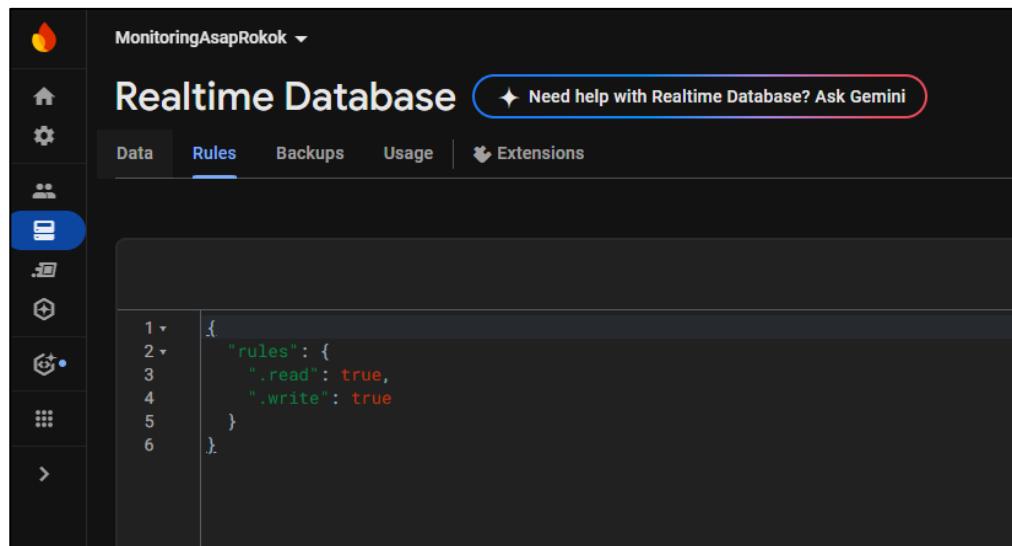


Gambar 5. 9 Tampilan data *real-time* pada *Firebase Realtime Database*

*Firebase Realtime Database* digunakan sebagai media penyimpanan *cloud* untuk menyimpan data kualitas udara yang dideteksi oleh sensor. Gambar 5.10 memperlihatkan pengaturan aturan akses *database* (*rules*) yang digunakan pada sistem ini, yaitu mengatur izin baca (.read) dan tulis (.write) ke nilai true.

Pengaturan ini memungkinkan ESP32 untuk mengirim data ke *database* tanpa autentikasi tambahan, yang sesuai digunakan dalam tahap pengembangan dan pengujian sistem.

Aturan akses ini ditampilkan dalam bentuk struktur JSON, di mana nilai *true* pada atribut *.read* dan *.write* menandakan bahwa semua pengguna memiliki hak akses penuh terhadap database tanpa batasan autentikasi.



```

1  {
2    "rules": {
3      ".read": true,
4      ".write": true
5    }
6  }

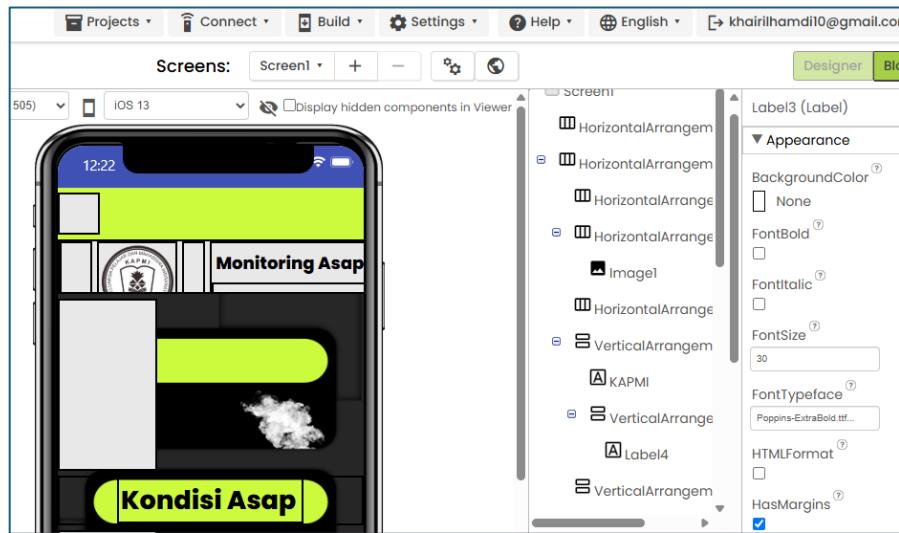
```

Gambar 5. 10 Pengaturan rules *Firebase Realtime Database*

### 5.1.2.3 Mit App Inventor

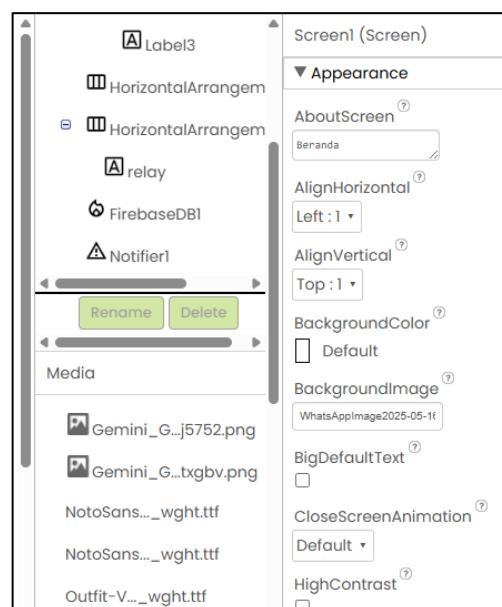
Antarmuka aplikasi disusun menggunakan beberapa komponen visual utama, yaitu komponen *Label*, *Image*, dan *Layout*. Komponen *Label* digunakan untuk menampilkan data kualitas udara dan status *relay* secara *real-time*. Komponen *Image* digunakan sebagai elemen visual tambahan seperti logo organisasi dan gambar simbolik asap. Untuk mengatur posisi dan tata letak antar elemen secara rapi, digunakan komponen *HorizontalArrangement* dan *VerticalArrangement*.

Tata letak dirancang agar pengguna dapat dengan mudah membaca data dan memahami kondisi secara langsung dari aplikasi. Desain keseluruhan aplikasi ditunjukkan pada Gambar 5.11.



Gambar 5. 11 Tampilan Desain Aplikasi pada MIT App Inventor

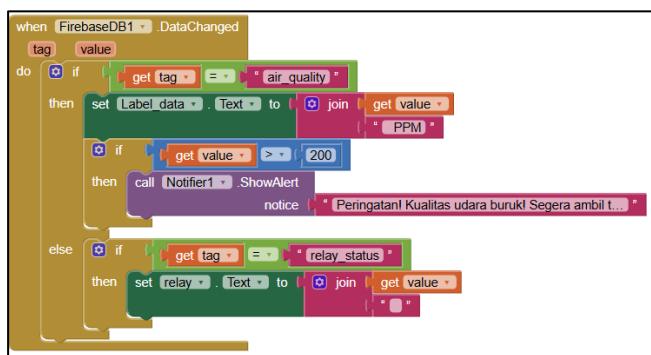
Komponen *FirebaseDB1* menghubungkan aplikasi ke *Firebase*. Dalam komponen ini, pengaturan URL *database Firebase* pada bagian *FirebaseURL*, token akses pada *FirebaseToken*, dan nama cabang data utama (bucket), yaitu monitoring, pada bagian *ProjectBucket*, dilakukan. Dengan konfigurasi ini, aplikasi dapat membaca dan menampilkan data dari tag *air\_quality* dan *relay\_status*, yang dikirim oleh mikrokontroler ESP32. Gambar 5.12 menunjukkan tampilan konfigurasi *Firebase* pada aplikasi.



Gambar 5. 12 Konfigurasi Komponen *Firebase* di MIT App Inventor

Untuk mendukung fungsi notifikasi, aplikasi menggunakan komponen bawaan Notifier dari MIT App Inventor. Komponen ini digunakan untuk menampilkan peringatan secara langsung di layar aplikasi saat kualitas udara terdeteksi buruk, yaitu ketika nilai PPM melebihi 100. Notifikasi ini bertujuan untuk memberikan peringatan kepada pengguna agar segera mengambil tindakan terhadap kondisi udara yang tidak sehat.

Seperti yang ditunjukkan pada Gambar 5.13, logika pemrosesan data dirancang menggunakan blok visual MIT App Inventor. Aplikasi akan melihat tag data yang diubah ketika data *Firebase* berubah. Nilai kualitas udara ditampilkan dan dibandingkan dengan ambang batas jika tag *air\_quality* diterima. Jika nilainya lebih dari 100, aplikasi secara otomatis mengirimkan notifikasi. Jika tag *relay\_status* diterima, aplikasi menampilkan status *relay* ON atau OFF.



Gambar 5. 13 Blok Diagram Aplikasi

Blok program pada aplikasi dirancang untuk merespons perubahan data yang terjadi di *Firebase Realtime Database* secara *real-time*. Ketika data berubah, blok *FirebaseDB1.DataChanged* akan dipicu dan menyesuaikan respons aplikasi berdasarkan nilai tag yang diterima. Jika tag yang berubah adalah *air\_quality*, maka nilai kualitas udara (dalam satuan PPM) akan ditampilkan pada label tampilan aplikasi. Selain itu, jika nilai kualitas udara tersebut mencapai atau melebihi 100 PPM, aplikasi akan secara otomatis menampilkan notifikasi peringatan kepada pengguna menggunakan komponen Notifier1 dengan metode *ShowAlert*. Notifikasi tersebut bertujuan untuk memberikan informasi bahwa kualitas udara dalam keadaan buruk sehingga

pengguna dapat segera mengambil tindakan. Sementara itu, jika tag yang diterima adalah *relay\_status*, maka sistem akan memperbarui status tampilan *relay* sesuai dengan nilai yang diterima dari *Firebase*, misalnya ON atau OFF. Fitur ini menunjukkan status aktif atau tidaknya perangkat seperti kipas penyedot asap yang dikendalikan secara otomatis melalui sistem.

Secara keseluruhan, aplikasi ini mendukung pemantauan kualitas udara dari asap rokok secara *real-time* dan responsif dengan memberikan informasi langsung dan peringatan dini melalui notifikasi.

## 5.2 Pengujian Hardware

Subbab ini membahas hasil pengujian perangkat keras (hardware) yang digunakan dalam sistem, guna memastikan bahwa setiap komponen berfungsi sesuai dengan perancangan dan mampu menjalankan tugasnya dengan baik dalam kondisi nyata.

### 5.2.1 Pengujian Akurasi Sensor MQ-135

Untuk mengetahui seberapa akurat sensor MQ-135 dalam membaca konsentrasi gas (PPM), dilakukan proses kalibrasi dengan cara membandingkan nilai pembacaan sensor terhadap alat ukur standar kualitas udara (air quality meter). Pengujian dilakukan selama 10 menit dengan membaca nilai dari kedua alat setiap satu menit sekali. Selisih antara pembacaan sensor dan alat standar kemudian dihitung untuk mengetahui standar yang terjadi. Selain itu, dilakukan pula perhitungan persentase error untuk menilai seberapa jauh nilai sensor menyimpang dari standar yang seharusnya, berikut rumus penghitungan errornya.

$$\text{Error} = \left( \frac{\text{MQ135} - \text{Alat Ukur Standar}}{\text{Alat Ukur Standar}} \right) \times 100\%$$

Tabel 5. 1 Akurasi Sensor MQ135 dan Alat Ukur Standar

No.	Waktu (Menit)	MQ-135 (PPM)	Alat Ukur Standar (PPM)	Selisih (PPM)	Error (%)
1	1	87	83	+4	4.82%
2	2	108	103	+5	4.85%
3	3	147	140	+7	5.00%
4	4	174	169	+5	2.96%
5	5	203	196	+7	3.57%
6	6	239	232	+7	3.02%
7	7	266	259	+7	2.70%
8	8	288	281	+7	2.49%
9	9	305	298	+7	2.35%
10	10	319	312	+7	2.24%

Hasil pengujian kalibrasi yang ditunjukkan pada tabel di atas menunjukkan bahwa nilai pembacaan sensor MQ-135 secara umum cukup dekat dengan nilai yang ditunjukkan oleh alat ukur standar kualitas udara. Selisih pada setiap pengujian berkisar antara 4 hingga 7 PPM, dengan persentase error antara 2,24% hingga 5,00%. Nilai error ini masih tergolong rendah, menunjukkan bahwa sensor MQ-135 dapat mendeteksi konsentrasi gas dengan cukup akurat, terutama dalam hal pemantauan kualitas udara umum.

### 5.2.2 Pengujian Buzzer

Pengujian ini dilakukan untuk mengetahui respon bagaimana sistem memberikan peringatan melalui buzzer. Pengujian dilakukan sebanyak tiga kali dengan kondisi udara yang bervariasi, dan data pembacaan sensor dicatat setiap menit selama lima menit. Hasil pengujian ditunjukkan pada Tabel 5.2. berikut.

Tabel 5. 2 Pengujian Sensor Mq-135 dan Buzzer

Menit	Pengujian 1		Pengujian 2		Pengujian 3	
	Nilai	Buzzer	Nilai	Buzzer	Nilai	Buzzer
1	0	OFF	56	OFF	37	OFF
2	14	OFF	224	ON	105	ON
3	64	OFF	210	ON	266	ON
4	150	ON	255	ON	234	ON
5	205	ON	180	OFF	285	ON

Berdasarkan hasil pengujian pada Tabel 5.2, terlihat bahwa buzzer mulai aktif ketika nilai pembacaan sensor MQ-135 melebihi ambang batas tertentu, yang pada sistem ini ditetapkan secara eksperimen di kisaran nilai 100. Pada Pengujian 2 dan Pengujian 3, *buzzer* menyala lebih cepat karena kemungkinan paparan gas lebih tinggi sejak awal. Hal ini menunjukkan bahwa sistem berhasil merespon perubahan kualitas udara dengan baik melalui notifikasi *buzzer* yang berfungsi secara otomatis.

### 5.2.3 Pengujian *Relay* dan *Fan*

Pengujian *relay* dilakukan untuk memastikan bahwa modul *relay* dapat menerima dan merespons sinyal logika digital dari mikrokontroler (ESP32) dengan baik. Dalam pengujian ini, sistem dikonfigurasi untuk mengaktifkan (ON) dan menonaktifkan (OFF) *relay* secara bergantian dengan interval waktu 2 detik. Status indikator LED pada modul *relay*, bunyi klik sebagai tanda aktifnya mekanisme saklar, serta kondisi beban (kipas) diamati untuk memastikan bahwa proses *switching* bekerja sesuai dengan perintah logika yang diberikan.

Tabel 5. 3 Pengujian *Relay* dan *Fan*

No	<i>Relay</i>	<i>Fan</i>
1	HIGH	ON
2	LOW	OFF
3	HIGH	ON
4	LOW	OFF
5	HIGH	ON
6	LOW	OFF

Berdasarkan hasil pengujian pada Tabel 5.3, dapat dilihat bahwa modul *relay* berhasil merespons sinyal logika yang dikirimkan dari mikrokontroler.

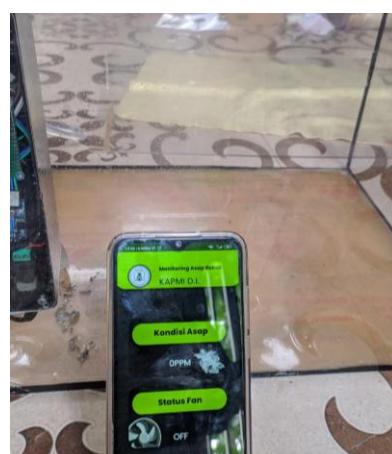
Ketika logika yang diberikan adalah HIGH, kipas menyala (ON), dan ketika logika berubah menjadi LOW, kipas dalam keadaan mati (OFF). Hal ini menunjukkan bahwa sistem *switching* bekerja secara normal, di mana status beban (kipas) mengikuti sinyal logika digital dari ESP32 dengan interval pengujian setiap 2 detik. Respon ini juga mengindikasikan bahwa koneksi antara ESP32, modul *relay*, dan kipas telah berfungsi dengan baik tanpa adanya delay atau gangguan pada *switching*.

### 5.3 Pengujian Alat Secara Keseluruhan

Pengujian deteksi asap menilai sensitivitas dan keakuratan sensor MQ-135 untuk mendeteksi berbagai jenis asap dari berbagai sumber. Tujuan dari pengujian ini adalah untuk memastikan bahwa sistem dapat mengenali perubahan kualitas udara secara tepat dan memberikan respons otomatis, seperti mengaktifkan *buzzer* dan mengontrol *relay* kipas. Selama pengujian, nilai kualitas udara dihitung secara konsisten selama 5 menit. Hasil dilacak melalui tampilan aplikasi, bersama dengan perubahan status *fan* dan notifikasi. Pengujian ini menguji kemampuan sistem untuk mengenali berbagai jenis Asap dan menanggapinya secara *real-time*.

#### 5.3.1 Pengujian Dengan Asap Pembakaran Kertas

Pengujian pertama dilakukan menggunakan asap dari pembakaran kertas yang diarahkan ke sensor MQ-135 dengan jarak sekitar 5–10 cm. Nilai kualitas udara diamati setiap 1 menit selama 5 menit melalui aplikasi.



Gambar 5. 14 Kondisi Normal



Gambar 5. 15 Pengujian alat dengan Asap Pembakaran Kertas

Pada gambar 5.14 menunjukkan kondisi alat masih dalam keadaan normal dengan nilai sensor asap 55 ppm kondisi asap normal sehingga *fan* dan *buzzer* tidak menyala, sedangkan gambar 5.15 menunjukkan pengujian dengan asap pembakaran kertas dengan nilai sensor 210 ppm merupakan nilai pertama yang tampil pada aplikasi setelah diuji dengan asap pembakaran kertas, kondisi asap berbahaya sehingga *fan* dan *buzzer* menyala.

Tabel 5. 4 Pengujian dengan asap pembakaran kertas

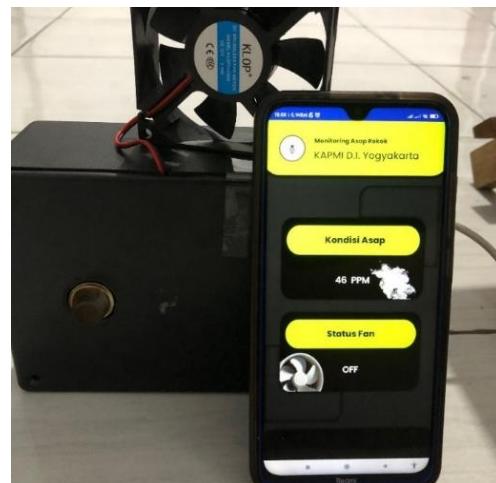
Pengujian ke (per-1 menit)	Tampilan			<i>Buzzer</i>	Keterangan
	Kondisi Asap	Status <i>Fan</i>	Notifikasi		
0	55 PPM	OFF	Tidak	Tidak	Normal
1	210 PPM	ON	Terkirim	Bunyi	Bahaya
2	220 PPM	ON	Terkirim	Bunyi	Bahaya
3	250 PPM	ON	Terkirim	Bunyi	Bahaya
4	100 PPM	ON	Terkirim	Bunyi	Bahaya
5	180 PPM	OFF	Tidak	Tidak	Sedang

Berdasarkan Tabel 5.4, hasil pengujian dengan asap pembakaran kertas menunjukkan bahwa pada menit ke-0, kualitas udara berada dalam kondisi normal dengan nilai 55 PPM dan sistem tidak mengaktifkan kipas, mengirim notifikasi maupun mengaktifkan *buzzer*. Mulai menit ke-1 hingga menit ke-4,

terdeteksi peningkatan kadar asap di atas ambang batas 100 PPM, yang mengakibatkan kipas (*relay*) menyala, notifikasi terkirim, dan *buzzer* berbunyi sebagai peringatan bahaya. Pada menit ke-5, kadar asap menurun menjadi 180 PPM, sehingga sistem otomatis mematikan *buzzer* dan *relay*, dan status berubah menjadi sedang.

### 5.3.2 Pengujian Dengan Asap Rokok

Pengujian kedua dilakukan dengan menggunakan asap rokok sebagai sumber emisi. Sensor MQ-135 didekatkan pada asap rokok, dan data diambil setiap 1 menit selama 5 menit. Tujuan dari pengujian ini adalah untuk melihat respon sistem terhadap paparan asap rokok dalam durasi tertentu.



Gambar 5. 16 Kondisi Sebelum pengujian Asap Rokok



Gambar 5. 17 Pengujian dengan Asap Rokok

Pada gambar 5.16 kondisi alat dalam keadaan normal dengan nilai sensor 46 ppm, sehingga *fan* dan *buzzer* tidak akan menyala. Pada gambar 5.17 pengujian dengan asap rokok dengan nilai sensor 250 ppm nilai ini melebihi ambang batas, sehingga *fan* dan *buzzer* akan menyala.

Tabel 5. 5 Pengujian dengan Asap Rokok

Pengujian ke (per-1 menit)	Tampilan			<i>Buzzer</i>	Keterangan
	Kondisi Asap	Status <i>Fan</i>	Notifikasi		
0	46 PPM	OFF	Tidak	Mati	Normal
1	250 PPM	ON	Terkirim	Bunyi	Bahaya
2	280 PPM	ON	Terkirim	Bunyi	Bahaya
3	350 PPM	ON	Terkirim	Bunyi	Bahaya
4	300 PPM	ON	Terkirim	Bunyi	Bahaya
5	205 PPM	ON	Terkirim	Bunyi	Bahaya

Berdasarkan hasil pengujian dengan asap rokok seperti yang ditunjukkan pada Tabel 5.5, diketahui bahwa kadar asap meningkat secara signifikan setelah menit pertama, mencapai angka tertinggi sebesar 350 PPM pada menit ke-3. Sensor mendeteksi kondisi berbahaya mulai dari menit ke-1, yang memicu aktifnya kipas,terkirimnya notifikasi, dan bunyi *buzzer* sebagai peringatan. Hingga menit ke-5, sistem terus menunjukkan kondisi berbahaya, menandakan bahwa asap rokok memberikan respon yang kuat terhadap sensor MQ-135 dan sistem bekerja dengan baik dalam merespons paparan asap tersebut.

### 5.3.3 Pengujian Dengan Asap *Vape*(Rokok Elektrik)

Pengujian selanjutnya dilakukan dengan menggunakan asap dari vape atau rokok elektrik. Pengujian ini bertujuan untuk mengetahui seberapa sensitif sistem dalam mendeteksi asap buatan dari rokok elektrik yang memiliki karakteristik berbeda dengan asap rokok konvensional. Sensor akan mengambil data setiap 1 menit selama 5 menit untuk mengamati perubahan kadar asap dan respon sistem secara keseluruhan.



Gambar 5. 18 Kondisi sebelum pengujian dengan *Vape*



Gambar 5. 19 Pengujian dengan Asap *Vape* (Rokok Elektrik)

Pada gambar 5.18 menunjukkan alat pada kondisi awal dengan nilai sensor 77 ppm, sehingga kontrol *buzzer* dan fan tidak menyala. Pada gambar 5.19 pengujian asap *vape* mencapai nilai 499 ppm nilai ini melebihi ambang batas, sehingga dapat memicu kontrol fan dan *buzzer*.

Tabel 5. 6 Pengujian dengan Asap Vape(Rokok Elektrik)

Pengujian ke (per-1 menit)	Tampilan			<i>Buzzer</i>	Keterangan
	Kondisi Asap	Status Fan	Notifikasi		
0	77 PPM	OFF	Tidak	Mati	Normal
1	170 PPM	ON	Terkirim	Bunyi	Bahaya
2	344 PPM	ON	Terkirim	Bunyi	Bahaya
3	450 PPM	ON	Terkirim	Bunyi	Bahaya
4	480 PPM	ON	Terkirim	Bunyi	Bahaya
5	499 PPM	ON	Terkirim	Bunyi	Bahaya

Hasil pengujian dengan rokok elektrik atau asap vape menunjukkan bahwa sistem cukup sensitif terhadap jenis asap ini. Pada menit pertama, kadar asap masih dalam batas normal (77 PPM), sehingga kipas, notifikasi, dan *buzzer* tidak aktif. Namun, pada menit kedua, nilai PPM meningkat pesat dan melebihi ambang batas 100 PPM, sehingga kipas dan *buzzer* otomatis menyala sebagai peringatan. Kondisi ini bertahan hingga menit kelima, ketika kadar asap mencapai tingkat tertinggi sebesar 499 PPM. Ini menunjukkan bahwa sistem dapat mendeteksi dan memberikan peringatan yang tepat tentang asap rokok elektrik.

#### **5.4 Analisis Hasil Pengujian**

Setelah pengujian dengan berbagai jenis sumber asap selesai, sistem menunjukkan respons yang berbeda terhadap karakteristik asap yang dihasilkan. Setiap jenis sumber asap menunjukkan nilai PPM dan waktu pemulihan yang berbeda. Tujuan analisis ini adalah untuk mengevaluasi kemampuan sistem monitoring untuk mendeteksi kualitas udara berdasarkan parameter yang telah ditetapkan.

##### **5.4.1 Analisis Pengujian Asap Pembakaran Kertas**

Pada pengujian ini, asap dari pembakaran kertas menunjukkan kenaikan nilai PPM yang cukup signifikan di menit ke-1 hingga ke-4, yaitu mencapai puncaknya pada 250 PPM. Sistem merespons dengan mengaktifkan *buzzer* dan fan sebagai indikator kondisi berbahaya. Namun pada menit ke-5, nilai PPM kembali turun menjadi 180 PPM, yang menunjukkan bahwa sistem berhasil memulihkan kondisi ke keadaan normal. Hal ini membuktikan bahwa sistem dapat mendeteksi keberadaan asap dan memberikan respons cepat serta mampu memulihkan status jika polusi menurun.

##### **5.4.2 Analisis Pengujian Asap Rokok**

Pengujian dengan asap rokok memulai dengan nilai PPM yang tinggi, langsung pada 250 PPM dan meningkat hingga 350 PPM. Selama pengujian, tidak ada pemulihan ke nilai normal. Asap rokok memiliki konsentrasi zat

berbahaya yang lebih tinggi dan lebih lama berada di udara dibandingkan asap kertas; oleh karena itu, *buzzer* dan fan terus-menerus dihidupkan, menunjukkan bahwa sistem bekerja dengan baik untuk memberikan peringatan saat kondisi udara menjadi berbahaya.

#### **5.4.3 Analisis Pengujian Asap Vape**

Asap dari rokok elektrik (vape) menghasilkan tren peningkatan PPM yang paling tinggi dan konsisten dibandingkan pengujian lainnya. Nilai awal sebesar 170 PPM terus meningkat secara signifikan hingga menyentuh 499 PPM di akhir pengujian. Tidak hanya mencapai angka tertinggi, asap vape juga tidak menunjukkan adanya penurunan konsentrasi selama lima menit pengujian. Sistem bekerja dengan baik untuk mendeteksi situasi ini dan secara terus-menerus mengaktifkan peringatan melalui *buzzer* dan fan. Hal ini membuktikan bahwa sensor cukup sensitif dalam mendeteksi partikel dari asap uap vape meskipun karakter partikelnya berbeda dari asap konvensional.

#### **5.4.4 Analisis Umum Sistem**

Asap dari rokok elektrik (vape) menghasilkan tren peningkatan PPM yang paling tinggi dan konsisten dibandingkan pengujian lainnya. Nilai awal sebesar 170 PPM terus meningkat secara signifikan hingga menyentuh 499 PPM di akhir pengujian. Tidak hanya mencapai angka tertinggi, asap vape juga tidak menunjukkan adanya penurunan konsentrasi selama lima menit pengujian. Sistem bekerja dengan baik untuk mendeteksi situasi ini dan secara terus-menerus mengaktifkan peringatan melalui *buzzer* dan notifikasi begitupun mengaktifkan fan. Hal ini membuktikan bahwa sensor cukup sensitif dalam mendeteksi partikel dari asap uap vape meskipun karakter partikelnya berbeda dari asap konvensional.

### **5.5 Kesesuaian Dengan Tujuan**

Berdasarkan hasil pengujian dan analisis, sistem pemantauan asap rokok yang menggunakan sensor MQ-135 telah memenuhi tujuan awalnya secara efektif. Ini adalah pilihan yang tepat karena sensor ini sensitif terhadap bahan kimia dalam asap rokok dan memiliki kemampuan untuk mendeteksi secara

akurat. Sistem dapat mendeteksi asap rokok dengan cepat, mengirim data ke *Firebase* secara teratur, dengan akurat menampilkan informasi di aplikasi Android, dan mengaktifkan peringatan audio dan visual saat terdeteksi aktivitas merokok.

Keberhasilan ini menunjukkan bahwa sensor MQ-135 dapat digunakan secara efektif untuk memantau asap rokok secara *real-time*. Menggabungkannya dengan platform *Firebase* dan aplikasi Android, arsitektur sistem IoT yang dipilih ini adalah pilihan yang tepat.

## BAB VI

## PENUTUP

### 6.1 Kesimpulan

Berdasarkan tujuan penelitian yang telah ditetapkan, yaitu "Merancang sistem monitoring asap rokok berbasis IoT menggunakan sensor MQ-135 yang dapat mendekripsi dan merespons keberadaan asap rokok secara real-time melalui aplikasi", dapat disimpulkan bahwa. Sistem pemantauan asap rokok yang dirancang dan dikembangkan telah berhasil memenuhi tujuan penelitian secara efektif. Hal ini ditunjukkan oleh beberapa aspek utama:

1. Sensor MQ-135 terbukti mampu mendekripsi keberadaan berbagai jenis asap, termasuk asap pembakaran kertas, asap rokok konvensional, dan asap vape (rokok elektrik). Sensor ini menunjukkan sensitivitas yang baik terhadap zat kimia dalam asap rokok.
2. Sistem mampu merespons secara otomatis dan *real-time* terhadap peningkatan konsentrasi asap. Ketika kadar asap melebihi ambang batas yang ditetapkan (100 PPM), mikrokontroler ESP32 akan mengaktifkan *buzzer* sebagai alarm suara dan menyalakan kipas ventilasi melalui modul *relay* untuk membantu mengurangi kadar asap.
3. Data hasil deteksi sensor, seperti nilai kadar asap dalam PPM dan status kipas (ON/OFF), dikirimkan secara *real-time* ke *Firebase Realtime Database* dan ditampilkan secara akurat pada aplikasi mobile yang dikembangkan menggunakan MIT App Inventor. Aplikasi ini juga mampu mengirimkan notifikasi peringatan kepada pengguna (Pengurus KAPMI) jika kualitas udara terdeteksi buruk.

Dengan demikian, integrasi sensor MQ-135, mikrokontroler ESP32, *Firebase*, dan aplikasi MIT App Inventor telah membentuk arsitektur sistem IoT yang fungsional dan responsif untuk pemantauan dan pengendalian asap rokok di lingkungan dalam ruangan.

## 6.2 Saran

Untuk pengembangan sistem lebih lanjut agar lebih optimal dan dapat diterapkan secara lebih luas, beberapa saran diberikan sebagai berikut:

1. Penambahan Parameter Kualitas Udara: Sistem saat ini hanya mendeteksi asap rokok menggunakan sensor MQ-1353. Disarankan untuk menambahkan sensor untuk mengukur parameter kualitas udara lain seperti PM2.5 dan PM10 untuk pemantauan yang lebih komprehensif.
2. Integrasi Fitur Pemurni Udara: Tindakan otomatis saat ini terbatas pada notifikasi, aktivasi kipas, dan alarm suara. Disarankan untuk mengintegrasikan fitur pemurni udara ke dalam sistem untuk penanganan asap yang lebih efektif.
3. Pengembangan Aplikasi yang Lebih Fleksibel: Fitur dan tampilan aplikasi saat ini masih terbatas dibandingkan platform pengembangan lainnya. Disarankan untuk mengembangkan aplikasi pemantauan yang lebih fleksibel dan kaya fitur dengan tampilan yang lebih dinamis untuk meningkatkan pengalaman pengguna.
4. Penerapan Teknologi Lanjutan: Sistem ini memiliki potensi untuk dikembangkan lebih lanjut dengan teknologi Artificial Intelligence (AI) atau diintegrasikan ke dalam sistem smart home yang lebih besar.

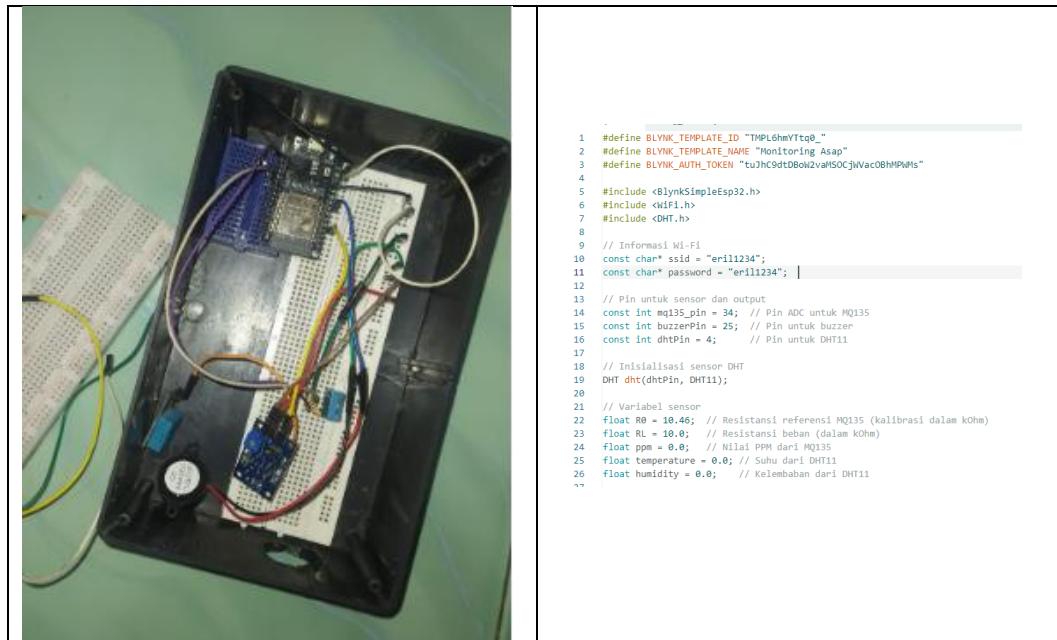
## DAFTAR PUSTAKA

- Basuki, A., & Efendi, E. (2022). *Prototipe Monitoring dan Penetralisir Asap Rokok Berbasis IoT*. 195–202. <http://journal.itny.ac.id/index.php/ReTII>
- Espressif. (t.t.). *ESP32 Series Datasheet Version 4.7 2.4 GHz Wi-Fi + Bluetooth ® + Bluetooth LE SoC Including*. [www.espressif.com](http://www.espressif.com)
- Hardika, D., & Nurfiana, N. (2019). Sistem Monitoring Asap Rokok Menggunakan Smartphone Berbasis Internet Of Things (IOT). *Explore: Jurnal Sistem informasi dan telematika*, 10(1). <https://doi.org/10.36448/jsit.v10i1.1221>
- Khesya, N. (2021). Mengenal Flowchart Dan Pseudocode Dalam Algoritma Dan Pemrograman. <https://doi.org/10.31219/osf.io/dq45e>
- Kong, S.-C., & Abelson, H. (2019). *Computational Thinking Education* (S.-C. Kong & H. Abelson, Ed.). Springer Singapore. <https://doi.org/10.1007/978-981-13-6528-7>
- Kopetz, H., & Steiner, W. (2022). Internet of Things. Dalam *Real-Time Systems* (hlm. 325–341). Springer International Publishing. [https://doi.org/10.1007/978-3-031-11992-7\\_13](https://doi.org/10.1007/978-3-031-11992-7_13)
- Menteri Lingkungan Hidup. (t.t.). *P\_14\_2020\_ISPU\_menlhk\_07302020074834*. Diambil 9 Desember 2024, dari <https://jdih.menlhk.go.id/>
- Muliadi, M., Imran, A., & Rasul, M. (2024). Pengembangan Tempat Sampah Pintar Menggunakan ESP32. *Jurnal Media Elektrik*, 17(2), 73–79. <https://doi.org/10.59562/metric.v17i2.5398>
- Nur Hakim, L. (2023). *Prototype Sistem Monitoring Asap Rokok Pada Ruangan Berbasis IoT Dan Wemos D1 R1 ESP 8266*. <http://repository.unissula.ac.id/id/eprint/31613>
- Nuryati, N., Sumeru, K., Setyawan, A., Hikmat, Y. P., Sumeru, H. A., & Sukri, M. F. bin. (2023). Pengaruh Asap Rokok pada Peningkatan Konsentrasi PM2.5 dan PM10 di Ruang Tamu Akibat Merokok di Dalam dan di Luar Rumah. *Jurnal Ilmu Lingkungan*, 22(1), 85–92. <https://doi.org/10.14710/jil.22.1.85-92>
- olimex.com. (t.t.). *DATASHEET SNS-MQ135*.
- Priyaungga, B. A., Aji, D. B., Syahroni, M., Aji, N. T. S., & Saifudin, A. (2020). Pengujian Black Box pada Aplikasi Perpustakaan Menggunakan Teknik Equivalence Partitions. *Jurnal Teknologi Sistem Informasi dan Aplikasi*, 3(3), 150. <https://doi.org/10.32493/jtsi.v3i3.5343>
- Rombang, I. A., Setyawan, L. B., & Dewantoro, G. (2022). Perancangan Prototipe Alat Deteksi Asap Rokok dengan Sistem Purifier Menggunakan Sensor MQ-

- 135 dan MQ-2. *Techné: Jurnal Ilmiah Elektroteknika*, 21(1), 131–144. <https://doi.org/10.31358/techne.v21i1.312>
- Septian, T., & Djamaruddin, S. (2022). Pengaruh Kebijakan Earmarking Cukai Hasil Tembakau Terhadap Pelayanan Kesehatan Masyarakat Di Indonesia. *Jurnal Perspektif Bea Dan Cukai*, 6(2), 243–257. <https://doi.org/10.31092/jpbc.v6i2.1692>
- Soleimani, F., Dobaradaran, S., De-la-Torre, G. E., Schmidt, T. C., & Saeedi, R. (2022). Content of toxic components of cigarette, cigarette smoke vs cigarette butts: A comprehensive systematic review. *Science of The Total Environment*, 813, 152667. <https://doi.org/10.1016/j.scitotenv.2021.152667>
- Trimbakrao Gaikwad Bharati Vidyapeeth, A., Chougale, P., Yadav, V., Gaikwad, A., & Vidyapeeth, B. (2022). FIREBASE-OVERVIEW AND USAGE. Article in *Journal of Engineering and Technology Management*. www.irjmets.com

## LAMPIRAN

Lampiran 1 Dokumentasi Pembuatan Alat



```
1 #define BLYNK_TEMPLATE_ID "TMPL6hmYTtq0_"
2 #define BLYNK_TEMPLATE_NAME "Monitoring Asap"
3 #define BLYNK_AUTH_TOKEN "tuJhC9dtDBow2v4SOcJiVacOBHMPMs"
4
5 #include <BlynkSimpleEsp32.h>
6 #include <WiFi.h>
7 #include <DHT.h>
8
9 // Informasi Wi-Fi
10 const char* ssid = "erill1234";
11 const char* password = "erill1234"; |
12
13 // Pin untuk sensor dan output
14 const int mq135_pin = 34; // Pin ADC untuk MQ135
15 const int buzzerPin = 25; // Pin untuk buzzer
16 const int dhtPin = 4; // Pin untuk DHT11
17
18 // Inisialisasi sensor DHT
19 DHT dht(dhtPin, DHT11);
20
21 // Variabel sensor
22 float R0 = 10.46; // Resistansi referensi MQ135 (kalibrasi dalam kohm)
23 float RL = 10.0; // Resistansi beban (dalam kohm)
24 float ppm = 0.0; // Nilai PPM dari MQ135
25 float temperature = 0.0; // Suhu dari DHT11
26 float humidity = 0.0; // Kelembaban dari DHT11
~~
```

Lampiran 2 Tempat Penelitian

