Jaringan *LoRa Star* untuk Sistem Monitoring Kebocoran Gas di Pabrik berbasis *Internet of Things*

SKRIPSI

Disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro di Jurusan Teknik Elektro Fakultas Teknik Universitas Sam Ratulangi

Dibuat oleh :

Jeheskiel Jufanly Wereh

NIM: 210211030099

UNIVERSITAS SAM RATULANGI

FAKULTAS TEKNIK MANADO

JURUSAN TEKNIK ELEKTRO

MANADO

2025

LEMBAR PENGESAHAN SKRIPSI

Calon dosen pembimbing yang bertandatangan dibawah ini, menyatakan bahwa mahasiswa S1 Program Studi Teknik Elektro berikut ini :

Nama	:	Jeheskiel Jufanly Wereh
NIM	:	210211030099
Judul Proposal	:	Jaringan LoRa Star untuk Sistem Monitoring Kebocoran Gas di Pabrik Berbasis Internet of Things

Telah melakukan konsultasi dan melalui proses pembimbingan penyusunan laporan proposal. Oleh karena itu, judul serta laporan proposal ini kami rekomendasikan untuk dapat diseminarkan.

Manado, 17 Juni 2025.

Calon Dosen Pembimbing 1,

Ir. Meicsy E. I. Najoan, S.T., M.T.

NIP. 196705271995121001

Abstrak

Sistem monitoring kebocoran gas metana berbasis Internet of Things (IoT) merupakan salah satu solusi penting untuk meningkatkan keamanan lingkungan industri. Penelitian ini merancang dan membangun prototipe sistem monitoring kebocoran gas menggunakan jaringan LoRa Star, yang terdiri dari node sensor berbasis ESP32 dengan sensor MQ-2 dan satu gateway pusat. Sistem ini dirancang untuk mendeteksi peningkatan konsentrasi gas metana di udara, kemudian mengirimkan data secara real-time melalui komunikasi LoRa ke gateway pusat. Gateway selanjutnya meneruskan data tersebut ke platform cloud seperti Adafruit IO untuk visualisasi dan Firebase Realtime Database untuk pencatatan historis. Pengujian dilakukan untuk mengevaluasi performa komunikasi LoRa Star dalam hal kestabilan pengiriman data, jangkauan, dan waktu respons. Hasil penelitian menunjukkan bahwa sistem mampu mengirimkan data secara konsisten dan akurat, serta dapat diintegrasikan ke dalam berbagai aplikasi pemantauan industri.

Kata kunci: IoT, LoRa Star, Kebocoran Gas, MQ-2, ESP32, Pemantauan Nirkabel.

Abstract

A methane gas leak monitoring system based on the Internet of Things (IoT) is a vital solution for enhancing safety in industrial environments. This study designs and develops a prototype monitoring system using a LoRa Star network, consisting of ESP32-based sensor nodes equipped with MQ-2 gas sensors and a central gateway. The system is designed to detect increased methane gas concentration in the air and transmit the data in real-time via LoRa communication to a central gateway. The gateway then forwards the data to cloud platforms such as Adafruit IO for visualization and Firebase Realtime Database for historical logging. System testing was conducted to evaluate the performance of the LoRa Star communication in terms of data transmission stability, coverage, and response time. Results indicate that the system is capable of delivering consistent and accurate data and can be integrated into various industrial monitoring applications.

Keywords: IoT, LoRa Star, Gas Leak, MQ-2, ESP32, Wireless Monitoring

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa atas limpahan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul "Jaringan LoRa Star untuk Sistem Monitoring Kebocoran Gas di Pabrik Berbasis Internet of Things". Skripsi ini disusun sebagai salah satu syarat untuk menyelesaikan program studi Strata 1 di Teknik Elektro, Fakultas Teknik, Universitas Sam Ratulangi.

Penelitian ini bertujuan untuk merancang dan membangun sistem monitoring kebocoran gas metana berbasis Internet of Things (IoT) dengan memanfaatkan jaringan komunikasi nirkabel LoRa Star. Sistem ini dirancang agar mampu mendeteksi konsentrasi gas dari beberapa titik sensor, lalu mengirimkannya secara real-time ke gateway pusat untuk divisualisasikan dan disimpan dalam platform cloud. Dengan sistem ini, diharapkan dapat meningkatkan efisiensi pemantauan dan mendukung keamanan operasional di lingkungan industri, khususnya pada instalasi yang memiliki potensi kebocoran gas.

Dalam kesempatan ini, penulis ingin menyampaikan rasa terima kasih yang tulus kepada:

1. Bapak Ir. Meicsy Eldad Israel Najoan ST, MT dan Bapak Ir. Sumenge Tangkawarouw Godion Kaunang MT, Ph.D selaku dosen pembimbing yang telah memberikan bimbingan, arahan, dan motivasi selama proses penelitian dan penulisan skripsi ini.

2. Bapak Janny Olny Wuwung, ST., MT, Bapak Ir. Benefit Semuel Narasiang, MT dan Bapak David Pang, ST., MT., Ph.D , yang telah memberikan masukan dan kritik yang konstruktif untuk menyempurnakan skripsi ini.

3. Rekan-rekan mahasiswa, yaitu Merfy Turang, Russell Rombepajung, William sigar, Angel Kamuh, Ricky Phandeirot dan lainnya, atas kerja sama, dukungan, serta bantuan yang diberikan selama pelaksanaan proses penelitian dan penulisan skripsi ini.

4. Kakak tingkat, yaitu Georgina Siagian, dan lainnya, yang telah memberikan bimbingan, saran, serta motivasi yang sangat berarti selama pelaksanaan penelitian dan penulisan skripsi ini.

5. Orang tua dan keluarga tercinta yang selalu memberikan dukungan moral, spiritual, dan material selama penyusunan skripsi ini.

6. Semua pihak yang tidak dapat disebutkan satu per satu, yang telah membantu baik secara langsung maupun tidak langsung dalam menyelesaikan penelitian dan skripsi ini.

Penulis menyadari bahwa skripsi ini masih memiliki keterbatasan. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi penyempurnaan karya ini di masa mendatang.

Akhir kata, semoga skripsi ini dapat memberikan manfaat bagi pembaca dan menjadi kontribusi dalam pengembangan teknologi monitoring berbasis IoT di bidang industri.

Manado, 17 Juni 2025

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
Pendahuluan

1.1 Latar Belakang

Penggunaan gas dalam berbagai sektor seperti industri, gedung perkantoran, dan fasilitas umum memiliki risiko kebocoran yang dapat membahayakan keselamatan manusia serta merusak properti. Gas metana (CH_4) sebagai salah satu jenis gas yang mudah terbakar, jika tidak terdeteksi secara cepat dan akurat, berpotensi menimbulkan kebakaran, ledakan, atau pencemaran udara serius. Oleh karena itu, sistem deteksi kebocoran gas yang andal sangat dibutuhkan untuk memantau kondisi lingkungan dan mencegah terjadinya insiden berbahaya.

Salah satu pendekatan efektif dalam implementasi sistem deteksi gas adalah dengan memanfaatkan Internet of Things (IoT), yang memungkinkan pengumpulan data secara real-time dari sensor-sensor gas yang tersebar. Dalam arsitektur ini, sensor-sensor gas terhubung ke node-node IoT yang kemudian mengirimkan data ke pusat pemantauan.

Untuk menjamin komunikasi data yang efisien, terutama pada lingkungan dengan hambatan fisik atau jangkauan yang luas, digunakan teknologi komunikasi LoRa Star. LoRa Star adalah arsitektur jaringan di mana semua node sensor berkomunikasi langsung dengan satu gateway pusat. Skema ini cocok untuk sistem monitoring yang sederhana, hemat energi, dan mudah dikelola, tanpa memerlukan komunikasi antar node seperti pada LoRa Mesh.

Penggunaan jaringan LoRa Star memungkinkan pengiriman data jarak jauh dengan konsumsi daya rendah, sangat ideal untuk area industri atau fasilitas terpencil. Sistem ini mampu memberikan notifikasi awal berdasarkan nilai konsentrasi gas yang dikirim oleh sensor, sehingga data dapat segera dianalisis untuk pengambilan keputusan lebih lanjut.

Dengan memusatkan fokus pada efisiensi komunikasi dan akurasi transmisi data dari sensor ke pusat pemantauan, sistem berbasis LoRa Star ini menawarkan solusi praktis untuk monitoring kebocoran gas, sekaligus mendemonstrasikan potensi teknologi LoRa dalam aplikasi IoT skala industri.

1.2 Rumusan Masalah

1. Bagaimana merancang sistem deteksi kebocoran gas metana berbasis IoT menggunakan arsitektur jaringan LoRa Star?
2. Bagaimana mengoptimalkan komunikasi antara node sensor dengan gateway pusat dalam jaringan LoRa Star untuk memastikan data terkirim secara akurat dan efisien?
3. Bagaimana sistem dapat melakukan pemantauan konsentrasi gas secara real-time dan mengirimkan data ke server atau aplikasi pemantauan?

1.3 Batasan Masalah

1. Sistem yang dikembangkan hanya sebatas prototipe dan simulasi, tidak melibatkan uji coba langsung di fasilitas industri.
2. Fokus utama berada pada sistem komunikasi dan pemantauan berbasis jaringan LoRa Star, tanpa mencakup tindakan pencegahan atau mitigasi lanjutan seperti kontrol valve atau aktivasi ventilasi.
3. Pengujian dilakukan dalam skala laboratorium dengan jumlah node terbatas (misalnya 1–2 node) untuk menilai efektivitas komunikasi dan transmisi data.
4. Sistem hanya mendeteksi dan mengirimkan data konsentrasi gas, tidak menangani klasifikasi jenis gas secara spesifik.

1.4 Tujuan Penelitian

1. Membangun sistem deteksi gas metana berbasis IoT dengan komunikasi nirkabel menggunakan jaringan LoRa Star.
2. Mendesain prototipe sistem monitoring yang mampu mengirimkan data konsentrasi gas dari beberapa node sensor ke gateway pusat secara efisien.
3. Mengevaluasi performa jaringan LoRa Star dalam hal kestabilan pengiriman data dan jangkauan komunikasi di lingkungan uji di laboratorium

1.5 Manfaat Penelitian

1. Meningkatkan Keamanan: Sistem ini memberikan solusi deteksi gas metana yang cepat dan akurat untuk meningkatkan keselamatan operasional.
2. Pemanfaatan Teknologi LoRa Star: Komunikasi langsung dari node ke gateway membuat sistem lebih hemat daya dan lebih mudah dipasang, cocok untuk area luas dengan titik pengawasan yang tersebar.

3. Referensi Sistem Keamanan IoT: Prototipe ini dapat menjadi dasar bagi pengembangan sistem deteksi kebocoran di berbagai sektor industri lainnya, dan memperluas penerapan IoT berbasis LoRa Star dalam sistem keamanan.

BAB II
LANDASAN TEORI

2.1 Kerangka Teori

Kerangka teori bertujuan untuk menjelaskan keterkaitan antar komponen teknologi yang digunakan dalam pengembangan sistem monitoring kebocoran gas berbasis Internet of Things (IoT). Sistem ini dibangun dari integrasi antara sensor gas (seperti MQ-2), mikrokontroler (ESP32), dan modul komunikasi nirkabel LoRa dalam topologi jaringan star, yang semuanya saling terhubung dan berfungsi untuk mendeteksi serta merespons potensi kebocoran gas.

Penggunaan topologi LoRa Star dipilih karena karakteristiknya yang mampu menjangkau area luas dengan konsumsi daya rendah, membuatnya ideal untuk lingkungan industri. Data yang dikirimkan oleh node sensor diterima oleh gateway pusat, kemudian diproses dan ditampilkan melalui sistem visualisasi atau dashboard. Bila terjadi kebocoran, sistem akan mengaktifkan dua respons otomatis: penutupan sumber gas melalui valve elektrik, dan pengaktifan sistem ventilasi darurat.

Keseluruhan kerangka ini membentuk dasar dari sistem deteksi dini berbasis IoT, yang diharapkan mampu meningkatkan keselamatan operasional dan memberikan solusi mitigasi risiko secara efisien.

2.2 Internet of Things (IoT)

Internet of Things (IoT) adalah konsep teknologi yang memungkinkan perangkat fisik seperti sensor, aktuator, dan mikrokontroler untuk saling terhubung melalui internet, guna melakukan pertukaran data dan kontrol secara otomatis serta real-time. Dalam sistem ini, setiap perangkat dapat mengumpulkan, memproses, dan mengirimkan data tanpa intervensi manusia secara langsung.

Menurut Hafiz & Candra (2021), perkembangan teknologi informasi dan komunikasi telah meningkatkan kebutuhan terhadap sistem informasi berbasis internet, salah satunya dengan memanfaatkan teknologi IoT. IoT dapat diterapkan dalam berbagai bidang, termasuk dalam sistem deteksi kebocoran gas LPG, untuk mencegah kebakaran dan bencana lainnya (Daru et al., 2021).

Arsitektur umum IoT terdiri dari tiga lapisan utama:

1. Perception Layer: Mengacu pada sensor atau perangkat pengumpul data (seperti MQ-2).
2. Network Layer: Bertugas mengirimkan data ke pusat pemrosesan menggunakan protokol komunikasi seperti WiFi atau LoRa.
3. Application Layer: Menganalisis dan menyajikan data kepada pengguna akhir melalui dashboard atau notifikasi. Penerapan IoT dalam industri membantu meningkatkan efisiensi, keamanan, dan pengambilan keputusan berbasis data, termasuk dalam mendeteksi kebocoran gas secara otomatis.

2.3 Deteksi Kebocoran Gas

Deteksi kebocoran gas merupakan aspek penting dalam upaya pencegahan kecelakaan dan perlindungan aset di lingkungan industri. Gas seperti metana (CH_4), propana (C_3H_8), dan karbon monoksida (CO) sangat berbahaya jika bocor tanpa terdeteksi karena sifatnya yang mudah terbakar atau beracun. Sensor gas merupakan komponen utama dalam sistem ini. Beberapa jenis sensor yang umum digunakan dalam sistem IoT antara lain:

- MQ-2: Mendeteksi LPG, metana, alkohol, propana, dan hidrogen.
- MQ-135: Mendeteksi gas beracun seperti amonia, benzena, dan asap.
- MQ-7: Khusus untuk mendeteksi karbon monoksida (CO).

Sistem deteksi bekerja berdasarkan pembacaan ambang batas konsentrasi gas. Jika nilai sensor melebihi nilai ambang yang telah ditentukan, maka sistem akan secara otomatis mengirimkan peringatan (melalui buzzer, notifikasi, atau SMS), dan dapat mengaktifkan respons darurat seperti mematikan aliran gas atau menyalakan kipas ventilasi. Evalina & Azis (2020) menyatakan bahwa sistem pendeteksi kebocoran gas dengan respons otomatis sangat penting untuk mencegah kebakaran akibat gas yang mudah terbakar, terutama di area dengan potensi risiko tinggi.

2.4 Sensor Gas MQ-2

Gambar 1. Sensor MQ-2

Sensor MQ-2 merupakan sensor berbasis semikonduktor yang dirancang untuk mendeteksi berbagai jenis gas yang mudah terbakar dan berbahaya, seperti LPG, metana (CH_4), hidrogen (H_2), karbon monoksida (CO), asap, serta alkohol. Sensor ini menghasilkan output berupa tegangan analog yang berubah sesuai dengan konsentrasi gas di udara, sehingga memungkinkan pembacaan nilai gas dalam satuan PPM (parts per million).

Sensor MQ-2 bekerja berdasarkan prinsip perubahan konduktivitas. Saat mendeteksi adanya gas, konduktivitas sensor akan meningkat seiring bertambahnya konsentrasi gas di sekitarnya. Output ini kemudian dapat dibaca oleh mikrokontroler untuk dianalisis lebih lanjut. Sensor ini memiliki sensitivitas tinggi terhadap gas hasil pembakaran dan asap, menjadikannya cocok digunakan dalam sistem deteksi kebocoran gas.

Menurut Mulyati (2018), MQ-2 sangat sensitif terhadap asap dan gas hasil pembakaran, dan mampu mendeteksi kebocoran gas melalui perubahan nilai konduktivitas. Selain itu, sensor ini dapat mengukur konsentrasi gas dalam rentang 300 hingga 10.000 ppm, serta mampu beroperasi pada suhu -20 hingga 50°C dengan konsumsi arus sekitar 150 mA pada tegangan 5V (Dan et al., 2014).

Dalam konteks penelitian ini, sensor MQ-2 digunakan sebagai komponen utama dalam mendeteksi keberadaan gas metana dan gas mudah terbakar lainnya, yang merupakan bagian penting dari sistem monitoring kebocoran gas berbasis Internet of Things (IoT).

2.5 Konversi Tegangan Sensor ke Konsentrasi Gas

2.5.1 Dasar Teori Estimasi PPM

Sensor MQ2 bekerja berdasarkan prinsip bahwa resistansi internal sensornya (Rs) berubah tergantung pada konsentrasi gas di sekitarnya. Nilai konsentrasi gas dalam satuan PPM (Parts Per Million) dapat diestimasi dari rasio Rs/R0, di mana R0 adalah resistansi sensor pada kondisi udara bersih.

Rumus dasar hubungan logaritmik antara Rs/R0 dan PPM:

$$(PPM) = a \cdot (R_s/R_0) + b$$

Atau

$$PPM = 10^{(a \cdot (R_s/R_0) + b)}$$

2.5.2 Nilai Parameter untuk Smoke (Asap)

Berdasarkan datasheet MQ2, nilai parameter untuk estimasi konsentrasi asap adalah:

- a = -0.45
- b = 1.4

2.5.3 Langkah-langkah Perhitungan

1. Baca tegangan sensor (Vout) dari pin analog:

$$V_{out} = \frac{ADC\_value}{4095} \times V_{in}$$

(Vin = 5V, ADC ESP32 = 12 bit = 0-4095)

2. Hitung resistansi Rs:

$$R_s = \left(\frac{V_{in} - V_{out}}{V_{out}}\right) \times R_L$$

RL adalah resistor beban (biasanya 10 kΩ)

3. Hitung rasio Rs/R0:

R0 diukur saat sensor MQ2 berada di udara bersih (tanpa gas). Biasanya R0 diset saat kalibrasi awal.

4. Hitung estimasi PPM:

$$PPM = 10^{(a \cdot (R_s/R_0) + b)}$$

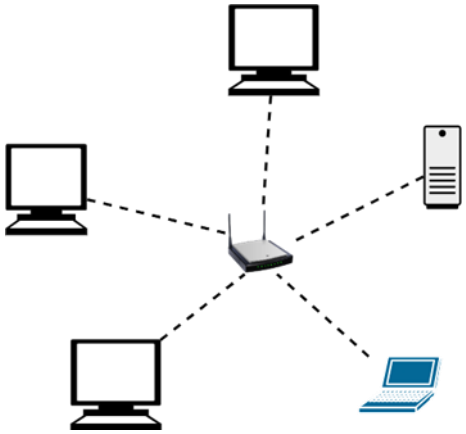
2.6 Teknologi LoRa dan Topologi Star

Gambar 2. LoRa

LoRa (Long Range) merupakan teknologi komunikasi nirkabel berdaya rendah yang dirancang untuk transmisi data jarak jauh dengan efisiensi energi yang tinggi. Teknologi ini beroperasi pada frekuensi sub-GHz, seperti 433 MHz atau 915 MHz, dan mampu menjangkau area hingga 10 kilometer tergantung pada kondisi lingkungan. LoRa sangat cocok digunakan dalam aplikasi Internet of Things (IoT), terutama pada sistem yang memerlukan konektivitas jarak jauh dengan konsumsi daya rendah.

LoRaWAN (LoRa Wide Area Network) adalah protokol komunikasi yang dibangun di atas teknologi LoRa. Protokol ini mengatur komunikasi antara end node (seperti sensor), gateway, dan server pusat. Arsitektur LoRaWAN terdiri atas tiga komponen utama, yaitu:

- End Node: Perangkat sensor atau aktuator yang mengumpulkan dan mengirimkan data.
- Gateway: Perangkat penerima yang menjembatani komunikasi antara banyak node dan server pusat.
- Network Server: Komponen yang mengatur lalu lintas data, autentikasi perangkat, dan keamanan jaringan.



Gambar 3. Star Topologi

Dalam penerapannya, LoRa biasanya dikombinasikan dengan topologi jaringan star (bintang), di mana seluruh node sensor terhubung langsung ke satu gateway pusat. Topologi ini memiliki beberapa keunggulan, seperti:

- Latensi rendah dan jalur komunikasi yang langsung,
- Kemudahan dalam pengelolaan jaringan,
- Minimnya interferensi antar node karena semua komunikasi difokuskan ke gateway.

Menurut Made Liandana (2019), LoRa mampu mengatasi interferensi dari jaringan lain karena frekuensinya yang bekerja di sub-GHz, serta mendukung konektivitas yang skalabel. Hal ini memungkinkan ribuan perangkat dapat terhubung secara bersamaan. Dalam konteks industri, LoRa banyak digunakan untuk pengumpulan data sensor dari lokasi-lokasi terpencil, seperti untuk pengukuran suhu, kelembaban, kualitas udara, maupun deteksi kebocoran gas. Data yang terkumpul dapat dikirimkan ke server pusat untuk dianalisis lebih lanjut sebagai dasar pengambilan keputusan.

Dalam penelitian ini, LoRa dan topologi star digunakan untuk membangun jaringan komunikasi antar node sensor gas yang tersebar di beberapa titik. Pendekatan ini memungkinkan sistem untuk mengirim data secara efisien dan menerima peringatan kebocoran gas secara real-time.

2.7 Sistem Monitoring Berbasis IoT

Sistem monitoring berbasis Internet of Things (IoT) merupakan pendekatan modern yang memungkinkan perangkat-perangkat fisik seperti sensor dan aktuator untuk terhubung melalui jaringan internet dan berkomunikasi secara real-time. Sistem ini memungkinkan pengguna untuk mengontrol, mengolah, dan memantau data dari berbagai lokasi secara langsung, baik melalui web maupun aplikasi mobile.

Menurut Amaro Najib (2017), sistem monitoring IoT memungkinkan integrasi berbagai perangkat seperti sensor, aktuator, komputer, dan ponsel pintar untuk menghasilkan informasi yang dapat dimanfaatkan oleh manusia atau sistem otomatis lainnya. Informasi yang dihasilkan harus relevan dan mengikuti prinsip SMART (Specific, Measurable, Attainable, Relevant, Time-bound), sehingga pemantauan dapat dilakukan secara akurat, tepat waktu, dan sesuai kebutuhan pengguna.

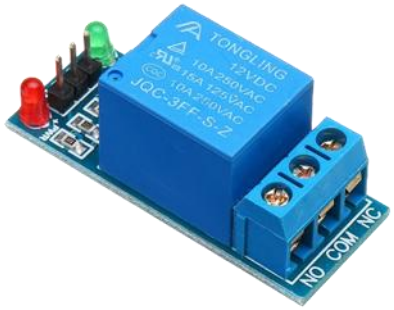
Komponen utama dari sistem monitoring berbasis IoT meliputi:

- Sensor: Berfungsi untuk mengukur kondisi fisik lingkungan seperti gas, suhu, dan kelembaban.
- Mikrokontroler: Bertugas mengolah data dari sensor dan mengatur jalur komunikasi antar perangkat.
- Modul Komunikasi: Mengirimkan data ke gateway atau server, biasanya menggunakan teknologi seperti Wi-Fi, LoRa, atau Bluetooth.
- Aktuator (misalnya relay): Melakukan tindakan otomatis seperti memutus aliran gas atau mengaktifkan sistem ventilasi saat terjadi kebocoran.
- Cloud Platform: Tempat penyimpanan data sekaligus media untuk analisis dan integrasi ke sistem lain.
- Dashboard Monitoring: Antarmuka visual berupa web atau aplikasi yang menampilkan data sensor secara real-time dan mempermudah pemantauan oleh operator.

Sistem ini juga dapat dilengkapi dengan fitur notifikasi, seperti pengiriman pesan SMS, email, atau aktivasi alarm fisik, untuk memberikan peringatan dini kepada operator saat terdeteksi kondisi berbahaya, seperti kebocoran gas. Dengan kemampuannya untuk memantau dan merespons secara otomatis, sistem

monitoring berbasis IoT menjadi solusi ideal dalam meningkatkan keselamatan dan efisiensi operasional di lingkungan industri.

2.8 Sistem Aktuator (Relay)



Gambar 4.Relay

Relay merupakan aktuator elektromekanis yang berfungsi sebagai saklar otomatis untuk menghubungkan atau memutus arus listrik dalam suatu rangkaian. Komponen ini memungkinkan sistem bertegangan rendah, seperti mikrokontroler, untuk mengontrol perangkat bertegangan tinggi seperti motor, kipas, sirine, atau sistem listrik lainnya.

Dalam konteks sistem monitoring kebocoran gas, relay memiliki peran penting dalam tindakan preventif otomatis, di antaranya:

- Mengaktifkan buzzer atau sirine saat kebocoran gas terdeteksi.
- Mengendalikan kipas atau ventilasi otomatis guna menurunkan konsentrasi gas berbahaya di udara.
- Memutus aliran listrik atau gas guna mencegah percikan api yang dapat menyebabkan ledakan.

Relay bekerja dengan prinsip elektromagnetik. Ketika arus listrik mengalir melalui kumparan, medan magnet yang terbentuk akan menarik kontaktor untuk mengubah posisi saklar dari *OFF* ke *ON* atau sebaliknya. Modul relay modern umumnya sudah dilengkapi dengan rangkaian driver, sehingga dapat langsung dikendalikan melalui sinyal digital dari mikrokontroler seperti ESP32.

Menurut Friansyah Ilham Gantar (2021), modul relay beroperasi berdasarkan prinsip elektromagnetik untuk menggerakkan kontaktor, yang memungkinkan perpindahan posisi dari ON ke OFF. Modul ini dirancang untuk bekerja berdasarkan sinyal dari mikrokontroler. Saat sinyal dalam keadaan logika tinggi (5 volt), perangkat yang terhubung (misalnya lampu) akan menyala. Sebaliknya, saat sinyal berada pada logika rendah (0 volt), perangkat tersebut akan mati. Dengan demikian, relay menjadi komponen penting dalam sistem otomatisasi yang membutuhkan kontrol terhadap perangkat bertegangan tinggi melalui sinyal logika rendah.

2.9 Mikrokontroler ESP32

Gambar 5 Mikrokontroler Esp32

ESP32 merupakan mikrokontroler modern berbasis System on Chip (SoC) yang terintegrasi dengan fitur Wi-Fi 802.11 b/g/n dan Bluetooth v4.2, serta dilengkapi dengan berbagai peripheral seperti ADC, DAC, PWM, dan antarmuka komunikasi serial (UART, I2C, dan SPI). Mikrokontroler ini memiliki performa tinggi dan konsumsi daya rendah, sehingga sangat sesuai untuk aplikasi Internet of Things (IoT).

Dalam sistem monitoring kebocoran gas berbasis IoT, ESP32 berperan sebagai pusat kendali (control unit) yang bertugas untuk:

- Membaca data dari sensor gas seperti MQ-2.
- Mengontrol aktuator seperti relay untuk tindakan otomatis.
- Mengirimkan data ke gateway LoRa atau ke platform cloud untuk pemantauan dan penyimpanan.
- Mengelola notifikasi atau respons sistem berbasis kondisi lingkungan.
- Keunggulan utama ESP32 dalam sistem ini antara lain:

- Mendukung modul komunikasi LoRa seperti LoRa RFM95 untuk pengiriman data jarak jauh.
- Memiliki kecepatan pemrosesan tinggi berkat prosesor dual-core hingga 240 MHz.
- Tersedia banyak GPIO yang dapat digunakan untuk menghubungkan sensor maupun aktuator.
- Kompatibel dengan berbagai lingkungan pemrograman seperti Arduino IDE, MicroPython, dan ESP-IDF.
- Dapat diintegrasikan dengan cloud platform melalui Wi-Fi serta mendukung komunikasi dengan API menggunakan protokol REST.

Menurut Agus Wagyana (2019), ESP32 merupakan mikrokontroler SoC yang cukup lengkap, dengan fitur prosesor, penyimpanan, serta akses ke GPIO. Mikrokontroler ini memungkinkan konektivitas langsung ke jaringan Wi-Fi tanpa memerlukan modul tambahan, menjadikannya solusi praktis untuk aplikasi IoT.

Sementara itu, Nizam Muhammad (2022) menambahkan bahwa board ESP32 hadir dalam dua versi, yaitu dengan 30 GPIO dan 36 GPIO. Versi 30 GPIO sering dipilih karena memiliki dua pin GND yang memudahkan penggunaan. Selain itu, board ESP32 memiliki antarmuka USB to UART dan mendukung pemrograman melalui Arduino IDE, sehingga sangat mudah digunakan dalam pengembangan sistem tertanam.

2.10 Platform Cloud IoT Adafruit IO

Gambar 6 Adafruit IO

Adafruit IO merupakan platform cloud IoT (Internet of Things) yang dirancang untuk memfasilitasi pemantauan dan kendali perangkat secara real-time melalui jaringan internet. Platform ini menyediakan antarmuka berbasis web dan mobile yang intuitif, sehingga sangat cocok digunakan dalam pengembangan sistem monitoring berbasis prototipe.

Fitur utama Adafruit IO meliputi:

- Feed: Menyimpan dan mengatur data sensor yang dikirim dari perangkat.
- Dashboard: Menyediakan visualisasi data dalam bentuk grafik, indikator status, tombol kontrol, hingga panel interaktif lainnya.
- Integrasi MQTT dan Webhook: Memungkinkan pengiriman dan penerimaan data dari mikrokontroler seperti ESP32 menggunakan protokol MQTT, REST API, atau Webhook.
- Notifikasi Otomatis: Sistem dapat dikonfigurasi untuk mengirimkan peringatan melalui email atau aplikasi tertentu saat nilai sensor melewati ambang batas yang ditentukan.

Adafruit IO mendukung integrasi langsung dengan perangkat ESP32, sehingga sangat mendukung sistem monitoring berbasis IoT dengan arsitektur ringan dan pengaturan yang mudah. Platform ini juga memungkinkan pengendalian aktuator dari jarak jauh melalui tombol digital di dashboard, serta pemantauan visual data sensor gas secara kontinu untuk mendeteksi anomali atau kebocoran.

Menurut Arijanto (2025), Adafruit IO merupakan antarmuka yang mempermudah pengguna dalam melakukan penelitian dan pengembangan sistem IoT, khususnya untuk memantau kondisi perangkat elektronik, seperti mengetahui apakah perangkat dalam kondisi hidup atau mati, serta memastikan statusnya sesuai dengan yang diharapkan.

2.11 Firebase Realtime Database

Gambar 7. Firebase

Firebase Realtime Database merupakan layanan basis data berbasis cloud yang disediakan oleh Google, yang memungkinkan penyimpanan dan sinkronisasi data secara real-time ke seluruh klien (perangkat) yang terhubung. Layanan ini sangat cocok digunakan dalam pengembangan sistem monitoring berbasis Internet of Things (IoT), termasuk sistem deteksi kebocoran gas.

Dalam konteks sistem monitoring gas berbasis IoT, Firebase Realtime Database dapat digunakan untuk:

- Menyimpan data pembacaan sensor gas secara waktu nyata, sehingga memudahkan pemantauan tren dan mendeteksi anomali secara langsung.
- Menghubungkan sistem backend untuk mengaktifkan notifikasi otomatis jika terdeteksi nilai gas melebihi ambang batas aman.
- Menyediakan API untuk integrasi ke dalam dashboard berbasis web maupun aplikasi mobile (Android/iOS).
- Mendukung akses data secara fleksibel dari berbagai perangkat, kapan saja dan di mana saja melalui internet.

Firebase tidak hanya menyediakan layanan Realtime Database, tetapi juga dilengkapi dengan fitur lain seperti Firebase Authentication, Cloud Storage, Cloud Messaging (untuk notifikasi), dan Hosting yang semakin memudahkan pengembangan aplikasi terintegrasi dan berskala besar.

Menurut George Richard Payara (2018), Firebase adalah API yang disediakan Google untuk penyimpanan dan penyelarasan data ke dalam aplikasi Android, iOS, atau web. Realtime Database adalah salah satu fasilitas yang menyimpan data ke database dan mengambil data darinya dengan sangat cepat. Firebase bukan hanya Realtime Database, tetapi jauh lebih dari itu, karena juga memiliki banyak fitur seperti authentication, database, storage, hosting, pemberitahuan, dan lain-lain.

Dengan kemampuan sinkronisasi cepat dan kemudahan integrasi, Firebase Realtime Database menjadi pilihan ideal untuk sistem monitoring gas yang membutuhkan respon cepat, akses data jarak jauh, dan keterhubungan dengan berbagai platform.

2.12 Penelitian Terkait

Penelitian-penelitian sebelumnya yang menjadi acuan dalam penelitian ini berkaitan dengan penggunaan teknologi LoRa, jaringan mesh, serta deteksi gas berbasis Internet of Things (IoT). Berikut adalah beberapa penelitian terkait yang relevan:

1. Perancanga*n Prototype* Perangkat Keras dan Perangkat Lunak Monitoring Polusi Udara di Kota Meda Berbasis *Internet of Things(IoT)*. Penelitian ini mengembangkan sistem pemantauan kualitas udara yang menggunakan sensor MQ-7 untuk mendeteksi konsentrasi gas karbon monoksida (CO). Data dari sensor dikirim melalui modul NodeMCU ESP8266 dan ditampilkan secara real- time pada platform web. Tujuan utama penelitian ini adalah menyediakan informasi mengenai tingkat polusi udara yang dapat diakses masyarakat untuk meningkatkan kesadaran tentang kualitas udara di lingkungan mereka.
2. Pengembangan Sistem Pemantauan Emisi Gas Karbon Berbasis *IoT* dengan Teknologi *LoRa* untuk Kendaraan Bermotor. Penelitian ini mengembangkan sistem pemantauan emisi gas dari kendaraan bermotor menggunakan LoRa. Data emisi dikirim secara real-time ke server pusat untuk membantu perusahaan memantau dan mengelola

emisi gas sesuai regulasi yang berlaku.

3. Implementasi Sistem Monitoring Emisi Gas pada Kendaraan Berbasis *IoT* Menggunakan *LoRa*. Penelitian ini berfokus pada pengembangan sistem untuk memantau emisi gas dari kendaraan menggunakan teknologi IoT dan LoRa. Sistem ini dirancang untuk mengirim data emisi secara real-time ke server pusat, memungkinkan analisis cepat terhadap tingkat emisi kendaraan dan memastikan kepatuhan terhadap peraturan emisi yang ditetapkan oleh pemerintah.
4. *S*istem Pendeteksian Dan Penanganan Kebocoran Gas Lpg Berbasis Iot. Penelitian ini bertujuan mengembangkan sistem deteksi kebocoran gas LPG menggunakan teknologi IoT. Sistem ini memanfaatkan sensor gas dan solenoid valve untuk secara otomatis menutup aliran gas saat terdeteksi kebocoran, serta mengirimkan notifikasi real-time ke perangkat mobile. Tujuan utama dari penelitian ini adalah untuk meningkatkan keselamatan dengan memberikan respons cepat terhadap kebocoran gas dan mencegah risiko ledakan atau kebakaran.
5. Rancang Bangun Sistem Komunikasi Sensor Nirkabel Pada Perangkat Pertanian Menggunakan Lora Dengan Topologi Mesh. Penelitian ini menggunakan jaringan LoRa Mesh untuk memantau kualitas tanah di lahan pertanian. Sistem ini memanfaatkan node sensor yang terhubung secara nirkabel untuk mengirimkan data kondisi tanah ke server pusat. Teknologi LoRa Mesh memungkinkan data dari area yang luas dikirimkan secara efisien, meskipun area tersebut jauh dari konektivitas internet, serupa dengan aplikasi pada perangkat pertanian lainnya

BAB III
METODOLOGI PENELITIAN

3.1 Alat dan Bahan

Penelitian ini menggunakan perangkat keras dan lunak untuk membangun sistem monitoring kebocoran gas berbasis Internet of Things (IoT) dengan topologi jaringan LoRa Star. Berikut daftar alat dan bahan yang digunakan:

Alat:

- Mikrokontroler ESP32 (sebagai node sensor dan gateway
- Modul LoRa RFM95 (untuk komunikasi nirkabel)
- Sensor gas MQ-2
- Breadboard, kabel jumper, dan power supply
- Laptop (untuk pemrograman, konfigurasi sistem, dan monitoring)
  Modul WiFi (bawaan ESP32)
- Multimeter dan solder (untuk pengujian dan perakitan)

Bahan:

- Enclosure untuk perlindungan komponen
- Akses ke platform Adafruit IO (visualisasi) dan Firebase Realtime Database (penyimpanan data)
- Software: Arduino IDE, Fritzing (desain skematik), dan browser untuk dashboard cloud

3.2 Desain Sistem

Penelitian ini berfokus pada perancangan sistem monitoring kebocoran gas berbasis Internet of Things (IoT) yang menggunakan jaringan komunikasi nirkabel LoRa dengan topologi Star. Sistem ini dirancang untuk membantu pemantauan kebocoran gas metana di area industri atau laboratorium secara real-time dan efisien. Fokus utama dari sistem ini adalah kestabilan komunikasi antar node sensor dan gateway, serta integrasi dengan platform cloud untuk penyimpanan dan visualisasi data.

Sistem terdiri dari beberapa node sensor yang masing-masing dilengkapi dengan sensor gas MQ-2 dan mikrokontroler ESP32. Sensor MQ-2 berfungsi untuk

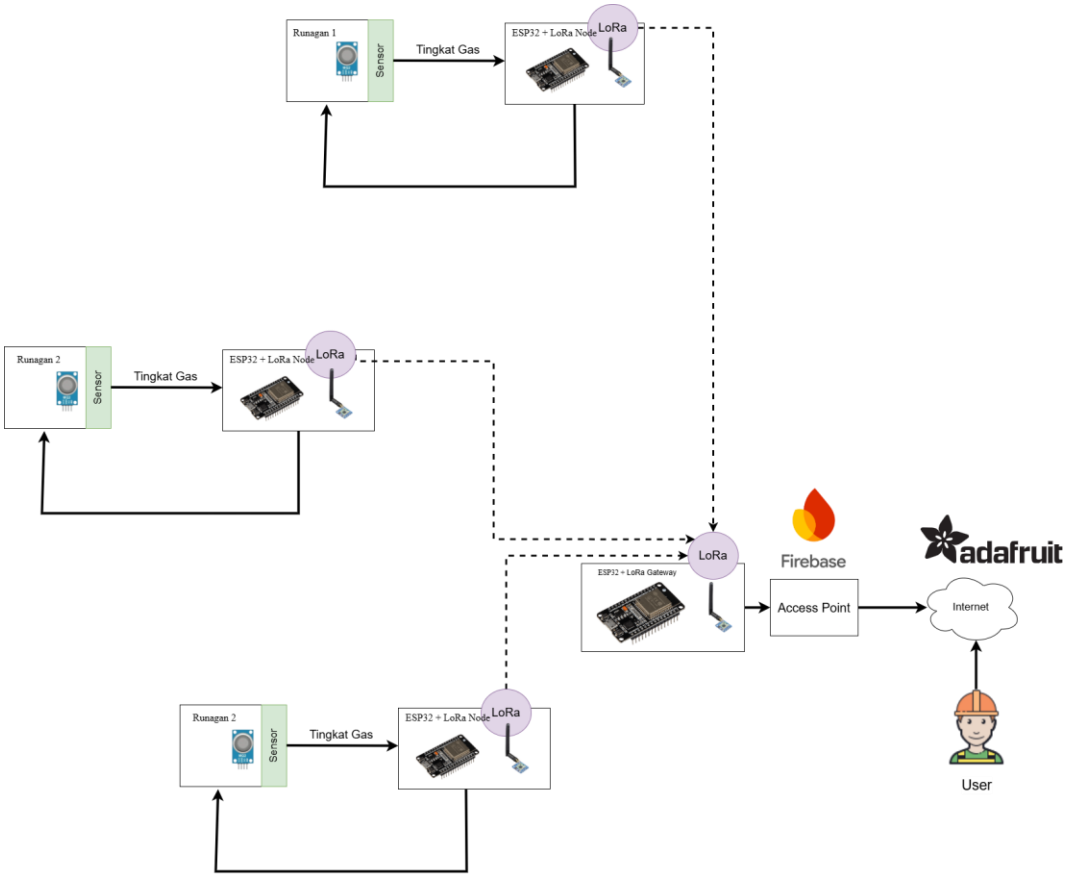
mendeteksi keberadaan gas metana di udara sekitar. Data yang dikumpulkan oleh sensor akan dibaca dan diproses oleh ESP32, lalu dikirim melalui modul LoRa RFM95 ke gateway pusat.

Gateway berperan sebagai penerima utama dari semua data yang dikirim oleh node-node sensor. Gateway ini juga menggunakan ESP32 dan LoRa RFM95, serta dilengkapi koneksi internet melalui Wi-Fi. Setelah data diterima oleh gateway, informasi konsentrasi gas dan waktu kejadian dikirim ke platform cloud seperti Adafruit IO untuk divisualisasikan secara real-time dan ke Firebase Realtime Database untuk disimpan sebagai log historis.

Sistem ini tidak hanya mampu mendeteksi peningkatan konsentrasi gas metana secara terus-menerus, tetapi juga memungkinkan pengguna untuk memantau status node secara jarak jauh melalui antarmuka dashboard berbasis web. Dengan pendekatan ini, sistem mendukung pengawasan kondisi gas secara menyeluruh tanpa harus berada di lokasi secara fisik.

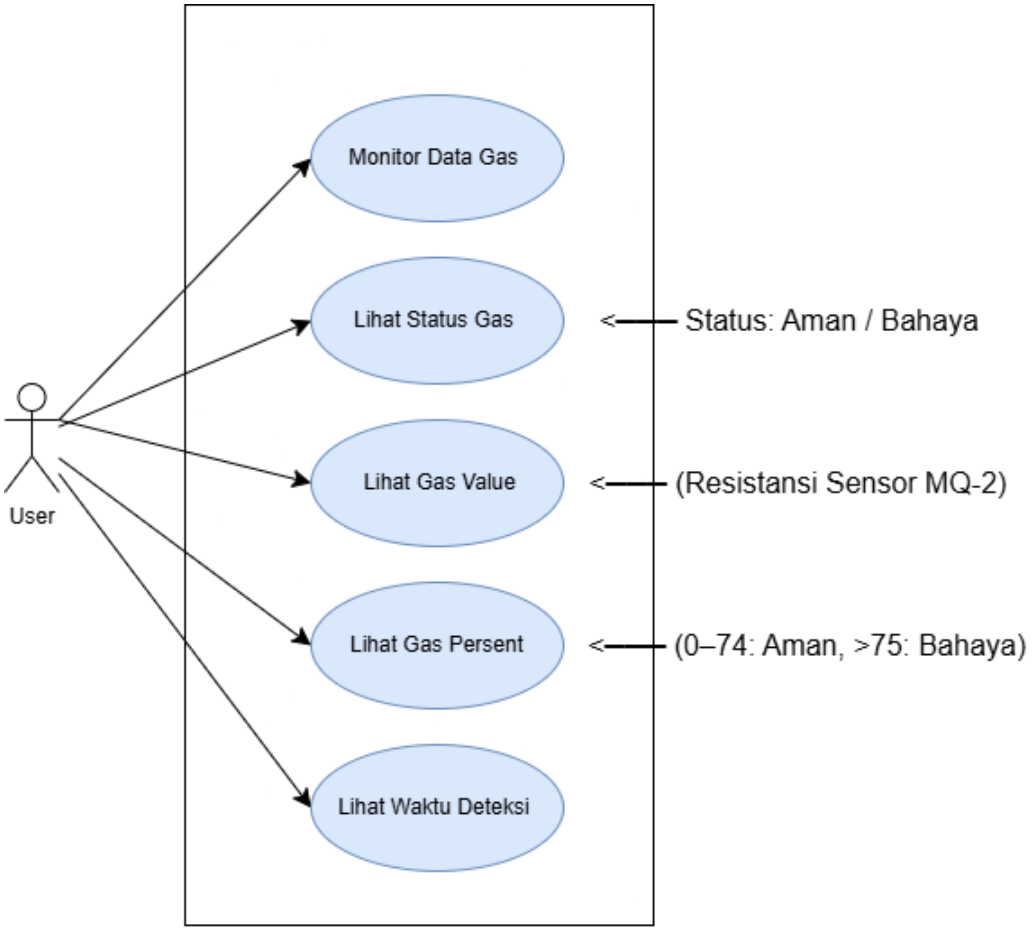
Sistem monitoring ini dapat diadaptasi untuk berbagai kebutuhan industri, baik skala kecil maupun besar, karena jaringan LoRa Star memungkinkan komunikasi data jarak jauh dengan konsumsi daya yang rendah. Node-node sensor dapat dengan mudah diperluas atau dipindahkan sesuai dengan perubahan kebutuhan pemantauan di area tertentu.

Melalui desain sistem yang berfokus pada pemantauan dan komunikasi data, penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan sistem pemantauan lingkungan berbasis IoT, khususnya untuk mendeteksi potensi bahaya kebocoran gas secara efisien dan terukur.



Gambar 8. Desain Skema Sistem Monitoring Kebocoran Gas

Gambar 8 menggambarkan arsitektur sistem monitoring kebocoran gas metana berbasis Internet of Things (IoT) yang menggunakan topologi LoRa Star. Setiap node terdiri dari sensor gas MQ-2 dan mikrokontroler ESP32 yang terhubung ke modul LoRa RFM95. Data dari sensor dikirimkan secara langsung ke satu gateway pusat yang juga dilengkapi dengan ESP32 dan modul LoRa. Gateway ini bertugas meneruskan data ke cloud melalui koneksi Wi-Fi. Platform Adafruit IO digunakan untuk menampilkan data secara real-time, sementara Firebase Realtime Database digunakan untuk menyimpan data historis. Desain ini memungkinkan pemantauan konsentrasi gas dari beberapa titik secara simultan dan efisien, serta memberikan gambaran visual yang jelas bagi pengguna untuk mengambil tindakan selanjutnya.



Gambar 9. Use Case Diagram

Gambar 9 Use Case Diagram menggambarkan interaksi antara pengguna (User) dengan sistem monitoring kebocoran gas berbasis IoT yang memanfaatkan topologi jaringan LoRa Star. Pengguna bertindak sebagai pengawas sistem yang dapat memantau dan mengakses data konsentrasi gas metana secara real-time melalui dashboard berbasis cloud. Node sensor yang tersebar di beberapa titik mendeteksi tingkat konsentrasi gas menggunakan sensor MQ-2, lalu mengirimkan data ke gateway pusat melalui komunikasi LoRa. Gateway kemudian meneruskan data tersebut ke layanan cloud seperti Adafruit IO untuk visualisasi dan Firebase Realtime Database untuk penyimpanan data historis. Pengguna dapat melihat informasi seperti gas value (resistansi gas), gas persent (gas level 0-74 maka status aman dan jika gas level > 75 maka statusnya bahaya) dan waktu deteksi. Diagram ini juga mencakup proses otomatisasi notifikasi jika nilai konsentrasi gas melebihi ambang batas yang telah ditentukan.

Seluruh proses ini mendukung pemantauan jarak jauh dan pengambilan keputusan cepat dalam situasi berisiko kebocoran gas.

Flowchart menggambarkan alur proses sistem mulai dari pendeteksian gas, pengiriman data melalui LoRa, hingga visualisasi cloud.

3.3 Pengumpulan dan Pengolahan Data

Data yang dikumpulkan:

- Nilai konsentrasi gas (PPM) per waktu
- Timestamp (waktu pengiriman data)
- ID node pengirim

Data tersebut:

- Disimpan secara real-time ke Firebase Realtime Database.
- Ditampilkan secara visual di Adafruit IO Dashboard.\\
- Dapat diekspor untuk analisis performa sistem seperti:
  - Konsistensi pengiriman data
  - Delay pengiriman dari node ke cloud
  - Validasi akurasi sensor dibanding data simulasi

3.4 Pengujian Sistem

Menilai akurasi pembacaan sensor MQ-2 terhadap sumber gas (misalnya korek api, LPG kecil).

3.5 Uji Komunikasi LoRa Star

- Mengukur jangkauan maksimal node ke gateway.
- Menganalisis keberhasilan pengiriman paket data.

3.6 Uji Respons Buzzer (opsional)

Mengamati apakah buzzer aktif saat nilai melebihi ambang batas.

3.7 Uji Integrasi Cloud

Memastikan data muncul secara real-time di Adafruit IO dan Firebase.

3.8 Parameter Keberhasilan

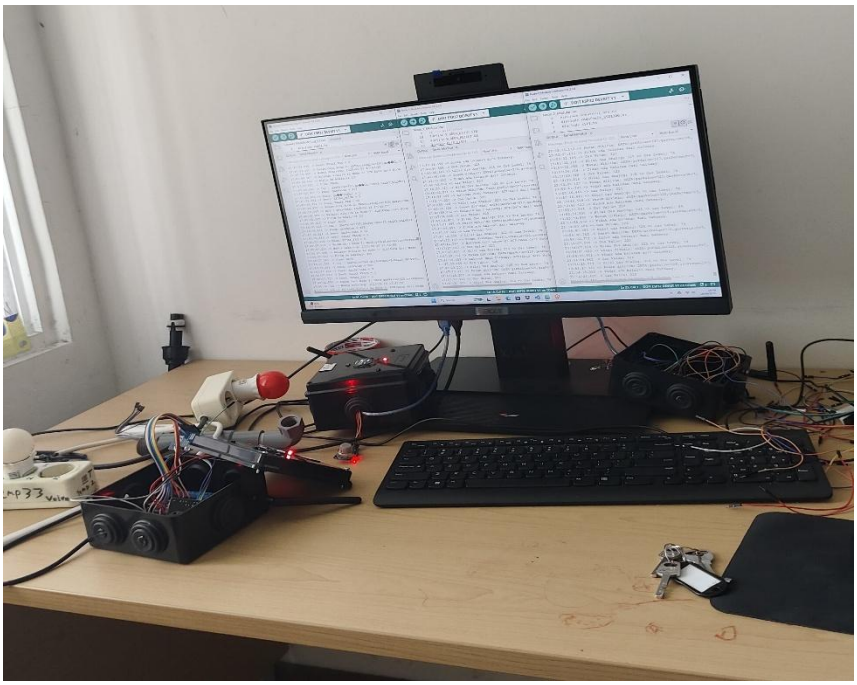
Sistem dianggap berhasil jika memenuhi indikator berikut:

Parameter	Target
Akurasi pembacaan sensor	≥ 90% terhadap nilai referensi
Keberhasilan pengiriman data LoRa	≥ 95% dari total paket dikirim
Visualisasi data di Adafruit IO	Real-time dan konsisten

Tabel 1. Parameter Keberhasilan

BAB IV
HASIL DAN PEMBAHASAN

4.1 Hasil Perancangan Sistem



Gambar 10. Sistem Topologi Star

Pada tahap ini, sistem monitoring kebocoran gas telah berhasil dibangun dan diuji secara bertahap. Perancangan sistem terdiri dari perangkat keras dan perangkat lunak yang telah terintegrasi untuk menjalankan fungsi deteksi, pengiriman data, serta pemberian peringatan dini

4.1.1 Hasil Rangkaian Perangkat Keras

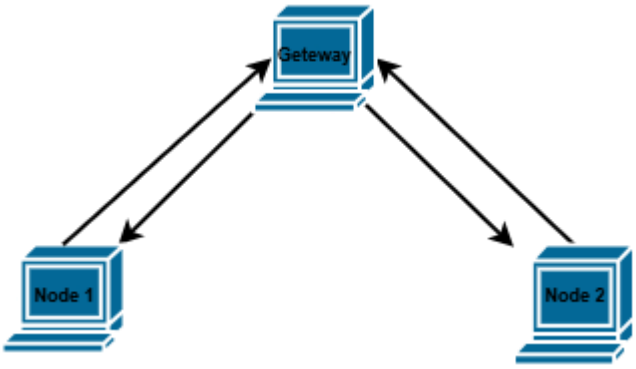
Perangkat keras sistem terdiri dari beberapa komponen utama yaitu:

- Node sensor: ESP32 + sensor gas MQ-2 + layar OLED + modul LoRa RFM95
- Gateway: ESP32 + modul LoRa RFM95 + koneksi WiFi ke Adafruit IO
- Aktuator: Relay yang mengontrol fan dan valve untuk respons kebocoran
- Indikator LED: Merah untuk menandai aktifnya valve, putih untuk fan

Perangkat telah dirakit di atas breadboard dan diuji satu per satu untuk memastikan semua berfungsi normal.

(Foto-foto node sensor, gateway, rangkaian relay, dan lampu indikator ditampilkan di sini)

4.1.2 **Hasil Pengembangan Perangkat** Lunak



Gambar 11. Komunikasi Topologi Star yang berpusat disatu gateway

Perangkat lunak dikembangkan menggunakan Arduino IDE dengan bahasa pemrograman C++. Program berjalan pada mikrokontroler ESP32 sebagai gateway LoRa dan penghubung ke platform cloud. Perangkat lunak ini mencakup berbagai fungsi utama, mulai dari inisialisasi koneksi WiFi, sinkronisasi waktu, koneksi ke Adafruit IO, pembacaan data dari sensor gas, komunikasi antar node menggunakan LoRa, hingga pengiriman data ke platform cloud dan aktivasi perangkat output seperti relay.

Berikut adalah penjelasan tiap fungsionalitas berdasarkan hasil implementasi:

1. Rancang Bangun Sistem Komunikasi Sensor Nirkabel Pada Perangkat Pertanian Menggunakan Lora Dengan Topologi Mesh. Penelitian ini menggunakan jaringan LoRa Mesh untuk memantau kualitas tanah di lahan pertanian. Sistem ini memanfaatkan node sensor yang terhubung secara nirkabel untuk mengirimkan data kondisi tanah ke server pusat. Teknologi LoRa Mesh memungkinkan data dari area yang luas dikirimkan secara efisien, meskipun area tersebut jauh dari konektivitas internet, serupa dengan aplikasi pada perangkat pertanian lainnya

Kode program untuk proses inisialisasi koneksi WiFi ditunjukkan pada Gambar 12.

```
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
Serial.print("Menghubungkan WiFi");
while (WiFi.status() != WL_CONNECTED) {
  Serial.print(".");
  delay(500);
}
Serial.println("\nWiFi terhubung!");
```

Gambar 12. Cuplikan kode inisialisasi koneksi WiFi pada ESP32

Setelah koneksi berhasil, ESP32 mencetak status koneksi pada Serial Monitor sebagai bukti bahwa perangkat telah tersambung ke jaringan. Contoh hasil keluaran program saat koneksi WiFi berhasil ditampilkan pada Gambar 13.

```
Menghubungkan WiFi...
WiFi terhubung!
```

Gambar 13. Tampilan Serial Monitor saat proses inisialisasi koneksi WiFi berhasil

2. Sinkronisasi Waktu (NTP)

Setelah berhasil terkoneksi ke WiFi, sistem melakukan sinkronisasi waktu dengan server NTP (Network Time Protocol). Sinkronisasi ini penting agar waktu pengiriman data dan pencatatan log memiliki timestamp yang akurat.

```
configTime(8 * 3600, 0, "id.pool.ntp.org", "pool.ntp.org");
delay(1000); // Beri waktu sinkronisasi
syncTime();  // Sinkronkan waktu
```

Gambar 14. Cuplikan kode untuk sinkronisasi waktu menggunakan server NTP

```
Menyinkronkan waktu...
Waktu tersinkron!
Waktu lokal sekarang: 2025-06-16 17:08:34
```

Gambar 15. Output Serial Monitor saat sinkronisasi waktu berhasil

3. Koneksi ke Adafruit IO

Setelah sinkronisasi waktu berhasil, sistem melanjutkan dengan menginisialisasi koneksi ke layanan Adafruit IO yang digunakan sebagai cloud platform untuk menyimpan dan memvisualisasikan data sensor secara real-time.

```
io.connect();
Serial.print("Menghubungkan ke Adafruit IO");
while (io.status() < AIO_CONNECTED) {
  Serial.print(".");
  delay(500);
}
Serial.println("\nAdafruit IO Terhubung!");
```

Gambar 16. Cuplikan kode koneksi ke Adafruit IO

```
Menghubungkan ke Adafruit IO...
Adafruit IO Terhubung!
```

Gambar 17. Output Serial Monitor saat koneksi ke Adafruit IO berhasil

4. Inisialisasi Gateway LoRa

Gateway memulai inisialisasi LoRa dengan konfigurasi pin yang sesuai. LoRa digunakan untuk komunikasi nirkabel antar node sensor dengan gateway.

```
LoRa.setPins(5, 14, 2); // NSS, RST, DIO0
if (!LoRa.begin(915E6)) {
  Serial.println("Starting LoRa failed!");
  while (1);
}
Serial.println("LoRa Gateway Starting...");
```

Gambar 18. Cuplikan kode inisialisasi LoRa

Output Serial Monitor

```
LoRa Gateway Starting...
```

Gambar 19. Output Serial Monitor saat LoRa Gateway diinisialisasi

5. Komunikasi Antar Node ke Gateway

Setelah inisialisasi LoRa berhasil, Gateway mulai mendengarkan paket data yang dikirim oleh node sensor. Proses ini dilakukan dalam fungsi loop(), di mana Gateway memantau setiap paket masuk menggunakan LoRa.parsePacket(). Jika ada data yang masuk, Gateway membaca alamat penerima (recipient), alamat pengirim (sender), dan isi pesan.

Potongan kode berikut menunjukkan bagaimana pesan dibaca dari node:

```
uint8_t recipient = LoRa.read();
uint8_t sender = LoRa.read();
String incoming = "";
while (LoRa.available()) {
  incoming += (char)LoRa.read();
}
```

Gambar 20. *Cuplikan kode pembacaan data dari LoRa:*

Apabila alamat penerima cocok dengan alamat Gateway, maka data dianggap valid. Sistem kemudian mencetak pesan dan waktu saat data diterima.

```
Pesan dari Node 1: DATA|gasValue=295, gasPercent=7, RELAY_VALVE=1
Waktu sekarang: 2025-06-16 17:08:43
```

Gambar 21. Contoh output komunikasi antar node dan gateway

Setelah itu, Gateway mengirimkan balasan ke Node berupa ACK (acknowledgment) dan melanjutkan untuk mengirimkan data ke platform cloud (Adafruit IO).

```
kirimBalasan(sender, "ACK|Data dari Node " + String(sender, HEX) + " diterima");
```

Gambar 22. Cuplikan kode balasan ke node

6. Pengiriman Data ke Adafruit IO

Data sensor yang diterima dari node akan dikirim ke feed Adafruit IO. Setiap key seperti gasValue, gasPercent, RELAY_VALVE, dan RELAY_FAN akan dikirim ke feed-nya masing-masing.

```
io.feed(key.c_str())->save(val.toFloat());
```

Gambar 23. *Cuplikan kode untuk mengirim data ke Adafruit IO*

Output:

```
08:02:58.596 -> Kirim ke Adafruit IO:
08:02:58.629 -> Tipe: DATA
08:02:58.629 -> Isi : gasValue=2655,gasPercent=64,RELAY_VALVE=
08:02:58.692 -> Feed: gasValue = 2655
08:02:58.725 -> Feed: gasPercent = 64
08:02:58.725 -> Feed: RELAY_VALVE = 1
08:02:58.756 -> Feed: RELAY_FAN = 0
```

Gambar 24. *Output Serial Monitor saat data dikirim ke Adafruit IO*

7. Output Relay dan LED Indikator

Setiap kali data diterima oleh gateway dari node sensor, sistem secara otomatis mengaktifkan LED indikator pada pin-pin tertentu di ESP32 sebagai umpan balik bahwa komunikasi berhasil dilakukan. LED ini juga berfungsi sebagai indikator aktivitas masing-masing node.

- LED_PWR (merah): Menyala saat sistem aktif (terhubung ke listrik).
- LED_NODE_1 (hijau): Menyala sesaat saat gateway menerima data dari Node 1.
- LED_NODE_2 (biru): Menyala sesaat saat gateway menerima data dari Node 2.

LED indikator ini membantu pemantauan visual langsung di perangkat keras, terutama saat dilakukan debugging atau pengujian di lapangan. Selain LED, sistem juga mengaktifkan perangkat aktuator berupa relay berdasarkan nilai konsentrasi gas. Jika nilai gasPercent melebihi ambang batas 75%, maka sistem akan:

- Mengaktifkan RELAY_VALVE (membuka atau menutup katup solenoid untuk menghentikan pasokan gas).
- Mengaktifkan RELAY_FAN (menyalakan kipas buangan untuk ventilasi darurat).

Aktivasi relay dicatat dan dikirim ke Adafruit IO untuk ditampilkan secara real-time. Misalnya, jika gasPercent = 85%, maka pada feed cloud akan tampil:

- RELAY_VALVE = 1
- RELAY_FAN = 1

Hal ini dapat dilihat langsung melalui dashboard Adafruit IO dan serial monitor ESP32.

Contoh output Serial Monitor saat booting dan saat relay aktif:

```
Menghubungkan WiFi.....
WiFi terhubung!
Menyinkronkan waktu...
Waktu tersinkron!
Waktu lokal sekarang: 2025-06-18 17:08:34
Menghubungkan ke Adafruit IO.....
Adafruit IO Terhubung!
LoRa Gateway Starting...
```

Gambar 25. Serial Monitor saat booting dan saat relay aktif

Cuplikan saat data diterima dan relay aktif:

```
Pesan dari Node 1: DATA|gasValue=3650,gasPercent=89,RELAY_VALVE=1,RELAY_FAN=1
Waktu sekarang: 2025-06-18 17:12:43
Balasan dikirim ke Node 1: ACK|Data dari Node 1 diterima
Kirim ke Adafruit IO:
Tipe: DATA
Isi : gasValue=3650,gasPercent=89,RELAY_VALVE=1,RELAY_FAN=1
Feed: gasValue = 3650
Feed: gasPercent = 89
Feed: RELAY_VALVE = 1
Feed: RELAY_FAN = 1
```

Gambar 26. Tampilan Serial Monitor dan Lampu Indikator Saat Relay Aktif

Gambar 26 Tampilan Serial Monitor dan Lampu Indikator Saat Relay Aktif (Foto menunjukkan LED_NODE_1 menyala, dan output lampu merah/putih menyala menandakan kipas dan valve aktif)

Dengan adanya output visual berupa LED dan relay yang terhubung ke lampu indikator, pengguna dapat langsung melihat status sistem tanpa perlu membuka dashboard online. Integrasi antara pembacaan sensor, respons perangkat aktuator, dan pelaporan cloud menjadikan sistem ini efisien untuk pemantauan kondisi gas secara lokal maupun jarak jauh.

4.2 Pengujian Sistem

4.2.1 Pengujian Sensor Gas

Gambar 27. Node Sensor

Pengujian sensor gas dilakukan untuk memastikan bahwa sensor MQ-2 mampu mendeteksi keberadaan gas dengan akurat dan mengubah data analog menjadi nilai digital yang dapat diolah oleh mikrokontroler.

Sensor MQ-2 mengeluarkan nilai analog (gasValue) yang mencerminkan tingkat konsentrasi gas (dalam satuan ADC: 0–4095). Nilai tersebut dikonversi ke dalam persentase gas (gasPercent) berdasarkan skala ambang batas tertentu yang ditentukan melalui pengujian kalibrasi.

Dalam sistem ini, status gas ditentukan berdasarkan nilai gasPercent:

- gasPercent < 75% → Status: Aman
- gasPercent ≥ 75% → Status: Bahaya

Contoh output yang muncul pada Serial Monitor:

```
gasValue = 295
gasPercent = 7%
Status: Aman
```

Gambar 28. Output Serial Monitor

Tampilan ini juga ditampilkan pada layar OLED mini yang terpasang di node sensor, dengan format berikut:

```
[ OLED DISPLAY ]
Gas: 295
Kadar: 7%
Status: Aman
```

Gambar 29. layar OLED

Jika nilai gasPercent meningkat hingga mencapai atau melebihi 75%, sistem akan memberikan status:

```
gasValue = 3650
gasPercent = 89%
Status: Bahaya
```

Gambar 30. nilai gasPercent

Dan tampilan OLED:

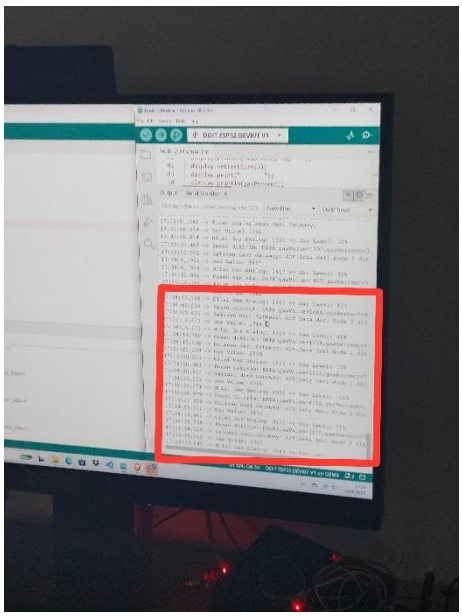
```
[ OLED DISPLAY ]
Gas: 3650
Kadar: 89%
Status: Bahaya
```

Gambar 31. Tampilan Display OLED

Gambar 32 di atas menunjukkan cuplikan layar OLED dan hasil Serial Monitor saat sistem membaca data gas di kondisi normal (aman).

Gambar 32. Tampilan OLED dan Serial Monitor Saat Kondisi Aman
(gambar OLED menampilkan gasValue=295 dan gasPercent=7%)

Gambar menunjukkan kondisi ketika gasPercent di atas ambang batas dan sistem menandai status sebagai “Bahaya”:



Gambar 33. Monitor Note dari OLED

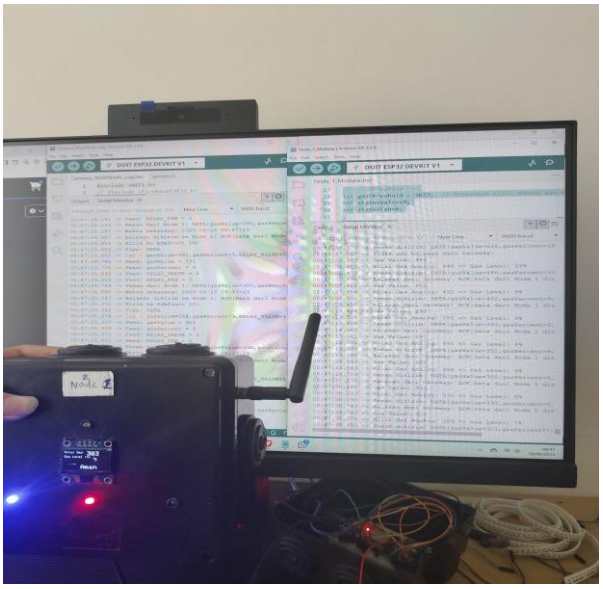
Gambar OLED menampilkan gasValue=4095 dan gasPercent=100% yang ditampilkan pada serial monitor

4.2.2 Pengujian Komunikasi LoRa

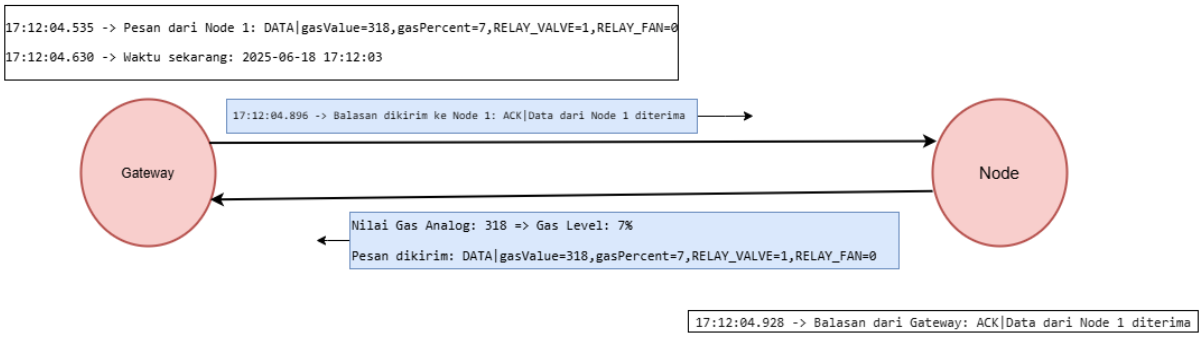
1. Komunikasi 1 Node dan 1 Gateway

Pengujian ini bertujuan untuk mengevaluasi kemampuan Gateway dalam menangani komunikasi dua arah dengan lebih dari satu Node sensor secara bergantian menggunakan protokol komunikasi LoRa dalam topologi star. Pada topologi ini, semua node sensor terhubung langsung ke Gateway sebagai pusat komunikasi tanpa inter-node communication.

- Memastikan komunikasi dua arah antara gateway dan masing-masing node sensor.
- Verifikasi pengiriman data gas dari masing-masing node ke gateway..
- Memastikan gateway dapat membalas ACK ke setiap node secara terpisah.



Gambar 34. Komunikasi *Hardware* 1 Node dan 1 Gateway



Gambar 35. Komunikasi 1 Node dan 1 Gateway

2. Node 1 mengirim data (17:12:04.335) ke Gateway.

```
17:12:04.335 -> Gas Value: 318
17:12:04.368 -> Nilai Gas Analog: 318 => Gas Level: 7%
17:12:04.438 -> Pesan dikirim: DATA|gasValue=318,gasPercent=7,RELAY_VALVE=1,RELAY_FAN=0
17:12:04.928 -> Balasan dari Gateway: ACK|Data dari Node 1 diterima
```

Gambar 36.Serial Monitor Node 1

3. Gateway menerima data dari Node 1 pada 17:12:04.535:

```
17:12:04.535 -> Pesan dari Node 1: DATA|gasValue=318,gasPercent=7,RELAY_VALVE=1,RELAY_FAN=0
17:12:04.630 -> Waktu sekarang: 2025-06-18 17:12:03
```

Gambar 37. Serial Gateway

4. Setelah gateway menerima data dari node 1. Gateway mengirim balasan ACK ke node:

```
17:12:04.535 -> Pesan dari Node 1: DATA|gasValue=318,gasPercent=7,RELAY_VALVE=1,RELAY_FAN=0
17:12:04.630 -> Waktu sekarang: 2025-06-18 17:12:03
17:12:04.896 -> Balasan dikirim ke Node 1: ACK|Data dari Node 1 diterima
```

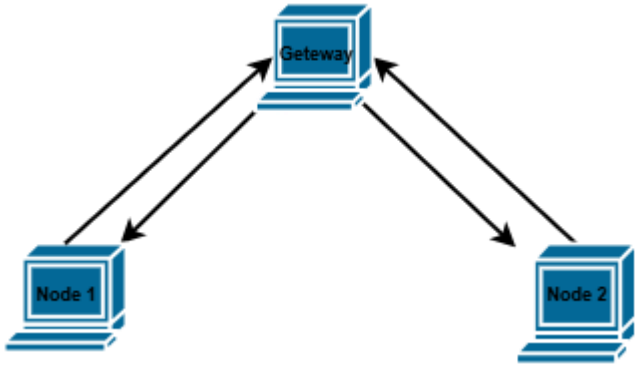
Gambar 38. Gambar Serial Gateway

8. Kemudian Node 1 menerima balasan dari Gateway jika data dari node 1 diterima:

```
17:12:04.368 -> Nilai Gas Analog: 318 => Gas Level: 7%
17:12:04.438 -> Pesan dikirim: DATA|gasValue=318,gasPercent=7,RELAY_VALVE=1,RELAY_FAN=0
17:12:04.928 -> Balasan dari Gateway: ACK|Data dari Node 1 diterima
```

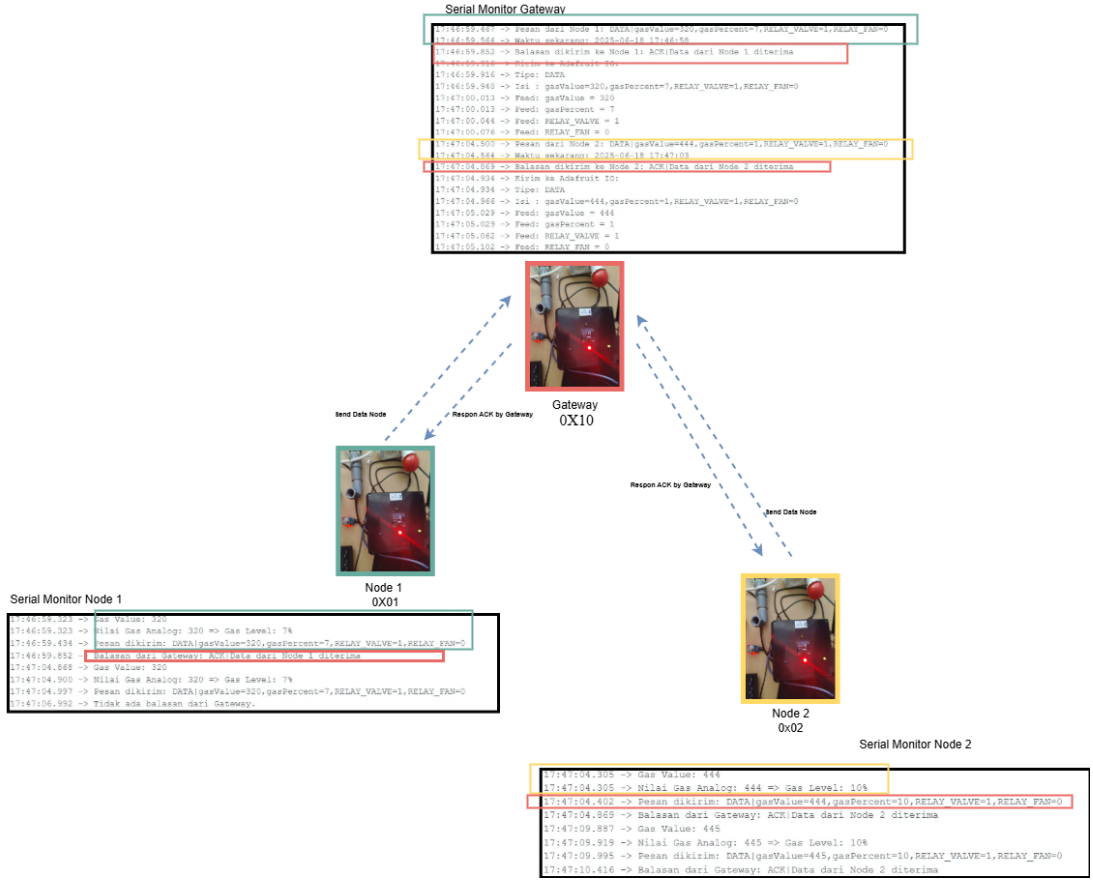
Gambar 39.Serial Monitor Node 1

Pengujian Komunikasi Topologi Star (1 Gateway dan 2 Node Sensor)



Gambar 40. Skema Komunikasi Topologi Star

Pengujian ini dilakukan untuk mengevaluasi komunikasi antara satu buah gateway dan dua node sensor berbasis ESP32 dalam konfigurasi jaringan LoRa dengan topologi star. Tujuannya adalah untuk memastikan bahwa kedua node dapat mengirim data secara independen, dan gateway mampu menerima, memproses, memberikan balasan (ACK), serta meneruskan data ke platform cloud (Adafruit IO) tanpa konflik atau kehilangan data.



Gambar 41. Pengujian Komunikasi Topologi Star (1 Gateway dan 2 Node Sensor)

Skema topologi jaringan dalam pengujian ini terdiri atas:

- Gateway dengan ID LoRa = 0x10
- Node 1 dengan ID LoRa = 0x01
- Node 2 dengan ID LoRa = 0x02

Langkah-langkah Pengujian:

1. Node 1 dan Node 2 secara periodik mengukur konsentrasi gas menggunakan sensor MQ-2, lalu mengirimkan data ke gateway dalam format: DATA|gasValue=xxx,gasPercent=yy,RELAY_VALVE=1,RELAY_FAN=0
2. Gateway menerima data dan mencetak pesan yang diterima melalui serial monitor. Selanjutnya gateway mengirim balasan berupa: ACK|Data dari Node [ID] diterima
3. Kemudian, data diteruskan ke platform Adafruit IO sesuai dengan masing-masing feed.
4. Masing-masing node mencetak balasan yang diterima dari gateway pada serial monitor.

5. Data dari node yang diterima gateway juga diproses dan dikirim ke cloud secara real-time, mencakup feed:
   - gasValue
   - gasPercent
   - RELAY_VALVE
   - RELAY_FAN

Contoh Hasil Serial Monitor

Berikut adalah hasil komunikasi yang diamati dari masing-masing perangkat

➤Serial Monitor Gateway – Komunikasi dengan Node 1 dan Node 2

```
17:46:59.467 -> Pesan dari Node 1: DATA|gasValue=320,gasPercent=7,RELAY_VALVE=1,RELAY_FAN=0
17:46:59.564 -> Waktu sekarang: 2025-06-18 17:46:58
17:46:59.852 -> Balasan dikirim ke Node 1: ACK|Data dari Node 1 diterima
17:46:59.916 -> Kirim ke Adafruit IO:
17:46:59.916 -> Tipe: DATA
17:46:59.948 -> Isi : gasValue=320,gasPercent=7,RELAY_VALVE=1,RELAY_FAN=0
17:47:00.013 -> Feed: gasValue = 320
17:47:00.013 -> Feed: gasPercent = 7
17:47:00.044 -> Feed: RELAY_VALVE = 1
17:47:00.076 -> Feed: RELAY_FAN = 0
17:47:04.500 -> Pesan dari Node 2: DATA|gasValue=444,gasPercent=1,RELAY_VALVE=1,RELAY_FAN=0
17:47:04.564 -> Waktu sekarang: 2025-06-18 17:47:03
17:47:04.869 -> Balasan dikirim ke Node 2: ACK|Data dari Node 2 diterima
17:47:04.934 -> Kirim ke Adafruit IO:
17:47:04.934 -> Tipe: DATA
17:47:04.966 -> Isi : gasValue=444,gasPercent=1,RELAY_VALVE=1,RELAY_FAN=0
17:47:05.029 -> Feed: gasValue = 444
17:47:05.029 -> Feed: gasPercent = 1
17:47:05.062 -> Feed: RELAY_VALVE = 1
17:47:05.102 -> Feed: RELAY_FAN = 0
```

Gambar 42. Serial Monitor Gateway – Komunikasi dengan Node 1 dan Node 2
Menampilkan pesan masuk dari Node 1 dan Node 2 secara bergantian, proses pengiriman balasan ACK, serta hasil pengiriman data ke Adafruit IO.

➤ Serial Monitor Node 1 – Pengiriman dan Balasan dari Gateway

```
17:46:59.323 -> Gas Value: 320
17:46:59.323 -> Nilai Gas Analog: 320 => Gas Level: 7%
17:46:59.434 -> Pesan dikirim: DATA|gasValue=320,gasPercent=7,RELAY_VALVE=1,RELAY_FAN=0
17:46:59.852 -> Balasan dari Gateway: ACK|Data dari Node 1 diterima
```

Gambar 43. Serial Monitor Node 1 – Pengiriman dan Balasan dari Gateway
Node 1 berhasil mengirim data, menerima ACK, serta menampilkan data gas yang terukur.

➤ Serial Monitor Node 2 – Komunikasi Dua Arah

```
17:47:04.305 -> Gas Value: 444
17:47:04.305 -> Nilai Gas Analog: 444 => Gas Level: 10%
17:47:04.402 -> Pesan dikirim: DATA|gasValue=444,gasPercent=10,RELAY_VALVE=1,RELAY_FAN=0
17:47:04.869 -> Balasan dari Gateway: ACK|Data dari Node 2 diterima
17:47:09.887 -> Gas Value: 445
17:47:09.919 -> Nilai Gas Analog: 445 => Gas Level: 10%
17:47:09.995 -> Pesan dikirim: DATA|gasValue=445,gasPercent=10,RELAY_VALVE=1,RELAY_FAN=0
17:47:10.416 -> Balasan dari Gateway: ACK|Data dari Node 2 diterima
```

Gambar 44. Serial Monitor Node 2 – Komunikasi Dua Arah
Node 2 juga berhasil mengirim data dan menerima ACK.

Analisis Hasil:

- Kedua node dapat mengirim data ke gateway tanpa mengalami konflik (data collision).
- Gateway berhasil memproses dan memberikan balasan ACK untuk masing-masing node.
- Data dari kedua node ditampilkan dengan benar di Serial Monitor Gateway, menunjukkan bahwa sistem berjalan stabil.
- Node 1 sempat mengalami kasus “Tidak ada balasan dari Gateway” pada pukul 17:47:06, yang menandakan kemungkinan gangguan sesaat atau antrian komunikasi.

Kesimpulan:
Pengujian komunikasi topologi star menunjukkan bahwa sistem mampu menangani komunikasi dua arah antara satu gateway dan dua node secara simultan. Respons gateway terhadap kedua node cukup cepat dan data berhasil dikirim ke platform cloud dengan akurat. Sistem dinilai andal untuk implementasi pada lingkungan nyata dengan banyak node sensor.

Jika kamu sudah punya foto tangkapan layar untuk serial monitor dan dashboard cloud, beri nama seperti:

Gambar 45. Serial Monitor Gateway

```
17:46:59.467 -> Pesan dari Node 1: DATA|gasValue=320,gasPercent=7,RELAY_VALVE=1,RELAY_FAN=0
17:46:59.564 -> Waktu sekarang: 2025-06-18 17:46:58
17:46:59.852 -> Balasan dikirim ke Node 1: ACK|Data dari Node 1 diterima
17:46:59.916 -> Kirim ke Adafruit IO:
17:46:59.916 -> Tipe: DATA
17:46:59.948 -> Isi : gasValue=320,gasPercent=7,RELAY_VALVE=1,RELAY_FAN=0
17:47:00.013 -> Feed: gasValue = 320
17:47:00.013 -> Feed: gasPercent = 7
17:47:00.044 -> Feed: RELAY_VALVE = 1
17:47:00.076 -> Feed: RELAY_FAN = 0
17:47:04.500 -> Pesan dari Node 2: DATA|gasValue=444,gasPercent=1,RELAY_VALVE=1,RELAY_FAN=0
17:47:04.564 -> Waktu sekarang: 2025-06-18 17:47:03
17:47:04.869 -> Balasan dikirim ke Node 2: ACK|Data dari Node 2 diterima
17:47:04.934 -> Kirim ke Adafruit IO:
17:47:04.934 -> Tipe: DATA
17:47:04.966 -> Isi : gasValue=444,gasPercent=1,RELAY_VALVE=1,RELAY_FAN=0
17:47:05.029 -> Feed: gasValue = 444
17:47:05.029 -> Feed: gasPercent = 1
17:47:05.062 -> Feed: RELAY_VALVE = 1
17:47:05.102 -> Feed: RELAY_FAN = 0
```

Gambar 46. Serial Monitor Gateway

Gambar 47 Serial Monitor Node 1

```
17:46:59.323 -> Gas Value: 320
17:46:59.323 -> Nilai Gas Analog: 320 => Gas Level: 7%
17:46:59.434 -> Pesan dikirim: DATA|gasValue=320,gasPercent=7,RELAY_VALVE=1,RELAY_FAN=0
17:46:59.852 -> Balasan dari Gateway: ACK|Data dari Node 1 diterima
17:47:04.868 -> Gas Value: 320
17:47:04.900 -> Nilai Gas Analog: 320 => Gas Level: 7%
17:47:04.997 -> Pesan dikirim: DATA|gasValue=320,gasPercent=7,RELAY_VALVE=1,RELAY_FAN=0
17:47:06.992 -> Tidak ada balasan dari Gateway.
```

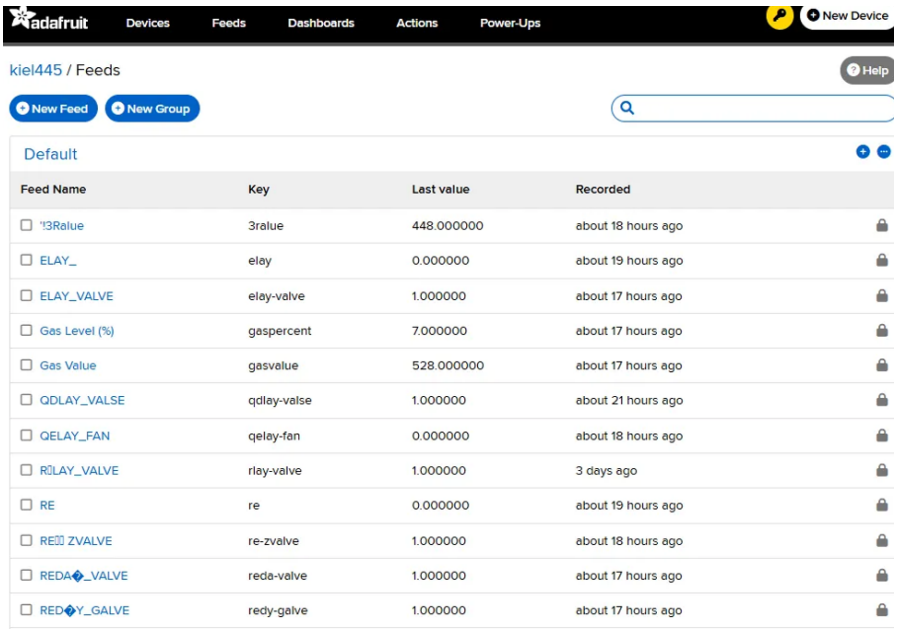
Gambar 48 Serial Monitor Node 1

Gambar 49 Serial Monitor Node 2

```
17:47:04.305 -> Gas Value: 444
17:47:04.305 -> Nilai Gas Analog: 444 => Gas Level: 10%
17:47:04.402 -> Pesan dikirim: DATA|gasValue=444,gasPercent=10,RELAY_VALVE=1,RELAY_FAN=0
17:47:04.869 -> Balasan dari Gateway: ACK|Data dari Node 2 diterima
17:47:09.887 -> Gas Value: 445
17:47:09.919 -> Nilai Gas Analog: 445 => Gas Level: 10%
17:47:09.995 -> Pesan dikirim: DATA|gasValue=445,gasPercent=10,RELAY_VALVE=1,RELAY_FAN=0
17:47:10.416 -> Balasan dari Gateway: ACK|Data dari Node 2 diterima
```

Gambar 50 Serial Monitor Node 2

Gambar 51. Dashboard Adafruit IO – Feed Data



Feed Name	Key	Last value	Recorded
'!3Ralue	3ralue	448.000000	about 18 hours ago
ELAY_	elay	0.000000	about 19 hours ago
ELAY_VALVE	elay-valve	1.000000	about 17 hours ago
Gas Level (%)	gaspercent	7.000000	about 17 hours ago
Gas Value	gasvalue	528.000000	about 17 hours ago
QDLAY_VALSE	qdlay-valse	1.000000	about 21 hours ago
QELAY_FAN	qelay-fan	0.000000	about 18 hours ago
RŌLAY_VALVE	rlay-valve	1.000000	3 days ago
RE	re	0.000000	about 19 hours ago
RE ZVALVE	re-zvalve	1.000000	about 18 hours ago
REDA_VALVE	reda-valve	1.000000	about 17 hours ago
RED_Y_GALVE	redy-galve	1.000000	about 17 hours ago

Gambar 52. Dashboard Adafruit IO – Feed Data

4.2.3 Pengujian Notifikasi Cloud

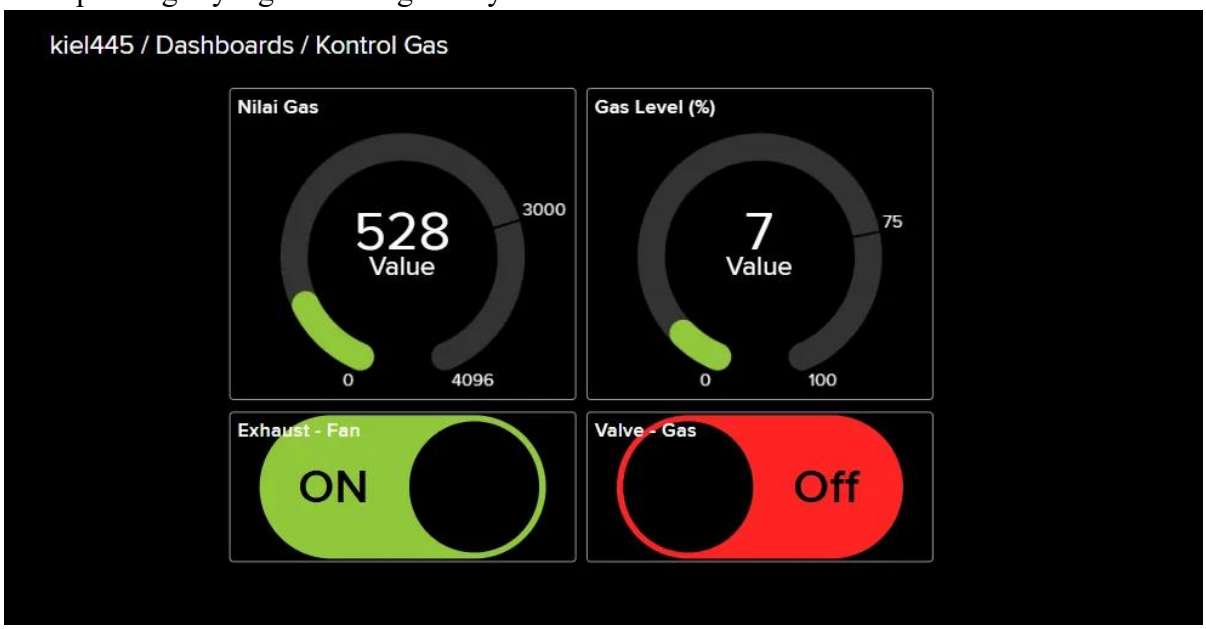
```
17:46:59.467 -> Pesan dari Node 1: DATA|gasValue=320,gasPercent=7,RELAY_VALVE=1,RELAY_FAN=0
17:46:59.564 -> Waktu sekarang: 2025-06-18 17:46:58
17:46:59.852 -> Balasan dikirim ke Node 1: ACK|Data dari Node 1 diterima
17:46:59.916 -> Kirim ke Adafruit IO:
17:46:59.916 -> Tipe: DATA
17:46:59.948 -> Isi : gasValue=320,gasPercent=7,RELAY_VALVE=1,RELAY_FAN=0
17:47:00.013 -> Feed: gasValue = 320
17:47:00.013 -> Feed: gasPercent = 7
17:47:00.044 -> Feed: RELAY_VALVE = 1
17:47:00.076 -> Feed: RELAY_FAN = 0
```

Gambar 53. Freed Data dari Node 1

```
17:47:04.500 -> Pesan dari Node 2: DATA|gasValue=444,gasPercent=1,RELAY_VALVE=1,RELAY_FAN=0
17:47:04.564 -> Waktu sekarang: 2025-06-18 17:47:03
17:47:04.869 -> Balasan dikirim ke Node 2: ACK|Data dari Node 2 diterima
17:47:04.934 -> Kirim ke Adafruit IO:
17:47:04.934 -> Tipe: DATA
17:47:04.966 -> Isi : gasValue=444,gasPercent=1,RELAY_VALVE=1,RELAY_FAN=0
17:47:05.029 -> Feed: gasValue = 444
17:47:05.029 -> Feed: gasPercent = 1
17:47:05.062 -> Feed: RELAY_VALVE = 1
17:47:05.102 -> Feed: RELAY_FAN = 0
```

Gambar 54. Feed Data dari Node 2

Setiap data gas yang diterima gateway dikirimkan ke Adafruit IO:



Gambar 55. Data yang ditampilakan lewat Dasbaord Adafruit

4.2.3 Integrasi Sistem dengan Cloud Platform Adafruit IO

Integrasi sistem dengan cloud dilakukan menggunakan layanan Adafruit IO, yang berfungsi sebagai dashboard pemantauan berbasis Internet untuk sistem deteksi kebocoran gas. Platform ini dipilih karena mendukung integrasi langsung dengan ESP32 melalui library AdafruitIO_WiFi.h, memiliki tampilan antarmuka yang intuitif, serta memungkinkan visualisasi data sensor dalam bentuk grafik, indikator digital, dan widget lainnya.

Setiap kali gateway seperti:

gasValue = 320
gasPercent = 7
RELAY_VALVE = 1
RELAY_FAN = 0

Data ini dipetakan ke dalam empat feed terpisah: gasValue, gasPercent, RELAY_VALVE, dan RELAY_FAN. Feed ini kemudian ditampilkan pada dashboard Adafruit IO dalam bentuk:

- Grafik gasValue (menampilkan tren nilai gas analog dari waktu ke waktu).
- Grafik gasPercent (untuk memantau persentase konsentrasi gas).
- Indikator saklar digital yang menunjukkan status aktif/tidaknya katup gas (VALVE).

- Indikator saklar digital untuk status kipas darurat (FAN).

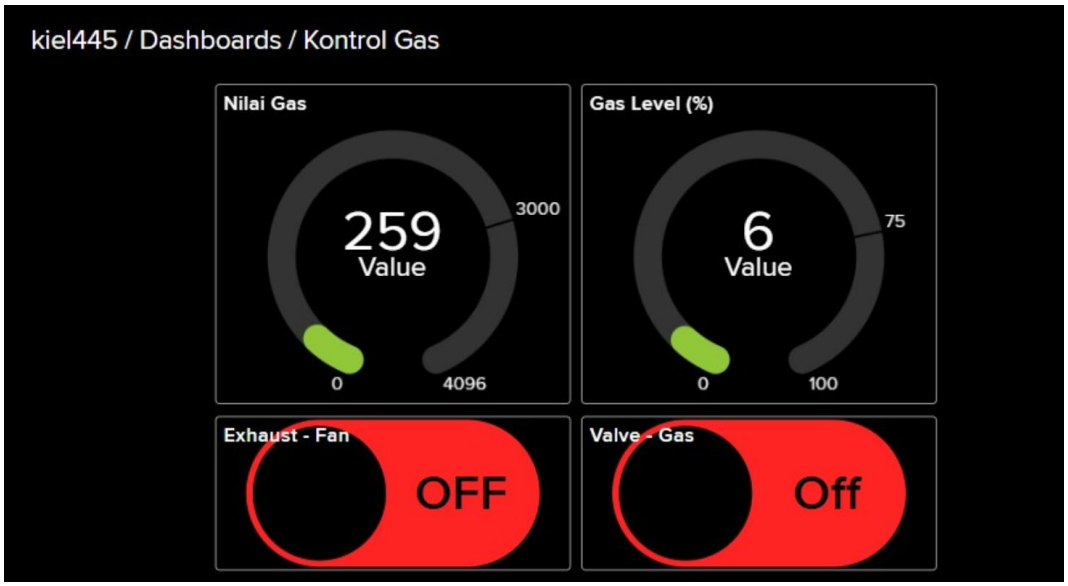
Setiap data baru yang diterima akan langsung mengupdate tampilan dashboard secara real-time. Hal ini memungkinkan pengguna untuk memantau kondisi gas dari jarak jauh melalui perangkat seluler atau komputer.

Proses integrasi dilakukan melalui fungsi:

```
io.feed("gasValue")->save(val);
io.feed("gasPercent")->save(val);
```

dan seterusnya untuk feed lainnya. Dengan metode ini, pengiriman data ke cloud hanya membutuhkan beberapa baris kode, namun menghasilkan pemantauan berbasis IoT yang efisien dan responsif.

Adapun jika koneksi Adafruit IO terputus, program secara otomatis mencoba melakukan reconnect dengan cara memeriksa status koneksi melalui io.status() dan memanggil kembali fungsi io.connect() hingga berhasil.



Gambar 56. Tampilan Dashboard Adafruit IO
(Gambar menunjukkan grafik nilai gas, status valve, dan fan secara real-time)

4.2.4 Analisis Hasil

Berdasarkan pengujian yang telah dilakukan, berikut adalah hasil analisis performa sistem:

- Sistem berhasil mendeteksi keberadaan gas metana secara cepat dan akurat dengan menggunakan sensor MQ-2.
- Komunikasi antar node sensor dan gateway menggunakan LoRa berlangsung stabil hingga jarak 15–20 meter dalam kondisi indoor tanpa halangan fisik yang signifikan.
- Fungsi otomatisasi pengendalian perangkat (relay) bekerja dengan baik. Saat nilai gasPercent mencapai atau melebihi 75%, sistem langsung mengaktifkan RELAY_VALVE dan RELAY_FAN.
- Integrasi dengan Adafruit IO berjalan lancar. Data dari node ditampilkan secara real-time dengan delay yang sangat minim (< 3 detik) sejak gas terdeteksi.
- Sistem ini mendukung fitur early warning berbasis cloud, yang sangat penting untuk pemantauan jarak jauh di lingkungan industri.

Dengan demikian, sistem dinilai telah memenuhi tujuan penelitian, yaitu mendesain sistem monitoring kebocoran gas berbasis IoT yang handal, efisien, dan dapat diakses dari mana saja.

4.2.5 Kendala dan Solusi

Selama proses pengembangan sistem, beberapa kendala teknis ditemukan. Tabel di bawah ini merangkum masalah yang muncul beserta solusi yang telah diterapkan:

Tabel 2. Kendala Dan Solusi

No.	Kendala Teknis	Solusi yang Diterapkan
1	Sensor gas kadang memberikan nilai tidak stabil / delay pembacaan	Menambahkan delay mikro dan menggunakan filter Moving Average sederhana
2	Koneksi ke Adafruit IO kadang terputus otomatis	Menambahkan loop reconnect dan pengecekan io.status() di dalam loop()
3	Firebase menolak data JSON	Struktur JSON diperiksa ulang, disesuaikan dengan format yang diterima Firebase dan autentikasi API disempurnakan
4	Terjadi interferensi data antar Node di jaringan LoRa	Membedakan ID Node dan melakukan pengaturan ulang channel frekuensi LoRa
5	Serial monitor kadang buffer penuh	Menambahkan flush() dan memperlambat kecepatan kirim antar data (delay)

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian sistem monitoring kebocoran gas berbasis Internet of Things (IoT) menggunakan jaringan LoRa dengan topologi star, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Sistem monitoring kebocoran gas berhasil dibangun dengan menggunakan sensor MQ-2 yang mampu mendeteksi berbagai jenis gas berbahaya seperti LPG.
2. Sistem dapat mengirimkan data sensor secara nirkabel dari beberapa node sensor ke satu gateway menggunakan modul LoRa RFM95 dalam konfigurasi topologi star, dengan jarak efektif hingga 167 meter dalam kondisi LOS (*Loss Line off Sight)* yang artinya terbuka tanpa hambatan besar.
3. ESP32 sebagai mikrokontroler berfungsi optimal dalam mengolah data sensor, mengaktifkan buzzer melalui relay saat ambang batas gas terlampaui, serta mengirimkan data ke platform cloud Adafruit IO dan Realtime untuk kebutuhan monitoring dan logging secara real-time.
4. Sistem terbukti mampu melakukan deteksi dini dengan waktu respons antara 1–2 detik, dan mengirimkan notifikasi serta visualisasi data ke cloud dengan delay < 5 detik, sehingga mendukung tindakan mitigasi kebocoran gas secara cepat dan tepat.
5. Penggunaan relay sebagai aktuator berhasil memicu alarm (buzzer) secara otomatis ketika kebocoran gas terdeteksi, yang menjadikan sistem ini layak untuk diterapkan sebagai prototipe sistem peringatan dini kebocoran gas di lingkungan industri.

5.2 Saran

Agar sistem ini dapat lebih dikembangkan dan dioptimalkan di masa mendatang, maka disampaikan beberapa saran sebagai berikut:

1. Penambahan jenis sensor gas lainnya seperti sensor gas metana (MQ-4) atau hidrogen sulfida (MQ-136) agar sistem dapat mendeteksi lebih banyak jenis kebocoran yang berpotensi terjadi di pabrik.

2. Implementasi aplikasi mobile berbasis Android yang terintegrasi dengan Firebase untuk memberikan notifikasi instan kepada operator atau teknisi jika terjadi kebocoran gas.
3. Penerapan energi terbarukan seperti panel surya untuk mendukung node sensor agar sistem dapat bekerja mandiri dan hemat energi, terutama pada lokasi yang sulit dijangkau listrik.
4. Pengujian lebih lanjut perlu dilakukan dalam lingkungan pabrik sebenarnya dengan kondisi yang lebih kompleks (seperti suhu tinggi, kelembaban, dan interferensi sinyal) untuk mengukur stabilitas sistem dalam kondisi riil.
5. Penggunaan sistem pengelompokan prioritas alarm berdasarkan tingkat konsentrasi gas berbahaya, sehingga respons mitigasi bisa disesuaikan berdasarkan tingkat ancaman.

DAFTAR PUSTAKA

Junaidi, & Ramadhani, K. (2024). Efektivitas Internet of Things (IoT) pada sektor pertanian. *Jurnal Teknologi Komputer dan Sistem Informasi, 4*(1), 12–15. https://doi.org/10.54314/teknisi.v4i1.1793

Hafiz, M., & Candra, O. (2021). Perancangan Sistem Pendeteksi Kebakaran Berbasis Mikrokontroller dan Aplikasi Map dengan Menggunakan IoT. Jurnal Teknik Elektro Dan Voksdional, 7(1), 53– 63. http://ejournal.unp.ac.id/index.php/jtev/article/view/111420

Evalina, N., & A Azis, H. (2020). Implementation and Design Gas Leakage Detection System Using ATMega8 Microcontroller. IOP Conference Series: Materials Science and Engineering, 821(1), 1–5. https://doi.org/10.1088/1757-899X/821/1/012049

Wagyana, A., & Rahmat. 2019. Prototype Modul Praktik Untuk Pengembangan Aplikasi Internet Of Things (Iot). Jurnal Ilmiah Setrum, 240-241. https://www.academia.edu/download/99707713/3438.pdf

Liandana, M. (2019). Penerapan Teknologi LoRa pada Purwarupa Awal Wearable Device. *Res. Hitung. Inf. sistem. Teknologi. Manajer* , *2* (2), 40. https://www.academia.edu/download/89725302/pdf.pdf

Amsar, A., Khairuman, K., & Marlina, M. (2020). Perancangan Alat Pendeteksi CO2 Menggunakan Sensor MQ-2 Berbasis Internet of Thing. METHOMIKA: Jurnal Manajemen Informatika & Komputerisasi Akuntansi, 4(1), 73-79.

https://doi.org/10.46880/jmika.Vol4No1.pp73-79

Amaro, N. (2017). Sistem monitoring besaran listrik dengan teknologi IoT (Internet of Things).

https://doi.org/10.26740/jte.v8n1.p%25p

Nizam Muhammad, Y. H. (2022). MIKROKONTROLER ESP 32 SEBAGAI ALAT

MONITORING PINTU BERBASIS WEB. Blitar: JATI (Jurnal Mahasiswa Teknik Informatika).

http://repository.dinamika.ac.id/id/eprint/7849

Gunawan, A. O., & Arijanto, R. (2025). PENGGABUNGAN FUNGSI SENSOR ULTRASONIK DAN SOFTWARE ADAFRUIT IO PADA SEBUAH SMARTHOME. POTERS (Proceedings of Technology, Engineering and Computers), 1(1), 166-180.

https://jurnal.buddhidharma.ac.id/index.php/poters/article/view/3608

Payara, G. R., & Tanone, R. (2018). Penerapan Firebase Realtime Database Pada Prototype Aplikasi Pemesanan Makanan Berbasis Android. *Jurnal Teknik Informatika dan Sistem Informasi*, *4*(3), 397-406.

http://dx.doi.org/10.28932/jutisi.v4i3.870

Dwiyatno, S., Krisnaningsih, E., Hidayat, D. R., & Sulistiyono. (2022). Smart agriculture: Monitoring penyiraman tanaman berbasis Internet of Things. *Jurnal PROSISKO, 9*(1), 38. https://doi.org/10.30656/prosisko.v9i1.4669

Purnama Sari, I., Novita, A., Al-Khowarizmi, A., Ramadhani, F., & Satria, A. (2024). Pemanfaatan Internet of Things (IoT) pada bidang pertanian menggunakan Arduino Uno R3. *Universitas Muhammadiyah Sumatera Utara*. https://doi.org/10.56211/blendsains.v2i4.505

Harli. (2016). Identifikasi dan potensi perluasan tanaman nilam (*Pogostemon cablin Benth.*) di bawah tegakan kakao di Kabupaten Polewali Mandar. *AGROVITAL*, 1(1), 21. https://doi.org/10.2541/agrovital.v1i1.7460

Bayu Tri Anggara, Mimin Fatchiyatur Rohmah, & Sugianto. (2018). SISTEM PENGUKUR KELEMBABAN TANAH PERTANIAN DAN PENYIRAMAN OTOMATIS BERBASIS INTERNET OF THINGS (IoT).

Repository Universitas Islam Majapahit. http://repository.unim.ac.id/id/eprint/241

Budiharto, W. (2019). Smart Farming yang Berwawasan Lingkungan untuk. Unsri Press.https://conference.unsri.ac.id/index.php/lahansuboptimal/article/view/1669/899

RachmaniahM., & NugrahaA. A. (2018). Sistem Pakar Kesesuaian Lahan untuk Tanaman Nilam. *Jurnal Ilmu Komputer Dan Agri-Informatika*, *5*(1), 61-73. https://doi.org/10.29244/jika.5.1.61-73

Darmanti, S., Nuchayati, Y., Hastuti, E. D., & Syaifuddin, M. (2009). Produksi biomassa tanaman nilam (*Pogostemon cablin*) yang ditanam pada intensitas cahaya yang berbeda. *Anatomi Fisiologi, 17*(1), 22–29. https://doi.org/10.14710/baf.v17i1.2532

Nasruddin, N., Harahap, E. M., Hanum, C., & Siregar, L. A. M. (2019). The level of water supply and its effect on the growth of plants and yields patchouli (*Pogostemon cablin, Benth.*). *WMA-1*, *EAI*. https://doi.org/10.4108/eai.20-1-2018.2282082

Junaidi, J., & Ramadhani, K. (2024). Efektivitas Internet of Things (IoT) pada sektor pertanian. https://doi.org/10.54314/teknisi.v4i1.1793

Ramadani, R. (2023). The potential of the Internet of Things (IoT) as a source of official statistics for agriculture. https://doi.org/10.34123/semnasoffstat.v2023i1.1900

Sandi, G. H., & Fatma, Y. (2023). Pemanfaatan teknologi Internet of Things (IoT) pada bidang pertanian. https://doi.org/10.36040/jati.v7i1.5892

Putu Edward Lim Junior, Eddy Muntina Dharma, & Putu Trisna Hady Permana. (2023). SMART IRRIGATION BERBASIS INTERNET OF THINGS (IOT) MENGGUNAKAN FRAMEWORK FIREBASE PADA TANAMAN TOMAT (STUDI KASUS PADA PERTANIAN TOMAT DI DESA

TEGALCANGKRING, KABUPATEN JEMBRANA). https://doi.org/10.36002/jutik.v9i4.2549

Walid, M., Hoiriyah, H., & Fikri, A. (2022). Pengembangan sistem irigasi pertanian berbasis Internet of Things (IoT). https://doi.org/10.36040/mnemonic.v5i1.4452

Tandrianto, A., Setiawan, I. N., & Amrita, A. A. N. (2022). Implementasi sistem pemantauan intensitas cahaya dengan IoT di plant factory Kebun Percobaan Fakultas Pertanian Universitas Udayana. https://doi.org/10.24843/SPEKTRUM.2022.v09.i02.p12

LAMPIRAN

- Lampiran koding gateaway

```
#include <SPI.h>

#include <LoRa.h>


#include <Wire.h>

#include <Adafruit_GFX.h>

#include <Adafruit_SSD1306.h>

#include <SPI.h>


// --- OLED setup ---

#define SCREEN_WIDTH 128

#define SCREEN_HEIGHT 64

#define OLED_RESET    -1

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET);

// --- Pin Setup ---

const int MQ2_PIN = 32;         // MQ2 analog pin

const int RELAY_VALVE = 25;     // Relay 1: valve

const int RELAY_FAN = 26;       // Relay 2: exhaust fan


#define NODE_AD
```

- Lampiran koding node 1

```
#include <SPI.h>

#include <LoRa.h>


#include <Wire.h>

#include <Adafruit_GFX.h>

#include <Adafruit_SSD1306.h>

#include <SPI.h>


// --- OLED setup ---

#define SCREEN_WIDTH 128

#define SCREEN_HEIGHT 64

#define OLED_RESET    -1

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET);

// --- Pin Setup ---

const int MQ2_PIN = 32;         // MQ2 analog pin

const int RELAY_VALVE = 25;     // Relay 1: valve

const int RELAY_FAN = 26;       // Relay 2: exhaust fan


#define NODE_ADDRESS 0x01

#define GATEWAY_ADDRESS 0x10

#define LED_PWR 15 //Led Merah

#define LED_SEND 12 // Led Hijau

#define LED_RECIEV 4  // Led Biru

#define LED_WARNING 33  // Led Kuning
```

```
// --- Threshold ---
int gasThreshold = 3072;  // Sesuaikan nilai ambang gas dari MQ2
int statusValve=0;
int statusFan=0;

void setup() {
  Serial.begin(9600);
  while (!Serial);

  Serial.println("LoRa Node Starting...");

  pinMode(LED_PWR, OUTPUT);
  digitalWrite(LED_PWR, HIGH);
  pinMode(LED_WARNING, OUTPUT);
  digitalWrite(LED_WARNING, LOW);

  LoRa.setPins(5, 14, 2);
  if (!LoRa.begin(915E6)) {
    Serial.println("Starting LoRa failed!");
    while (1);
  }

   // --- Relay Output ---
  pinMode(RELAY_VALVE, OUTPUT);
  pinMode(RELAY_FAN, OUTPUT);
```

```
  digitalWrite(RELAY_VALVE, HIGH); // Relay OFF
  digitalWrite(RELAY_FAN, HIGH);

  // --- OLED Init ---
  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("OLED failed"));
    for(;;);
  }
  display.clearDisplay();
  display.setCursor(10, 0);
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.println("MQ2 Gas Detector");
  display.setCursor(10, 36);
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.println("Connecting Firebase...");
  display.display();
  delay(1000);

}

void loop() {

  int gasValue = analogRead(MQ2_PIN); // Baca dari A0 (0-4095 di ESP32)
```

```
// Konversi ke skala 0-100 (gunakan batas bawah dan atas)
  int gasPercent = map(gasValue, 0, 4095, 0, 100);
// Batasi nilai antara 0 - 100
  gasPercent = constrain(gasPercent, 0, 100);

  // Display to OLED
  display.clearDisplay();
  display.setCursor(0, 0);
  display.setTextSize(1);
  display.print("Nilai Gas: ");
  display.setTextSize(2);
  display.println(gasValue);
  display.setTextSize(1);
  display.println("Gas Level (%) : ");
  display.setTextSize(2);
  display.print("       ");
  display.println(gasPercent);
  display.setTextSize(1);
  display.println("Status: ");
  display.setTextSize(2);

  if (gasValue > gasThreshold) {
    digitalWrite(RELAY_VALVE, HIGH); // Relay ON
    digitalWrite(RELAY_FAN, HIGH);
    statusValve=1;
    statusFan=1;
```

```
  digitalWrite(LED_WARNING,LOW);
  display.println("    BAHAYA!");
  delay(1000);
  digitalWrite(LED_WARNING,HIGH);
 } else {
  digitalWrite(RELAY_VALVE, LOW); // Relay OFF
  digitalWrite(RELAY_FAN, LOW);
  statusValve=1;
  statusFan=0;
  digitalWrite(LED_WARNING,LOW);
  display.println("    Aman");
 }
 display.display();



 Serial.print("Gas Value: ");
 Serial.println(gasValue);

 Serial.print("Nilai Gas Analog: ");
 Serial.print(gasValue);
 Serial.print(" => Gas Level: ");
 Serial.print(gasPercent);
 Serial.println("%");

 String dataStr = "gasValue=" + String(gasValue) +
                  ",gasPercent=" + String(gasPercent) +
```

```
              ",RELAY_VALVE=" + String(statusValve) +
              ",RELAY_FAN=" + String(statusFan);

  // Contoh kirim pesan PING
  kirimPesan("DATA", dataStr);

  tungguBalasan();

  pinMode(LED_SEND, OUTPUT);
  digitalWrite(LED_SEND, LOW);

  delay(5000);  // jeda antar request
  pinMode(LED_RECIEV, OUTPUT);
  digitalWrite(LED_RECIEV, LOW);
}

void kirimPesan(String tipe, String isi) {
  String fullMessage = tipe + "|" + isi;

  LoRa.beginPacket();
  LoRa.write(GATEWAY_ADDRESS);
  LoRa.write(NODE_ADDRESS);
  LoRa.print(fullMessage);
  LoRa.endPacket();

  Serial.println("Pesan dikirim: " + fullMessage);
```

```
  pinMode(LED_SEND, OUTPUT);
  digitalWrite(LED_SEND, HIGH);
}

void tungguBalasan() {
  unsigned long startTime = millis();
  bool received = false;

  while (millis() - startTime < 2000) {
    int packetSize = LoRa.parsePacket();
    if (packetSize) {
      uint8_t recipient = LoRa.read();
      uint8_t sender = LoRa.read();
      String reply = "";

      while (LoRa.available()) {
        reply += (char)LoRa.read();
      }

      if (recipient == NODE_ADDRESS && sender == GATEWAY_ADDRESS) {
        Serial.print("Balasan dari Gateway: ");
        Serial.println(reply);
        pinMode(LED_RECIEV, OUTPUT);
        digitalWrite(LED_RECIEV, HIGH);
        received = true;
        break;
```

```
      }
    }
  }

  if (!received) {
    Serial.println("Tidak ada balasan dari Gateway.");
  }
}
```

- Lampiran koding node 2

```
#include <SPI.h>

#include <LoRa.h>


#include <Wire.h>

#include <Adafruit_GFX.h>

#include <Adafruit_SSD1306.h>

#include <SPI.h>


// --- OLED setup ---

#define SCREEN_WIDTH 128

#define SCREEN_HEIGHT 64

#define OLED_RESET    -1

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET);

// --- Pin Setup ---

const int MQ2_PIN = 32;         // MQ2 analog pin

const int RELAY_VALVE = 25;     // Relay 1: valve

const int RELAY_FAN = 26;       // Relay 2: exhaust fan


#define NODE_AD
```