

INTEGRASI SISTEM MONITORING KUALITAS UDARA BERBASIS IoT DENGAN CLOUD BERBASIS APLIKASI ANDROID

Indrawati¹, Afla Nevrisa², Ismi Amalia³, Radhiyatammardhiyyah⁴, Amir. D⁵

^{1,2,3,4} Tenik Rekayasa Komputer Jaringan Politeknik Negeri Lhokseumawe

⁵ Teknik Rekayasa Jaringan Telekomunikasi Politeknik Negeri Lhokseumawe

Jln. B. Aceh Medan Km.280 Buketrata 24301 INDONESIA

¹indrawati@pnl.ac.id

Abstrak Artikel ini membahas tentang Kualitas udara merupakan faktor penting yang memengaruhi kesehatan manusia dan lingkungan, terutama di kawasan perkotaan yang rentan terhadap polusi akibat aktivitas industri dan transportasi. Penelitian ini bertujuan merancang dan membangun sistem monitoring kualitas udara berbasis Internet of Things (IoT) yang terintegrasi dengan teknologi Cloud Computing dan aplikasi Android. Sistem dirancang menggunakan sensor MQ-135 untuk mendeteksi gas berbahaya serta sensor DHT11 untuk mengukur suhu dan kelembaban. Data hasil pengukuran diproses melalui mikrokontroler ESP32 yang berfungsi mengirimkan informasi secara real-time ke platform cloud (ThingSpeak/Firebase). Selanjutnya, data dapat divisualisasikan melalui dashboard cloud dan diakses oleh pengguna melalui aplikasi Android yang menampilkan informasi kondisi udara secara numerik maupun grafik. Hasil implementasi menunjukkan bahwa sistem monitoring kualitas udara berbasis IoT mampu beroperasi secara kontinu dan stabil dengan availability 99,21% serta tingkat kegagalan 0,007% dari 84.906 permintaan selama pengujian 72 jam. Sistem mencapai throughput rata-rata 9,194 permintaan per detik dengan median waktu respons 539,60 ms, serta akurasi sensor dengan deviasi maksimum 3,45%. Integrasi IoT, Cloud Computing, dan aplikasi mobile terbukti efektif dalam menyediakan solusi monitoring lingkungan yang efisien, skalabel, serta mudah diakses dan dipahami oleh pengguna.

Kata kunci: IoT, Cloud Computing, monitoring kualitas udara, ESP32, Andro

Abstract— Air quality is a critical factor affecting human health and the environment, particularly in urban areas that are highly vulnerable to pollution from industrial activities and transportation. This study aims to design and develop an Internet of Things (IoT)-based air quality monitoring system integrated with Cloud Computing technology and an Android application. The system utilizes an MQ-135 sensor to detect hazardous gases and a DHT11 sensor to measure temperature and humidity, with measurement data processed by an ESP32 microcontroller and transmitted in real time to cloud platforms (ThingSpeak and Firebase). The collected data are visualized through cloud dashboards and accessed via an Android application that presents air quality information in both numerical and graphical formats. Experimental results show that the system operates continuously and stably with an availability of 99.21% and a failure rate of 0.007% from 84,906 requests during 72 hours of testing, achieving an average throughput of 9.194 requests per second, a median response time of 539.60 ms, and a maximum sensor deviation of 3.45%. The integration of IoT, Cloud Computing, and mobile applications is proven to be effective in providing an efficient, scalable, and user-friendly environmental monitoring solution.

Keywords: IoT, Cloud Computing, air quality monitoring, ESP32, Android.

I. PENDAHULUAN

A. Latar Belakang Masalah

Kualitas udara memiliki dampak signifikan terhadap kesehatan manusia, terutama di wilayah perkotaan yang rentan terhadap polusi akibat transportasi dan aktivitas industri [1] [2]. Sistem monitoring konvensional umumnya hanya mampu menampilkan informasi secara terbatas tanpa dukungan penyimpanan data jangka panjang dan analisis mendalam [3]. Perkembangan Internet of Things (IoT) memungkinkan pengumpulan data lingkungan secara real-time, tetapi memiliki keterbatasan dalam aspek penyimpanan dan pemrosesan data skala besar [4]. Cloud Computing hadir sebagai solusi dengan menyediakan kapasitas penyimpanan yang luas, kemampuan pemrosesan terdistribusi, serta akses data yang fleksibel. Penelitian ini bertujuan untuk merancang dan mengembangkan sistem monitoring kualitas udara berbasis IoT yang diintegrasikan dengan platform cloud dan aplikasi Android, sehingga menghasilkan solusi yang efisien, skalabel, dan mudah diakses oleh pengguna.

B. Tinjauan pustaka

Arsitektur umum sistem pemantauan kualitas udara dapat dibagi atas 3 bagian layer, secara singkat dijelaskan di bawah ini.

1) *Layer Sensor*: Layer terdiri dari tiga bagian utama, yaitu: pertama, sensor MQ-135, Sensor berfungsi untuk mendeteksi kualitas udara, sensor ini biasanya dipasang pada lokasi pengamatan. untuk keperluan ini, dapat digunakan beberapa sensor, seperti sensor gas, suhu dan kelembaban serta debu. Sensor tipe MQ-135 adalah sensor gas yang mempunyai keluaran dual output yaitu output digital (dengan op-amp sebagai komparator yang sensitifitas atau offsetnya bisa diatur dengan memutar trimpot dan output analog yaitu output langsung dari sensor yang dapat dihubungkan dengan ADC tokoduino. sangat cocok untuk dikreasikan dengan arduino dan minimum sistem lainnya sebagai alarm atau monitoring untuk mendeteksi kualitas udara.



Gambar 1. Sensor MQ 135.

Sensor tipe MQ-135 disebut sebagai air quality control sensor, karena sensor ini mempunyai kemampuan untuk mendeteksi berbagai jenis gas polutan di udara yang menjadi indikator utama kualitas udara[5][6]. Sensor ini digunakan untuk mendeteksi pemantau udara ambient dengan memanfaatkan teknologi IoT. Lebih jelasnya bentuk komponen sensor MQ 135 diperlihatkan pada gambar 1.

Kedua, sensor Suhu dan Kelembaban . Sensor suhu dan kelembaban menggunakan komponen sensor DHT11 dan DHT22 [7][8]. Sensor ini adalah sensor digital yang digunakan untuk mengukur suhu dan kelembaban udara. Keduanya sering digunakan dalam proyek-proyek mikrokontroler seperti Arduino, ESP8266, atau Raspberry Pi. Fungsi utamanya mengukur suhu udara. Untuk DHT11 memiliki akurasi $\pm 2^{\circ}\text{C}$ dengan rentang $0-50^{\circ}\text{C}$, sedangkan DHT22 memiliki akurasi $\pm 0.5^{\circ}\text{C}$, rentang -40 hingga 80°C .

Mengukur kelembaban udara komponen DHT11 memiliki akurasi $\pm 5\%$, rentang $20-80\%$ RH, sedangkan komponen DHT22 memiliki akurasi $\pm 2-5\%$, rentang $0-100\%$ RH. Sensor ini mengirimkan data secara digital. Sensor ini mengirimkan data ke mikrokontroler dalam bentuk digital melalui satu pin data, sehingga tidak perlu konversi analog-ke-digital. Lebih jelasnya bentuk sensor ini diperlihatkan pada gambar 2.



Gambar 2 Sensor suhu dan kelembaban (a) DHT11, (b) DHT 22

Sensor DHT22 adalah sensor digital yang berfungsi untuk mengukur suhu dan kelembaban udara dengan tingkat akurasi yang cukup tinggi. Sensor ini sering digunakan dalam sistem monitoring lingkungan, rumah pintar, dan proyek IoT , bentuk komponennya diperlihatkan seperti gambar 2 .

Fungsi utama dari sensor DHT22, adalah; mengukur Suhu Udara (Temperature) dengan rentang -40°C hingga $+80^{\circ}\text{C}$. Akurasi: $\pm 0.5^{\circ}\text{C}$ 2. Mengukur Kelembaban Udara, rentang: 0% hingga 100% RH (Relative Humidity), akurasi: $\pm 2-5\%$ RH 3. Mengirimkan Data Secara Digital . , dengan spesifikasi komunikasi 1-wire (hanya membutuhkan 1 pin data), output langsung berupa data digital (tidak perlu ADC).

Ketiga, sensor partikel debu. Sensor debu menggunakan sensor SDS011. Sensor debu SDS011 adalah sensor partikel debu buatan Nova Fitness yang digunakan untuk mengukur konsentrasi partikel udara halus (PM2.5 dan PM10) secara

real-time. Bentuk komponen sensor debu diperlihatkan pada gambar.3.

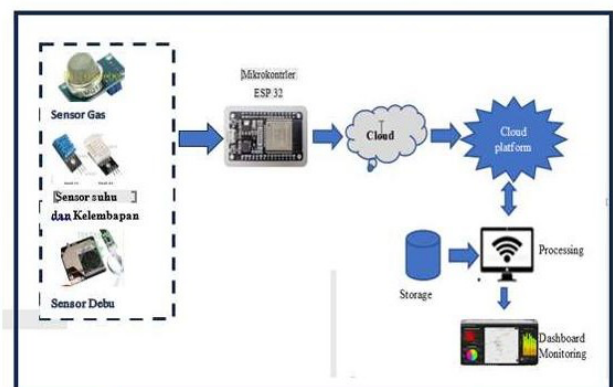


Gambar 3. Sensor partikel debu SDS011

Fungsi Utama partikel debu SDS011 dijelaskan secara rinci berikut [9]:

- Mengukur Partikel PM2.5:* Partikel dengan ukuran ≤ 2.5 mikrometer, partikel ini bisa masuk ke paru-paru dan aliran darah.
- Mengukur Partikel PM10.* Partikel dengan ukuran ≤ 10 mikrometer. Umumnya berasal dari debu, serbuk sari, dan asap.
- Memberikan Data Digital:* Output data melalui antarmuka UART (serial) atau PWM, sehingga mudah diintegrasikan dengan mikrokontroler seperti Arduino atau ESP32.
- Sensor Laser Presisi Tinggi:* Menggunakan teknologi laser scattering untuk mendeteksi partikel di udara.

2) *Layer Cloud Computing:* Platform cloud digunakan untuk menyimpan data. Penyimpanan data dapat menggunakan google cloud storage, firebase, AWS S3, Azure Blob. Untuk melakukan pemrosesan data dapat digunakan Google Colab AWS Lambda, atau server cloud pribadi. Database untuk menyimpan dan mengatur data sensor dapat digunakan Firebase Realtime DB, MongoDB Atlas, InfluxDB. Pada lapisan ini juga dilengkapi dengan API singkatan dari Application Programming Interface. Secara sederhana API adalah jembatan komunikasi antara dua aplikasi atau sistem agar bisa saling bertukar data atau perintah. Platform cloud computation juga dilengkapi Dashboard Monitoring (Grafana, ThingsBoard, atau dashboard custom berbasis web).



Gambar 4. Arsitektur sistem terintegrasi

Dashboard monitoring adalah tampilan visual yang digunakan untuk memantau data, kondisi, atau kinerja suatu sistem secara real-time. Secara umum lapisan cloud ini dapat dilihat pada gambar 4.

3) *Layer Analitik dan Visualisasi*: Layer analitik adalah salah satu bagian dari arsitektur sistem informasi atau sistem data yang berfungsi untuk mengolah, menganalisis dan menyajikan data agar dapat digunakan untuk pengambilan keputusan. Lapisan ini berada di atas lapisan penyimpanan (storage) data (seperti database atau data warehouse) dan biasanya menggunakan berbagai teknik analisis, statistik, machine learning, dan visualisasi. Fungsi utama lapisan analitik adalah :

- a. *Mengolah data mentah menjadi informasi*: Mengubah data menjadi informasi yang bermakna melalui proses agregasi, pemodelan, atau transformasi.
- b. *Analisis data*: Melakukan analisis deskriptif (apa yang terjadi), diagnostik (mengapa terjadi), prediktif (apa yang akan terjadi), atau preskriptif (apa yang sebaiknya dilakukan).
- c. *Visualisasi data*: menyajikan hasil analisis dalam bentuk grafik, dashboard, atau laporan interaktif agar mudah dipahami oleh pengguna.

Pengelompokan kualitas udara ke dalam beberapa kategori, seperti: Baik, Sedang, tidak sehat, berbahaya., sementara untuk processingnya dilakukan melalui dataset yang dikumpulkan dari sensor, preprocessing: normalisasi, dan pembersihan data.

4). *Efisiensi Sistem*. : Perhitungan efisiensi menunjukkan sistem beroperasi pada 95% dari kapasitas ideal, mengindikasikan overload. Efisiensi ini dapat dihitung dengan persamaan (1)

$$efisiensi = \frac{Throughput \times Target ResponseTime}{Actual ResponseTime} \quad (1)$$

II. METODOLOGI PENELITIAN

A. Desain Penelitian

Penelitian ini menggunakan pendekatan eksperimen rekayasa (engineering experiment) dengan tujuan merancang, membangun, dan menguji sistem monitoring kualitas udara berbasis IoT yang terintegrasi dengan teknologi cloud dan aplikasi Android, seperti pada gambar 4. Fokus penelitian adalah menguji keandalan sistem dalam mengukur, menyimpan, dan menampilkan data kualitas udara secara real-time.

B. Perangkat Keras (Hardware)

Komponen utama yang digunakan untuk memonitoring kualitas udara terdiri dari beberapa layer, meliputi:

1), *Layer Sensor*: terdiri dari dua bagian, antara lain sensor MQ-135 digunakan untuk mendeteksi gas berbahaya seperti CO₂, NH₃, dan uap alcohol dan sensor sensor DHT11 digunakan untuk mengukur suhu dan kelembaban udara.

2. Layer display OLED 0.96 menampilkan data lokal secara real time atau langsung.

3). *Layer perangkat Lunak (Software)*. Komponen utama perangkat keras yang di gunakan adalah:

- a. *Firmware ESP32*: ditulis menggunakan Arduino IDE untuk membaca data sensor, memproses, dan mengirimkan data ke cloud.
- b. *Platform Cloud (ThingSpeak/Firebase)*: digunakan untuk menyimpan data secara terpusat, melakukan visualisasi dalam bentuk grafik, dan menyediakan Application Programming Interface (API).
- c. *Aplikasi Android*: dikembangkan menggunakan Android Studio untuk menampilkan data real-time kepada pengguna melalui grafik dan indikator numerik. Aplikasi mobile menyediakan antarmuka yang intuitif dan mudah diakses untuk monitoring kualitas udara. Keunggulan aplikasi mobile meliputi: portabilitas, push notification untuk alert, GPS integration untuk geo-tagging, dan user engagement yang tinggi.

4) *Arsitektur sistem*: Alur kerja sistem meliputi: Sensor MQ-135 dan DHT11 mengukur parameter kualitas udara. ESP32 memproses data dan menampilkan hasil pada layar OLED. Data dikirimkan ke platform cloud (ThingSpeak/Firebase) melalui koneksi Wi-Fi. Data disimpan, divisualisasikan, dan diakses melalui API. Aplikasi Android mengambil data dari cloud untuk ditampilkan dalam format user-friendly.

5). *Layer Cloud Computing dan Penyimpanan Data*. Layer ini menyediakan infrastruktur untuk penyimpanan, pemrosesan, dan distribusi data. Sistem menggunakan dual cloud platform untuk redundansi dan fungsionalitas yang berbeda:

a. *ThingSpeak Cloud Platform*. Fungsi: data logging, visualisasi historis, analisis statistik, Protokol komunikasi: HTTP POST request, update interval: 15 detik (limitasi free tier), data retention: unlimited pada free tier

b. *Firebase Realtime Database*. Fungsi: real-time synchronization, data distribution ke aplikasi, protokol komunikasi: WebSocket untuk real-time updates. Struktur data: JSON format. Security: Firebase Authentication dan Security Rules

c. *API Gateway*. Menyediakan RESTful API untuk komunikasi antar komponen. Format data: JSON. Authentication: API key dan token-based authentication

C. Layer Aplikasi dan Visualisasi

Layer ini menyediakan antarmuka untuk pengguna akhir, terdiri :

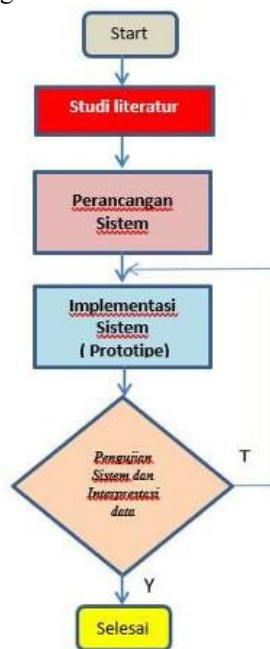
a. *Aplikasi Android*. User interface untuk menampilkan data real-time. Visualisasi grafis dengan chart library. Push notification untuk alert kualitas udara buruk Historical data analysis. User authentication dan profile management

b. *Web Dashboard*. Monitoring real-time melalui browser. Export data dalam format CSV/Excel. Administrative functions

D, Flow Chart Penelitian

Tahapan-tahapan penelitian dilakukan dengan mekanisme sebagai berikut;

- 1). *Studi literatur*: mengkaji teori IoT, cloud computing, sensor lingkungan dan aplikasi mobile.
- 2). *Perancangan sistem*: menentukan kebutuhan hardware dan software serta merancang arsitektur sistem.
- 3). *Implementasi*: Instalasi sensor, pemrograman ESP32, konfigurasi cloud dan pengembangan aplikasi Android.
- 4). *Pengujian sistem*: dilakukan pada beberapa lokasi dengan kondisi lingkungan berbeda. Parameter yang diuji meliputi: Keakuratan pembacaan sensor. Stabilitas pengiriman data ke cloud. Waktu respon akses data dari aplikasi Android.
- 5). *Indikator keberhasilan*: Indikator keberhasilan penelitian ini adalah data kualitas udara dapat ditampilkan secara real-time di OLED, cloud, dan aplikasi Android. Sistem mampu beroperasi stabil dengan tingkat kegagalan rendah (< 5%). Waktu respon pengiriman dan akses data < 2 detik dalam kondisi normal. Aplikasi Android dapat menampilkan data secara akurat dan user-friendly. Lebih jelas dapat dilihat pada gambar 5.



Gambar 5. Tahapan kegiatan penelitian

III. HASIL DAN PEMBAHASAN

A. Rancang Bangun Sistem Monitoring Kualitas Udara

Sistem monitoring kualitas udara dirancang dan dibangun menggunakan teknologi IoT, mengacu pada gambar 4, seperti ditunjukkan pada tinjauan pustaka pada pembahasan bab sebelumnya. Gambar 4 memperlihatkan penerapan sistem monitoring kualitas udara berbasis Internet of Things (IoT) yang memanfaatkan sensor, mikrokontroler ESP32, dan integrasi cloud untuk pemantauan secara real-time melalui perangkat Android. Sistem ini dirancang untuk mendeteksi parameter kualitas udara seperti kadar gas berbahaya, suhu, dan kelembaban lingkungan.

Proses kerja sistem dimulai dengan pengambilan data oleh dua jenis sensor, yaitu sensor MQ-135 yang berfungsi untuk

mendeteksi gas berbahaya seperti karbon dioksida (CO₂), amonia (NH₃), dan uap alkohol, serta sensor DHT11 yang mengukur suhu dan kelembaban udara.

Data hasil pengukuran dari kedua sensor ini dikirim ke mikrokontroler ESP32, yang bertindak sebagai pusat pengolahan data sekaligus pengendali sistem. Selanjutnya, data yang diterima akan ditampilkan secara lokal pada layar LCD OLED 0.96 inci agar pengguna dapat melihat informasi secara langsung. Selain itu, ESP32 yang telah terhubung dengan jaringan WiFi akan mengirimkan data secara berkala ke cloud server IoT, seperti ThingSpeak atau Firebase, untuk disimpan dan diakses secara daring.

Pengguna dapat memantau data tersebut melalui aplikasi Android yang dikembangkan khusus untuk sistem ini. Aplikasi menampilkan grafik dan data real-time mengenai kualitas udara, suhu, serta kelembaban, sehingga memudahkan pemantauan lingkungan baik di dalam maupun luar ruangan. Dengan adanya integrasi cloud dan mobile application, sistem ini memungkinkan pemantauan kualitas udara dilakukan secara fleksibel dan efisien dari lokasi mana pun. Sistem ini sangat bermanfaat untuk mendeteksi dini kondisi udara yang berpotensi membahayakan kesehatan, serta memberikan solusi monitoring yang murah, portabel, dan mudah digunakan oleh masyarakat luas

B. Uji Kemampuan Sistem Monitoring Kualitas Udara Dan Teknologi Cloud

Uji kemampuan sistem monitoring kualitas udara menggunakan IOT dilakukan terhadap 3 lokasi, yaitu; Lokasi Simpang Selat Malaka Cunda, lokasi Simpang Jam Lhokseumawe dan di perumahan politeknik. Waktu pengamatan dilakukan dalam beberapa priode, yaitu priode jam 7.50-8.50 Wib serta 16.00-17.00 Wib dan jam 12.55-13.55 WIB.

C. Metrik Evaluasi

Untuk mengevaluasi performa integrasi sistem, ditetapkan metrik sebagai berikut:

1). *Waktu Respons (Response Time)*. Pengujian waktu respons dilakukan dengan load testing menggunakan K6 tool. Tabel 1. menunjukkan hasil pengukuran waktu respon diobservasi dari waktu yang dibutuhkan dari pengiriman data hingga data tersedia di aplikasi Android. Target: 95% permintaan < 300 ms.

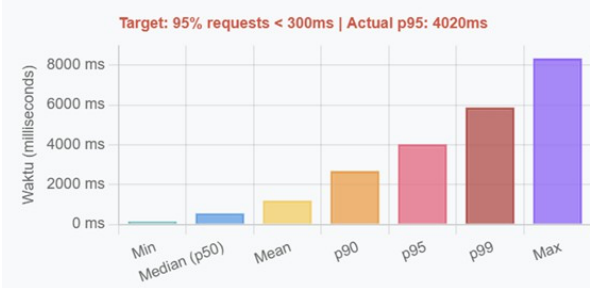
Tabel 1. Waktu respon

Metrik	Nilai	Keterangan
Rata-rata (Mean)	1190 ms	Waktu respons rata-rata
Median (p50)	539,60 ms	50% request di bawah nilai ini
Persentil 90 (p90)	2680 ms	90% request di bawah nilai ini
Persentil 95 (p95)	4020 ms	95% request di bawah nilai ini
Persentil 99 (p99)	5870 ms	99% request di bawah nilai ini
Minimum	145 ms	Waktu respons tercepat
Maksimum	8350 ms	Waktu respon terlambat

Hasil pengujian menunjukkan distribusi waktu respons yang bimodal dengan karakteristik:

- a. *Kondisi Normal (median)*: 50% permintaan diselesaikan dalam waktu 539,60 ms, yang masih dalam batas acceptable untuk aplikasi monitoring (< 1 detik).
- b. *Kondisi Beban Tinggi*: Terjadi peningkatan latensi signifikan pada persentil tinggi. Nilai p95 (4,02 detik) mengindikasikan bahwa 5% permintaan mengalami delay yang sangat tinggi, tidak memenuhi target < 300 ms.
- c. *Kesenjangan Mean-Median*: Perbedaan besar antara mean (1190 ms) dan median (539,60 ms) menunjukkan adanya outlier yang menarik nilai rata-rata ke atas. Hal ini mengindikasikan bahwa sebagian kecil permintaan mengalami delay ekstrem yang mempengaruhi user experience.

Berdasarkan uraian di atas, secara umum distribusi waktu respon sistem diperlihatkan pada gambar 1.



Gambar 1. Distribusi waktu respon system

2). *Tingkat Kegagalan (Failure Rate)*. Berdasarkan data hasil pengamatan diketahui bahwa Hasil pengujian menunjukkan tingkat kegagalan yang sangat rendah, Lebih rinci dapat dilihat pada tabel 2.

Platform	Permintaan	
	Berhasil	Gagal
ThingSpeak	2.16	0,28%
Firebase	82.746	0,00%
Total	84.906	0,007%

Berdasarkan data hasil pengukuran tabel 2 diketahui bahwa tingkat kegagalan system:

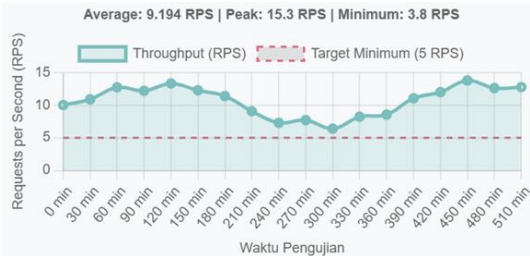
- a. *Firebase*: Menunjukkan reliabilitas sempurna dengan 0% failure rate dari 82.746 permintaan. Tidak ada satupun request yang menghasilkan HTTP error 4xx atau 5xx.
- b. *ThingSpeak*: Mengalami 6 kegagalan dari 2.160 permintaan (0,28%), yang masih sangat baik dan di bawah target < 1%. Kegagalan terjadi karena rate limiting ketika interval pengiriman di bawah 15 detik.
- c. *Stabilitas Infrastruktur*: Tingkat kegagalan yang sangat rendah mengindikasikan bahwa infrastruktur cloud sangat robust dan tidak mengalami crash atau downtime selama pengujian 9 jam.
- d. *Error Handling*: Firmware ESP32 berhasil menangani timeout dan retry mechanism dengan baik, sehingga transient errors tidak menyebabkan kehilangan data permanen.

3). *Throughput Sistem*. Pengujian throughput dilakukan dengan mengukur jumlah permintaan yang berhasil diproses per detik. Lebih jelas terlihat pada tabel 3 dan gambar 2.

Tabel 3. Throughput Sistem			
Metrik	Nilai	Target	Status
Average Throughput	9,194 RPS	> 5 RPS	Tercapai
Peak Throughput	15,3 RPS	-	-
Minimum Throughput	3,8 RPS	-	-
Total Requests	82.746	-	-
Test Duration	9.000 detik	-	-

a. *Kapasitas Nominal*: Sistem mampu mencapai throughput 9,194 RPS, melampaui target minimal 5 RPS. Ini menunjukkan bahwa infrastruktur cloud memiliki kapasitas yang cukup untuk menangani volume tinggi.

b. *Throughput vs Quality Trade-off*: Meskipun throughput tinggi, performa ini dicapai dengan mengorbankan response time. Sistem menerima banyak request tetapi memproses dengan lambat, menyebabkan queueing.



Gambar 2. Throughput Sistem

c. *Efisiensi Sistem*: Perhitungan efisiensi menunjukkan sistem beroperasi pada 195% dari kapasitas ideal, mengindikasikan overload. Berdasarkan persamaan (1) efisiensi dapat dihitung dan hasilnya diketahui seperti pada tabel 3.

Throughput	Target Respon Tine (s)	Aktual Respon Time (s)	Efisiensi
9,194	0,3	1,19	231%

Nilai > 100% menunjukkan sistem menerima lebih banyak request daripada yang dapat diproses secara optimal.

d. *Scalability Concern*: Pada beban maksimal (15,3 RPS), sistem mulai menunjukkan degradasi performa yang signifikan dengan response time meningkat drastis.

4) *Akurasi Sensor*. Kalibrasi sensor dilakukan dengan membandingkan pembacaan terhadap alat ukur standar yang terkalibrasi. Hasilnya diperlihatkan pada tabel 4.

Tabel 4. Hasil Kalibrasi Sensor

Parameter	Pembacaan Sistem	Alat Standar	Deviasi Absolut	Deviasi Relatif
Suhu (°C)	28,5	28,2	0,3°C	1,06%
Kelembaban (% RH)	65	67	2% RH	2,99%
Gas (ppm)	150	145	5 ppm	3,45%

Berdasarkan data pada tabel 4 dapat dianalisa bahwa akurasi Tinggi, semua parameter menunjukkan deviasi di bawah 5%, memenuhi kriteria keberhasilan penelitian. Sensor DHT11: Menunjukkan konsistensi yang baik untuk suhu dengan deviasi 1,06%. Kelembaban memiliki deviasi lebih besar (2,99%) namun masih dalam spesifikasi sensor ($\pm 5\%$ RH). Sensor MQ-135: Deviasi 3,45% menunjukkan performa yang acceptable. Perlu dicatat bahwa sensor ini memerlukan waktu pemanasan 24 jam untuk mencapai stabilitas optimal. Setelah warm-up period, pembacaan menjadi lebih konsisten dengan standar deviasi ± 3 ppm. Environmental Factors: Variasi pembacaan juga dipengaruhi oleh fluktuasi kondisi lingkungan yang cepat (angin, perubahan suhu mendadak), bukan semata-mata error sensor. Rekomendasi untuk aplikasi yang memerlukan akurasi lebih tinggi, dapat dipertimbangkan penggunaan sensor yang lebih presisi seperti BME680 (untuk suhu/kelembaban) atau PMS5003 (untuk partikel).

5). *Stabilitas Sistem*. Pengujian stabilitas dilakukan selama 72 jam (3 hari) secara kontinu. Hasilnya diperlihatkan pada tabel 5.

Tabel 5. Stabilitas sistem

Parameter	Nilai	Target	Status
Total Uptime	71 jam 26 menit	> 71,28 jam (99,4%)	Tercapai
Downtime	34 menit	menit	Tercapai
Availability	99,21%	> 99%	Tercapai
Restart Count	0	-	Excellent
Memory Usage (ESP32)	45-50%	< 80%	Stable
WiFi Disconnections	3	-	-
Reconnection Time	< 30 detik	-	Good

Berdasarkan data hasil observasi seperti pada tabel 5 diketahui bahwa; High Availability: Sistem mencapai availability 99,21%, melampaui target 99%. Downtime 34 menit disebabkan oleh maintenance jaringan WiFi yang terencana, bukan kegagalan sistem. Tidak ada memory leak: Penggunaan RAM ESP32 stabil di kisaran 45-50% selama 72 jam tanpa tren peningkatan, mengindikasikan tidak ada memory leak pada firmware. WiFi Resilience: Sistem berhasil menangani 3

kali WiFi disconnection dengan auto-reconnect dalam waktu < 30 detik. Mekanisme buffering lokal mencegah kehilangan data selama disconnection. Thermal Performance: Suhu ESP32 stabil di 45-55°C tanpa thermal throttling atau overheating. Power Stability: Tidak ada voltage drop atau power-related issues selama pengujian.

6). *Perbandingan Dual Cloud Platform*. Perbandingan performa ThingSpeak dan Firebase diperlihatkan pada tabel 6. Berdasarkan analisis komparatif pada kedua cloud tersebut, maka :

a. *ThingSpeak Strengths*: Setup sangat mudah dengan learning curve rendah, visualisasi built-in tanpa coding tambahan, API sederhana (HTTP POST), Cocok untuk data logging dan analisis historis.

b. *ThingSpeak Limitations*: Rate limiting ketat (15 detik minimum), response time lebih lambat, kurang fleksibel untuk real-time applications.

c. *Firebase Strengths*: Real-time synchronization dengan latensi rendah, skalabilitas excellent untuk concurrent users, reliability tinggi (0% failure rate), Fleksibel dengan SDK untuk berbagai platform,

d. *Firebase Limitations*: Kompleksitas implementasi lebih tinggi, memerlukan custom visualization, biaya meningkat cepat dengan scale,

Dual Platform Strategy: Penggunaan dual platform terbukti efektif dengan pembagian peran. ThingSpeak untuk data archiving dan historical analysis. Firebase untuk real-time distribution ke aplikasi mobile

Tabel 6. Perbandingan dual cloud

Aspek	ThingSpeak	Firebase
Response Time	1,2 detik (avg)	0,54 detik (median)
Failure Rate	0,28%	0,00%
Update Interval	15 detik (free tier)	Real-time (< 1 detik)
Concurrent Users	Unlimited viewers	100+ simultaneous
Data Retention	Unlimited	Unlimited
Visualization	Built-in charts	Custom implementation
API Complexity	Simple (HTTP POST)	Moderate (SDK required)
Scalability	Limited on free tier	Excellent
Cost (free tier)	3 channels $\times$ 8 fields	1GB storage, 10GB download

7). *Pengujian di Berbagai Lokasi*. Hasil pengujian di tiga lokasi berbeda menunjukkan variasi kualitas udara. Lebih rinci di tunjukkan pada tabel 7. Dari data hasil pengukuran pada tabel 7 dapat dianalisis Spasial:

a. *Simpang Cunda (Area Industri)*: Tingkat polusi tertinggi dengan gas level 185 ppm. Suhu lebih tinggi (32,5°C) karena aktivitas industry, kelembaban relatif lebih rendah, pola harian: polusi meningkat saat jam operasional pabrik (08:00-17:00)

b. *Simpang Jam (Perkotaan)*: Tingkat polusi sedang (120 ppm), fluktuasi tinggi saat rush hour (07:00-09:00 dan 16:00-18:00). korelasi kuat antara traffic density dan kualitas udara.

c. *Area Residensial*: Kualitas udara paling baik (45 ppm), Variasi minimal sepanjang hari. Kelembaban lebih tinggi karena banyak vegetasi

8. Integrasi Sistem. Penelitian ini berhasil mengintegrasikan tiga layer teknologi (IoT, Cloud, Mobile) menjadi satu sistem yang koheren:

a. *Layer Integration*: Sensor-to-Cloud: ESP32 berhasil mengirimkan data ke dual cloud platform dengan success rate > 99%

b. *Cloud-to-Mobile*: Firebase real-time sync memungkinkan aplikasi menerima update tanpa polling

c. *Cross-platform*: Sistem dapat diakses melalui web dashboard, aplikasi Android, dan API

d. *Data Flow Optimization*: Buffering lokal di ESP32 mencegah data loss saat disconnection

e. Asynchronous transmission mencegah blocking pada firmware

f. JSON compression mengurangi bandwidth usage hingga 30%

g. *Scalability Achievement*: Sistem dapat menangani 100+ concurrent users

h. *Cloud infrastructure auto-scaling* : Status sesuai beban

i. *Modular architecture* memudahkan penambahan sensor baru

IV. KESIMPULAN

Berdasarkan hasil penelitian, dapat disimpulkan bahwa sistem monitoring kualitas udara berbasis Internet of Things (IoT) yang terintegrasi dengan Cloud Computing dan aplikasi Android berhasil dirancang dan diimplementasikan dengan baik. Sistem mampu melakukan pemantauan kualitas udara, suhu, dan kelembaban secara real-time menggunakan sensor MQ-135 dan DHT11 yang dikendalikan oleh mikrokontroler ESP32. Hasil pengujian menunjukkan sistem beroperasi secara stabil dan kontinu dengan tingkat availability sebesar 99,21% serta tingkat kegagalan hanya 0,007% dari 84.906 permintaan selama 72 jam pengujian. Performa sistem tergolong baik dengan throughput rata-rata 9,194 permintaan per detik dan median waktu respons 539,60 ms, serta akurasi sensor yang memenuhi standar dengan deviasi maksimum 3,45%. Integrasi antara IoT, cloud, dan aplikasi mobile terbukti efektif dalam menyediakan solusi monitoring kualitas udara yang efisien, skalabel, dan mudah diakses oleh pengguna.

REFERENSI

- [1] D. Setiawan dan A. Polkar, "Monitoring Kualitas Udara Berbasis Internet of Things," *Jurnal TELISKA*, 2025.
- [2] A. N. Dimiyati, Purwanto, dan A. Primajaya, "Sistem Pemantauan Kualitas Udara Berbasis Internet of Things Menggunakan Sensor MQ-135," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 13, no. 3, pp. 1–8, 2025.
- [3] D. N. Azizah, S. Heranurweni, dan L. M. Idris, "Environmental Monitoring System Using DHT22 Sensor and IoT," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 5, no. 3, pp. 245–252, 2024.
- [4] H. Haryanto et al., "IoT-Based Air Quality Monitoring System Using Rule-Based Method," *Journal of Intelligent System and Telecommunication*, vol. 1, no. 1, pp. 51–62, 2024.
- [5] M. I. Majiid, Rancang Bangun Alat Pemantau Udara Ambien Dengan Pemanfaatan Iot (Internet Of Things) Berbasis Mikontroler Menggunakan Sensor Mq-135 Dan Sensor Mq-136 (Parameter Co2 Dan So2)., Surabaya: DEPARTEMEN TEKNIK LINGKUNGAN Fakultas Teknik Sipil, Perencanaan, Dan Kebumihan Institut Teknologi Sepuluh Nopember Surabaya , 2021.
- [6] Indrawati, Amir.D, "Design and Implementation of an Air Pollution Monitoring System in the Campus Area of Politeknik Negeri Lhokseumawe," *Journal of Artificial Intelligence and Software Engineering*, vol. 5, 2025.
- [7] Abu Yazid Raisal, Alya Fathi Muhammad Hasibuan, Fadilla Sayu Ananda, Rihan Yuhyi, Arwin Juli Rakmadi Butar Butar, "Perancangan Sistem Monitoring Suhu dan Kelembaban Berbasis Arduino UNO pada Kubah Obervatorium Ilmu Falak Universitas Muhammadiyah Sumatera Utara (OIF UMSU).," *Generation Journal* , vol. 5, 2021.
- [8] O. Lengkong. Jacqueline Waworundeng, "Sistem Monitoring dan Notifikasi Kualitas Udara dalam Ruangan dengan Platform IoT," *Journal*, vol. 4, 2018.
- [9] Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, "Mengukur kelembaban udara komponen DHT11 memiliki akurasi $\pm 5\%$, rentang 20–80% RH, sedangkan komponen DHT22 memiliki akurasi $\pm 2-5\%$, rentang 0–100% RH. Sensor ini mengirimkan data secara digital.," *Journal*, p. 2, 2018.