

Implementasi Sistem Monitoring Greenhouse Berbasis IoT dan Evaluasi Kinerja Server Menggunakan Node-RED dan MQTT

Rudi Hermawan^{1*)}, Siti Maesaroh², Dewanto Rosian Adhy³

^{1,2,3} Teknik Informatika, Fakultas Teknik, Universitas Mayasari Bakti, Tasikmalaya

¹ Jalan Tamansari Blok Rahayu I, RT.004 / RW.004, Kel. Sukahurip, Kec. Tamansari, Kota Tasikmalaya 46191
email: ¹rudihermawan@mayasaribakti.ac.id, ²siti.maesaroh40@gmail.com, ³dewanto_ra@mayasaribakti.ac.id

Abstract This study develops an Internet of Things (IoT)-based greenhouse monitoring system that operates entirely on a local server using an edge computing approach. The system is designed to address challenges in rural areas with limited network infrastructure, such as unstable internet connections. Utilizing Node-RED as a visual platform for data integration and MQTT as a lightweight communication protocol, the system can collect, store, and display environmental data in real time. Additionally, it can automatically control devices such as water pumps based on predefined logic. The implementation is carried out on a virtual machine environment running Debian 12 OS, with data stored locally in MySQL. Testing results indicate that the system can operate stably over extended periods even without internet connectivity and provides quick responses to environmental changes. This approach offers an efficient and autonomous solution for modern agricultural environmental monitoring, especially in regions with limited digital infrastructure.

Keywords --- Internet of Things, greenhouse, edge computing, Node-RED, MQTT, environmental monitoring, local server

Abstrak --- Penelitian ini mengembangkan sistem monitoring greenhouse berbasis Internet of Things (IoT) yang dijalankan sepenuhnya di server lokal dengan pendekatan edge computing. Sistem ini dirancang untuk mengatasi tantangan di wilayah pedesaan yang memiliki keterbatasan infrastruktur jaringan, seperti koneksi internet yang tidak stabil. Dengan memanfaatkan Node-RED sebagai platform visual untuk integrasi data dan protokol MQTT untuk komunikasi data ringan, sistem ini mampu mengumpulkan, menyimpan, dan menampilkan data lingkungan secara real-time. Selain itu, sistem dapat mengontrol perangkat seperti pompa air secara otomatis berdasarkan logika kontrol yang ditentukan. Implementasi dilakukan pada lingkungan virtual machine dengan sistem operasi Debian 12, dan data disimpan secara lokal menggunakan MySQL. Hasil pengujian menunjukkan bahwa sistem mampu beroperasi secara stabil dalam jangka waktu lama, meskipun tanpa koneksi internet, serta memberikan respon cepat terhadap perubahan kondisi lingkungan. Pendekatan ini menawarkan solusi efisien dan mandiri untuk pengawasan lingkungan pertanian modern, khususnya di daerah dengan keterbatasan infrastruktur digital.

Kata Kunci ---- Internet of Things, greenhouse, edge computing, Node-RED, MQTT, monitoring lingkungan, server lokal

I. PENDAHULUAN

Pertanian modern saat ini menghadapi tantangan besar dalam meningkatkan produktivitas secara efisien, terutama di wilayah pedesaan yang minim akses terhadap infrastruktur teknologi. Salah satu pendekatan yang mulai banyak diterapkan adalah penggunaan sistem greenhouse berbasis Internet of Things (IoT) untuk memantau dan mengontrol variabel lingkungan seperti suhu, kelembaban, intensitas cahaya, serta pH tanah [1]. Namun, implementasi sistem IoT yang sepenuhnya bergantung pada layanan cloud sering kali tidak sesuai untuk daerah rural karena beberapa kendala nyata di lapangan.

Permasalahan utama yang sering muncul antara lain: keterbatasan infrastruktur jaringan internet di wilayah pedesaan, rendahnya kemampuan petani dalam mengoperasikan sistem berbasis cloud, dan tingginya biaya operasional untuk menjaga koneksi serta layanan cloud secara berkelanjutan. Hal ini mendorong perlunya pendekatan alternatif yang dapat beroperasi secara mandiri, hemat biaya, dan tetap andal dalam melakukan fungsi monitoring serta kontrol lingkungan pertanian secara real-time.

Salah satu solusi yang potensial adalah penerapan arsitektur edge computing, yakni dengan memindahkan proses pengolahan dan pengambilan keputusan awal ke server lokal yang terhubung langsung dengan sensor dan aktuator. Dengan pendekatan ini, sistem dapat tetap berjalan meskipun koneksi internet terputus, serta memungkinkan respon otomatis terhadap kondisi lingkungan tanpa menunggu instruksi dari server pusat atau cloud. Edge computing juga membuka peluang penggunaan perangkat keras hemat daya seperti mini PC atau virtual machine di laptop/lokal server dengan spesifikasi rendah.

Penelitian ini mengusulkan desain dan implementasi sistem monitoring greenhouse berbasis IoT yang menggunakan Node-RED sebagai alat integrasi alur data, serta MQTT sebagai protokol komunikasi data ringan antara sensor, server, dan antarmuka pengguna (UI). Sistem ini tidak hanya mampu mengumpulkan dan menyimpan data dari sensor secara lokal, tetapi juga menampilkan data secara real-time melalui dashboard dan mengendalikan perangkat seperti pompa berdasarkan logika yang diatur pada sisi server.

Tujuan utama dari penelitian ini adalah:

1. Merancang sistem monitoring greenhouse berbasis IoT yang dapat berjalan secara lokal menggunakan arsitektur edge computing.
2. Mengimplementasikan protokol MQTT dan platform Node-RED sebagai alat utama dalam pengolahan dan visualisasi data.
3. Mengevaluasi kinerja server lokal dalam menangani data sensor secara real-time serta kestabilannya saat dijalankan secara terus-menerus.

Dengan pendekatan ini, penelitian diharapkan dapat memberikan solusi yang relevan dan aplikatif bagi pengembangan pertanian cerdas di daerah-daerah dengan keterbatasan infrastruktur teknologi.

II. PENELITIAN YANG TERKAIT

Penggunaan teknologi Internet of Things (IoT) dalam bidang pertanian telah banyak dikaji dan dikembangkan sebagai bagian dari upaya menciptakan sistem pertanian presisi. Dalam sistem monitoring berbasis cloud diperkenalkan untuk memantau kelembaban tanah dan mengontrol irigasi otomatis. Meskipun sistem ini efektif dalam pengumpulan data dan

pengendalian jarak jauh, ketergantungan pada koneksi internet menjadi kendala signifikan di daerah dengan jaringan tidak stabil [2].

Sementara itu, pendekatan edge computing mulai dilirik sebagai solusi alternatif. Seperti pada penelitian sebelumnya, sistem berbasis edge computing dikembangkan untuk mendeteksi kondisi suhu dan kelembaban di peternakan ayam[3][4]. Pengolahan data dilakukan secara lokal menggunakan Raspberry Pi, sehingga memungkinkan sistem tetap berjalan meski tanpa koneksi internet. Penelitian ini menunjukkan efisiensi tinggi dari sisi respon waktu dan penghematan biaya cloud. Penelitian lain juga menunjukkan bahwa penerapan edge computing pada sistem pemantauan tanaman dapat mengurangi latensi dan meminimalkan beban jaringan, sekaligus menjaga keberlanjutan monitoring meskipun tidak terhubung ke internet [5]

Node-RED digunakan sebagai alat visual pemrograman untuk integrasi sensor dan aktuator dalam sistem smart home. Node-RED terbukti memudahkan integrasi antar perangkat dan mampu menangani alur data real-time secara ringan. Penelitian ini memberikan dasar yang kuat dalam memilih Node-RED sebagai platform utama dalam penelitian ini [6]

Selain itu, protokol MQTT telah banyak diakui karena efisiensinya dalam komunikasi data IoT. MQTT dirancang untuk komunikasi berbasis publish-subscribe yang ringan, sehingga sangat sesuai untuk perangkat IoT dengan keterbatasan bandwidth dan daya pemrosesan[7]

Berbagai penelitian di atas menunjukkan adanya tren pergeseran dari sistem berbasis cloud penuh ke sistem hybrid atau lokal (edge). Namun, belum banyak penelitian yang secara eksplisit mengevaluasi kinerja server lokal dalam jangka panjang serta integrasi penuh antara MQTT, Node-RED, dan sistem monitoring berbasis greenhouse dalam konteks edge computing. Oleh karena itu, penelitian ini bertujuan untuk mengisi celah tersebut dengan menggabungkan elemen praktis dan analitis [8].

III.METODE PENELITIAN

A. Rancangan Sistem

Penelitian ini menggunakan pendekatan eksperimen implementatif, dengan merancang dan membangun sistem monitoring greenhouse berbasis IoT yang dijalankan sepenuhnya pada server lokal (edge server). Sistem ini dirancang untuk membaca data dari beberapa sensor (suhu, kelembaban, intensitas cahaya, dan pH) serta mengontrol aktuator berupa pompa air secara otomatis berdasarkan parameter tertentu.

Pengolahan data dilakukan secara lokal menggunakan platform Node-RED yang diinstal pada sistem operasi Debian 12 (64-bit) yang berjalan di dalam lingkungan virtual machine (VM). Server lokal ini bertugas untuk:

1. menerima data dari sensor melalui protokol MQTT
2. menyimpan data ke MySQL
3. menampilkan data dan kontrol melalui dashboard web lokal
4. mengatur logika kontrol otomatis pompa

B. Arsitektur Sistem

Sistem terdiri atas tiga komponen utama:

1. Perangkat Sensor dan Aktuator
Terdiri dari NodeMCU ESP8266 yang terhubung ke sensor suhu, kelembaban, cahaya, dan pH. Data dikirim melalui protokol MQTT ke broker lokal.
2. Broker MQTT (Mosquitto)

dijalankan di server lokal untuk menerima dan meneruskan pesan antara sensor dan dashboard/logic controller.

3. Server Edge (Node-RED + MySQL)

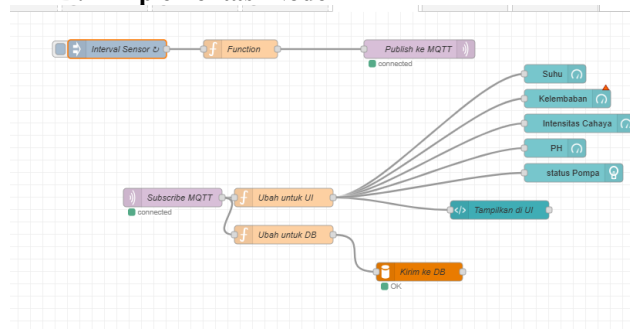
Digunakan untuk mengatur aliran data, menyimpan data, serta menyediakan dashboard UI untuk visualisasi dan kontrol.

C. Spesifikasi Perangkat

Server edge dijalankan dalam VM dengan spesifikasi:

1. Base Memory: 1469 MB
2. Video Memory: 20 MB
3. OS: Debian 12 64-bit
4. Host: Laptop Lenovo Ideapad
5. CPU: Intel Core i3-5005U @ 2.00GHz
6. RAM: 4 GB

D. Implementasi Node-RED



Gambar 1. Rancangan Flow Node Red

Node-RED digunakan untuk membangun alur data (flow) secara visual. Alur ini mencakup:

1. Node MQTT-in untuk menerima data dari sensor
2. Node function untuk parsing dan pemrosesan data
3. Node MySQL untuk menyimpan data ke database
4. Node UI gauge/chart/text untuk visualisasi parameter
5. Node logic untuk pengaturan status pompa

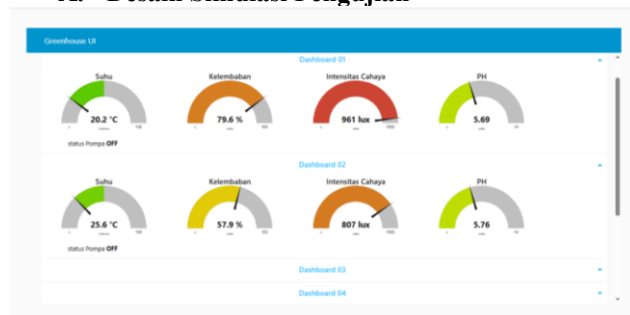
E. Pengujian Kinerja Server

Evaluasi dilakukan untuk menilai keandalan dan performa server lokal, terutama dalam:

1. Kemampuan menerima dan memproses data secara real-time
2. Latensi antara data masuk dan aksi (otomatisasi pompa)
3. Konsistensi tampilan UI
4. Penggunaan sumber daya sistem (CPU, memori, storage, network, mqtt dan Mysql)

IV.HASIL DAN PEMBAHASAN

A. Desain Simulasi Pengujian



Gambar 2. Tampilan Sensor Data Melalui UI Node RED

Untuk mengevaluasi kinerja sistem monitoring greenhouse berbasis IoT, dilakukan simulasi menggunakan 10 flow Node-RED yang masing-masing mensimulasikan satu node sensor. Setiap flow mengirimkan data suhu, kelembaban, pH, intensitas cahaya, dan status pompa ke broker MQTT setiap 10, 7 dan 5 detik. Data ini kemudian dikonsumsi oleh Node-RED untuk ditampilkan di dashboard dan disimpan ke database MySQL lokal.

Topologi sistem dirancang menggunakan pendekatan publisher-subscriber dengan protokol MQTT, di mana inject node bertindak sebagai publisher dan mqtt in node sebagai subscriber. Komponen backend mencakup:

- Node-RED sebagai orchestrator dan UI builder
- Mosquitto sebagai broker MQTT
- MySQL sebagai sistem penyimpanan data
- Dashboard UI berbasis ui_template, ui_gauge, dan ui_text

B. Metodologi Pengumpulan dan Analisis Data

Untuk mengamati performa sistem secara real-time dan komprehensif, digunakan tools System Activity Reporter (sar) yang merupakan bagian dari paket sysstat di Linux. sar memiliki kemampuan untuk mencatat berbagai metrik sistem seperti penggunaan CPU, memori, swap, disk, dan jaringan secara berkala dan otomatis.

sar dipilih sebagai alat pencatat log karena memiliki sejumlah keunggulan, yaitu:

- Multidimensional Monitoring

sar mendukung pencatatan berbagai aspek performa sistem dalam satu tool, mencakup CPU (-u), memori (-r), swap (-B), disk (-d), dan jaringan (-n DEV), sehingga sangat efisien dalam pengumpulan data terintegrasi.

- Real-time dan Interval Presisi Tinggi

Tool ini memungkinkan pengambilan data dengan interval rendah, misalnya setiap 1 detik, yang ideal untuk pengujian performa pada sistem real-time seperti arsitektur **edge computing** dalam IoT

- Ringan dan Noninvasif

Dibandingkan tools lain seperti top atau htop, sar memiliki jejak memori (footprint) yang sangat kecil dan tidak membebani sistem saat proses pencatatan, sehingga hasil pengamatan tidak terdistorsi oleh keberadaan tool itu sendiri.

- Kompatibilitas Tinggi

sar merupakan tool standar di hampir semua distribusi Linux, menjadikannya solusi lintas sistem operasi yang mudah direplikasi dan didokumentasikan.

Logging dilakukan dengan menjalankan skrip shell otomatis selama server berjalan. Setelah semua proses pencatatan selesai, data disimpan dalam file .txt dan dianalisis menggunakan Python dengan bantuan library Pandas untuk analisis statistik dan Matplotlib/Seaborn untuk pembuatan grafik.

C. Analisis Kinerja Sistem

Setelah data log terkumpul menggunakan sar, dilakukan analisis terhadap enam komponen utama sistem: CPU, memori, swap, disk, jaringan, mysql dan broker MQTT. Analisis dilakukan untuk mengevaluasi efisiensi, stabilitas, serta kemampuan sistem dalam menangani beban kerja dari 10 flow Node-RED yang mengirim data secara periodik dengan variasi interval 10, 7, dan 5 detik.

Hasil analisis disajikan dalam bentuk tabel statistik dan grafik visualisasi untuk memberikan gambaran yang lebih menyeluruh mengenai performa sistem selama simulasi.

D. Evaluasi Kinerja CPU

TABEL I
RANGKUMAN KINERJA CPU

Metrik	Nilai Rata-rata	Maksimum	Interpretasi
%user	0.2%	3.3%	Beban ringan meskipun 10 flow aktif
%system	1.0%	3.3%	Sistem stabil
%idle	>95%	100%	CPU sebagian besar waktu tidak digunakan

Tabel diatas menunjukkan bahwa aktivitas CPU selama simulasi berlangsung berada pada tingkat yang sangat rendah. Hal ini menunjukkan efisiensi proses dari arsitektur event-driven yang digunakan oleh Node-RED dan protokol MQTT. Persentase waktu idle yang tinggi (>95%) membuktikan bahwa sistem memiliki kapasitas yang cukup untuk menangani trafik IoT tanpa mengalami overloading.

Setelah memastikan bahwa CPU bekerja secara efisien dilakukan juga analisis apakah memori juga dikelola dengan baik, terutama dalam sistem edge yang memiliki keterbatasan RAM. Berikut adalah tabel hasil pengujian dari penggunaan memori pada saat aktivitas simulasi di jalakan.

E. Evaluasi Penggunaan Memori

TABEL II
PENGUNAAN MEMORI

Metrik	Nilai	Keterangan
%memused	36.4%	Stabil, tanpa lonjakan memori
Cached	±490 MB	Digunakan untuk optimalisasi akses data
%commit	±17.4%	Beban komitmen memori rendah

Penggunaan memori menunjukkan kestabilan yang signifikan dengan rata-rata di bawah 40%. Ini mengindikasikan bahwa sistem mampu menjalankan semua proses tanpa mengalami tekanan memori, serta tidak bergantung pada swap untuk alokasi memori tambahan. Cached memory yang cukup besar menunjukkan efisiensi sistem dalam mengelola akses data secara cepat.

F. Evaluasi Penggunaan Swap

Untuk memastikan bahwa memory benar benar tidak menggunakan swap, maka dilakuakn juga proses pengujian terhadap swap memory yang dijadikan sebagai cadangan. Berikut adalah hasil pengujian dari penggunaan swap pada saat aktivitas pengujian dilaksanakan :

TABEL III
AKTIVITAS SWAP

Metrik	Nilai	Interpretasi
pgpgin/s	0.00	Tidak ada penggunaan swap
pgpgout/s	0.00–72.00	Sporadis, sangat ringan
majflt/s	0.00	Tidak ada page fault mayor

Tabel diatas menunjukan nilai aktivitas swap sangat rendah hingga mendekati nol, bahkan pada saat semua flow aktif. Tidak adanya major fault mengindikasikan bahwa sistem memiliki ketersediaan memori yang cukup, dan tidak mengalami kekurangan sumber daya yang mengakibatkan page swapping.

G. Evaluasi Pemanfaatan Disk

Karena tidak terjadi tekanan pada memori maupun penggunaan swap, maka perhatian berikutnya diarahkan pada aktivitas

input/output (I/O) terhadap media penyimpanan. Hal ini menjadi krusial mengingat dalam sistem IoT berbasis edge computing, data sensor yang dikirimkan secara periodik perlu dicatat ke dalam sistem penyimpanan lokal secara berkelanjutan.

Dengan beban kerja dari 10 flow Node-RED yang aktif mengirim data tiap 10, 7, dan 5 detik, diharapkan terjadi aktivitas penulisan (write) yang konsisten ke dalam database lokal MySQL. Oleh karena itu, pemantauan terhadap aktivitas disk menjadi penting untuk memastikan bahwa proses pencatatan data tidak menimbulkan bottleneck atau latensi tinggi yang dapat mengganggu performa sistem secara keseluruhan.

TABEL IV
PEMANFAATAN DISK

Metrik	Rata-rata	Maksimum	Keterangan
wkB/s	0-72.0 kB/s	72.0	Penulisan terjadi saat penyimpanan data sensor ke DB
%util	0.4%	1.58%	Utilisasi disk rendah, tidak menunjukkan bottleneck

Berdasarkan hasil pencatatan log menggunakan `strace`, aktivitas tulis ke disk (kB/s) terjadi secara sporadis namun konsisten selama simulasi berlangsung. Rata-rata nilai write berkisar antara 0 hingga 72 kB/s, yang umumnya terjadi saat proses INSERT ke database MySQL dilakukan oleh Node-RED. Nilai ini tergolong sangat ringan, dan bahkan pada titik beban puncak (72.0 kB/s), sistem tidak menunjukkan gejala penurunan performa atau delay.

Lebih lanjut, nilai persentase utilisasi disk (%util) yang sangat rendah, yakni hanya berkisar 0.4% hingga maksimum 1.58%, menunjukkan bahwa subsistem penyimpanan memiliki kapasitas jauh di atas kebutuhan sistem saat ini. Hal ini sangat penting karena pada banyak sistem edge dengan hardware terbatas (misalnya menggunakan SSD SATA atau eMMC), bottleneck I/O sering menjadi penyebab utama kegagalan dalam pemrosesan data real-time.

Efisiensi pemanfaatan disk juga memperlihatkan bahwa arsitektur sistem yang memisahkan proses pengambilan, pengolahan, dan penyimpanan data (`publish-process-store`) dapat berjalan secara sinergis tanpa saling mengganggu performa satu sama lain.

H. Evaluasi Kinerja Jaringan dan Broker MQTT

Salah satu komponen vital dalam sistem monitoring berbasis IoT adalah efektivitas komunikasi data antar node. Dalam implementasi ini, komunikasi dilakukan secara nirkabel dalam jaringan lokal menggunakan protokol MQTT yang berjalan di atas protokol TCP/IP. Oleh karena itu, penting untuk mengevaluasi dua aspek kunci yang sangat menentukan kelancaran sistem:

1. Aktivitas jaringan (network throughput) sebagai media transportasi antar node dan broker
2. Performa broker MQTT (Mosquitto) sebagai pusat pengelola lalu lintas pesan (`publish` dan `subscribe`)

Pengujian dilakukan dengan mencatat data lalu lintas jaringan menggunakan perintah `strace` DEV, sedangkan performa Mosquitto dianalisis dengan memantau konsumsi CPU dan memori prosesnya. Hasilnya disajikan dalam dua tabel berikut.

TABEL V

AKTIVITAS JARINGAN SISTEM

Metrik	Rata-rata	Interpretasi
--------	-----------	--------------

rxkB/s	≈ 0.06	Rata-rata data masuk dari broker ke subscriber
txkB/s	≈ 0.52	Rata-rata data keluar dari flow ke broker

hasil pencatatan menunjukkan bahwa aktivitas jaringan yang ditimbulkan oleh sistem monitoring sangat ringan, dengan nilai rata-rata hanya sekitar 0.06 kB/s untuk penerimaan data dan 0.52 kB/s untuk pengiriman data. Hal ini merupakan konsekuensi positif dari penggunaan protokol MQTT, yang memang dirancang untuk komunikasi berbasis event dengan payload kecil.

Efisiensi jaringan ini menunjukkan bahwa sistem ini sangat cocok digunakan dalam kondisi jaringan terbatas, seperti di daerah pedesaan, perkebunan, atau greenhouse yang tidak memiliki infrastruktur internet kuat. Dengan total beban jaringan yang tidak melebihi 1 kB/s, sistem ini tidak memerlukan bandwidth tinggi, dan tetap mampu bekerja secara real-time.

Namun demikian, aktivitas jaringan yang ringan harus didukung oleh kinerja broker MQTT yang andal, karena semua data dikirim dan diteruskan melalui satu titik pusat yaitu Mosquitto broker. Oleh karena itu, bagian berikut ini membahas analisis terhadap performa proses broker MQTT secara langsung.

TABEL VI
KINERJA BROKER MQTT (MOSQUITTO)

Metrik	Rata-rata	Maksimum	Interpretasi
Penggunaan CPU	0.06%	38.0%	Sangat efisien, lonjakan kecil saat beban puncak
Penggunaan Memori	0.55%	0.55%	Stabil, tidak mengalami peningkatan drastis

Broker Mosquitto menjalankan fungsinya dengan sangat stabil dan efisien. Rata-rata penggunaan CPU hanya sekitar 0.06%, menunjukkan bahwa proses ini tidak memberatkan sistem meskipun menangani pengiriman data dari 10 flow berbeda. Lonjakan hingga 38% hanya terjadi sesekali dan berlangsung sangat singkat, kemungkinan saat seluruh flow aktif secara bersamaan pada interval waktu yang sama.

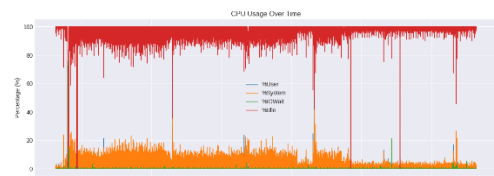
Penggunaan memori juga sangat rendah dan tidak menunjukkan fluktuasi berarti, yakni konstan di 0.55% dari total memori sistem. Stabilitas ini membuktikan bahwa Mosquitto cocok untuk digunakan dalam sistem edge computing yang membutuhkan keandalan komunikasi antar perangkat tanpa overutilization.

I. Visualisasi Grafik Kinerja Sistem

Untuk memperkuat hasil analisis numerik yang telah disajikan pada subbab sebelumnya, dilakukan pula visualisasi data log dalam bentuk grafik. Tujuan dari penyajian grafik ini adalah untuk:

- Memperlihatkan pola fluktuasi dan kestabilan performa sistem secara visual
- Membantu identifikasi spike, idle time, atau anomali performa selama pengujian
- Memberikan representasi intuitif bagi pembaca non-teknis

Data log yang digunakan diambil dari hasil pencatatan `strace`, dan grafik dihasilkan menggunakan library Matplotlib dan Seaborn di Python.



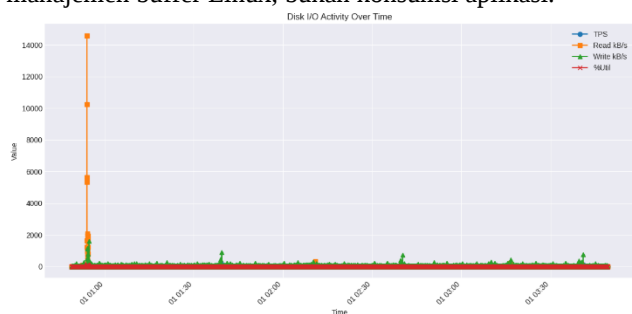
Gambar 3. Grafik CPU Usage Overtime

Grafik ini memperlihatkan perubahan nilai %user, %system, dan %idle selama proses simulasi berjalan. Nampak bahwa penggunaan CPU didominasi oleh status idle (>95%), dengan sedikit fluktuasi pada saat data dikirim secara bersamaan dari flow Node-RED.



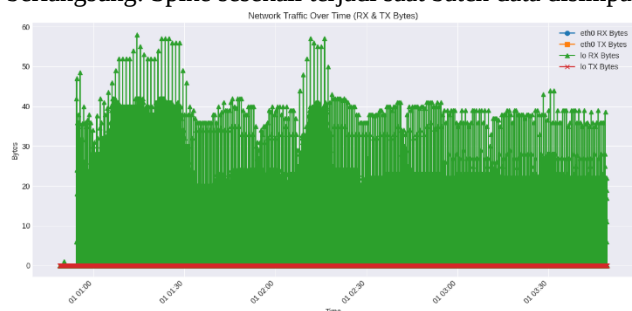
Gambar 4 Grafik Memory Usage Overtime

Grafik ini menunjukkan penggunaan memori total dan cache sepanjang waktu. Tidak ada lonjakan signifikan yang terjadi. Fluktuasi kecil yang tampak mencerminkan mekanisme manajemen buffer Linux, bukan konsumsi aplikasi.



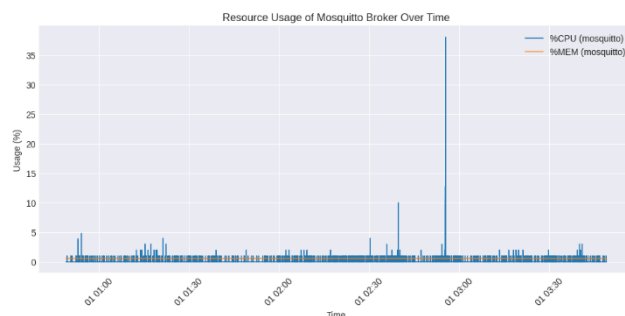
Gambar 5 Disk I/O Activity Over Time

Grafik ini menunjukkan kecepatan penulisan ke disk (kB/s) selama proses pencatatan data sensor ke dalam database berlangsung. Spike sesekali terjadi saat batch data disimpan.



Gambar 6 Network Traffic Over time

Grafik ini menampilkan nilai rxkB/s dan txkB/s yang menggambarkan aliran data antar node dan broker MQTT. Lalu lintas berlangsung secara konstan dan teratur.



Gambar 7. Grafik Beban CPU dari proses Mosquitto

Graphic ini menampilkan beban CPU yang digunakan oleh proses mosquitto. Terlihat spike sesekali, namun tetap dalam durasi pendek dan tidak memengaruhi performa sistem secara keseluruhan.

Berdasarkan hasil simulasi dan analisis performa sistem monitoring greenhouse berbasis IoT yang dibangun menggunakan Node-RED dan protokol MQTT, dapat disimpulkan bahwa sistem berjalan secara efisien dan stabil meskipun dijalankan dengan 10 flow aktif yang mengirim data dalam interval 10, 7, dan 5 detik. Pemanfaatan sumber daya sistem—seperti CPU, memori, disk, dan jaringan—berada dalam kategori rendah, tanpa menunjukkan gejala bottleneck atau degradasi performa. Aktivitas memori stabil tanpa tekanan swap, aktivitas disk hanya terjadi saat penulisan ke database, dan lalu lintas jaringan tetap ringan berkat efisiensi protokol MQTT. Broker Mosquitto juga menunjukkan performa optimal dengan penggunaan CPU dan memori yang sangat rendah serta stabil. Keseluruhan hasil ini menunjukkan bahwa arsitektur edge computing yang diterapkan terbukti efektif dan layak untuk mendukung sistem pemantauan pertanian modern secara real-time, bahkan pada perangkat dengan sumber daya terbatas dan koneksi jaringan yang minimal.

V.KESIMPULAN

Berdasarkan hasil simulasi dan analisis terhadap sistem monitoring greenhouse berbasis Internet of Things (IoT) menggunakan Node-RED dan protokol MQTT, dapat ditarik kesimpulan sebagai berikut:

1. Sistem berjalan secara efisien dan stabil bahkan dengan 10 flow aktif yang mengirim data dalam interval 10, 7, dan 5 detik. Tidak ditemukan gangguan performa pada CPU, memori, jaringan, maupun penyimpanan selama pengujian berlangsung.
2. Penggunaan sumber daya sistem tergolong ringan, dengan rata-rata pemanfaatan CPU di bawah 1%, memori dalam kondisi stabil, serta penggunaan swap memory yang nyaris nol. Hal ini menjadikan sistem sangat ideal untuk dijalankan pada perangkat edge dengan spesifikasi rendah.
3. Protokol MQTT terbukti efisien dalam menangani pertukaran data antar node. Rata-rata lalu lintas jaringan yang rendah membuktikan bahwa sistem ini mampu bekerja dalam jaringan lokal dengan bandwidth minimal tanpa mengorbankan kecepatan dan keandalan komunikasi.
4. Broker Mosquitto bekerja optimal, dengan konsumsi CPU dan memori yang sangat rendah serta tetap stabil meskipun menangani trafik dari banyak publisher secara simultan.
5. Arsitektur edge computing yang diterapkan efektif, memungkinkan sistem bekerja secara real-time tanpa ketergantungan pada cloud, sehingga sangat cocok untuk diterapkan di lokasi-lokasi terpencil yang memiliki keterbatasan akses internet.

Adapun penelitian selanjutnya dapat dilakukan beberapa hal seperti :

1. Melakukan pengujian jangka panjang untuk mengevaluasi ketahanan sistem terhadap penggunaan terus-menerus.
2. Mengembangkan sistem agar mendukung pengiriman data secara hybrid (lokal dan cloud) sebagai cadangan (redundansi data).
3. Menambah skala jumlah node untuk menguji batas skalabilitas dan beban maksimum sistem.
4. Menambahkan fitur notifikasi otomatis atau kendali pompa berbasis logika threshold dari data sensor.
5. Mengintegrasikan sistem keamanan dasar (seperti autentikasi MQTT dan enkripsi TLS) untuk skenario implementasi di lingkungan nyata.

UCAPAN TERIMA KASIH

Penulis menyampaikan terima kasih kepada Lembaga Penelitian dan Pengabdian kepada Masyarakat (LPPM) Universitas Mayasari Bakti atas dukungan pendanaan melalui hibah penelitian yang diberikan, serta kepada Universitas Mayasari Bakti yang telah menyediakan fasilitas dan infrastruktur pendukung selama proses penelitian berlangsung. Apresiasi juga disampaikan kepada seluruh rekan sejawat dan pihak-pihak terkait yang telah berkontribusi secara langsung maupun tidak langsung dalam mendukung kelancaran proses penelitian yang dilaksanakan.

DAFTAR PUSTAKA

- [1] R. Hermawan *et al.*, “Formulasi Penempatan Sensor Suhu Pada Green House,” *Power Elektronik: Jurnal Orang Elektro*, vol. 13, no. 3, pp. 342–350, 2024.
- [2] R. Dhega Ramanda, A. Sofwan, and D. M. Arfan, “PERANCANGAN APLIKASI ANTARMUKA SMART GREENHOUSE BERBASIS INTERNET OF THINGS (IOT) PADA PERANGKAT BERGERAK ANDROID,” Mar. 2020. [Online]. Available: <https://ejournal3.undip.ac.id/index.php/transient>
- [3] A. Rakhman *et al.*, “Pemanfaatan Narrowband IoT (NB-IoT) dalam Peningkatan Produktivitas Peternakan melalui Monitoring Otomatis,” *Jurnal pengembangan IT (JPIT)*, vol. 8, no. 3, 2023.
- [4] J. Elektronika and D. Komputer, “SISTEM MONITORING SUHU DAN KELEMBABAN KANDANG AYAM BERBASIS INTERNET OF THINGS (IOT),” vol. 16, no. 2, pp. 278–285, 2023, doi: 10.51903/elkom.v16i2.1153.
- [5] M. Hannats, H. Ichsan, and P. Korespondensi, “ANALISIS KINERJA JARINGAN SENSOR NIRKABEL UNTUK EDGE COMPUTING MENGGUNAKAN LORA SX1278,” *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, vol. 8, no. 5, pp. 887–894, Oct. 2021, doi: 10.25626/jtiik.202183631.
- [6] J. Homepage, G. A. Maizar, B. Hidayat, and O. B. Kharisma, “Home Electrical Intelligent Control using Node-Red Based on Raspberry Pi-3 B +,” *IJEERE: Indonesian Journal of Electrical Engineering and Renewable Energy*, vol. 1, no. 1, pp. 1–7, Jun. 2021.
- [7] A. Banks, E. Briggs, K. Borgendale, and R. Gupta, “MQTT Version 5.0,” Mar. 2019. Accessed: May 16, 2025.

[Online]. Available: [url{https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.pdf}](https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.pdf)
[8] C. Fredy Naa, “Greenhouse Monitoring System using ESP32, Raspberry Pi, MQTT and Node-RED,” *Jurnal Teknik Elektro dan Komputer*, vol. 11, no. 3, pp. 133–138, Dec. 2022.