

Sistem Monitoring Gudang Berbasis IoT dengan MQTT dan Rule-Based Notification

Carolus Hengki Dora Perdana¹, Aditya Prapanca²

^{1,2} Program Studi S1 Teknik Informatika, Universitas Negeri Surabaya

¹124051204256@mhs.unesa.ac.id

²adityaprapanca@unesa.ac.id

Abstrak— Monitoring kondisi Gudang penyimpanan barang yang masih dilakukan secara manual memiliki potensi adanya keterlambatan dalam mendeteksi perubahan suhu, kelembaban dan deteksi Gerakan sehingga dapat mempengaruhi kualitas dan keamanan barang yang disimpan. Penelitian ini bertujuan untuk mengembangkan sistem monitoring Gudang berbasis IoT menggunakan NodeMCU ESP8266, sensor DHT22 dan sensor PIR dengan arsitektur sistem terbentuk dari empat layer, yaitu *Data Acquisition Layer*, *Edge Processing Layer*, *Communication Layer* serta *Application Layer*. Sistem menggunakan protokol MQTT untuk komunikasi dengan broker Thinger.io dan HTTP untuk pengiriman notifikasi Telegram. Kebaruan penelitian terletak pada penerapan arsitektur empat layer yang dipadukan dengan mekanisme *rule-based notification* serta pemisahan protokol MQTT untuk monitoring dan HTTP untuk pengiriman notifikasi. Evaluasi sistem dilakukan melalui pengujian *blackbox*, pengujian latensi komunikasi dan pengujian akurasi sensor. Hasil akhir didapatkan rata – rata latensi MQTT sebesar 1,8 detik, rata – rata latensi Telegram sebesar 1,6 detik yang menunjukkan sistem mampu mendukung monitoring secara cepat dan *real-time*. Penelitian ini menyimpulkan bahwa sistem monitoring Gudang berbasis IoT dengan MQTT dan Telegram mampu melakukan monitoring Gudang secara *real-time* serta mengirimkan notifikasi otomatis berdasarkan kondisi lingkungan yang terdeteksi.

Kata Kunci— *Internet of Things*, Monitoring Gudang, MQTT, Telegram, *rule-based notification*.

I. PENDAHULUAN

Perkembangan teknologi Internet of Things telah membawa perubahan signifikan terutama dalam mendorong transformasi sistem monitoring dari metode konvensional menjadi sistem yang dapat melakukan akuisisi data, pemrosesan data serta penyajian data secara *real-time*. Pemanfaatan sensor, perangkat komputasi serta internet memungkinkan proses monitoring dilakukan secara berkelanjutan sehingga mendukung pengambilan keputusan yang cepat dan akurat. Gudang – gudang modern saat ini telah banyak menggunakan ekosistem cerdas berbasis data yang memungkinkan otomatisasi, visibilitas realtime dan pengambilan keputusan yang fleksibel[1].

Gudang penyimpanan barang merupakan fasilitas yang memerlukan pengawasan terhadap suhu, kelembaban serta keamanannya karena perubahan suhu atau kelembaban yang terlalu jauh dari standar gudang dapat mempengaruhi kualitas barang yang disimpan. Pada gudang milik Dinas Kebudayaan, Kepemudaan dan Olahraga serta Pariwisata Kota Surabaya,

proses monitoring suhu, kelembaban dan gerakan masih dilakukan secara manual sehingga perubahan kondisi lingkungan tidak bisa terpantau secara *real-time*. Penerapan sistem monitoring gudang berbasis IoT adalah langkah yang tepat dalam penyediaan informasi kondisi gudang secara *real-time* sehingga resiko kerusakan barang bisa diperkecil.

Berbagai penelitian telah mengembangkan sistem monitoring berbasis IoT dengan protokol MQTT[2][3] dan HTTP[4]. MQTT digunakan untuk monitoring *real-time* karena memiliki karakteristik *publish-subscribe* yang sangat ringan. MQTT dirancang khusus untuk jaringan dengan bandwidth terbatas dan koneksi yang tidak stabil [5]. Karakteristik tersebut menjadikan MQTT dapat melakukan komunikasi data secara efisien serta latensi yang relatif kecil. HTTP digunakan untuk membuat permintaan dan menerima jawaban dari server Telegram saat pengiriman notifikasi.

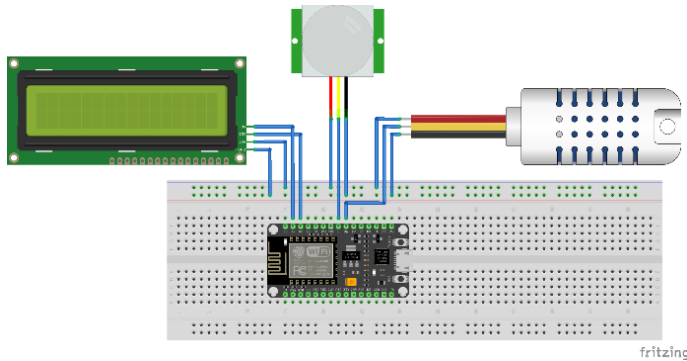
Sistem monitoring gudang yang tidak hanya mampu melakukan pemantauan kondisi lingkungan secara *real-time*, tetapi juga memiliki mekanisme pemrosesan data sebelum dikirim ke layanan cloud sangat diperlukan. Oleh karena itu, penelitian ini mengembangkan sistem monitoring gudang berbasis Internet of Things dengan menerapkan arsitektur empat lapisan, yaitu *Data Acquisition Layer*, *Edge Processing Layer*, *Communication Layer* dan *Application Layer*. Mekanisme *rule-based notification* digunakan untuk mendukung proses monitoring dan penyampaian informasi yang lebih efisien.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk mengembangkan sistem monitoring gudang berbasis Internet of Things yang menerapkan mekanisme *edge processing* untuk melakukan validasi data dan *rule-based decision* sebelum data dikirim ke server. Protokol komunikasi MQTT digunakan untuk mendukung monitoring *real-time* melalui platform Thinger.io sedangkan protokol HTTP digunakan untuk mengirimkan notifikasi melalui Telegram.

II. METODOLOGI PENELITIAN

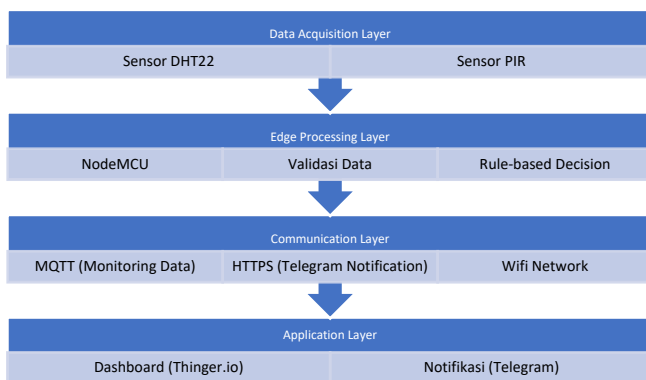
Penelitian ini menggunakan metode prototype dengan uji lapangan. Tahapan penelitian meliputi studi literatur, perancangan dan implementasi sistem serta pengujian sistem. Studi literatur digunakan untuk mengumpulkan data dan referensi dengan mempelajari buku – buku, literatur, catatan serta laporan yang berkaitan dengan masalah yang sedang dibahas[6]. Tahap perancangan menghasilkan arsitektur sistem empat layer, alur proses monitoring serta mekanisme

komunikasi data. Implementasi dilakukan dengan mengintegrasikan NodeMCU ESP8266, sensor DHT22, sensor PIR, LCD 16x2, platform Thinger.io serta Telegram.



Gbr 1 Desain Rancangan Prototype

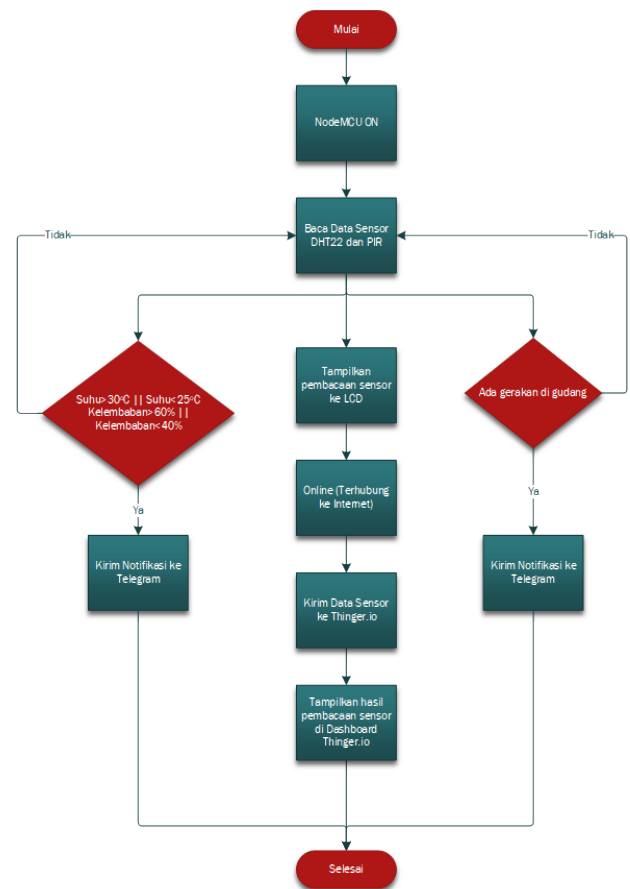
Sistem dirancang menggunakan arsitektur empat layer yang terdiri atas *Data Acquisition Layer*, *Edge Processing Layer*, *Communication Layer* dan *Application Layer*. *Data Acquisition layer* bertugas untuk memperoleh data suhu, kelembaban dan Gerakan menggunakan sensor DHT22 dan PIR. Data yang diperoleh kemudian di proses pada *Edge Processing Layer* oleh NodeMCU ESP8266 melalui proses validasi data dan mekanisme *rule-based notification*. *Communication Layer* bertugas mengirimkan data menggunakan protokol MQTT menuju platform Thinger.io dan menggunakan protokol HTTP untuk pengiriman notifikasi Telegram. Data hasil monitoring yang sudah dikirim, kemudian ditampilkan pada *Application Layer* melalui dashboard Thinger.io serta aplikasi Telegram sehingga operator gudang dapat memantau kondisi gudang secara *real-time*.



Gbr 2 Diagram Arsitektur Layer

Edge processing device dikembangkan dengan menggunakan NodeMCU ESP8266 yang terhubung dengan sensor suhu dan kelembaban DHT22 serta sensor PIR untuk deteksi Gerakan. Data hasil pembacaan sensor diproses oleh NodeMCU sebelum dikirimkan ke platform Thinger.io

menggunakan protokol MQTT. Dashboard Thinger.io adalah subscriber dari topik yang di publish oleh NodeMCU dan berfungsi sebagai penampil data monitoring sensor secara interaktif.



Gbr 3 Diagram Alir Sistem

Pengiriman data menggunakan mekanisme *rule-based notification*, yaitu proses validasi yang dilakukan oleh *edge processing device* terhadap nilai suhu, kelembaban dan Gerakan berdasarkan aturan atau batas yang telah ditentukan[7]. Data sensor dikirim secara berkala ke Dashboard Thinger.io sebagai media monitoring, namun notifikasi Telegram hanya akan dikirim jika suhu > 30°C atau suhu < 25°C, kelembaban > 60% atau kelembaban < 40%, serta ketika terdeteksi gerakan. Pendekatan ini bertujuan untuk meningkatkan komunikasi dan mengurangi *spam* notifikasi.

Pengujian system dilakukan melalui pengujian *Blackbox*, pengujian latensi komunikasi MQTT dan Telegram serta pengujian akurasi sensor. Pengujian *Blackbox* akan menguji apakah fungsi-fungsi perangkat tersebut sesuai dengan spesifikasi yang telah ditentukan tanpa melihat keseluruhan program[6]. Pengujian latensi digunakan untuk mengukur berapa lama waktu yang dibutuhkan dari sebuah data yang dikirimkan NodeMCU untuk sampai ke dashboard Thinger.io atau Telegram. Latensi dihitung dengan rumus:

$$\text{Latensi} = W_t - W_k[8] \quad (1)$$

Dimana:

W_k = Waktu data dikirim

W_t = Waktu data diterima

Latensi MQTT dan Telegram dihitung dengan menggunakan manual timer yang dimulai pada saat data muncul pada LCD dan berakhir pada saat data muncul di Dashboard Thinger.io dan Notifikasi Telegram. Pengujian akurasi dilakukan dengan membandingkan hasil pembacaan sensor DHT22 terhadap 2 alat ukur referensi.

III. HASIL DAN PEMBAHASAN

A. Hasil implementasi sistem

Berdasarkan rancangan sistem yang telah disusun, implementasi sistem berhasil dilakukan dengan mengintegrasikan antara perangkat keras dan perangkat lunak menjadi satu sistem monitoring. Sistem mampu mendeteksi dan membaca nilai suhu, kelembaban serta pergerakan yang terjadi di dalam Gudang.



Gbr 4 Prototype sistem monitoring

Gambar 3 menunjukkan hasil implementasi dari perangkat keras yang terdiri dari sensor DHT22 (dalam frame hitam di sisi samping kotak), sensor PIR (dalam frame putih di sisi atas kotak), LCD 16x2 yang ditempatkan pada sisi depan kotak untuk monitoring yang lebih mudah terlihat, NodeMCU menempel diatas breadboard yang berada di dalam kotak. Wiring menggunakan kabel jumper yang dipasang pada kaki komponen dengan breadboard. Catu daya menggunakan kabel USB type C yang terhubung langsung dengan NodeMCU. Seluruh komponen berhasil dirangkai sesuai dengan desain

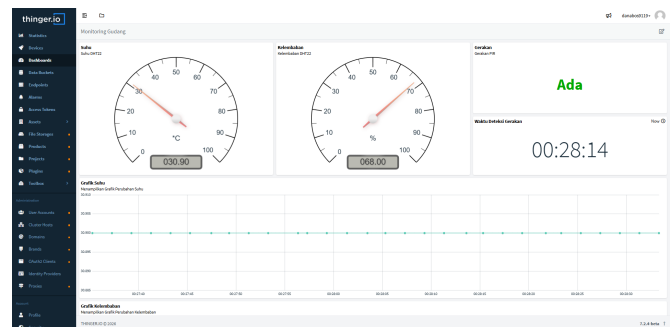
sistem sehingga mampu bekerja untuk melakukan proses akuisisi data, pemrosesan data serta komunikasi dengan platform Thinger.io dan telegram melalui jaringan internet. Protokol komunikasi yang digunakan adalah MQTT untuk Thinger.io dan HTTP untuk telegram.

Sistem monitoring ini menggunakan konsep *data pipeline*, dimana data berpindah dari satu bagian menuju bagian lain secara terus menerus. Berikut adalah diagram konsep *data pipeline* yang digunakan



Gbr 5 Diagram Data Pipeline

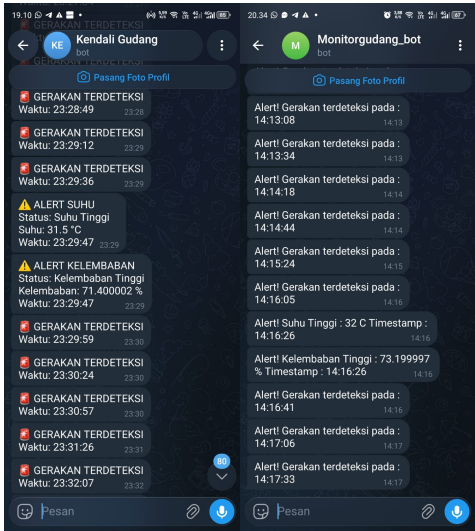
Data yang terbaca pada tahap input (*data acquisition*), berpindah menuju tahap *processing* yang dilakukan oleh *edge processing device* (NodeMCU) untuk dilakukan validasi data. Setelah data divalidasi, data berpindah ke proses *Transmission*, yaitu pengiriman data melalui protokol MQTT atau HTTP. Setelah data ditransmisikan, maka pada tahap *output* akan menampilkan data visual pada dashboard Thinger.io maupun notifikasi Telegram.



Gbr 6 Tampilan Dashboard Thinger.io

Dashboard Thinger.io mampu menampilkan data suhu, kelembaban serta peringatan Gerakan di dalam gudang secara *real-time*. Operator dapat melakukan monitoring dengan melihat data yang ditampilkan dalam visual yang menarik. Data suhu dan kelembaban ditampilkan pada widget dengan bentuk tachometer yang memudahkan dalam menganalisa nilainya. Untuk peringatan Gerakan menggunakan teks yang akan menampilkan “Ada” dan “Tidak Ada”. Waktu deteksi sensor PIR ditampilkan di bawah peringatan Gerakan untuk mengetahui waktu pada saat Gerakan terdeteksi, misal pada pukul 15:03:14 (hh:mm:ss).

Pada tampilan dashboard juga dimunculkan grafik suhu dan kelembaban. Grafik ini berfungsi untuk menganalisa kenaikan serta penurunan suhu dan kelembaban yang terjadi sehingga operator dapat membuat keputusan segera dari kondisi yang terjadi.



Gbr 7 Notifikasi Telegram

Selain monitoring menggunakan dashboard Thinger.io, sistem juga mampu mengirimkan notifikasi melalui pesan Telegram. Tidak semua kondisi akan mengirim notifikasi ke telegram karena sistem menggunakan mekanisme *rule-based notification*. NodeMCU sebagai *edge device* hanya akan mengirimkan notifikasi ke Telegram jika kondisi terpenuhi. Berikut adalah aturan yang ditentukan dalam sistem monitoring

Tabel 1 Rule-Based Notification

Kondisi	Aturan	Aksi
Suhu normal	IF suhu > 25°C && suhu < 30°C	Tampilkan di dashboard
Kelembaban Normal	IF kelembaban > 40°C && kelembaban < 60°C	Tampilkan di dashboard
Suhu > batas atas	IF suhu > 30°C	Tampilkan di dashboard, Kirim Notifikasi
Suhu < batas bawah	IF suhu < 25°C	Tampilkan di dashboard, Kirim Notifikasi
Kelembaban > batas atas	IF humidity > 60%	Tampilkan di dashboard, Kirim Notifikasi
Kelembaban < batas bawah	IF humidity < 40%	Tampilkan di dashboard, Kirim Notifikasi
Gerakan Terdeteksi	PIR = HIGH	Tampilkan di dashboard, Kirim Notifikasi
Gerakan tidak terdeteksi	PIR = LOW	Tampilkan di dashboard

Mekanisme *rule-based notification* ini mensyaratkan kondisi terpenuhi untuk pengiriman notifikasi. Jika kondisi tidak terpenuhi, maka notifikasi tidak akan dikirim. Hal ini akan menghemat bandwidth karena tidak semua data diproses untuk notifikasi Telegram. Dengan begitu, mekanisme ini

memungkinkan proses monitoring berlangsung secara berkelanjutan tanpa menghasilkan notifikasi yang berlebihan.

Pada penelitian ini pengiriman notifikasi Telegram tidak diproses langsung oleh NodeMCU, melainkan melalui endpoint Thinger.io. Prosesnya adalah NodeMCU memanggil endpoint Thinger.io yang telah dibuat sebelumnya, kemudian endpoint Thinger.io yang akan membuat HTTP request ke Telegram. Keuntungan menggunakan endpoint Thinger.io adalah NodeMCU tidak perlu memproses HTTP request di dalam kodenya, sehingga akan lebih menghemat memori karena NodeMCU hanya meminta Thinger.io untuk melakukan HTTP Request. Mekanisme ini juga lebih memudahkan dalam melakukan *troubleshooting* karena kode program menjadi lebih sederhana.

Hasil implementasi menunjukkan bahwa seluruh komponen sistem telah terintegrasi dengan baik dan dapat menjalankan fungsi monitoring maupun pengiriman notifikasi sesuai dengan desain rancangan sistem.

B. Pengujian Fungsional Sistem

Pengujian fungsional sistem dilakukan dengan menggunakan metode pengujian *Blackbox*. Pengujian dilakukan untuk memastikan setiap fungsi sistem bekerja dengan baik sesuai kebutuhan yang telah dirancang tanpa memperhatikan struktur program.

Berikut adalah tabel hasil pengujian *BlackBox* yang dilakukan pada setiap komponen untuk mengetahui komponen tersebut berfungsi atau tidak

Tabel 2 Hasil pengujian Blackbox

No	Komponen yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1	Sensor DHT22	Membaca suhu dan kelembaban	Data terbaca	Berhasil terbaca
2	Sensor PIR	Deteksi gerakan	Gerakan terdeteksi	Berhasil mendeteksi
3	LCD	Menampilkan data sensor	Data tertampil di layar LCD	Berhasil menampilkan
4	Koneksi WiFi	Koneksi ke jaringan wifi	Terhubung ke jaringan WiFi	Berhasil terhubung
5	MQTT Thinger.io	Kirim data sensor ke server	Data sensor tampil di dashboard	Data sensor terkirim
6	Notifikasi Telegram	Kirim notifikasi	Notifikasi terkirim	Berhasil kirim notifikasi

Seluruh skenario pengujian yang dilakukan mendapatkan hasil pengujian yang baik dan sesuai dengan yang diharapkan. Sensor DHT22 mampu membaca suhu dan kelembaban dengan baik, sensor PIR mampu mendeteksi Gerakan dengan baik sampai pada jarak 5 meter, LCD dapat menampilkan informasi yang diperlukan. Koneksi internet juga berjalan dengan baik menggunakan wifi. MQTT terhubung dengan baik, terbukti data dapat tertampil pada Dashboard Thinger.io. Pengiriman notifikasi Telegram juga berjalan dengan lancar.

Dengan demikian, sistem monitoring gudang berbasis IoT ini sudah memenuhi kebutuhan fungsional untuk melakukan monitoring gudang secara *real-time*.

C. Analisis Performa Komunikasi

Analisis performa komunikasi dilakukan dengan menguji latensi pengiriman data dari NodeMCU sampai data ditampilkan pada Aplikasi. Terdapat dua protokol komunikasi yang dilakukan pengujian, yaitu MQTT dan HTTP. Pengukuran latensi dilakukan secara manual dengan menggunakan stopwatch.

Pada MQTT, latensi dihitung berdasarkan selisih waktu antara data pertama kali ditampilkan pada LCD dengan waktu saat data muncul di dashboard. berikut adalah tabel pengukuran latensi MQTT yang diuji sebanyak 15 kali

Tabel 3 Hasil Pengujian Latensi MQTT

Tes	Waktu Deteksi (LCD) hh:mm:ss	Waktu Data Tampil (Dashboard) hh:mm:ss	Latensi (detik)
Tes 1	15:19:32	15:19:34	2
Tes 2	15:19:56	15:19:58	2
Tes 3	15:20:19	15:20:20	1
Tes 4	15:21:14	15:21:16	2
Tes 5	15:22:01	15:22:02	1
Tes 6	15:22:26	15:22:28	2
Tes 7	15:22:54	15:22:56	2
Tes 8	15:23:23	15:23:24	1
Tes 9	15:23:51	15:23:52	1
Tes 10	15:25:13	15:25:16	3
Tes 11	15:26:32	15:26:34	2
Tes 12	15:26:55	15:26:57	2
Tes 13	15:27:18	15:27:20	2
Tes 14	15:27:41	15:27:43	2
Tes 15	15:28:12	15:28:15	2

Pengujian yang dilakukan menunjukkan latensi terendah adalah 1 detik dan latensi tertinggi 3 detik dengan rata – rata latensi 1,8 detik. Nilai median adalah 2 dan nilai modus adalah 2 yang menunjukkan sebagian besar percobaan menghasilkan latensi sebesar 2 detik. Dari 15 kali percobaan yang dilakukan, 5 percobaan (33,3%) memiliki latensi 1 detik, 9 percobaan (60%) memiliki latensi 2 detik dan 1 percobaan (6,7%) memiliki latensi 3 detik. Hasil ini menunjukkan bahwa 93,3% pengujian berhasil mengirimkan data dengan waktu maksimal 2 detik sedangkan hanya ada 1 yang mengalami latensi lebih dari 2 detik.

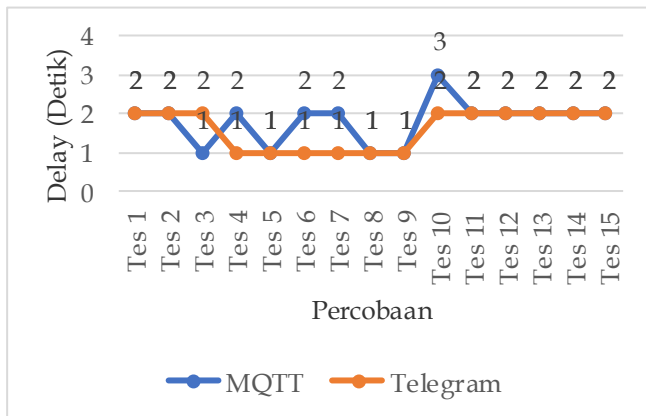
Jika pada MQTT latensi dihitung mulai data tertampil di LCD hingga data tertampil di Dashboard, maka pada Telegram, latensi dihitung berdasarkan selisih waktu antara data pertama kali ditampilkan pada LCD dengan waktu saat data muncul sebagai notifikasi di pesan Telegram. Berikut adalah tabel pengukuran latensi Telegram yang diuji sebanyak 15 kali

Tabel 4 Hasil Pengujian Latensi Telegram

Tes	Waktu deteksi (LCD) hh:mm:ss	Waktu Notifikasi Diterima (Telegram) hh:mm:ss	Latensi
Tes 1	15:19:32	15:19:11	2
Tes 2	15:19:56	15:19:35	2
Tes 3	15:20:19	15:19:58	2
Tes 4	15:21:14	15:20:20	1
Tes 5	15:22:01	15:21:15	1
Tes 6	15:22:26	15:22:27	1
Tes 7	15:22:54	15:22:55	1
Tes 8	15:23:23	15:23:24	1
Tes 9	15:23:51	15:23:52	1
Tes 10	15:25:13	15:25:15	2
Tes 11	15:26:32	15:26:34	2
Tes 12	15:26:55	15:26:57	2
Tes 13	15:27:18	15:27:20	2
Tes 14	15:27:41	15:27:43	2
Tes 15	15:28:12	15:28:14	2

Pengujian yang dilakukan menunjukkan latensi terendah adalah 1 detik dan latensi tertinggi 2 detik dengan rata – rata latensi 1,53 detik. Nilai median adalah 2 dan nilai modus adalah 2 yang menunjukkan sebagian besar percobaan menghasilkan latensi sebesar 2 detik. Dari 15 kali percobaan yang dilakukan, 7 percobaan (46,7%) memiliki latensi 1 detik dan 8 percobaan (53,3%) memiliki latensi 2 detik. Hasil ini menunjukkan bahwa 100% pengujian berhasil mengirimkan data dengan waktu maksimal 2 detik.

Dari hasil percobaan, rata rata latensi yang dihasilkan dari dua protokol yang berbeda ini tidak terlalu jauh selisihnya. Namun Telegram terlihat lebih cepat dibanding MQTT. Hal ini dapat disebabkan oleh logika sistem yang digunakan oleh NodeMCU dalam melakukan eksekusi kode. NodeMCU melakukan *rate limiting* untuk eksekusi data yang dikirim melalui MQTT. Hal ini dilakukan untuk menjaga performa sistem agar tetap stabil saat digunakan untuk monitoring. Berikut perbandingan latensi MQTT dan Telegram



Gbr 8 Grafik perbandingan latensi MQTT dan Telegram

Terlihat pada grafik, saat tes ke 10 terjadi latensi sebesar 3 detik pada MQTT, sedangkan Telegram menghasilkan latensi sekitar 1 sampai 2 detik saja. Selain *Rate limiting* kondisi jaringan juga sangat berpengaruh terhadap besarnya latensi yang terjadi. nMaun, dengan latensi pengiriman yang rata – rata kurang dari 2 detik, sistem ini sudah mampu untuk melakukan pemantauan kondisi gudang secara *real-time*.

D. Analisis Akurasi Sensor

Analisis akurasi sensor dilakukan untuk mengevaluasi seberapa akurat hasil pembacaan sensor. Pengujian sensor DHT22 dilakukan dengan membandingkan hasil pembacaan sensor dengan dua alat ukur suhu dan kelembaban lain yang dijual di pasaran. Perbandingan dilakukan pada tempat yang sama dan posisi sama, sehingga parameter lingkungan berada pada status yang sama. Karena suhu dan kelembaban dihitung terpisah oleh DHT22, maka hasil dalam tabel disusun terpisah antara pengujian suhu dan kelembaban. Berikut adalah hasil pengujian akurasi sensor DHT22

Tabel 5 Hasil pengujian akurasi sensor suhu

No	Suhu Sensor (°c)	Suhu Referensi 1 (°c)	Selisih (°c)	Suhu Referensi 2 (°c)	Selisih (°c)
1	28,9	29	0,1	29,5	0,6
2	29,4	29	0,4	29,6	0,2
3	30	30	0	29,8	0,2
4	23	26,4	3,4	26,8	3,8
5	23,3	25,5	2,2	26,3	3,0

Pengujian dilakukan sebanyak lima kali pada lokasi yang berbeda dengan kondisi setiap alat ukur saling berdekatan. Dari hasil pengujian diperoleh selisih rata – rata sensor dengan alat ukur pertama adalah 1,02°C dan selisih rata – rata dengan alat ukur kedua sebesar 1,35°C. Hasil pengukuran

sensor juga tidak selalu lebih rendah. Selisih dengan alat ukur pembanding tidak terlalu jauh. Perbedaan toleransi dan jenis sensor yang digunakan sangat berpengaruh terhadap kepekaan dan hasil pembacaan sensor.

Tabel 6 Hasil pengujian akurasi sensor kelembaban

No	Nilai Sensor (%)	Nilai Referensi 1 (%)	Selisih (%)	Nilai Referensi 2 (%)	Selisih (%)
1	58	59	1	58	0
2	57	56	1	57	0
3	53	56	3	54	1
4	51	51	0	53	2
5	51	50	1	48	3

Pengujian dilakukan sebanyak lima kali pada lokasi yang berbeda dengan kondisi alat ukur yang berdekatan. Dari hasil pengujian diperoleh selisih rata – rata sensor dengan alat ukur pertama adalah 1,2% dan selisih rata – rata dengan alat ukur kedua sebesar 1,2%. Hasil pengukuran sensor juga tidak selalu lebih rendah. Selisih dengan alat ukur pembanding tidak terlalu jauh. Perbedaan toleransi dan jenis sensor yang digunakan sangat berpengaruh terhadap kepekaan dan hasil pembacaan sensor. Hasil pengukuran menunjukkan nilai yang konsisten tanpa ada penyimpangan yang ekstrim.

Pengujian sensor PIR dilakukan dengan menguji jarak deteksi sensor terhadap gerakan di depan sensor. Pengujian dilakukan sebanyak 11 kali dengan obyek deteksi yang berbeda. Berikut adalah hasil pengujian sensor PIR

Tabel 7 Hasil pengujian sensor PIR

Tes Ke	Jarak Objek (M)	Deteksi Sensor	Keterangan
1	1	Terdeteksi	Gerakan manusia
2	2	Terdeteksi	Gerakan manusia
3	3	Terdeteksi	Gerakan manusia
4	4	Terdeteksi	Gerakan manusia
5	5	Terdeteksi	Gerakan manusia
6	6	Tidak Terdeteksi	Gerakan manusia
7	7	Tidak Terdeteksi	Gerakan manusia
8	1	Terdeteksi	Bola menggelinding
9	2	Terdeteksi	Bola menggelinding
10	3	Terdeteksi	Bola menggelinding
11	4	Tidak Terdeteksi	Bola menggelinding

Hasil pengujian sensor PIR menunjukkan, sensor masih dapat mendeteksi Gerakan dengan baik pada jarak 1 sampai dengan 5 meter. Pada jarak 6 meter sudah tidak dapat mendeteksi gerakan. Hal ini menunjukkan jarak efektif sensor PIR dalam mendeteksi gerakan manusia adalah 1 sampai dengan 5 meter. Sedangkan ketika dilakukan tes menggunakan bola yang digelindingkan di depan sensor, jarak efektif deteksinya adalah 1 sampai dengan 3 meter saja. Pada 4 meter, sensor tidak mendeteksi ada pergerakan.

Perbedaan hasil deteksi ini dapat terjadi karena karakteristik sensor PIR yang mendeteksi perubahan radiasi inframerah, dimana manusia menghasilkan panas sehingga lebih mudah terdeteksi sedangkan bola tidak memancarkan panas sehingga lebih sulit untuk dideteksi.

Hasil pengujian juga menunjukkan bahwa semakin jauh jarak objek dari sensor, maka akan semakin sulit bagi sensor untuk mendeteksi. Dengan demikian, sensor PIR bekerja optimal untuk mendeteksi objek yang menghasilkan panas seperti manusia dengan jarak efektif hingga 5 meter.



Gbr 9 Pengujian akurasi sensor

IV. KESIMPULAN

Berdasarkan hasil perancangan, implementasi dan pengujian dapat disimpulkan sebagai berikut

1. Sistem pemantauan berbasis IoT telah berhasil dirancang dan diimplementasikan dengan memanfaatkan NodeMCU V3, sensor DHT22, sensor PIR, platform Thingier.io untuk visualisasi data secara real-time, serta Telegram sebagai sarana notifikasi untuk peringatan otomatis saat terjadi situasi abnormal.
2. Sistem pemantauan ini dijalankan secara efektif dengan NodeMCU V3 sebagai unit *edge processing*, yang terhubung dengan platform Thingier.io dan Telegram di *application layer*, dan mampu memfasilitasi akses data secara real-time. Thingier.io beroperasi sebagai broker dan platform visualisasi data, sedangkan Telegram berfungsi sebagai sistem pemberitahuan untuk kondisi tidak normal dan telah terbukti efektif
3. Hasil pengujian untuk sistem pemantauan ini menunjukkan kinerja yang memuaskan sesuai dengan kondisi pengujian yang ditetapkan. Latensi transfer data rata-rata untuk MQTT adalah 1,8 detik, yang menandakan koneksi yang responsif, sedangkan latensi untuk notifikasi Telegram adalah 1,6 detik, yang masih berada dalam rentang responsif. Sistem berhasil mengirimkan pemberitahuan, dan sensor yang digunakan memiliki tingkat presisi yang tinggi, sehingga sistem ini dapat diandalkan untuk pemantauan gudang secara real-time

REFERENSI

- [1] M. Ruqnuzzaman, M. M. R. Opu, M. R. Ahmed, M. M. R. Shamim, and M. Nuruzzaman, "A systematic review of recent advances in IoT-based sensor networks for warehouse management," *Results Eng.*, vol. 29, p. 109195, Mar. 2026, doi: 10.1016/J.RINENG.2026.109195.

- [2] M. A. Bakri, M. Farhan, A. Sujatmiko, and A. Firasanti, "Pemantauan Suhu dan Deteksi Gerak Obyek Berbasis IoT pada Ruang Server Menggunakan Thingier.IO," *TELKA - Telekomun. Elektron. Komputasi dan Kontrol*, vol. 8, no. 1, 2022, doi: 10.15575/telka.v8n1.74-81.
- [3] G. G. Salindeho and T. Wellem, "PERANCANGAN DAN IMPLEMENTASI SISTEM PENDETEKSI DAN PERINGATAN KEBAKARAN BERBASIS IOT MENGGUNAKAN NODEMCU ESP8266 DAN SENSOR API," *IT-Explore J. Penerapan Teknol. Inf. dan Komun.*, vol. 2, no. 3, 2023, doi: 10.24246/itexplore.v2i03.2023.pp179-191.
- [4] N. Hardi, R. Afuw Rouf Subyan, and A. Arbasyah, "Alat Berbasis IOT Smarthome Monitoring dan Kontrol via Telegram Menggunakan Nodemcu," *INSANTEK*, vol. 4, no. 1, 2023, doi: 10.31294/instk.v4i1.2018.
- [5] "MQTT dan CoAP: Protokol untuk IoT | D3 Teknologi Telekomunikasi." Accessed: Feb. 28, 2026. [Online]. Available: <https://dte.telkomuniversity.ac.id/mqtt-dan-coap-protokol-untuk-iot/>
- [6] R. Santosa, P. A. Sari, and A. T. Sasongko, "Sistem Monitoring Suhu dan Kelembaban Berbasis IoT (Internet of Thing) pada Gudang Penyimpanan PT Sakafarma Laboratories," *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 5, no. 4, 2023, doi: 10.47233/jteksis.v5i4.943.
- [7] A. Kathirgamanathan, M. De Rosa, E. Mangina, and D. P. Finn, "Data-driven predictive control for unlocking building energy flexibility: A review," *Renew. Sustain. Energy Rev.*, vol. 135, Jan. 2021, doi: 10.1016/j.rser.2020.110120.
- [8] R. A. Sianturi, F. Larosa, A. Gea, and H. Artikel, "Analisis QoS Routing OSPF IP Versi 4 dan OSPF IP Versi 6 Pada Mikrotik OS," *Methodika J. Ilm. Tek. Inform.*, vol. 2, no. 2, 2022.