

Implementasi Docker Container untuk Sistem Monitoring dan Pengontrolan Peralatan Listrik di Laboratorium Cerdas

Sahirul Alam^{a1}, Sri Lestari^{a2}, Anni Karimatul Fauziyyah^{a3}, Dzulfikar^{a4}

^aDepartemen Teknik Elektro dan Informatika, Sekolah Vokasi, Universitas Gadjah Mada
Gedung Herman Yohanes, Sekip Unit III, Catur Tunggal, Depok, Sleman, Yogyakarta, Indonesia

¹sahirul.alam@ugm.ac.id

²srilestari59@ugm.ac.id

³anni.karimatul.f@ugm.ac.id

⁴dzulfikar.sv@mail.ugm.ac.id

Abstract

Smart laboratory atau laboratorium cerdas adalah laboratorium yang dilengkapi dengan teknologi canggih seperti Internet of Things (IoT), robotika, dan kecerdasan buatan (Artificial Intelligence/AI) untuk meningkatkan efisiensi, akurasi, dan keamanan dalam melakukan penelitian dan pengujian. Dalam sebuah smart laboratory, perangkat IoT dapat digunakan untuk memantau suhu, kelembaban, dan kualitas udara di dalam ruangan, sehingga dapat memastikan kondisi lingkungan yang ideal untuk menjaga kualitas sampel yang diuji. IoT sendiri adalah sebuah konsep yang mengacu pada konektivitas antara berbagai perangkat atau objek yang terhubung ke internet, sehingga memungkinkan perangkat tersebut saling berkomunikasi dan bertukar data. Penelitian ini bertujuan untuk membangun sebuah sistem pemantauan dan pengontrolan peralatan listrik di dalam laboratorium. Sistem yang dibangun menerapkan teknologi IoT sehingga pemantauan dan pengontrolan peralatan listrik akan menjadi lebih mudah dan dapat dilakukan dari mana saja. Selain itu, sistem memanfaatkan teknologi docker container sehingga instalasi dan pengelolaan perangkat lunak dapat dilakukan dengan lebih mudah dan efisien.

Keywords: Docker Container, Internet of Things, Smart Laboratory, Monitoring System, Energy Efficiency

1. Pendahuluan

Internet of Things (IoT) adalah sebuah konsep yang mengacu pada konektivitas antara berbagai perangkat atau objek yang terhubung ke internet, sehingga memungkinkan perangkat tersebut untuk saling berkomunikasi dan bertukar data. Secara sederhana, IoT menggambarkan jaringan perangkat yang terhubung secara online, sehingga dapat saling berkomunikasi dan berinteraksi satu sama lain [1]. Perangkat IoT dapat berupa segala jenis perangkat elektronik, mulai dari sensor dan kamera hingga kendaraan dan peralatan rumah tangga, bahkan hewan peliharaan yang dikenakan alat pelacak.

Dalam aplikasi praktis, teknologi IoT dapat digunakan dalam berbagai macam industri, seperti pertanian [2], manufaktur, transportasi, kesehatan, dan rumah tangga. Salah satu manfaat utama dari IoT adalah memungkinkan pengumpulan data yang lebih banyak dan akurat, yang dapat digunakan untuk membantu membuat keputusan yang lebih baik dan meningkatkan efisiensi dalam bisnis atau aktivitas sehari-hari. Namun, keamanan dan privasi menjadi isu penting yang harus diperhatikan dalam pengembangan dan penggunaan teknologi IoT [3].

Smart laboratory atau laboratorium cerdas adalah laboratorium yang dilengkapi dengan teknologi canggih seperti IoT, robotika, dan kecerdasan buatan untuk meningkatkan efisiensi, akurasi, dan keamanan dalam melakukan penelitian dan pengujian. Dalam sebuah smart laboratory, perangkat IoT dapat digunakan untuk memantau suhu, kelembaban, dan kualitas udara di dalam ruangan [4], sehingga dapat memastikan kondisi lingkungan yang ideal untuk menjaga kualitas sampel yang diuji. Selain itu suhu dan kelembaban ruangan juga penting untuk dijaga agar peralatan di laboratorium tidak mengalami kerusakan akibat kelembaban tinggi seperti jamur dan karat.

Penelitian ini bertujuan untuk membangun sistem monitoring dan pengontrolan peralatan listrik di laboratorium cerdas. Peralatan listrik yang dikendalikan dengan IoT yaitu lampu penerangan dan AC. Sedangkan monitoring dilakukan untuk memantau dan mengumpulkan data terkait suhu dan

kelembaban ruang laboratorium. Penerapan IoT ke dalam sistem yang dibangun dengan memanfaatkan docker container. Docker merupakan seperangkat platform yang menggunakan teknologi virtualisasi pada level sistem operasi untuk menjalankan perangkat lunak dengan memuat semua dependensi yang diperlukan di dalam container [5]. Dengan menggunakan docker container, instalasi dan pengelolaan perangkat lunak IoT dapat dilakukan dengan lebih mudah dan efisien.

Beberapa penelitian terkait penerapan IoT untuk mengendalikan perangkat listrik antara lain untuk pengontrolan lampu [6][7] dan untuk pengontrolan kipas angin dan AC [8]. Teknologi IoT memang telah banyak diimplementasikan untuk pengendalian peralatan listrik. Namun, belum banyak penelitian yang membahas tentang penggunaan docker container untuk aplikasi IoT. Penelitian [9] menggunakan docker container untuk diterapkan pada pengontrolan lampu penerangan di gedung. Perbedaan penelitian tersebut dengan penelitian ini antara lain adalah penggunaan perangkat keras, perangkat lunak, dan juga pemanfaatan. Dari segi perangkat keras, penelitian [9] menggunakan komputer dengan spesifikasi menengah ke atas.

Beberapa penelitian terkait monitoring kualitas udara ruangan dapat ditemui di literatur. Penelitian [10] dan [11] memanfaatkan IoT untuk melakukan monitoring kualitas udara terkait kadar CO₂ dan resiko infeksi virus serta melakukan pengontrolan buka tutup jendela. Penelitian [12] memanfaatkan IoT dan CCTV untuk monitoring suhu dan kelembaban ruang pengering umbi bawang merah. Penelitian [13] memanfaatkan IoT untuk monitoring suhu dan kelembaban ruang produksi obat serta melakukan pengontrolan kipas dan dehumidifier. Selain itu, beberapa penelitian juga memanfaatkan IoT untuk monitoring suhu dan kelembaban udara di ruang sarang burung walet [14], ruang budidaya jamur tiram putih [15], ruang penyimpanan obat [16], hingga ruang server [17] [18]. Perbedaan utama penelitian ini dengan penelitian yang telah disebutkan adalah bahwa penelitian ini mengimplementasikan IoT untuk monitoring dan pengontrolan peralatan listrik di laboratorium, terutama untuk memantau suhu dan kelembaban ruang penyimpanan peralatan laboratorium. Selain itu, penelitian ini membahas implementasi docker container untuk sistem monitoring sebagai kebaruan yang belum pernah dibahas di penelitian yang sudah ada.

2. Metodologi

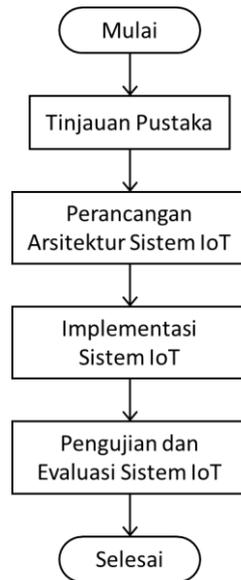
2.1 Alur Pengembangan Sistem

Pada penelitian ini, teknologi IoT akan diimplementasikan untuk mewujudkan konsep laboratorium cerdas (smart laboratory). IoT digunakan untuk memantau dan mengontrol peralatan listrik yang ada di laboratorium seperti lampu, air conditioner, dan LED proyektor. Pada konsep laboratorium cerdas, peralatan listrik tersebut harus dapat dikontrol secara otomatis untuk mendukung suasana laboratorium yang nyaman namun tetap memperhatikan efisiensi penggunaan daya listrik.

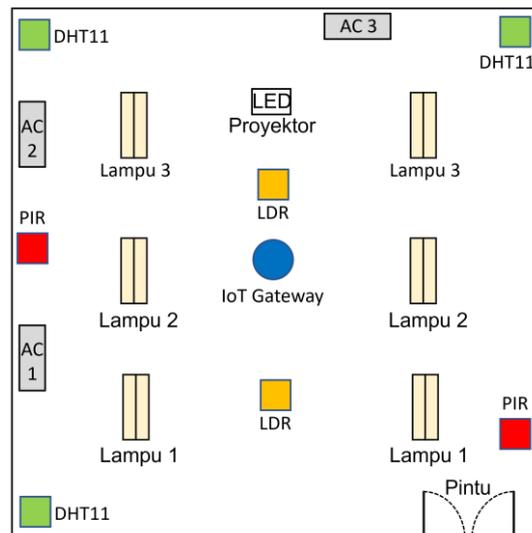
Untuk penerangan laboratorium, sistem IoT akan mengatur pengoperasian lampu dengan mempertimbangkan ada tidaknya orang dan intensitas cahaya di dalam ruangan. Oleh karena itu, dua jenis sensor akan digunakan yaitu sensor passive infra-red (PIR) dan light depending resistor (LDR). Selanjutnya, air conditioner akan dioperasikan untuk mengatur suhu dan kelembaban yang kondusif di laboratorium. Dalam hal ini, sistem akan menggunakan sensor DHT-11. Untuk memantau penggunaan daya listrik, kWh meter akan dipasang di sumber tegangan peralatan listrik. Secara sederhana, jalannya penelitian direncanakan seperti pada Gambar 1. Tinjauan pustaka dilakukan untuk mengetahui trend implementasi IoT terutama untuk pengontrolan peralatan listrik dan pemantauan kondisi ruangan. Perancangan sistem IoT meliputi pemilihan perangkat keras dan perangkat lunak IoT. Dalam hal ini, peralatan perlu menyesuaikan dengan kondisi ruangan tempat IoT akan diimplementasikan. Setelah sistem IoT diimplementasikan, maka pengujian perlu dilakukan untuk mengetahui fungsionalitas dan kinerja sistem IoT yang sudah dibuat.

2.2 Perakitan Perangkat Keras

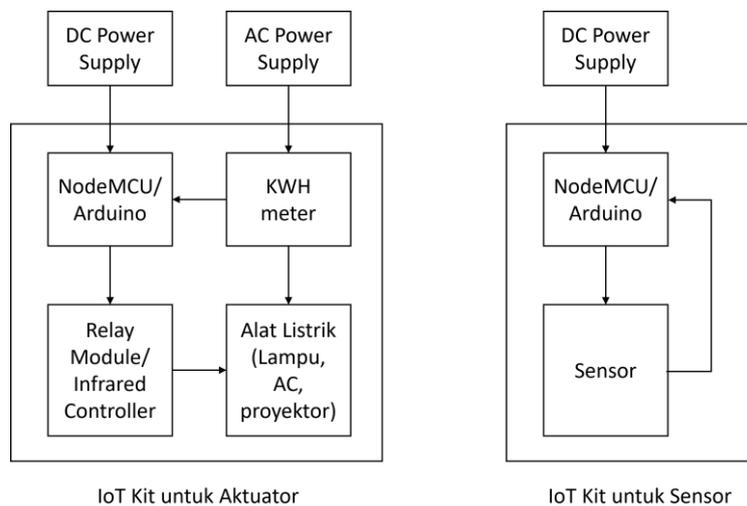
Perancangan arsitektur IoT perlu disesuaikan dengan kondisi ruangan. Dalam hal ini, sistem IoT akan diterapkan di Laboratorium Telekomunikasi Departemen Teknik Elektro dan Informatika, Sekolah Vokasi, Universitas Gadjah Mada. Ilustrasi rencana implementasi sistem IoT di laboratorium adalah seperti ditunjukkan pada Gambar 2. Setiap sensor dan aktuator di Gambar 2 dilengkapi dengan IoT kit. Sedangkan IoT kit sendiri tersusun dari beberapa komponen seperti ditunjukkan pada Gambar 3. Secara lebih spesifik, perangkat keras yang digunakan untuk sistem IoT ditunjukkan pada Tabel 1.



Gambar 1. Alur pengembangan sistem



Gambar 2. Rencana implementasi sistem IoT di laboratorium

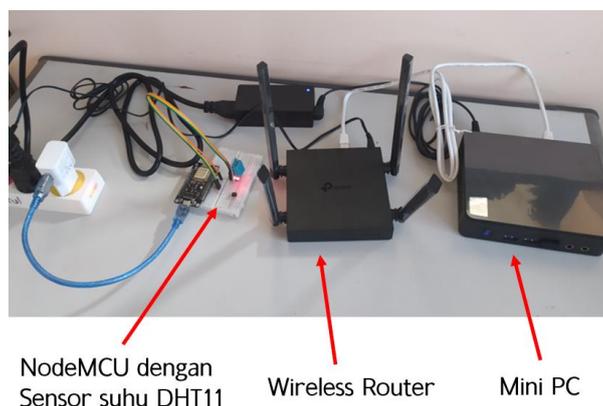


Gambar 3. Komponen penyusun IoT-kit untuk sensor dan aktuator

Tabel 1. Spesifikasi perangkat keras yang digunakan

Perangkat Keras	Spesifikasi/Jenis
NodeMCU board	ESP8266 Lolin
Relay Module	KY-019 1-Channel
Sensor Suhu dan Kelembaban	DHT11
Sensor Gerak	HC-SR501 Human Sensing PIR
IR Transmitter	KY-005
Sensor Tegangan AC	ZMPT101B
Sensor Arus AC	100A SCT-013-000 Non-invasive
IoT Gateway	Wireless Router TP-Link Archer C54
IoT Server	Intel NUC11ATKC4

Pengukuran konsumsi daya listrik menggunakan kWh meter yang terdiri dari sensor tegangan ZMPT101B dan sensor arus 100A SCT-013-000. IoT server yang sudah lebih dulu populer adalah menggunakan Raspberry. Namun, pada penelitian ini, IoT server menggunakan mini PC karena dengan kisaran harga yang hampir sama dengan Raspberry, namun mini PC memiliki skalabilitas yang lebih baik, misalnya dapat ditingkatkan kapasitas memori maupun penyimpanannya. IoT Gateway untuk komunikasi data semua perangkat IoT dan server menggunakan wireless router seperti ditunjukkan pada Gambar 4.

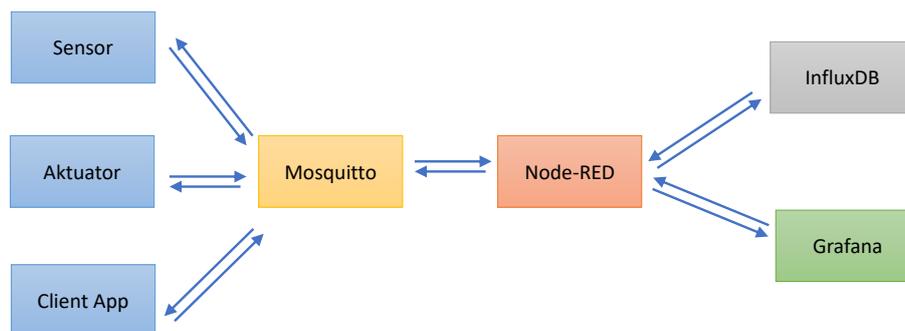


Gambar 4. Perangkat Keras IoT untuk Laboratorium Cerdas

2.3 Instalasi dan Konfigurasi Perangkat Lunak Menggunakan Docker Container

Perangkat lunak dalam sistem IoT digunakan untuk mengatur komunikasi data antara sensor, aktuator, server, dan perangkat pengguna. Selain itu, perangkat lunak juga digunakan untuk mengatur tampilan untuk monitoring kondisi ruangan. Perangkat lunak sistem IoT diinstall di dalam server yang menggunakan sistem operasi Ubuntu Server. Perangkat lunak tersebut diinstall dengan menggunakan docker container. Kelebihan metode instalasi ini yaitu dapat dilakukan dengan cepat dan lebih mudah untuk melakukan manajemen dan konfigurasi perangkat lunak. Secara umum, perangkat lunak yang dijalankan di dalam docker container serta ilustrasi alur komunikasi datanya ditunjukkan pada Gambar 5. Mosquitto adalah perangkat lunak yang digunakan sebagai broker IoT yang berperan mengatur komunikasi data antar perangkat IoT. InfluxDB merupakan perangkat lunak basis data yang digunakan untuk menyimpan data hasil pengukuran dari perangkat IoT. Grafana adalah perangkat lunak front end yang digunakan untuk menampilkan visualisasi data untuk keperluan monitoring kondisi ruangan. Data yang dikirim oleh Mosquitto memiliki format JSON. Oleh karena itu, Node-RED yang berperan sebagai penghubung komunikasi data antar perangkat lunak memiliki kemampuan untuk melakukan interpretasi data dalam format JSON.

Dari sistem operasi, IoTStack dapat diunduh dengan menggunakan command seperti pada Gambar 6, kemudian reboot perlu dilakukan setelah download selesai. Setelah itu, instalasi dapat dilanjutkan dengan menggunakan secure socket shell (SSH) dan membuka folder IoTStack. Menu instalasi akan muncul dan perangkat lunak yang dipilih untuk diinstall adalah Mosquitto, Node-RED, InfluxDB, Grafana, dan Portainer-CE. Untuk memulai proses pembuatan container setelah memilih perangkat lunak dapat dilakukan dengan command "Start Stack". Setelah proses instalasi selesai, pengecekan dapat dilakukan untuk memastikan semua program yang dipilih berhasil diinstall dengan menggunakan command "docker-compose ps".



Gambar 5. Arah komunikasi data dari perangkat keras dan perangkat lunak IoT

```
curl -fsSL https://raw.githubusercontent.com/SensorsIot/IOTstack/master/install.sh | bash  
sudo shutdown -r now
```

Gambar 6. Perintah untuk mengunduh IoTStack

Pembuatan database menggunakan InfluxDB dapat dilakukan dengan memasuki influx docker container, kemudian menambahkan database dengan perintah seperti Gambar 7. Data yang dikirimkan dari Mosquitto menggunakan format JSON, sehingga Node-RED perlu menginterpretasi data dalam format JSON tersebut.

```
docker exec -it influxdb influx  
CREATE DATABASE sensor_data  
quit
```

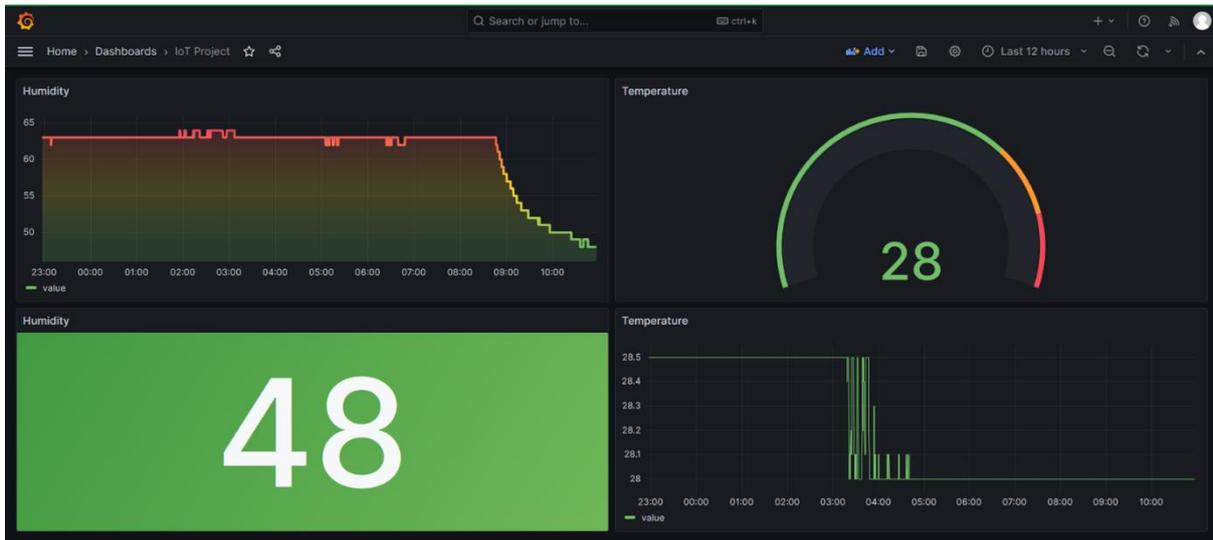
Gambar 7. Perintah untuk menambahkan database di InfluxDB

3. Hasil dan Pembahasan

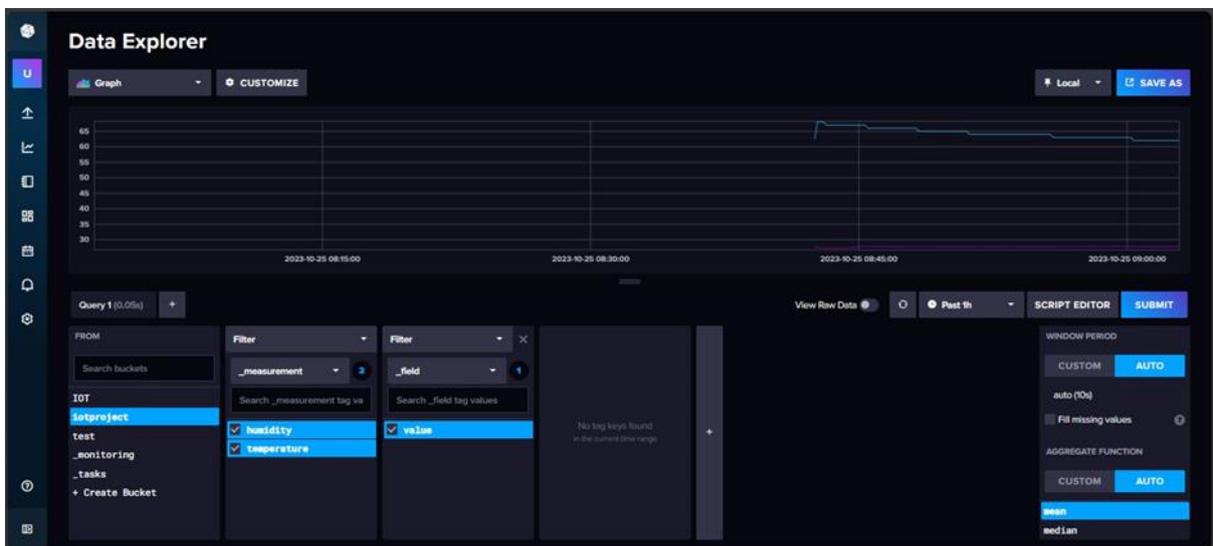
Pengujian sistem yang telah dibangun dilakukan dengan melakukan pengontrolan baik secara manual maupun otomatis. Pengontrolan manual dapat dilakukan melalui perangkat pengguna yang telah terlebih dahulu diinstall aplikasi MQTT client. Pengontrolan yang dapat dilakukan secara manual antara lain ON-OFF peralatan listrik seperti lampu, AC, dan LCD proyektor. Monitoring kondisi ruangan seperti suhu dan kelembaban udara dapat dilakukan dengan mengakses Grafana menggunakan browser. Caranya yaitu dengan mengakses alamat IP server kemudian diikuti port 3000. Langkah selanjutnya adalah proses autentikasi dengan memasukkan username dan password, kemudian dashboard grafana dapat diakses. Ilustrasi tampilan dashboard Grafana untuk monitoring suhu dan kelembaban udara ruangan laboratorium adalah seperti ditunjukkan pada Gambar 8.

Data hasil pengukuran sensor disimpan di dalam basis data. Data tersebut dapat diakses dengan membuka InfluxDB menggunakan browser melalui alamat IP server dengan port 8086. Tampilan InfluxDB ditunjukkan pada Gambar 9. Data yang tersimpan di basis data dapat diunduh dalam format CSV. Contoh log data temperatur dan kelembaban ruangan tanggal 11 Oktober 2023 mulai dari pukul 00:00 hingga 23:59 WIB ditunjukkan pada Tabel 2. Data dari file CSV menggunakan periode sampling 5 detik sehingga ada 5 data pengukuran tiap menit. Namun, data pada Tabel 2 yang diambil dari file CSV menggunakan 1 sampel per jam. Data pada Tabel 2 selanjutnya ditampilkan dalam plot grafik seperti ditunjukkan pada Gambar 10.

Grafik temperatur dan kelembaban udara ruangan di Gambar 10 menunjukkan keterkaitannya dengan okupansi atau ada tidaknya orang di dalam ruangan. Suhu dan kelembaban yang tinggi biasanya terjadi di luar jam kerja. Pada saat jam kerja atau ada kegiatan di dalam laboratorium, suhu dan kelembaban menurun karena AC dinyalakan. Berdasarkan data hasil monitoring tersebut, dapat disimpulkan bahwa pengontrolan kelembaban udara ruangan masih kurang optimal terutama untuk ruang penyimpanan peralatan laboratorium. Oleh karena itu, penelitian selanjutnya dapat diarahkan untuk mengoptimalkan pengontrolan kelembaban udara dengan mengatur pengoperasian AC namun tetap mempertimbangkan efisiensi konsumsi daya.



Gambar 8. Tampilan dashboard Grafana untuk monitoring kondisi udara ruangan



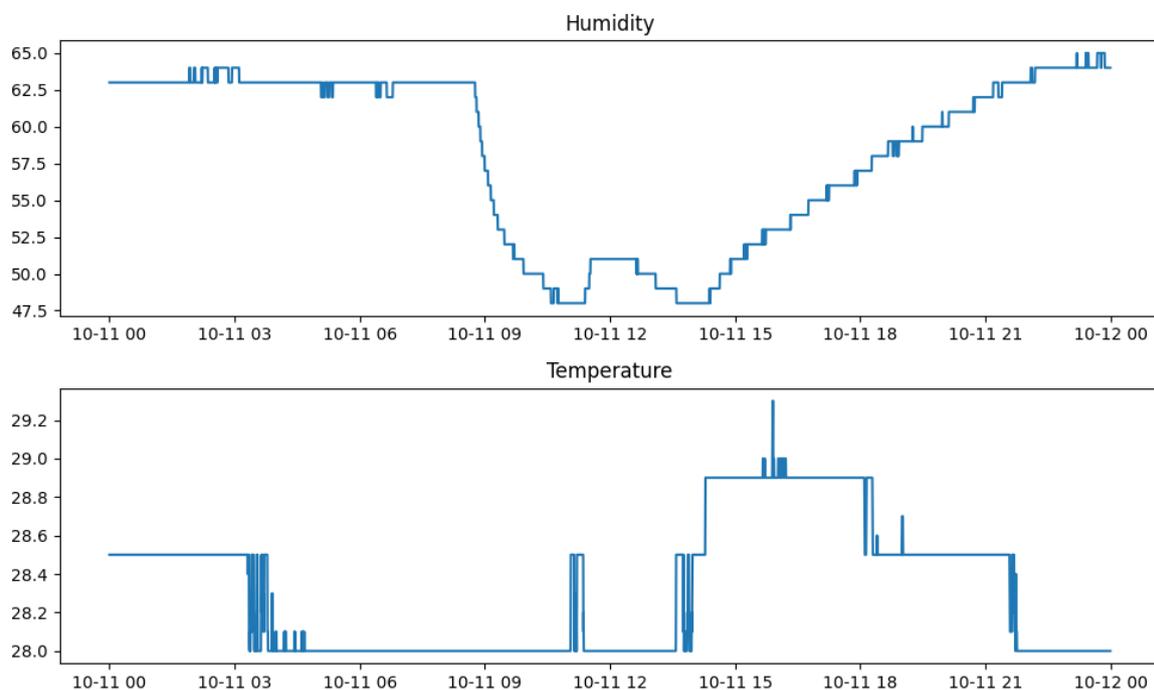
Gambar 9. Tampilan dashboard InfluxDB

4. Kesimpulan

Penelitian ini mengimplementasikan penggunaan docker container untuk membangun sistem IoT dengan cepat, karena pengguna dapat menggunakan beberapa perangkat lunak seperti basis data dan visualisasi data yang sudah tersedia. Pada penelitian ini, sistem IoT digunakan untuk melakukan pengontrolan peralatan listrik dan monitoring kondisi ruangan di dalam laboratorium. Penggunaan docker container juga mempermudah pengelolaan dan pengaturan konfigurasi perangkat lunak yang digunakan dalam sistem IoT. Dari hasil monitoring, kondisi udara dalam ruang laboratorium berhubungan dengan okupansi ruangan. Biasanya peralatan listrik dioperasikan untuk meningkatkan kenyamanan dalam ruang laboratorium. Namun, kelembaban udara juga perlu dikontrol terutama untuk ruang penyimpanan peralatan laboratorium. Oleh karena itu, penelitian selanjutnya disarankan untuk melakukan optimasi pengontrolan kelembaban udara namun dengan tetap memperhatikan efisiensi konsumsi daya.

Tabel 2. Log Data Hasil Pengukuran Sensor IoT

No.	Time	Humidity Value	Temperature Value
1	10/11/2023 0:00	63	28.5
2	10/11/2023 1:00	63	28.5
3	10/11/2023 2:00	63	28.5
4	10/11/2023 3:00	64	28.5
5	10/11/2023 4:00	63	28.1
6	10/11/2023 5:00	63	28
7	10/11/2023 6:00	63	28
8	10/11/2023 7:00	63	28
9	10/11/2023 8:00	63	28
10	10/11/2023 9:00	57	28
11	10/11/2023 10:00	50	28
12	10/11/2023 11:00	48	28
13	10/11/2023 12:00	51	28
14	10/11/2023 13:00	50	28
15	10/11/2023 14:00	48	28.5
16	10/11/2023 15:00	51	28.9
17	10/11/2023 16:00	53	28.9
18	10/11/2023 17:00	55	28.9
19	10/11/2023 18:00	57	28.9
20	10/11/2023 19:00	59	28.6
21	10/11/2023 20:00	60	28.5
22	10/11/2023 21:00	62	28.5
23	10/11/2023 22:00	63	28
24	10/11/2023 23:00	64	28



Gambar 10. Plot data temperatur dan kelembaban udara ruangan tanggal 11 Oktober 2023

References

- [1] A. F. Isnanto, A. Surriani, S. Lestari and U. Y. Oktiawati, "Prototype of Smart Home and Monitoring Application Based On Internet of Things (IoT) Using Android," *Jurnal Listrik, Instrumentasi, dan Elektronika Terapan (JuLIET)*, vol. 1, no. 1, 2020.
- [2] K. L. Rivaldo, I. K. A. Mogi, I. P. G. H. Suputra, N. A. Sanjaya, I. D. M. B. A. Darmawan and I. B. G. Dwidasmara, "Sistem Monitoring Tanaman Hidroponik Berbasis Internet of Things Menggunakan Restful API," *Jurnal Elektronik Ilmu Komputer Udayana*, vol. 11, no. 1, 2022.
- [3] M. H. Shirvani and M. Masdari, "A survey study on trust-based security in Internet of Things: Challenges and issues," *Internet of Things*, vol. 2023, no. 21, 2023.
- [4] H. Yin, "Smart Lab Technologies," in *Handbook of Mobile Teaching and Learning*, Berlin, Springer-Verlag Berlin Heidelberg, 2015, pp. 1-11.
- [5] B. S. Kim, S. H. Lee, Y. R. Lee, Y. H. Park and J. Jeong, "Design and Implementation of Cloud Docker Application Architecture Based on Machine Learning in Container Management for Smart Manufacturing," *Applied Science*, vol. 12, no. 13, 2022.
- [6] W. Istiana, "Sistem Pengendalian Lampu Menggunakan Raspberry Pi Internet of Things (IoT) Berbasis Mobile," *Jurnal Portal Data*, vol. 2, no. 6, 2022.
- [7] S. D. M. Panjaitan, "Prototype Pengendalian Lampu Jarak Jauh Dengan Jaringan Internet Berbasis Internet of Things (IoT) Menggunakan Raspberry Pi 3," *Jurnal Pendidikan Sains dan Komputer*, vol. 2, no. 2, 2022.
- [8] F. Z. Rachman, "Smart Home Berbasis IoT," in *Seminar Nasional Inovasi Teknologi Terapan (SNITT)*, Balikpapan, 2017.
- [9] Z. Iklima, "Rancang Bangun Prototipe Sistem Kendali Terdistribusi Instalasi Penerangan pada Gedung 3 Lantai berbasis IoTaaS (Internet of Things as a Service) menggunakan Docker Container," *Jurnal Teknologi Elektro*, vol. 10, no. 1, 2019.
- [10] A. Zivelonghi and A. Guiseppi, "Smart Healthy Schools: An IoT-enabled concept for multi-room dynamic air quality control," *Internet of Things and Cyber-Physical Systems*, vol. 4, pp. 24-31, 2024.
- [11] G. Rescio, A. Manni, A. Caroppo, A. M. Carluccio, P. Siciliano and A. Leone, "Multi-Sensor Platform for Predictive Air Quality Monitoring," *Sensors*, vol. 23, no. 11, pp. 1-17, 2023.
- [12] F. Y. Ontowirjo, V. C. Poekoel, P. D. Manembu and R. F. Robot, "Implementasi Internet of Things Pada Sistem Monitoring Suhu dan Kelembaban Pada Ruangan Pengereng Berbasis Web," *Jurnal Teknik Elektro dan Komputer*, vol. 7, no. 3, pp. 331-339, 2018.
- [13] A. A. M. Khalifa and K. Prawiroredjo, "Model Sistem Pengendalian Suhu dan Kelembaban Ruangan Produksi Obat Berbasis NodeMCU ESP32," *Jurnal ELTIKOM : Jurnal Teknik Elektro, Teknologi Informasi dan Komputer*, vol. 6, no. 1, pp. 13-25, 2022.
- [14] A. Iskandar, "Implementasi IoT Pada Sistem Monitoring dan Kendali Otomatis Suhu Dan Kelembaban Ruangan Sarang Burung Walet Berbasis Mikrokontroler," *Jurnal Cyber Tech*, vol. 4, no. 8, pp. 1-8, 2022.
- [15] I. Y. Syas and F. A. Rakhmadi, "Prototipe Sistem Monitoring serta Kendali Suhu dan Kelembapan Ruangan Budidaya Jamur Tiram Putih Menggunakan Sensor DHT22 Dan Mikrokontroler NodeMCU," *Sunan Kalijaga Journal of Physics*, vol. 1, no. 1, pp. 7-13, 2019.
- [16] S. H. W. Sasono, "IoT Smart Health Untuk Monitoring Dan Kontrol Suhu dan Kelembaban Ruang Penyimpan Obat Berbasis Android di Rumah Sakit Umum Pusat Dr. Sardjito," *ReTII*, pp. 53-62, 2020.
- [17] E. B. Raharjo, S. Marwanto and A. Romadhona, "Rancangan Sistem Monitoring Suhu Dan Kelembaban Ruang Server Berbasis Internet Of Things," *Teknika ATW*, vol. 6, no. 2, pp. 61-69, 2019.
- [18] N. F. Khobariah, P. D. S. Hermawan and R. S. Kusumadiarti, "Sistem Monitoring Suhu dan Kelembaban Ruang Server Berbasis Wemos D1," *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 7, no. 1, pp. 32-42, 2022.