Monitoring Development Board based on InfluxDB and Grafana

Monitoring Development Board pada Platform InfluxDB dan Grafana

Noprianto[1], Vivi Nur Wijayaningrum[2], Rokhimatul Wakhidah[3]

[1,2,3] Teknologi Informasi, Politeknik Negeri Malang, Indonesia

[1*]noprianto@polinema.ac.id, [2]vivinurw@polinema.ac.id, [3]wakhidah@polinema.ac.id
**: Penulis korenspondensi* (*corresponding author*)

Informasi Artikel
Received: August 2022
Revised: January 2023
Accepted: January 2023
Published: February 2023

Keywords:Monitoring; Development Board; InfluxDB; Grafana
Kata kunci: Pemantuan; Development Board; InfluxDB; Grafana

Abstract

Purpose: Designing a sensor data monitoring system using a time series database and monitoring platform on a Development Board device.
Design/methodology/approach: It begins with a requirement analysis, such as the preparation of the required software and hardware, followed by the creation of the system architecture that will be adopted. Then the development process from a predetermined design to the testing process to ensure the dashboard page can display data according to a predetermined scenario.
Findings/result: From the research that has been done, produces a design of sensor data that is sent using the MQTT protocol via Node-RED, then stored in a time series database (InfluxDB) and displayed on the Grafana dashboard display.
Originality/value/state of the art: Sensor data monitoring dashboard on Development Board devices

Abstrak

Tujuan: Merancang sebuah sistem pemantauan data sensor menggunakan *time series database* dan platform pemantuan pada sebuah perangkat development board.
Perancangan/metode/pendekatan: Diawali dengan analisis kebutuhan seperti penyiapan perangkat lunak serta perangkat keras yang dibutuhkan, dilanjutkan pembuatan arsitektur sistem yang akan diadopsi. Kemudian proses pengembangan dari rancangan yang telah ditentukan sampai proses pengujian untuk memastikan halaman *dashboard* bisa menampilkan data sesuai dengan skenario yang telah ditentukan.
Hasil: Dari penelitian yang telah dilakukan menghasilkan sebuah perancangan data sensor yang dikirimkan menggunakan protokol MQTT melalui Node-RED, selanjutnya disimpan pada sebuah database time series(InfluxDB) dan ditampilkan pada tampilan dashboard Grafana.
Keaslian/ *state of the art*: *Dashboard* pemantauan data sensor pada perangkat development board

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI:10.31515/telematika.v20i1.7643

1. Pendahuluan

Sebuah peradaban dunia fisik telah bertransformasi menjadi digital sehingga semua dapat saling terhubung, pertumbuhan yang sangat pesat pada bidang *smart device* dan teknologi mengakibatkan manusia dapat berkomunikasi kapan saja dan di mana saja. Revolusi Industri 4.0 meningkatkan tren teknologi *Internet of Things* yang berfokus pada interkonektivitas, otomatisasi, otonomi, *machine learning*, dan data *real time* [1]–[5]. Salah satu yang menjadi tantangan pada *Internet of Things* adalah proses pengiriman data secara real time dari *smart device* sebelum dilakukan pengolahan sampai dengan proses visualiasi data, dan pengiriman data *real time* yang memungkinkan pengambilan keputusan secara *real time* yang dapat meningkatkan pendapatan, produktivitas, dan efisiensi. Proses visualisasi menjadi penting karena dapat menggambarkan sebuah data yang sebelumnya tidak bermakna berubah menjadi sebuah informasi yang mudah untuk dibaca dan dipahami. Data visualisasi umumnya berasal dari *smart device* atau perangkat *Internet of Thing* yang telah terpasang sensor.

Beberapa peneliti terdahulu telah mengemukakan penelitiannya tentang pemantuan data, seperti yang dilakukan oleh [6] dalam melakukan *traffic* pemantuan dan log gangguan pada perangkat jaringan pada PT Cyber Network Indonesia, platform yang digunakan dalam melakukan pemantuan adalah Zabbix terintegrasi dengan Grafana dan Telegram. Grafana digunakan untuk memvisualisasikan metrik menjadi grafik-grafik yang menarik serta mudah dimengerti. Data secara *real time* mengenai penggunaan *bandwidth*, *log problem*, dan ketersediaan sumber daya dapat dilihat di manapun oleh seorang administrator jaringan. Kemudian ketika terjadi permasalahan, administrator jaringan akan mendapatkan notifikasi atau pemberitahuan melalui Telegram.

Penelitian yang sama juga memanfaatkan Grafana untuk memantau kondisi layanan (*service*) yang terdapat pada *server*. Grafana mendapatkan data kondisi *service* pada *server* dari Prometheus kemudian mengetahui status dari *service* yang dipantau menggunakan Telegram [7]. Secara spesifik, Prometheus mengumpulkan *data source* dalam bentuk metrik secara langsung ataupun melalui *gateway push* Prometheus. Namun, sebelumnya harus dipasang terlebih dahulu *exporter* pada setiap *node*.

Selain digunakan untuk melakukan pemantuan pada sebuah jaringan, Grafana digunakan untuk memantau curah hujan karena dapat memberikan visualisasi yang sangat baik dari hasil yang diperoleh dari sensor curah hujan. Selain itu, Grafana juga dapat menampilkan keadaan mikrokontroler untuk mendukung kinerja sensor karena lokasi mikrokontroler dan sensor yang berada di luar ruangan. Informasi tentang keadaan mikrokontroler dan sensor sangat dibutuhkan ketika terjadi anomali, sehingga dapat dengan cepat diidentifikasi dan ditindaklanjuti [8].

Grafana merupakan sebuah *tool* yang digunakan untuk pemantuan, akan tetapi butuh sebuah sumber data atau *data source* sebagai masukan dalam menampilkan data yaitu data yang berasal dari berbagai macam *database*. Salah satu yang sering digunakan yaitu InfluxDB. InfluxDB adalah *database* tidak terstruktur yang dirancang khusus untuk menyimpan data historis dan dapat memproses data dalam bentuk matriks untuk dianalisis. Karena masuk ke dalam kategori database tidak terstruktur, sehingga memiliki format yang sangat berbeda dan mendukung dalam memaksimalkan performa *database* khususnya data dalam bentuk *time series* [9]. Anih et al [10] menggunakan InfluxDB dalam menyimpan data *time series* berupa suhu harian minimum periode 10 tahun (1981-1990) di kota Melbourne, Australia. Tujuan dari penyimpanan data tersebut untuk melakukan *preprocessing* sebelum dilakukan penambangan data, karena *data preprocessing* merupakan modal awal dalam penambangan data namun para peneliti

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI:10.31515/telematika.v20i1.7643

mengabaikannya. Data diolah menggunakan bahasa pemrograman Python dan hasilnya diuraikan menjadi data tren, musiman, dan komponen residu.

InfluxDB menyediakan sebuah *Interface Programming Application* dengan banyak dukungan bahasa pemrograman dan juga platform *Internet of Things* seperti Node-RED. Node-RED merupakan sebuah *development tool* yang berbasis JavaScript dibuat menggunakan Node.js, yang biasanya digunakan dalam pengembangan perangkat Internet of Things dengan menghubungkan API secara bersama, perangkat keras, dan layanan online yang dikembangkan oleh IBM. Node-RED menyediakan *dashboard* dengan tampilan sistem berupa kontrol tombol, gauge maupun grafik [11]–[13]. Meela A et al [14] menggunakan Node-RED untuk memantau suhu industri menggunakan NodeMCU ESP8266 sebagai pengontrol dan sensor suhu terpasang dengan pengiriman data yang digunakan adalah Message Queue Telemetry Transport (MQTT). MQTT merupakan sebuah protokol yang digunakan dalam pengiriman data, alasan menggunakan MQTT adalah karena MQTT adalah salah satu protokol komunikasi ringan berbasis internet, sedangkan penyajian datanya menggunakan Node-RED. Pada penelitian yang dilakukan oleh Duraiswamy R et al [15] menggunakan Node-RED dalam merancang sebuah sistem pertanian otomatis di dalam ruangan berbasis *Internet of Things* yang memanfaatkan cloud IBM Bluemix. Parameter yang dipantau adalah yang berkontribusi pertumbuhan tanaman seperti suhu, kelembaban, suhu mesin, intensitas cahaya, dan tingkat pH agar tetap berada pada kondisi yang ideal. Ketika terjadi hal yang tidak ideal, sistem akan memperingatkan kepada pemilik perkebunan agar segera mengambil tindakan yang dibutuhkan.

Pada penelitian ini, peneliti akan menggabungkan sistem pemantuan perangkat development board menggunakan Node-RED, InfluxDB, serta Grafana yang diharapkan mampu untuk melakukan pemantuan data sensor secara real-time yang terdapat pada perangkat development board.

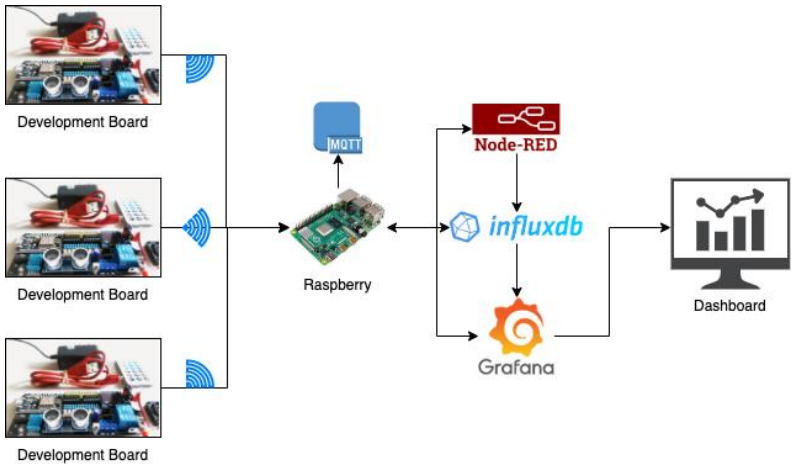
2. Metode/Perancangan

Ketika melakukan penelitian ini, beberapa kebutuhan yang harus dipenuhi untuk perangkat *hardware* yaitu seperangkat laptop dengan spesifikasi Macbook Pro 13” Intel core i3, RAM 16 GB, dan SSD 750GB, serta sebuah Raspberry Pi 4B 8 GB, dan seperangkat development board. Untuk perangkat lunak yang harus terpasang yaitu *message broker* menggunakan Mosquitto, InfluxDB, Node-RED, dan Grafana. Seperangkat laptop digunakan untuk melakukan pengembangan hingga pengujian, Raspberry difungsikan sebagai sebuah server untuk melakukan instalasi *message broker*, *database*, Node-RED, serta Grafana. Selanjutnya untuk development board yang telah terpasang sensor dan aktuator, akan mengirimkan datanya ke server, yaitu Raspberry.

Penggunaan mosquitto merupakan salah satu perangkat lunak yang mengimplementasikan *protocol* MQTT dengan kemampuannya yang ringan dan mudah untuk digunakan berbagai macam bahasa pemrograman dalam mengirimkan data. Sementara InfluxDB dikenal sebagai *database time series* atau masuk ke dalam database tidak struktur dalam fungsinya pada penyimpanan data sensor, selain open source, dan memiliki kecepatan tinggi dalam membaca atau menuliskan data. Dalam mengorganisasikan MQTT dan InfluxDB perannya dilakukan oleh Node-RED yang semua komponen telah tersedia tanpa harus banyak menuliskan kode program, terakhir penggunaan platform Grafana untuk menampilkan data dengan kemampuan dapat membaca dari berbagai sumber data seperti InfluxDB.

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI:10.31515/telematika.v20i1.7643

Setelah semua kebutuhan terpenuhi dan dilakukan konfigurasi atau instalasi, terlebih dahulu melakukan ujicoba *message broker*, InfluxDB, Node-RED, dan Grafana menggunakan data *dummy* atau data yang dihasilkan dengan cara di-*generate* menggunakan Node-RED. Hal ini dilakukan sebelum menggunakan perangkat yang sebenarnya yaitu development board. Setelah semua tidak terjadi kendala, data bisa tersimpan di *database* influxDB dan Grafana bisa menampilkan data tersebut dengan sebuah *dashboard* dan dilanjutkan dengan data dari sebenarnya yang dihasilkan dari development board. Secara umum, untuk skema atau rancangan dari alur di atas dapat disajikan menggunakan **Gambar 1**.



Gambar 1. Desain Arsitektur

Gambar 1 menunjukkan proses data yang dihasilkan dari development board yang mengirimkan data ke server sampai data tersebut bisa tampil pada sebuah dashboard pemantauan. Secara terperinci, diawali dengan development board mengirimkan data sensor suhu(DHT11), data sensor cahaya(LDR), dan sensor ultrasonik(HC-SR04) menggunakan koneksi WI-FI melalui protokol Message Queueing Telemetry Transport(MQTT) pada sebuah topik tertentu. Data sensor yang telah dikirimkan atau *publish* akan diterima oleh semua *client* yang melakukan *subscribe*, termasuk Node-RED. Selain melakukan *subscribe*, Node-RED sekaligus menyimpan data-data sensor pada sebuah database InfluxDB. Setelah datanya tersimpan ke dalam InfluxDB, kemudian data tersebut dijadikan *data source* pada Grafana yang selanjutnya membuat template *dashboard* sesuai dengan kebutuhan.

3. Hasil dan Pembahasan

Dalam melakukan percobaan pada penelitian ini menggunakan koneksi *wireless* yang menghubungkan development board dengan Raspberry sebagai server. Sementara untuk Mosquitto, *database*(InfluxDB), Node-RED, dan Grafana dilakukan instalasi pada server yang sama yaitu Raspberry untuk memudahkan konfigurasi agar proses pembuatan *dashboard* pemantauan data sensor dapat dengan mudah dilakukan. Khusus untuk Node-RED, tidak perlu dilakukan instalasi karena sistem operasi Raspbian pada Raspberry sudah terinstal khususnya untuk versi yang *desktop*. Praktik terbaiknya adalah untuk Mosquitto sampai dengan Grafana tidak dilakukan instalasi dalam mesin yang sama karena masalah keamanan dan pembagian sumber daya jika dimanfaatkan untuk melakukan pengelolaan data dengan volume yang banyak.

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI:10.31515/telematika.v20i1.7643

3.1. Data Sensor dan Aktuator Development Board

Development board yang digunakan berbasis mikrokontroler menggunakan NodeMCU ESP8266 dengan bahasa pemrograman C/C++. Data yang dikirimkan adalah data sensor cahaya, data sensor suhu kelembaban, dan data sensor ultrasonic. Ketiga sensor tersebut telah terpasang pada development board, sehingga untuk konfigurasi sensor tidak banyak dilakukan. Penggunaan sensor-sensor tersebut dengan alasan sensor yang sering digunakan karena harganya murah dan mudah untuk didapatkan di sekitar kita. Pengiriman data dilakukan setiap 5 detik menggunaan protokol MQTT. Pemilihan penggunaan protokol MQTT ini karena dikenal sebagai protokol yang ringan khususnya ketika digunakan untuk perangkat mikrokontroler dibandingkan menggunakan protokol HTTP yang harus terhubung ketika akan mengirimkan data. Data yang di-*publish* ke *message broker* menggunakan topik /smk-data. Topik adalah sebuah kata kunci yang digunakan oleh *client* lain agar dapat menerima data ketika melakukan *subscribe*. Format data yang digunakan ketika dikirimkan menggunakan format JSON dengan cara serialisasi. Serialisasi adalah sebuah proses mengubah data ke dalam sebuah objek untuk disimpan atau dikirimkan melalui media jaringan. Format data yang di-*publish* dapat dilihat pada **Gambar 2**.


```
{
 "mac": "DC:4F:22:76:35:2A",
 "IP": "192.168.0.110",
 "ldr": 66,
 "src04": 234,
 "temp": 31,
 "hum": 58
}
```


Gambar 2 Format data sensor

Dari **Gambar 2**, terlihat terdapat atribut seperti mac, IP, src04, temp, dan hum. Mac digunakan untuk menyimpan informasi *mac address* dari perangkat development board, IP adalah informasi tentang IP *address* dari development board, sedangkan ldr adalah untuk menyimpan informasi data sensor tingkat intensitas cahaya. Src04 digunakan untuk menyimpan informasi data sensor ultrasonik berupa jarak objek yang terdeteksi dalam satuan centimeter (CM), temp digunakan untuk menyimpan informasi temperatur dalam satuan Celcius, dan hum digunakan untuk menyimpan informasi *humidity* atau kelembaban dalam satuan persen (%). Perangkat development board yang dalam penelitian ini menggunakan 3 buah seperti ditunjukkan pada **Gambar 3** di bawah ini

Gambar 3. Perangkat Development Board

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI:10.31515/telematika.v20i1.7643

3.2. Struktur Data Sensor InfluxDB

Seperti yang telah disebutkan bahwa InfluxDB merupakan *database* tidak terstruktur sehingga ketika akan menyiapkan penyimpananpun tidak terlalu banyak proses yang dilakukan. Setelah dilakukan installasi InfluxDB, yang perlu dipersiapkan hanya nama *database* yang perlu dibuat secara manual karena akan dibutuhkan oleh Node-RED untuk menyimpan data dan dibutuhkan Grafana ketika akan menambahkan *data source* untuk menampilkan data sensor pada sebuah *dashboard*.

Pada InfluxDb tidak menggunakan konsep kolom, tetapi menggunakan istilah tags dan *fields*. Tags diibaratkan sebuah index kolom jika dianalogikan sebuah *database* relasional sedangkan *fields* seperti sebuah kolom. Adapun istilah tabel jika pada *database* relasional disebut sebagai measurement pada InfluxDB. **Gambar 4** adalah salah satu contoh penerapan struktur data sensor ultrasonic

```
> show field keys on monitoring from ultrasonic
name: ultrasonic
fieldKey fieldType
-------- ---------
value    float
> show tag keys on monitoring from ultrasonic
name: ultrasonic
tagKey
------
ip
mac
```

Gambar 4 Struktur penyimpanan InfluxDB

Dari **Gambar 4**, bahwa fieldKey dengan nama value dan menggunakan fieldType float adalah untuk menyimpan nilai sensor data ultrasonic. Kemudian untuk tagKey adalah IP dan mac, tagKey nantinya akan dimanfaatkan ketika pembuatan *dashboard* pemantauan menggunakan Grafana agar *dashboard* lebih dinamis. Untuk format data yang disimpan mirip dengan *database* struktural seperti ditunjukkan pada **Tabel 1** di bawah.

Tabel 1. Format data InfluxDB

Time	IP	MAC	Value
1646890021166413493	192.168.0.107	98:CD:AC:25:A1:88	43
1646890021196167680	192.168.0.110	DC:4F:22:76:35:2A	68
1646890023099970507	192.168.0.109	DC:4F:22:76:33:25	48

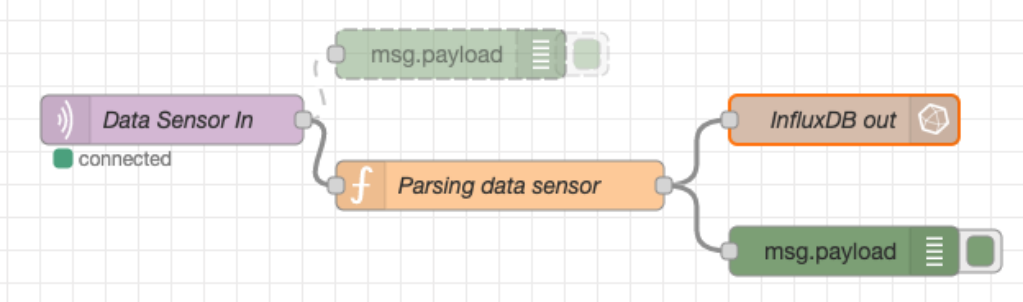
Jika dilihat dari stuktur data di atas, tidak jauh berbeda dengan database struktural. Yang membedakan adalah terdapat sebuah fieldKey dengan nama time. FieldKey tersebut terisi secara otomatis ketika data masuk ke *database* dan bersifat *unique* dalam format timestamps yang menjadi pembeda antara record atau baris yang lain.

Fitur penting yang terdapat pada InfluxDB yaitu diperkenankan untuk melakukan konfigurasi data retensi, yaitu sebuah kebijakan atau *rule* yang mendeskripsikan seberapa lama data disimpan, berapa banyak *copy* data yang disimpan dalam sebuah *cluster*, dan rentang waktu yang didukung oleh *shard group*(durasi shard group). Konfigurasi tersebut bisa berbeda pada setiap *database* yang dibuat.

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI:10.31515/telematika.v20i1.7643

3.3. Pembuatan Konfigurasi pada Node-RED

Secara mendasar, Node-RED berfungsi menyediakan komponen untuk menampilkan data sensor atau untuk melakukan kontrol aktuator. Selain penggunaannya yang sederhana yaitu dengan merangkai sebuah node sehingga terbentuk sebuah alur atau aliran, Node-RED juga menyediakan banyak sekali komponen yang siap digunakan tanpa harus membuat kode program yang panjang untuk fungsi tertentu. Sebagai contoh, ketika akan *publish* ataupun *subscribe* data dari protokol MQTT sudah tersedia dan yang perlu dilakukan adalah melakukan konfigurasi alamat *server broker* dan topiknya apa. Hal yang sama ketika akan mengirimkan atau menyimpan data pada database InfluxDB, dibutuhkan beberapa konfigurasi seperti alamat server InfluxDB, nama *database*, serta *measurements*(tabel). **Gambar 5** berikut adalah rancangan *flow* untuk menyimpan data ke *database* InfluxDB.



Gambar 5. Flow pada Node-RED

Dari **Gambar 5**, terdapat beberapa node yaitu Data Sensor In, msg.payload, Parsing data sensor, dan InfluxDB out yang memiliki masing-masing fungsi tertentu. Data Sensor In adalah sebuah node MQTT untuk melakukan *subscribe* data yang di-*publish* oleh development board, msg.payload merupakan node debug yang sangat membantu dalam testing atau ujicoba sebuah flow karena datanya akan ditampilkan pada jendela debugging Node-RED, Parsing data sensor merupakan sebuah node fungsi untuk melakukan parsing data JSON ke dalam format yang dibutuhkan oleh InfluxDB, kemudian yang terakhir yaitu InfluxDB out ialah node yang membantu dalam menyimpan data ke dalam database InfluxDB. Khusus untuk sebuah node *function* atau dalam hal ini adalah Parsing data sensor, harus penambahan sebuah baris perintah menggunakan bahasa pemrograman javascript sesuai yang dibutuhkan.

3.4. Desain Dashboard pada Grafana

Untuk membuat sebuah dashboard, langkah-langkah yang perlu dilakukan yaitu menambahkan *data source* pada Grafana. *Data source* yang dimaksud adalah *database* yang akan diambil datanya kemudian ditampilkan pada sebuah *dashboard*. Selain mendukung data source dengan model *database time series*, jenis database lain juga dapat ditambahkan seperti untuk logging menggunakan Elasticsearch, *database* struktural(MysQL), serta yang berbasis *cloud* seperti Google Cloud Monitoring.

Proses pembuatan dashboard dilakukan dengan menambahkan panel serta jenis visualisasi yang kita inginkan, visualisasi yang ditawarkan bisa berupa Graph, Gauge, Heatmap, bar gauge, dan yang lain. Panel ini nantinya yang akan menampilkan data pada dashboard, pada panel juga terdapat *query inspector* untuk mengambil data pada *data source* yang sebelumnya telah dikonfigurasi. Di dalam *query inspector* telah tersedia fungsi seperti agregasi(integral, *mean*, ataupun *median*), juga terdapat transformasi(derivatif, *difference*, stdev, dan yang lain).

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI:10.31515/telematika.v20i1.7643

Gambar 6 menampilkan hasil *dashboard* yang masing-masing menampilkan data sensor kelembaban, sensor LDR, dan sensor ultrasonic. Development board yang digunakan sebanyak 3 buah dengan masing-masing IP adalah 192.168.0.107, 192.168.0.109, dan 192.168.0.110. Terlihat bahwa data temperatur yang dihasilkan dari ketiga *developent board* nilainya rentang 31 - 32 °C yang mengakibatkan bentuk grafiknya lurus horisontal seperti ditunjukkan pada panel Sensor Temperatur. Sementara untuk grafik pada panel Sensor Humidity, Sensor LDR, dan Sensor Ultrasonic cenderung dinamis naik turun.

Gambar 6. Dashboard pada Grafana

Dari **Gambar 6** di atas, terlihat bahwa data-data sensor dari semua *develoment board* ditampilkan di *dashboard*. Untuk menampilkan beberapa atau hanya salah satu data-data sensor, dapat menggunakan *combo box* pada label *address*. Hal ini bisa dilakukan karena alamat IP atau mac adalah sebuah tagkey sehingga bisa dijadikan filter pada *dashboard* di Grafana. Selain itu, data yang ditampilkan pada *dashboard* bisa difilter berdasarkan waktu misalkan 5, 15, 30 menit terakhir atau bahkan jam dan hari. Agar *dashboard* terlihat perjalanan datanya, dapat dilakukan dengan mengubah waktu *refresh* pada menu pojok kanan atas dengan pilihan setiap 5 detik sampai dengan *refresh* per hari.

3.5. Skenario Pengujian Data

Dalam melakukan pengujian pada penelitian ini, skenario yang dilakukan adalah dengan melihat data yang dihasilkan dari sensor pada development board sampai data berhasil ditampilkan pada *dasboard* Grafana. **Tabel 2** berikut adalah daftar skenario masing-masing data tersebut

Tabel 2. Pengujian Data Sensor

No	Perangkat	Data	Skenario	Hasil
1	Development board	{"mac":"DC:4F:22:76:35:2A","IP":"192.168.0.110","ldr":66,"src04":234,"temp":31,"hum":58}	Data yang dihasilkan sensor dipublish ke message broker dengan menampilkan pada debug monitor	Berhasil
2	Node-RED	{"mac":"DC:4F:22:76:35:2A","IP":"192.168.0.110","ldr":66,"src04":234,"temp":31,"hum":58}	Data sensor dilakukan subscribe kemudian ditambahkan node debug untuk melihat message payload pada panel debugger	Berhasil
3	InfluxDB	time=1646890026318540918, ip=192.168.0.110, mac=DC:4F:22:76:35:2A,value=66	Data sensor dicek pada setiap measurement dengan cara melakukan query	Berhasil
4	Grafana	time=20122-03-10 12:28:12.500, ip=192.168.0.110, mac=DC:4F:22:76:35:2A,value=66	Data tampil di masing-masing panel pada halaman dashboard ketika grafik di hover menggunakan mouse dengan melihat waktu data tersebut masuk	Berhasil

Dari **Tabel 2** di atas menunjukkan bahwa setiap pengujian di masing-masing perangkat baik *software* atau *hardware* berhasil dilakukan. Hasil Berhasil berarti data sensor dapat ditampilkan dan dipantau pada halaman *dashboard* Grafana. Data-data sensor dapat dilihat pada setiap langkah sampai dengan tertampil di halaman *dashboard*.

4. Kesimpulan dan Saran

Dari pembahasan di atas, didapatkan kesimpulan bahwa untuk menampilkan data sensor dari development board yang merupakan perangkat *Internet of Things* dapat ditampilkan sesuai dengan informasi yang dibutuhkan dan dapat langsung terlihat ketika masing-masing data sensor mengalami perubahan. Perpaduan antara InfluxDB, Node-RED, dan Grafana bisa menjadi alternatif atau solusi ketika akan menampilkan data serta melakukan analisis data secara cepat.

Hanya saja pada penelitian ini, InfluxDB, Node-RED, dan Grafana masih diinstal dalam server yang sama yaitu Raspberry, sehingga ketika diterapkan pada kasus yang yang lebih kompleks akan timbul masalah, seperti komputasi dan pengolahan data menjadi terhambat karena keterbatasan perangkat Raspberry tersebut. Ke depan, ketiga komponen tersebut bisa dipasang pada mesin yang berbeda atau bisa juga menggunakan *container* agar tidak ada ketergantungan antara *platform* tersebut. Selain itu, kasus yang diterapkan dapat lebih kompleks lagi misalkan untuk melakukan pemantau sumber daya server seperti konsumsi RAM, kinerja prosesor, ketersediaan ruang penyimpanan, dan juga performa jaringan.

Daftar Pustaka

[1] S. Munirathinam, “Industry 4.0: Industrial Internet of Things (IIOT),” *Adv. Comput.*, vol. 117, no. 1, pp. 129–164, Jan. 2020.

[2] B. Wang, X. Liu, and Y. Zhang, "Internet of Things," *Internet Things BDS Appl.*, pp. 71–127, 2022.

[3] P. K. Malik *et al.*, "Industrial Internet of Things and its Applications in Industry 4.0: State of The Art," *Comput. Commun.*, vol. 166, pp. 125–139, Jan. 2021.

[4] B. Diène, J. J. P. C. Rodrigues, O. Diallo, E. H. M. Ndoye, and V. V. Korotaev, "Data management techniques for Internet of Things," *Mech. Syst. Signal Process.*, vol. 138, p. 106564, Apr. 2020.

[5] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, and E. K. Markakis, "A Survey on the Internet of Things (IoT) Forensics: Challenges, Approaches, and Open Issues," *IEEE Commun. Surv. Tutorials*, vol. 22, no. 2, pp. 1191–1221, Apr. 2020.

[6] M. A. Husna and P. Rosyani, "Implementasi Sistem Monitoring Jaringan dan Server Menggunakan Zabbix yang Terintegrasi dengan Grafana dan Telegram," *JURIKOM (Jurnal Ris. Komputer)*, vol. 8, no. 6, pp. 247–255, Dec. 2021.

[7] D. Rahman, H. Amnur, and I. Rahmayuni, "Monitoring Server dengan Prometheus dan Grafana serta Notifikasi Telegram," *JITSI J. Ilm. Teknol. Sist. Inf.*, vol. 1, no. 4, pp. 133–138, Dec. 2020.

[8] K. T. Widagdo, T. I. Bayu, and Y. A. Susetyo, "Pemodelan Sistem Monitoring Sensor Curah Hujan Menggunakan Grafana," *Indones. J. Comput. Model.*, vol. 2, no. 2, pp. 1–8, Dec. 2019.

[9] H. - and W. Andriyani, "STUDI KOMPARASI MENYIMPAN DAN MENAMPILKAN DATA HISTORI ANTARA DATABASE TERSTRUKTUR MARIADB DAN DATABASE TIDAK TERSTRUKTUR INFLUXDB," *J. Teknol. TECHNOSCIENTIA*, pp. 168–174, Feb. 2020.

[10] T. J. Anih, C. A. Bede, and C. F. Umeokpala, "Detection of Anomalies in a Time Series Data using InfluxDB and Python," Dec. 2020.

[11] P. MacHeso, T. D. Manda, S. Chisale, N. Dzupire, J. Mlatho, and D. Mukanyiligira, "Design of ESP8266 Smart Home Using MQTT and Node-RED," *Proc. - Int. Conf. Artif. Intell. Smart Syst. ICAIS 2021*, pp. 502–505, Mar. 2021.

[12] N. Bin Kamarozaman and A. H. Awang, "IOT COVID-19 Portable Health Monitoring System Using Raspberry Pi, Node-Red and ThingSpeak," *IEEE Symp. Wirel. Technol. Appl. ISWTA*, vol. 2021-August, pp. 107–112, 2021.

[13] S. Bharath and C. Khusi, "IoT Based Smart Traffic System Using MQTT Protocol: Node-Red Framework," *2021 2nd Glob. Conf. Adv. Technol. GCAT 2021*, Oct. 2021.

[14] P. S. B. MacHeso, T. D. Manda, A. G. Meela, J. S. Mlatho, G. T. Taulo, and J. C. Phiri, "Industrial Temperature Monitor Based on NodeMCU ESP8266, MQTT and Node-RED," *Proc. - 2021 3rd Int. Conf. Adv. Comput. Commun. Control Networking, ICAC3N 2021*, pp. 740–743, 2021.

[15] V. David, H. Ragu, R. K. Duraiswamy, and P. Sasikumar, "IoT based Automated Indoor Agriculture System Using Node-RED and IBM Bluemix," *Proc. 6th Int. Conf. Inven. Comput. Technol. ICICT 2021*, pp. 157–162, Jan. 2021.