Implementation of environmental monitoring based on KAA IoT platform

M. Udin Harun Al Rasyid, M. Husni Mubarrok, Jauari Akhmad Nur Hasim
Politeknik Elektronika Negeri Surabaya (PENS), Indonesia

Article Info

Article history:

Received Feb 24, 2020
Revised Apr 25, 2020
Accepted May 18, 2020

Keywords:

Environment monitoring
IoT
KAA platform
Raspberry Pi
Wireless sensor network

ABSTRACT

Wireless sensor network (WSN) is a key to access the internet of things (IoT). The popularity of IoT and the prediction that there will be more devices connected to the Internet cause difficulties in integrating and making connected devices. The problem of IoT implementation are the lack of real-time data collection, processing, and the inability to provide continuous monitoring. To overcome these problems, this paper proposes an IoT device for monitoring environmental conditions through the IoT KAA platform that can be monitored anywhere and anytime in real time. The end device node consists of several sensors such as as temperature, humidity, carbon monoxide (CO) and carbon dioxide (CO_2) sensors. The collected data from the end device node will be transmitted via a communication based on IEEE 802.15.4 to Raspberry Pi gateway, then sent to the KAA cloud server and saved into the database. The environmental data can be accessed via a web-based sensor application. We Analize the performance evaluation in terms of transaction, availability, data transfer, response time, transaction rate, throughput, and concurrency. The experimental result shows that the use of KAA IoT platform is better than that without platform.

Corresponding Author:

M. Udin Harun Al Rasyid,
Department of Informatics and Computer Engineering,
Politeknik Elektronika Negeri Surabaya (PENS),
Jl. Raya ITS, Kampus PENS, Keputih, Kec. Sukolilo, Surabaya, Jawa Timur 60111, Indonesia.
Email: udinharun@pens.ac.id

1. INTRODUCTION

The popularity of wireless sensor network (WSN) attracts world wide attention. WSN is the key that allows internet of things (IoT) to play an important role in the future of the internet in collecting information from environment and surrounding context. WSN and sensing tools will have a big influence on IoT technology. Some physical sensors and actuator devices are placed in the environment for monitoring and control operations. Information collected through the sensor device will be sent automatically to the monitoring center, and then an analysis of the data will be done when the data arrive at the monitoring center [1-3].

IoT is a computing system related to devices, mechanical machines, digital, animals, people, or objects, which are given unique identifiers and abilities to transfer data through a network without the need for human-to-human interactions or human-to-computer interactions [4]. WSN as part of IoT technology have been used in various intelligent applications and services such as smart healthcare, smart building, smart home, smart cities, and smart environment [5, 6]. IoT has a vision to connect massively different entities which involves anything, any service and any media that can be accessed anytime and anywhere. The main purpose of IoT is data and intelligent decisions that result from data. More and more data are obtained, stored, processed, and analyzed with the advancement of public cloud computing technology [7].

The increasing number of the development of connected devices creates new problems in the integration of WSN and manufacturing systems. The issues related to IoT environmental monitoring methods include the lack of real-time data collection and processing, and the inability to provide continuous monitoring, heterogeneity of communication protocol in the IoT world and hardware, interoperability and back-end services. Thus, it takes longer time to apply the IoT. In order to overcome the above-mentioned issues, some researchers have conducted several investigations related to WSN, IoT framework and its applications in various fields such as green house monitoring [8], smart home [9-12], flood prediction [13], electricity energy monitoring [14], and water quality monitoring [15].

Abraham *et al.* [16] proposed remote monitoring of air condition and environmental soil quality based on intelligent data collection. The data obtained is displayed in a web-based that displays real-time data. Warning information will be displayed if it exceeds the limit of predetermined parameters. The device uses MQ-2 gas sensor and arella soil hygrometer, Raspberry Pi 2, ethernet, and Wi-Fi connection. Montori *et al.* [17] proposed an IoT architecture called SenSquare that combines sensor data from various IoT platforms. SenSquare integrates government data sources, open data sources, and crowdsensed sources sent by users. Then SenSquare will classify data collected using natural language processing (NLP) to be displayed on one display screen both on mobile and desktop. Spandana *et al.* [18] developed an air pollution monitoring system that integrates IoT, ESP8266 module, carbon dioxide (CO) sensor, carbon monoxide (CO_2) sensor, dust particle sensor, and GPS location. Users can monitor pollution levels via android mobile devices and monitor air pollution locations through Google Maps.

Tong *et al.* [19] proposed a simulation framework for monitoring energy-efficient environments by considering operating costs and device computing capabilities. The system uses scheduling to balance the monitoring process and energy consumption under different environmental conditions with low power wide area (LPWA) based. Wu *et al.* [20] proposes an IoT wearable device called WE-Safe to monitor dangerous environmental conditions through the LORA wireless network. The WE-safe sensor node is composed of MCU devices, temperature sensors, humidity sensors, ultra violet sensors, and CO_2 sensors. Punniamoorthy [21] proposed a temperature condition monitoring system using an LM35 sensor, detection of dangerous gases using an MQ-2 sensor, humidity detection using a DHT11 sensor, fire detection, and water level. Data from the microcontroller will be sent to the base station using the GSM module. If an emergency condition occurs, the microcontroller will analyze and send information to the owner via SMS and email.

Malche *et al.* [22] proposed the architecture of the IoT security application system for monitoring environmental conditions and alert systems. The system monitors air quality and noise pollution in certain areas. The sistem contains several sensors MG811 CO_2, MQ6 LPG sensor, MQ7 CO, MQ2 smoke, MQ131 ozone (O_3), MQ136 sulfur dioxide (SO_2), DHT22 temperature, and humidity. Yavari *et al.* [23] proposed monitoring hydrocarbon pollution from fuel stations. The sensing results will be sent to the cloud via the LoRa communication network, then the data will be analyzed and display a pollution map directly. Yang *et al.* [24] proposed ocean of things (OoT) framework for monitoring the condition of the marine environment uses IoT technology to improve the efficient use of data and information about the marine environment. The OoT framework consists of a data acquisition layer, a fog layer, and a cloud layer. Ghoneim *et al.* [25] proposed an air pollution detection system (APDS) using CO, nitrogen dioxide (NO_2), O_3 and SO_2 sensors. Johnston *et al.* [26] proposed a prototype of the particulate matter (PM) monitoring system installed in a small temporary city.

Based on the above state of the arts of research in WSN and IoT platform, the goal of this research is to use the IoT platform in the field of environment health monitoring system as the extention of our previous research which has not used the IoT platform [27]. It is necessary to do innovation of the integration of WSN and IoT. IoT middleware platform is used to build, manage, and integrate applications to provide a standard approach for integration and interoperability across connected products.

In this paper, we collect the data sensor such as temperature, humidity, CO and CO_2 from the sensor node. After being transmitted via a communication based on IEEE 802.15.4 to Raspberry Pi, the sensor data is sent to the KAA server and saved into the database that has been embedded in the KAA server. The information provided based on real time data, to be able to determine how the state of air quality is good or not, it needs a system that can monitor air quality, in which the system can be tracked live the results of sensor data of CO, CO_2, humidity and temperature in the environment. The system automatically assigns the decision of the measurement results from the sensor data in the form of warning is very good, good, medium, bad, and dangerous indicator. The data can be accessed via a web-based sensor application. And then we conduct a service quality analysis in terms of transaction, availability, data transfer, response time, transaction rate, throughput, and concurrency.

2. RESEARCH METHOD

In this section, we discuss our proposed system for environment monitoring. It contains overall system architecture, component of system, sensor modules, fuzzification, configuration of waspmote sensor node, implementation of KAA SDK in Raspberry Pi, and configuration of KAA platform. Figure 1 shows the general overview of the system design process. There are several sensor nodes to be in pairs. A sensor node is connected to the battery as a power source and XBee receiver as a communication tool. Gas sensor nodes are placed in some points to test and collect data such as gas concentration of CO and CO_2 temperature and humidity in the environment.

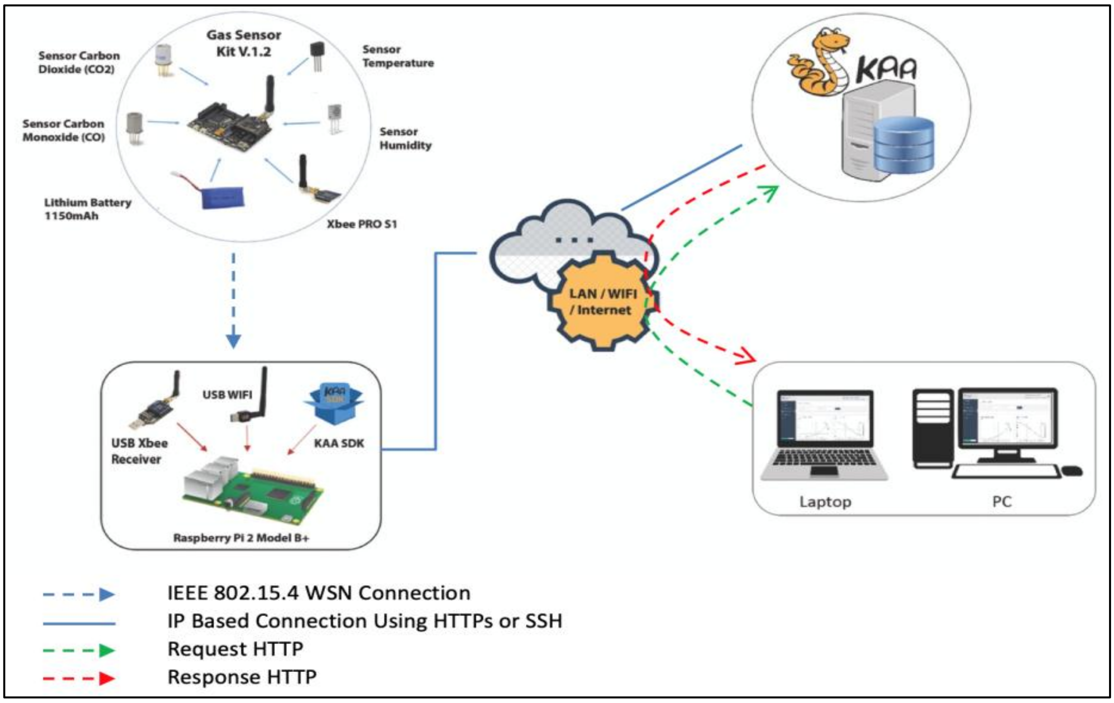


Figure 1. General system design

The sensors will capture the value of the concentration of CO, CO_2, temperature and humidity. They will send the value to the XBee transceiver using 802.15.4 ZigBee communication prior to the data transmission process. There is a decision support system using fuzzy logic as a decision making about environmental conditions on the measurement of sensor data, the data will be taken periodically every 5 minutes.

XBee receiver will connect to Raspberry Pi which would enable it to make communication between the receiver and the Raspberry Pi using a USB serial communications. The data received by the Raspberry Pi will be processed in the Raspberry Pi. In the raspberry pi [28], there is an application used as a communication and data processing. In getting the sensor, the data is received in the form of data frame which will be changed by separating the data corresponding to sensor data. This application uses libraries: RS232 and KAA SDK. KAA SDK is a library that provides communication, data marshalling, persistence, and other functions available in KAA for specific types of endpoints (eg, Java-based, based on C ++, C based) [29]. It is the responsibility of the KAA client to process structured data given by KAA server (configuration, notifications, etc.) to supply the data to interface. the data that has been in the process of KAA SDK will be stored by the database that is embedded in KAA server. In this case, we use MongoDB database.

The communication between the Raspberry Pi and KAA server is using the Internet and WiFi if accessed locally. The data stored in the MongoDB database that resides in the KAA server will be visualized in the form of web-based application. The application uses the servis of KAA server. This application will request KAA server to provide HTTP response in the form of JSON data. We use codeigniter framework (CI) to build web application.

User can access Web-based applications that have been created, and the website should be connected to the Internet to retrieve data from the server. From client side (front-end), user will send an HTTP request via web browser (when accessing the website) to web server that has been provided. Then the web service will send a response in the form of JavaScript object notation (JSON) to users who access the website. When the request is received, web service will retrieve data from the database to be displayed on web applications.

Figure 2 shows the flowchart of KAA platform configuration. The configuration of log schema, log appender, and SDK profiles that is set to KAA server through a web UI provided by KAA. This stage is the most important one to generate KAA SDK that will be implemented in the end-device. Before generating KAA SDK that will be implemented on the client, it is necessary to advance the configuration on KAA platform through KAA sandbox web console. Host IP configuration is required as the communication between the KAA server and KAA SDK/KAA client. This configuration is done in the sandbox KAA. KAA sandbox is a separated KAA environment that includes client demo application. KAA sandbox includes all the components required in a comfortable virtual environment that can be arranged in just 5 minutes.

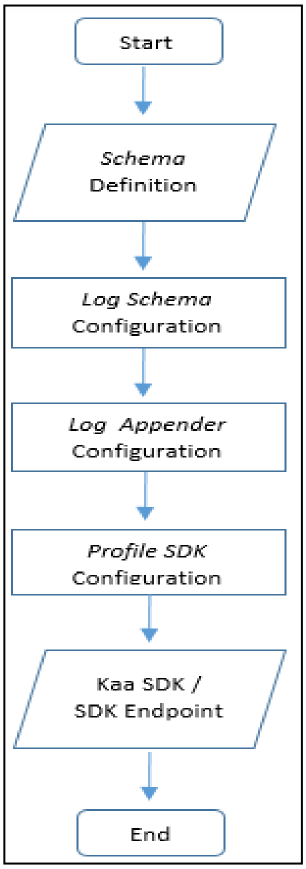
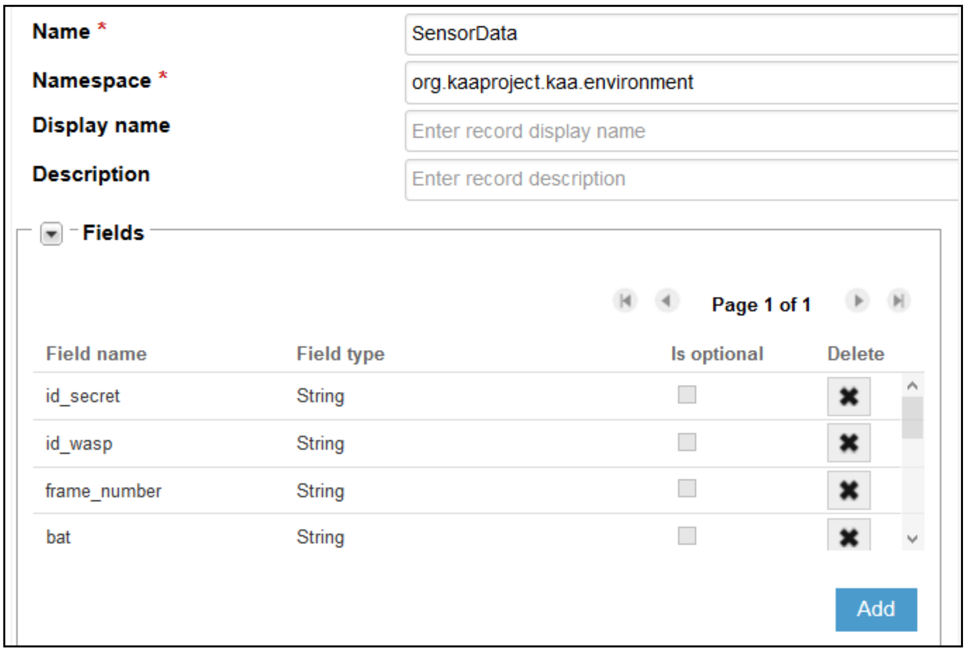

Start
Schema Definition
Log Schema Configuration
Log Appender Configuration
Profile SDK Configuration
Kaa SDK / SDK Endpoint
End

Figure 2. Flowchart to configure KAA

There are two ways to configure host IP KAA sandbox, through sandbox web UI and through a terminal console. Here is each way to configure the IP host KAA sandbox. The first is using the sandbox web UI. KAA sandbox web UI provides access to the KAA demo project and some basic configuration capabilities of KAA sandbox. After KAA sandbox is installed and opened, the web UI is available at the following URL (by default): http://127.0.0.1:9080/sandbox.

The second is using terminal console. Configuration via terminal console can be done by typing the command below using the access rights as root. The next discussion is about making schema constructor on Avro UI sandbox that has been provided by KAA platform. Manufacture schema can be done by filling the column that has been provided and by using JSON. The explanation about making schema using filling column is provided as shown in Figure 3 which illustrates the making with JSON schema that will be made into the column of form. After manufacturing schema to the Avro UI sandbox, the result is saved in the form of JSON. Then, the log schema and log appender are configured. The function of the log schema is to define the schema will be used to do versioning in the schema that has been made.

Log appender is used not only to determine the version of the log schema that will be used before it generates KAA SDK, but also log appender used to deliver the log records from a data collection application to local MongoDB instance, and then Collection will be made automatically.



Name *	SensorData
Namespace *	org.kaaproject.kaa.environment
Display name	Enter record display name
Description	Enter record description

Fields

Page 1 of 1

Field name	Field type	Is optional	Delete
id_secret	String	□	✖
id_wasp	String	□	✖
frame_number	String	□	✖
bat	String	□	✖

Figure 3. Schema constructor using field

Figure 4 shows the densing data from end device to KAA server. The data transmission process starting from sending waspmote sensor data and the resulted data. To send the data to the raspberry Pi that has been connected to the gateway (USB XBee receiver), waspmote transmit the data by using the Zigbee 802.15.4 protocol frame structure for each delivery. Frame structure read through Waspmote IDE will be written in hexadecimal format, but when initializing frame is sent as ASCII type, then the frame will also bring the translation frame information in the form of ASCII. As a result, the information in frame can be read more easily. One frame will be marked with a start delimiter and each of information will be separated by the separator "#". Such information should include id_Waspmote, id_frame, sensor type and value.

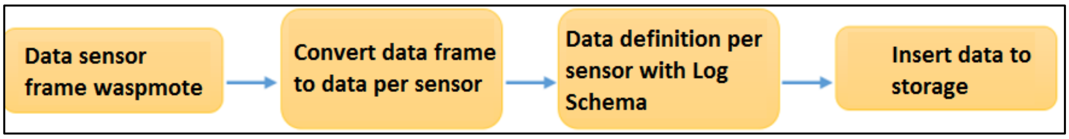

Data sensor frame waspmote
Convert data frame to data per sensor
Data definition per sensor with Log Schema
Insert data to storage

Figure 4. Flow of the data transmission process

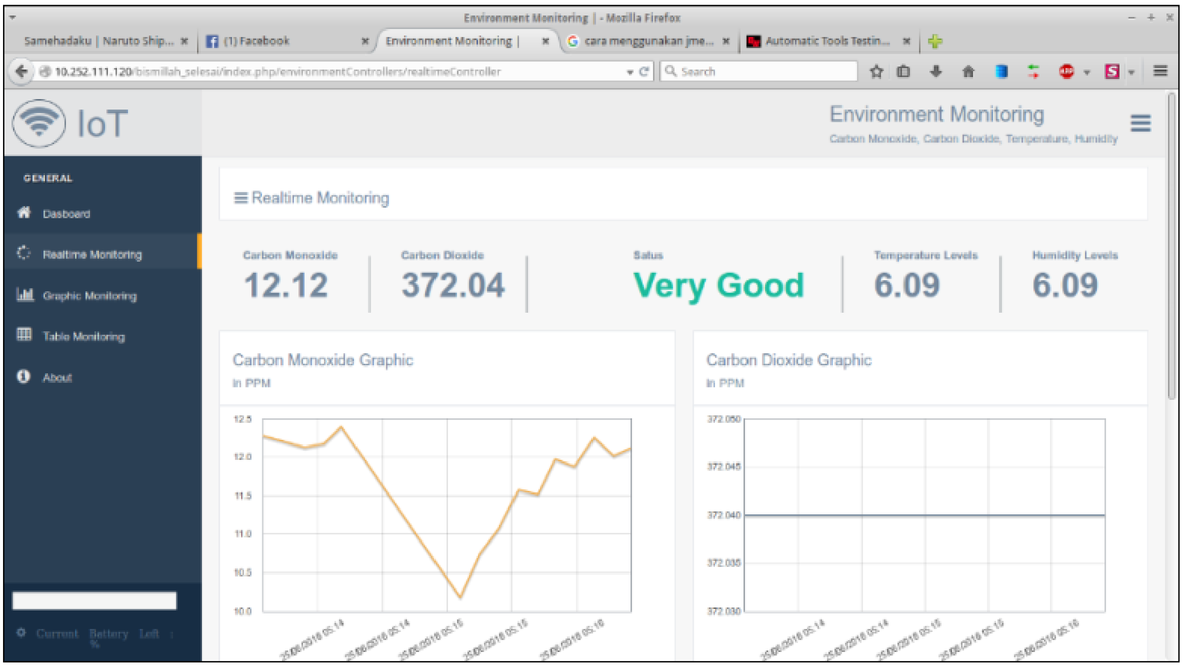
The data, in the form of a frame (STR), is received by the gateway (USB XBee receiver) through 802.15.4 Zigbee protocol. Data of the top frame will be converted by using the Raspberry Pi with C code which has been planted in it. This code not only implements KAA SDK, but also performs, from data frame into data per sensor. The data will be packaged in the form of frames (ASCII frame). Start delimiter " <=> " indicates that a new frame has been sent and become a marker for the next new frame. Each frame successfully received by the gateway will be converted by raspberry into data per sensor. Raspberry will define data per sensor with log schema; the purpose of this definition is a translation of the value of data per sensor to KAA SDK, so this SDK can communicate with KAA server and KAA SDK/client can perform data storage to the database already in KAA.

3. RESULTS AND DISCUSSION

This section discusses the testing process of the implementation of KAA IoT platform on the performance of the service provided. This test will compare the use of IoT application between KAA platform and the one using the platform in web-based application. The purpose of this test is to find out the performance of KAA with zwamp web server that includes the number of transactions, availability, data transferred, response time, transaction rate, throughput, and concurrency.

3.1. Web-based application

Figure 5 shows the realtime monitoring menu. The graphics on this application shows the data concentration levels of carbon monoxide, carbon dioxide, temperature and humidity by displaying the latest data. There is a numerical rating that displays not only 4 sensor data values of carbon monoxide, carbon dioxide, temperature and humidity, but also the most recent values. In addition, the application provides information of air condition which is very good, good, average, bad, and dangerous. Figure 6 shows realtime monitoring dashboard containing facts about air pollution, the description of CO, CO_2, temperature, humidity, and their impacts. There is also a real time digital data conditions updated continuesly evey 10 seconds.



Carbon Monoxide 12.12

Carbon Dioxide 372.04

Satus Very Good

Temperature Levels 6.09

Humidity Levels 6.09

Figure 5. Realtime monitoring menu

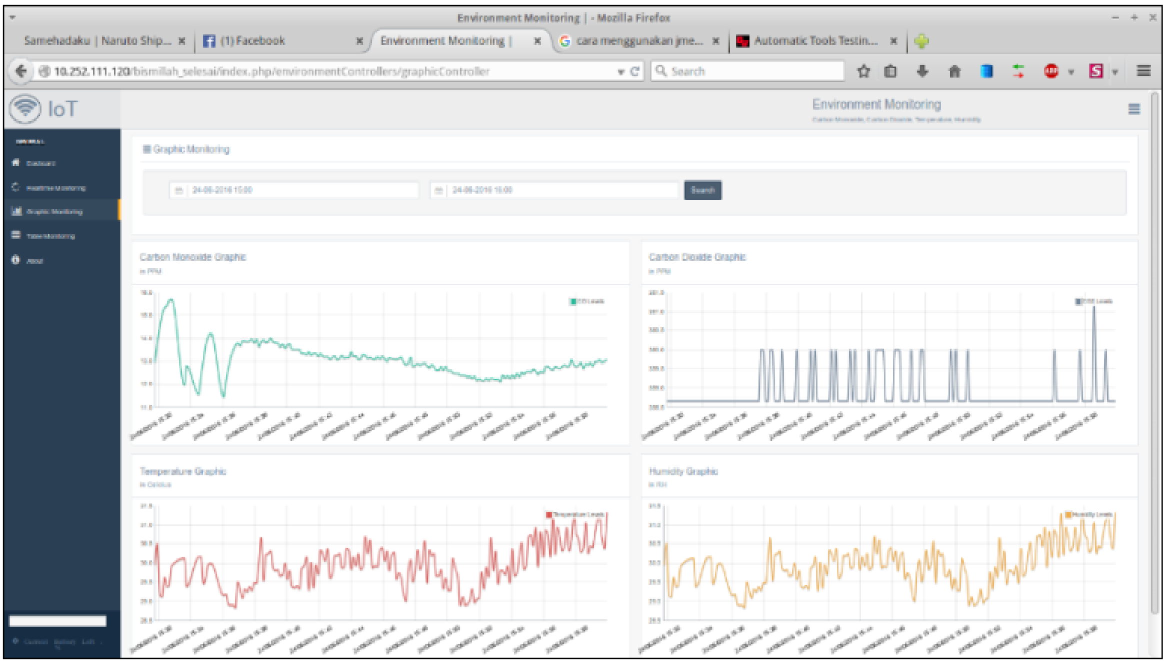


Figure 6. Graphic monitoring menu

3.2. Application performance comparation between KAA platform and without one

To test application performance using KAA platform and without platform, we simulated the upload data from sensing by KAA endpoint to KAA server and downloaded the results of sensing by the client from the server. We used a tool called "Siege" to test and define KAA performance. We tested the performance of the transaction, availability, data transfer, response time, transaction rate, throughput, and concurrency, as described bellow:

- Transaction is the number of server hits.
- Availability is readiness in terms of both servers and endpoints to exchange/send data in KAA platform.
- Data transfer is the number of data generated on the endpoint or the sensor to be sent to the server. It includes the header information as well as content. Because it includes header information, the number reported by siege will be larger than the number reported by the server. In internet mode, which hits random URLs in a configuration file, this number is expected to vary from run to run.
- Response time is the average time to respond to each simulated user's requests.
- Throughput is the average number of bytes transferred every second from the server to all the simulated users.
- Concurrency is the average number of simultaneous user connections, the greater number of concurrencies, the more decressing the server performance.

Table 1 shows that the transaction and availability of KAA platform is bigger than that without platform. Although the number of user concurrent increases, the availability of KAA platform is constant at 100% and gains more transactions than that without IoT platform.

Table 1. Result of transaction (hits) and availability (%)

User (Concurrent)	Transaction (hits)		Availability (%)	
	KAA platform	Without platform	KAA platform	Without platform
10	1162	1188	100%	100%
50	5888	5730	100%	100%
100	11702	11472	100%	100%
150	17502	14804	100%	100%
200	23364	14398	100%	99,65%
250	29144	14462	100%	99,87%
300	33766	14328	100%	99,18%
350	33416	14706	100%	100%
400	34022	14216	100%	99,9%

Table 2 shows that the data transfer and response time of KAA platform is less than that of without platform. KAA platform has its own method of sending data and models such as data encryption so that the size will be smaller and safer. If the size of the data to be sent is smaller, it is likely that such data will not lose data, and the data will be faster up and not spend a lot of bandwidth.

Table 2. Result of data transfer (MB) and response time (s)

User (concurrent)	Data transfer (MB)		Response time (s)	
	KAA platform	Without platform	KAA platform	Without platform
10	4,09	25,64	0,11	0,11
50	20,72	123,66	0,07	0,12
100	41,19	247,58	0,07	0,2
150	61,6	319,5	0,08	1,04
200	82,23	310,8	0,08	3,19
250	102,58	312,17	0,11	5,22
300	118,84	309,33	0,29	7,19
350	117,62	317,44	1,21	8,96
400	119,76	306,88	1,98	11,48

Table 3 shows that the transaction rate of KAA platform is higher than that without platform, while the use of throughput of KAA platform is lower than that without platform. It means using KAA platform can gain more transaction rate and reduce the use of bandwidth. Table 4 shows that the concurrency of KAA platform is lower than that without platform. Concurrency is the process that occurs at the same time. The more the number of concurrency, the harder the hardware works, causing the device to be hot. The fewer the number of concurrency, the better the performance is.

Table 3. Result of transaction rate (trans/s) and throughput (MB/s)

User (concurent)	Transaction rate (trans/s)		Throughput (MB/s)	
	KAA platform	Without platform	KAA platform	Without platform
10	1,94	1,98	0,01	0,04
50	9,82	9,56	0,03	0,21
100	19,53	19,13	0,07	0,41
150	29,19	24,67	0,1	0,53
200	38,96	24	0,14	0,57
250	48,63	24,11	0,17	0,52
300	56,34	23,88	0,2	0,52
350	55,75	24,54	0,2	0,53
400	56,71	23,73	0,2	0,51

Table 4. Result of concurrency

User (concurrent)	Concurrency	
	KAA platform	Without platform
10	0,21	0,21
50	0,72	1,13
100	1,46	3,73
150	2,23	25,63
200	2,98	76,52
250	5,22	125,93
300	16,07	171,8
350	67,57	220
400	112,5	272,3

4. CONCLUSION

This study proposes an IoT for monitoring environmental conditions through the IoT KAA platform base that can be monitored in real time anywhere and anytime. We collected the sensor data such as temperature, humidity, CO and CO_2 from the sensor nodes. After being transmitted via a communication based on IEEE 802.15.4 to Raspberry Pi, the sensor data is sent to the KAA server and saved into the database that has been embedded in the KAA server. Based on the experiment results, KAA is a reliable and flexible platform to use and develop, and it makes it easy for developers to build IoT applications. KAA platform has good scalability in terms of transaction, availability, data transfer, response time, transaction rate, throughput, and concurrency. With the implementation of KAA platform, the concentration of CO, CO_2, temperature, and humidity can be monitored remotely through web applications with better performance. For future works, mobile applications will be developed to monitor the condition of the air environment from anywhere and anytime.

ACKNOWLEDGEMENTS

This research was supported by research institute of Politeknik Elektronika Negeri Surabaya (PENS) Indonesia and EEPIS Wireless Sensor Network (EWSN) research group.

REFERENCES

[1] M. Altayeb, S. M. Sharif and S. Abdella, “The Internet-of-Things and Integration with Wireless Sensor Network Comprehensive Survey and System Implementation,” *2018 International Conference on Computer, Control, Electrical, and Electronics Engineering ICCCEEE*, Khartoum, pp. 1-6, 2018.

[2] R.A. Rahman, U.R. Hashim, S. Ahmad, “IoT based temperature and humidity monitoring framework,” *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 1, pp. 229-237, February 2020.

[3] M. Zhang, X. Li and L. Wang, “An Adaptive Outlier Detection and Processing Approach Towards Time Series Sensor Data,” in *IEEE Access*, vol. 7, pp. 175192-175212, 2019.

[4] O. Elijah, T. A. Rahman, I. Orikumhi, C. Y. Leow and M. N. Hindia, “An Overview of Internet of Things (IoT) and Data Analytics in Agriculture: Benefits and Challenges,” in *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3758-3773, Oct. 2018.

[5] Guobao Xu,Yanjun Shi,Xueyan Sun, Weiming Shen, “Internet of Things in Marine Environment Monitoring: A Review,” *Sensors*, vol. 19, no. 7, pp. 1-21, 2019.

[6] D. R. Tisna, M. Udin Harun Al Rasyid and S. Sukaridhoto, “AT-Mo: Wireless Data Collection System For Physiology Monitoring of Athlete,” *2019 International Electronics Symposium IES*, Surabaya, Indonesia, pp. 115-119, 2019.

[7] G. Marques, R. Pitarma, N.M. Garcia, N. Pombo, “Internet of Things Architectures, Technologies, Applications, Challenges, and Future Directions for Enhanced Living Environments and Healthcare Systems: A Review,” *Electronics*, vol. 8, pp. 1-27, 2019.

[8] Y. Liu, K. A. Hassan, M. Karlsson, Z. Pang, S. Gong, "A Data-Centric Internet of Things Framework Based on Azure Cloud," in *IEEE Access*, vol. 7, pp. 53839-53858, 2019.

[9] L. Liao *et al*., "Design and Validation of a Multifunctional Android-Based Smart Home Control and Monitoring System," in *IEEE Access*, vol. 7, pp. 163313-163322, 2019.

[10] M. U. Harun Al Rasyid, D. Enda and F. A. Saputra, "Smart Home System for Fire Detection Monitoring Based on Wireless Sensor Network," *2019 International Electronics Symposium IES*, Surabaya, Indonesia, pp. 189-194, 2019.

[11] D. Assante and C. Fornaro, "An Educational IoT-based Indoor Environment Monitoring System," *2019 IEEE Global Engineering Education Conference EDUCON*, Dubai, United Arab Emirates, pp. 1475-1479, 2019.

[12] W-T. Sung, S-J. Hsiao, J-A. Shih, "Construction of Indoor Thermal Comfort Environmental Monitoring System Based on the IoT Architecture," *Journal of Sensors*, pp. 1-16, 2019.

[13] Salami Ifedapo Abdullahi, Mohamed Hadi Habaebi, Noreha Abd Malik, "Intelligent flood disaster warning on the fly: developing IoT-based management platform and using 2-class neural network to predict flood status," *Bulletin of Electrical Engineering and Informatics*, vol. 8, no. 2, pp. 706-717, June 2019.

[14] Meysam Shamshiri, Chin Kim Gan, Kyairul Azmi Baharin, Mohd Arif Mohd Azman, "IoT-based electricity energy monitoring system at Universiti Teknikal Malaysia Melaka," *Bulletin of Electrical Engineering and Informatics*, vol. 8, no. 2, pp. 683-689, June 2019.

[15] Muhammad Aznil Ab Aziz, M. F. Abas, Mohamad Khairul Anwar Abu Bashri, N. Md. Saad, M. H. Ariff, "Evaluating IoT based passive water catchment monitoring system data acquisition and analysis," *Bulletin of Electrical Engineering and Informatics*, vol. 8, no. 4, pp. 1373-1382, December 2019.

[16] S. Abraham, J. Beard and R. Manijacob, "Remote environmental monitoring using Internet of Things (IoT)," *2017 IEEE Global Humanitarian Technology Conference GHTC*, San Jose, CA, pp. 1-6, 2017.

[17] F. Montori, L. Bedogni, L. Bononi, "A Collaborative Internet of Things Architecture for Smart Cities and Environmental Monitoring," *IEEE Internet of Things Journal*, vol. 5, pp. 592-605, April 2018.

[18] G. Spandana and R. Shanmughasundram, "Design and Development of Air Pollution Monitoring System for Smart Cities," *2018 Second International Conference on Intelligent Computing and Control Systems ICICCS*, Madurai, India, pp. 1640-1643, 2018.

[19] J. Tong, M. Cui, M. Tian and Y. He, "Surrogate Model-Based Energy-Efficient Scheduling for LPWA-Based Environmental Monitoring Systems," in *IEEE Access*, vol. 6, pp. 59940-59948, 2018.

[20] F. Wu, J. Redouté and M. R. Yuce, "WE-Safe: A Self-Powered Wearable IoT Sensor Network for Safety Applications Based on LoRa," in *IEEE Access*, vol. 6, pp. 40846-40853, 2018.

[21] D. Punniamoorthy, V. S. Kamadal, B. Srujana Yadav and V. S. Reddy, "Wireless Sensor Networks for Effective Environmental Tracking System Using IoT and Sensors," *2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2018 2nd International Conference on*, Palladam, India, pp. 66-69, 2018.

[22] T. Malche, P. Maheshwary, R. Kumar, "Environmental Monitoring System for Smart City Based on Secure Internet of Things (IoT) Architecture," *Wireless Personal Communications*, vol. 107, pp. 2143-2172, 2019.

[23] A. Yavari, D. Georgakopoulos, P. R. Stoddart and M. Shafiei, "Internet of Things-based Hydrocarbon Sensing for Real-time Environmental Monitoring," *2019 IEEE 5th World Forum on Internet of Things WF-IoT*, Limerick, Ireland, pp. 729-732, 2019.

[24] J. Yang, J. Wen, Y. Wang, B. Jiang, H. Wang and H. Song, "Fog-Based Marine Environmental Information Monitoring Toward Ocean of Things," in *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4238-4247, May 2020.

[25] M. Ghoneim and S. M. Hamed, "Towards a Smart Sustainable City: Air Pollution Detection and Control using Internet of Things," *2019 5th International Conference on Optimization and Applications ICOA*, Kenitra, Morocco, pp. 1-6, 2019.

[26] S. J. Johnston *et al*., "Testing Smart City environmental monitoring technology using small scale temporary cities," *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, Limerick, Ireland, pp. 578-583, 2019

[27] N. Fahmi, M.U.H. A. Rasyid, A. Sudarsono, "Adaptive Sleep Scheduling for Health Monitoring System Based on the IEEE 802.15.4 Standard," *EMITTER International Journal of Engineering Technology*, vol. 4, no. 1, pp. 91-124, 2016.

[28] P.K.Das, P.K. Malik, R. Singh, A. Gehlot, K.V. Gupta, A. Singh, "Industrial Hazard Prevention Using Raspberry Pi," *International Conference on Intelligent Computing and Smart Communication ICSC*, pp. 1487-1499, 2019.

[29] KAA IoT Platform. [Online]. Available: https://www.Kaaproject.org/

BIOGRAPHIES OF AUTHORS

M. Udin Harun Al Rasyid received the B.Sc. degree in the Informatics Engineering Department from Sepuluh Nopember Institute of Technology (ITS), Indonesia, in 2004 and the Ph.D. degree in Computer and Communication Network Program, College of Electrical Engineering and Computer Science (CECS) from National Taiwan University of Science and Technology (NTUST), Taiwan, in 2012. He is currently an Associate Professor at Informatics and Computer Engineering Department, Politeknik Elektronika Negeri Surabaya (PENS), Indonesia. He heads the research group of EEPIS Wireless Sensor Networks (EWSN) PENS. His Research Interest is mainly in Wireless Sensor Network (WSN), Wireless Body Area Network (WBAN), Internet of Things (IoT), and Web Technology.

M. Husni Mubarrok is a programmer who has received the Associate’s Degree (AMd) in Informatics Engineering Department from Electronic Engineering Polytechnic of Surabaya (EEPIS) or PENS, Indonesia in 2016. He was interested in research Internet of Things (IoT), Wireless Sensor Networks (WSN) and Web Technology. He is currently a Programmer wich is research in Internet of Things (IoT) and web technology to build medical technology platform specifically in hospital laboratories to help facilitate of medical personnel work for checking the patient's laboratory analysis.

Jauari Akhmad Nur Hasim received the Master Degree in the Informatics Engineering Department from Sepuluh November Institute of Technology (ITS), Indonesia 2018. He is currently lecturer at Informatics and Computer Engineering Department, Politeknik Elektronika Negeri Surabaya (PENS), Indonesia. He also member from Hazard and Disaster Research Center (HDRC). His Research Interest in Software Engineering, Geographic Information System (GIS), Internet of Things (IoT), Multimedia, Web Technology and Design Pattern.