Scientific Journal of Informatics
Vol. 8, No. 2, Nov 2021

p-ISSN 2407-7658 http://journal.unnes.ac.id/nju/index.php/sji e-ISSN 2460-0040

Real-Time Monitoring: Development of Low Power Fire Detection System for Dense Residential Housing Based on Internet of Things (IoT) and Cloud Messenger

Rifki Muhendra[1*], **and Aisyah Amin**[2]
[1]Industrial Engineering, Bhayangkara Jakarta Raya University, Indonesia
[2]Physics Department, Faculty of Mathematics and Natural Sciences, Universitas Halim Sanusi, Indonesia

Abstract.
Purpose: This paper reports on the development of a low-power fire potential monitoring system for densely populated housing based on IoT
Study design: This system consists of sensors integrated with a microcontroller and a Wi-Fi module that can provide data in real-time and can be accessed remotely. In addition, this system has also been simulated to analyze environmental conditions as notification data sent using cloud messenger. The grouping of potential fire hazards based on the legibility of physical parameter values is also displayed.
Findings: In performance measurement, the data loss of this system is less than 1% of the total data every day.
Value: It shows that this system is feasible and applied more broadly to anticipate fires in densely populated areas.

Keywords: Fire potential, Real-time monitoring, IoT, Cloud Messenger
Received June 2021 / **Revised** October 2021 / **Accepted** November 2021

INTRODUCTION

Fires in densely populated areas are regional disasters that continue to occur every year. It is not only materially detrimental but also the death toll continues to increase due to this fire. Data from the DKI Jakarta Provincial Fire and Rescue Service website stated that there were 3 factors causing fires in the DKI Jakarta area. There are unsafe electrical installations, the habit of disposing of cigarette butts carelessly, and neglecting the use of stoves, as shown in Figure 1 below[1].

Fires caused by negligence in using the stove must continue to be watched out for. This is because the stove is a dated home cooking utensil that is almost always present in a house. These fires are often caused by a gas leak that is not detected by the occupants and the possibility of a lighter and a large oxygen capacity[2],[3]. Especially in densely populated areas, a fire in one house will cause a bigger fire for nearby houses. In addition, because of the small access road, fire suppression takes a long time.

For this reason, as an anticipatory step, a real-time monitoring system is needed to provide accurate information about gas leaks and detect potential fires. Huang, 2020 has developed Fire Detection and Recognition Optimization Based on Virtual Reality Video Images [4]. Pasquale Foggia, 2015 also developed Real-Time Fire Detection for Video-Surveillance Applications Using a Combination of Experts Based on Color, Shape, and Motion[5] and Fire Emissivity Detection by a Microwave Radiometer by Petr Dvorak, 2015[6]. These systems can detect potential fires in real-time but is complicated to apply in densely populated areas.

Simpler real-time fire monitoring systems are being researched. Kharisma, 2017 has succeeded in designing a Early Fire Detection System manufactured Using ATmega8 [7]. Dewi, 2017 also succeeded in designing a Fire Detection System in Cases of Gas Leaks Based on SMS Gateway [8]. These systems are quite simple and easy to implement.

---

[*] Corresponding author
Email addresses: rifki.muhendra@dsn.ubharajaya.ac.id (Rifki), aisyahmuhendra55@gmail.com (Aisyah)
DOI: 10.15294/sji.v8i2.30811

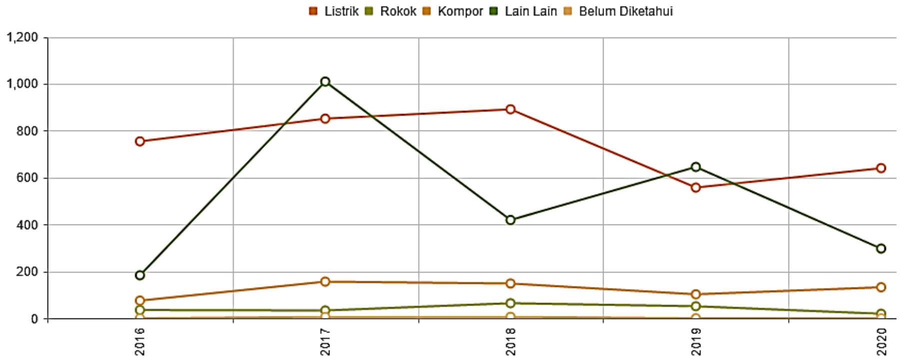


Figure 1. Graph of fire statistics based on the cause of the DKI Jakarta Provincial Fire and Rescue Service (www.jakartafire.net/statistic)

In this research, an IoT-based fire detection instrumentation system is proposed. This system develops electronic technology and information systems from previous technologies, namely sensor systems, microcontrollers, and Wi-Fi modules. This system can provide gas level information in real-time along with other physical parameters that affect the potential for fire. In addition, the system is also equipped with cloud messenger-based notifications as an emergency notification when there is a gas leak. The art of the state of this research is how to build a low power system, simple, and has a high speed of information delivery.

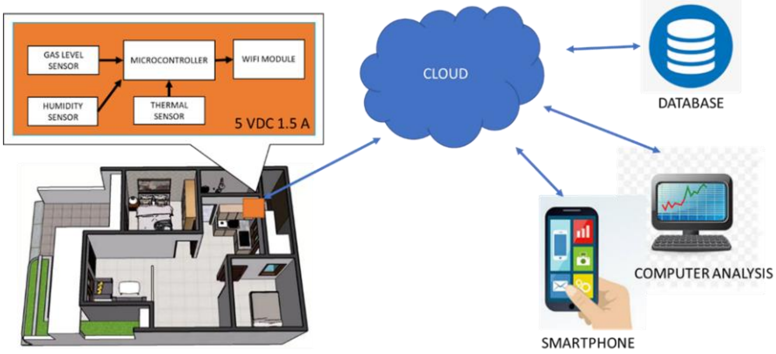


Figure 2. Low Power Fire Detection System Based on Internet Of Things (IoT) and Cloud Messenger (source: Internal)

DESIGN OF SYSTEM

The design of this system consists of three main stages, namely the design of low-power fire sensor hardware, an IoT cloud-based communication system, and notification of fire hazards using a messenger cloud, as shown in Figure 2. This system can measure the values of temperature, humidity, and environmental gas levels. These values are then processed by the microcontroller and then sent to cloud computing on the internet. This computing cloud is in the form of an IoT web server that can display measurement values in real-time and analyze data. If the sensor detects a potential fire based on temperature, humidity, and gas level values, the sensor will send a notification to the cloud messenger.

IoT based fire sensor and Cloud Messenger

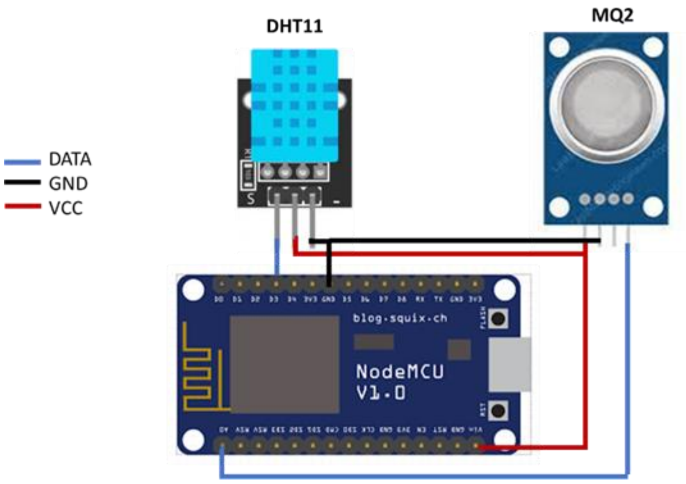


Figure 3. IoT based fire sensor and Cloud Messenger Hardware
(source: Internal)

In general, IoT and Cloud Messenger sensor hardware consists of 3 components: temperature and humidity sensors, gas level sensors, and a microcontroller + Wi-Fi module. The temperature and humidity sensor used in this system is the DHT11 sensor module. DHT11 works to measure air temperature and humidity simultaneously, which includes a thermistor type NTC (Negative Temperature Coefficient) to measure temperature, a humidity sensor with resistive characteristics to changes in air content. The sensor in the air, a chip inside performs several Analog to digital conversions and outputs output in single-wire bi-directional format. DHT11 works on 3v to 5v voltage with a maximum current consumption of 2.5mA when used during conversion (when requesting data). The gas level sensor used in this system is the MQ2 sensor module. MQ2 is a sensor MQ-2 is a sensor that is sensitive to cigarette smoke.

The main material of this sensor is SnO2 with low conductivity in clean air. If there is a leak, the conductivity of the sensor gas becomes higher, with every increase in gas concentration, the conductivity of the sensor also increases. The MQ-2 sensor is sensitive to LPG, Propane, Hydrogen, Carbon Monoxide, Methane and Alcohol gases, and other combustible gases in the air. MQ2 works at a voltage of 5 V with small current consumption of 150 mA. NodeMCU is an ESP8266 board that functions as a microcontroller and also a Wi-Fi-based communication module. ESP8266 is a Wi-Fi module that functions as an additional microcontroller such as Arduino to connect directly to Wi-Fi and make TCP/IP connections[9]. This versatile Wi-Fi module is already an SoC (System on Chip), so we can do programming directly to the ESP8266 without needing an additional microcontroller. The NodeMCU operates at 3.3 V and consumes no more than 150 mA of current.

NodeMCU is the central processing and communication in the system being built. NodeMCU is programmed using Arduino IDE to be able to read sensors, process data and send data to cloud IoT, and send notifications to cloud messenger. The flowchart of the NodeMCU programming algorithm can be seen in Figure 4.

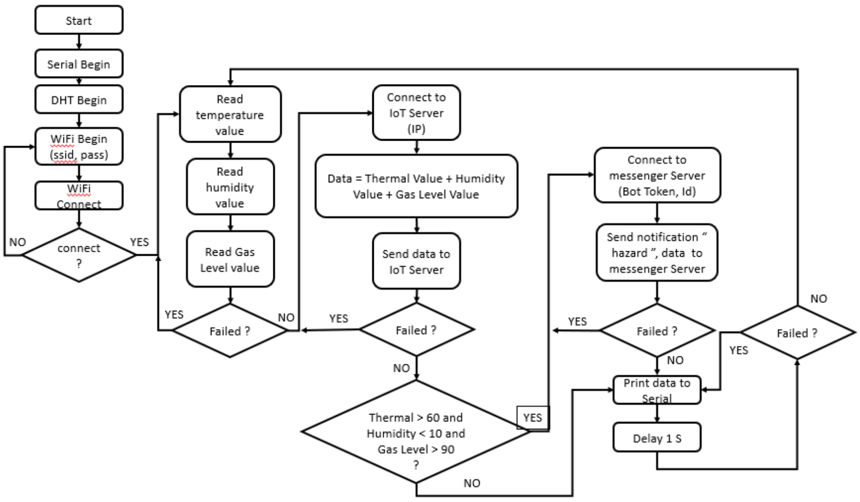


Figure 4. Flowchart of IoT-based fire sensor software and Cloud Messenger

ThingSpeak as IoT cloud and data analytics

ThingSpeak is an IoT analytics platform service that makes it possible to collect, visualize and analyze data streams directly in the cloud. ThingSpeak provides instant visualization of the data posted by the device to ThingSpeak. With the ability to execute MATLAB® code in ThingSpeak, online analysis and data processing at login are possible. ThingSpeak is often used for prototyping and proof-of-concept of IoT systems that require analytics[10].

Account creation is the first thing to do to be able to use ThingSpeak as a computing and analytics cloud. Then, the channel creation along with the settings is carried out. After that, the API key degenerates automatically. This code will be used to send data to ThingSpeak by NodeMCU. Users can use computers or mobile phones to access measurement data in real-time. In addition, for analysis purposes, it can be done manually by exporting data from ThingSpeak to a document file or you can also do it directly using the MATLAB-based analysis feature on ThingSpeak. ThingSpeak portal can be seen in Figure 5.

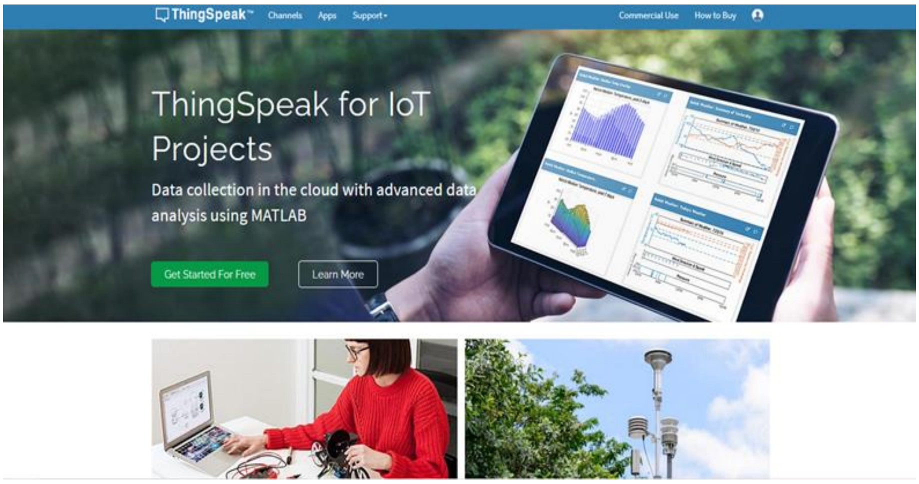
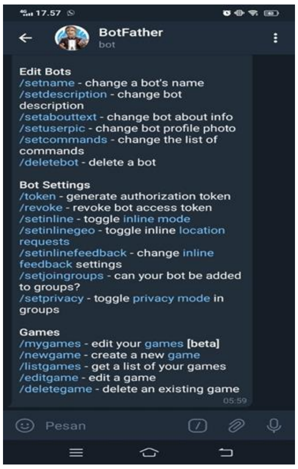


Figure 5. ThingSpeak Interface (source: https://ThingSpeak.com)

Fire emergency notification using the cloud messenger “Telegram”



Edit Bots
/setname - change a bot's name
/setdescription - change bot description
/setabouttext - change bot about info
/setuserpic - change bot profile photo
/setcommands - change the list of commands
/deletebot - delete a bot

Bot Settings
/token - generate authorization token
/revoke - revoke bot access token
/setinline - toggle inline mode
/setinlinegeo - toggle inline location requests
/setinlinefeedback - change inline feedback settings
/setjoingroups - can your bot be added to groups?
/setprivacy - toggle privacy mode in groups

Games
/mygames - edit your games [beta]
/newgame - create a new game
/listgames - get a list of your games
/editgame - edit a game
/deletegame - delete an existing game
05:59

Figure 6. Telegram Bot

Telegram is a free and non-profit cloud-based multi-platform instant messaging service application. Telegram clients are available for mobile phone devices (Android, iOS, Windows Phone, Ubuntu Touch) and computer system devices (Windows, OS X, Linux). Users can send messages and exchange photos, videos, stickers, audio, and other types of files. Telegram also provides optional end-to-end encrypted messaging. The Telegram bot is more specifically used for fire emergency notifications in this system because it is simple and supports IoT applications[11]–[14]. The telegram bot display can be seen in Figure 6.

Telegram Bot is a special account that does not require an additional phone number to set up. Messages, commands, and requests sent by the user are forwarded to the software running on the server. users communicate with these servers via a simple HTTPS interface that offers a simplified version of the Telegram API or commonly called API Bot. To connect the telegram bot with a sensor device, the user must create a bot account. After the bot account is created, Telegram will give you a Bot Token which will then be used in NodeMCU programming

RESULT AND DISCUSSION

The physical form of IoT-based sensor devices and cloud messengers can be seen in Figure 7. This sensor device is supplied with a 5 VDC adapter with a maximum current of 1.5 A. This supply value can be classified as low power because it has low voltage and current values but can still meet the operational needs of the sensor device. Each sensor also works at a voltage of 5 VDC which is taken from the Vin pin of the NodeMCU, while the ground sensors will be connected to the NodeMCU ground. The temperature and humidity sensor data pins are connected to one of the digital pins and the gas level sensor data pins are connected to the analog pins of the NodeMCU. In addition to measuring sensor values, NodeMCU also carries out data processing and delivery to cloud computing and cloud messenger-based notifications using the open-source IP protocol for the Wi-Fi module. Collecting data from sensors and sending data to cloud computing is repeated every 15 seconds. Meanwhile, sending notification of potential fires is only done if it fulfills the conditions as described in the programming algorithm.

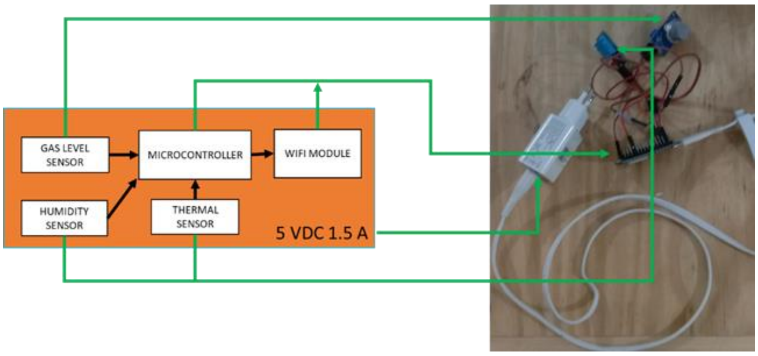


Figure 7. The physical form of the sensor device

The monitoring display of the real-time fire detection system that has been built can be seen in Figure 8. In general, this display is divided into two parts: the display in graphs and meters. Display in the form of a graph is used to see changes in sensor values based on the time of data receipt. This can help the user to interpret the state of the environment around the installation of the sensor device. The display in the form of a meter clarifies the user's reading at the last time the data was received. The display of this meter can be set according to the threshold value of each sensor. For example, the gas level meter where the range above the value of 90 can be categorized as a hazard.

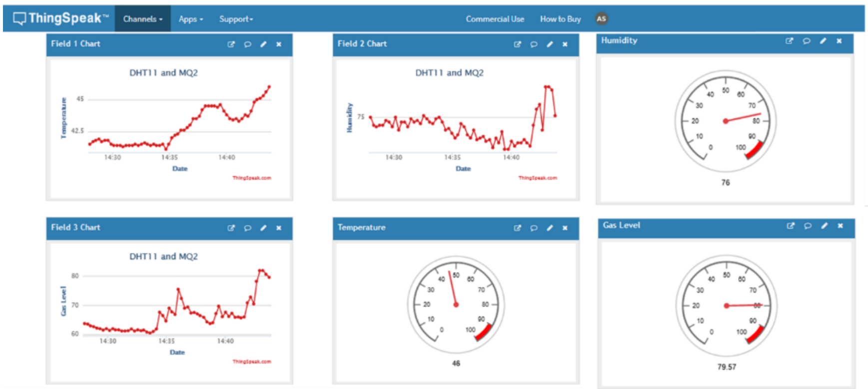


Figure 8. Display of fire detection monitoring in cloud computing using ThingSpeak

Simulation

This section reports on the simulation of the sensor device response in response to the surrounding environment. This simulation is divided into 2 parts: a simulation using matches and an initial simulation on installing sensors in the kitchen.

1) Simulation of sensor device response using lighter

```
Waiting...
Temperature: 34.30 degrees Celcius, Humidity: 95.00%, Gas Level: 54.84%. Send to Thingspeak.
Waiting...
Temperature: 34.30 degrees Celcius, Humidity: 95.00%, Gas Level: 55.52%. Send to Thingspeak.
Waiting...
Temperature: 34.30 degrees Celcius, Humidity: 95.00%, Gas Level: 54.94%. Send to Thingspeak.
Waiting...
Temperature: 34.30 degrees Celcius, Humidity: 95.00%, Gas Level: 55.91%. Send to Thingspeak.
Waiting...
Temperature: 34.30 degrees Celcius, Humidity: 95.00%, Gas Level: 55.72%. S
Waiting...
Temperature: 34.30 degrees Celcius, Humidity: 95.00%, Gas Level: 54.64%. S
```

```
Temperature: 60.00 degrees Celcius, Humidity: 8.00%, Gas Level: 61.19%. Send to Thingspeak.
Waiting...
Temperature: 60.00 degrees Celcius, Humidity: 8.00%, Gas Level: 61.19%. Send to Thingspeak.
Waiting...
Temperature: 60.00 degrees Celcius, Humidity: 9.00%, Gas Level: 61.29%. Send to Thingspeak.
Waiting...
Temperature: 60.00 degrees Celcius, Humidity: 9.00%, Gas Level: 61.49%. Send to Thingspeak.
Waiting...
Temperature: 60.00 degrees Celcius, Humidity: 11.00%, Gas Level: 62.46%. Send to Thingspeak.
Waiting...
Temperature: 60.00 degrees Celcius, Humidity: 11.00%, Gas Level: 62.85%. Send to Thingspeak.
Waiting...
Temperature: 60.00 degrees Celcius, Humidity: 15.00%, Gas Level: 62.95%. Send to Thingspeak.
Waiting...
Temperature: 60.00 degrees Celcius, Humidity: 15.00%, Gas Level: 63.05%. Send to Thingspeak.
```

Figure 9. Simulation of sensor device response using a lighter

In this simulation, the sensor readings are compared for the state without fire and the sensor being brought closer to the fire as shown in Figure 9. In a state without fire, the sensor readings include temperature around 34 0C, humidity 95%, and gas level. worth 55.91%. Sensor values like this can be categorized as normal state values. The changes up and down the value of the sensor readings are only influenced by natural conditions. In the state of the sensor near the fire, the sensor readings include: temperature is around 60 0C, humidity is 11% and gas level is 62.85%. Sensor values like this can be categorized as fire hazard values. In this case, the sensor device is set to send fire alert notifications to cloud messenger. With this notification, it is hoped that there will be direct handling steps by the user to anticipate fires.

2) Simulation on the installation of sensors in the kitchen

Figure 10. Simulation of the installation of sensor devices in the kitchen

In this section, a simulation of the installation of sensor devices in the kitchen is carried out as shown in Figure 10. This simulation aims to determine the response of sensor devices to activities carried out in the kitchen, especially in terms of cooking. The sensor device automatically measures the temperature, humidity, and gas level values then sends them to ThingSpeak. This simulation was carried out at 14.58 to 20.21 WIB. The results of this simulation measurement can be seen in Figure 11.

The graph in Figure 11 shows the measurement values for temperature, humidity, and gas levels based on activities in the kitchen and the surrounding environment. The units for temperature measurement are degrees Celsius, percent (%) for humidity and gas levels. From 2:59 p.m. to around 4:52 p.m., the sensor shows relatively the same value. This is a response to measuring environmental conditions without any cooking activity. while at 17.14-17.59, there was a significant change in sensor values, namely the temperature increased to 10 0C, humidity decreased to 25% and gas level increased to 15%. These values are sensor responses to cooking activities in the kitchen.

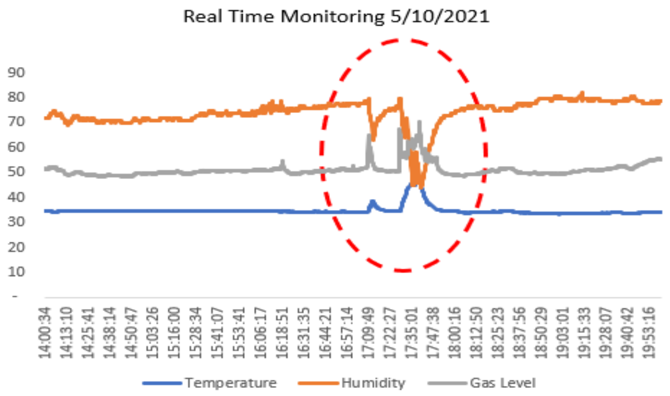


Figure 11. Graph of measurement results of temperature, humidity, and gas level in a simulation in the kitchen

Based on the two simulations above, a categorization of the analysis of the conclusions of environmental conditions is carried out based on the sensor reading values as shown in Figure 12.

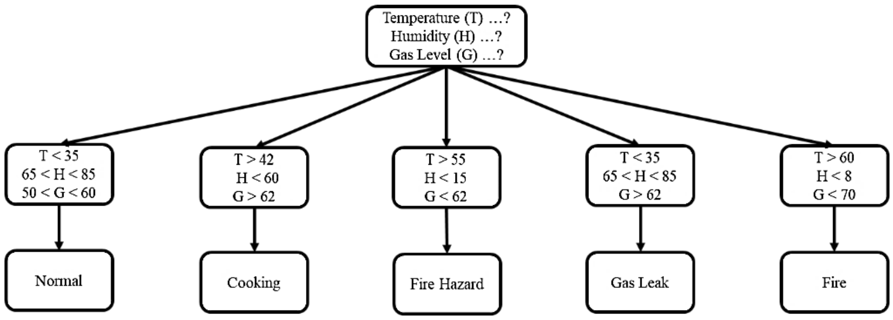


Figure 12. Categorization of sensor reading values

Figure 12 shows five categories of environmental analysis conclusions based on sensor reading values. The normal category is concluded if the readability value of temperature is small from 350˚ Celsius, humidity is between 65 to 85% and gas level is between 50 to 60%. This normal category shows that the sensor reading value only depends on environmental conditions such as the placement of the sensor placement room and environmental values based on areas such as the city of Jakarta, Indonesia. The cooking category is concluded if the temperature readability is greater than 420C, the humidity is small from 60% and the gas level is large from 62. The Cooking category is taken completely so the conclusions are generated from the second simulation. The cooking category can also be interpreted that there is a potential for fire but it is still under control. The fire hazard category is when the temperature is above 550C, the humidity is less than 15% and the gas level is greater than 62%.

The fire hazard category indicates that there is a fire that should be a concern or alert for users. A fire hazard can be defined as the presence of fire activity within the controlled and uncontrolled threshold. The controllable threshold can be exemplified as cooking activity exceeds the normal cooking state. The uncontrollable threshold can be interpreted as the potential for a fire to occur if it is not handled as soon as possible. The Gas Leak category is concluded if the temperature is still below 350C, humidity is between 65 and 85% and the gas level is 62%. The gas leak also includes a fire threat if it is not controlled as soon as possible. Fire can not only arise from cooking activities but also other sources that can produce fire. The state of the gas leak can also have the potential for explosion. The last category and also the most dangerous is fire. Fire is concluded if the temperature reads above 600C, the humidity is small from 8% and the gas level is greater than 70%. Fire is an event that has the potential to damage the environment and its surroundings.

A fire potential information system is something that is further developed based on the categorization of the values of this sensor reading. In this system, each sensor reading value will be automatically sent to ThingSpeak, but if the sensor reading indicates it is in the fire hazard, Gas Leak, and Fire categories, the sensor device will immediately send a telegram notification with the sensor readability values. With the popularity of using Telegram among the public, such as in the city of Jakarta, the response to fire hazards is faster. This is the advantage of the system built.

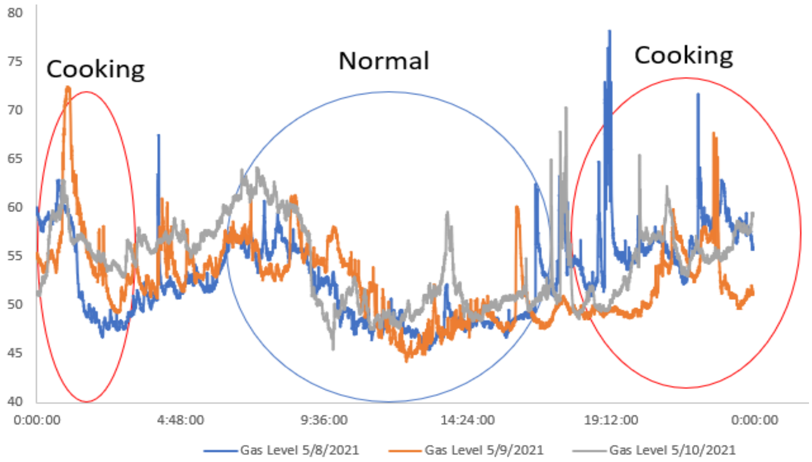


Figure 13. Comparison graph of gas level readings for 3 days

Figure 13 shows a comparison of gas level readings for 3 days, from 8 to 10 May 2021. This comparison can help information managers to find out anomalous activities around the device based on the sensor reading value. In general, two conditions can be concluded in the comparison of these values, namely cooking at 00.00 to 1.00 and 16.00 to 20.00 and normal from 5.00 to 14.30. This indicates that there is a similarity of activity where the sensor is installed.

Table 1 shows the statistical value of gas level readings for 3 days, namely the maximum, minimum, and average values. The lowest value of gas level reading is on May 9, 2021. The highest value of gas level reading is on May 8, 2021. And the highest average gas level reading is on May 10, 2021. This average value can indicate that on May 10, 2021, May 10, 2021, the use of LPG Gas for cooking is greater than other days. However, after three days the reading of this gas level value can still be categorized as safe and under control.

Table 1. Statistics on gas level readings for 3 days

Day	Lowest value	Highest value	Average value
Gas Level 5/8/2021	45.45	78.30	52.91
Gas Level 5/9/2021	44.18	72.53	52.38
Gas Level 5/10/2021	45.45	70.38	54.34

Table 1 shows the statistical value of gas level readings for 3 days, namely the maximum, minimum, and average values. The lowest value of gas level reading is on May 9, 2021. The highest value of gas level reading is on May 8, 2021. And the highest average gas level reading is on May 10, 2021. This average value can indicate that on May 10, 2021, May 10, 2021, the use of LPG Gas for cooking is greater than other days. However, from three days the reading of this gas level value can still be categorized as safe and under control.

The system has been functional for more than 2 months 24 hours a day. The number of incoming data per day is 5760 with data updates every 15 seconds from the last data. Based on the consistency of sensor measurement values and data transmission to Cloud IoT, it can be said that this system has good performance. With a simple design, low power, and also easy to duplicate, this system can be applied for monitoring with a larger coverage using wireless networks such as radio frequency [15]–[17] and Wi-Fi[18]–[21]. In addition, this system can be installed in open spaces.

Real-time data performance analysis

Real-time data performance analysis is important to determine the reliability of the system in measuring the state of visualizing data on the web server. For this reason, real-time data collection of systems that have been installed for 10 consecutive days was carried out. The system as described in the system design section will measure the gas level value every 15 seconds and then send it to the destination web server. The data that has been sent on the web server can be exported for further analysis.

Figure 14 shows a comparison graph of the amount of data sent on the web server for 10 consecutive days. The blue bar is the amount of data every day while the orange bar is the amount of data that should be sent on the web server. In general, it can be seen that the amount of data sent every day is in the range of 5351-5392 total data. While the amount of data that should be sent is 5760 data. For further analysis can be seen in Table 2.

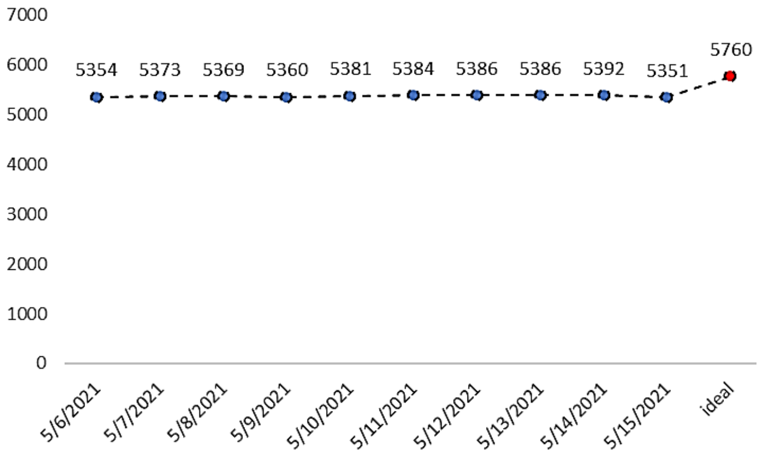


Figure 14. Comparison graph of the amount of data sent by the webserver for 10 consecutive days

Table 2 Statistical analysis of the amount of data sent

Statistics	Data Loss	Data Loss (%)
Max	409	0.071
Min	368	0.064
Average	386.4	0.067

The lowest amount of data sent to the web server was on 5/15/2021 with a total of 5351 data. The amount of data loss is 409 data with the percentage value of data loss is 0.071%. The highest amount of data sent to the web server was on 5/14/2021 with a total of 5392. Total data loss was 368 data with a data loss percentage value of 0.064%. The average amount of data sent to the web server starting from 6 – 14/5/2021 is 5373 data. The amount of data loss is 386.4 data with the percentage value of data loss is 0.067%. From the statistical analysis of the amount of data sent, it can be concluded that the system has a very good performance, with a data loss percentage of less than 1%. This shows that the system is feasible to be applied daily to monitor the level of gas leakage.

CONCLUSION

The development of a low-power fire detection system for densely populated housing based on the Internet of Things (IoT) and Cloud Messenger has been successfully carried out. This system automatically generates gas level information in real-time. This system is also equipped with a cloud-based messenger system as an emergency notification when there is a gas leak and a potential fire situation. This system has also been tested in a simulation to get an analysis of the environmental conditions where the system is applied such as normal conditions, cooking, gas leaks, and potential fires. With the consistency of measuring gas level values along with temperature and humidity as well as sending data to the IoT cloud and notifications using cloud messengers, it is concluded that this system has good performance.

REFERENCES

[1] N. Sudiana, O. Rofara, and Astisiasari, “Urban fire hazard analysis of DKI Jakarta Province,” *J. Sains dan Teknol. Mitigasi Bencana*, vol. 13, no. 2, 2018.

[2] L. Mistry, S. Susarla, R. Kharade, A. Kamble, and M. Mokashi, “Home security: - Water leakage and LPG gas leakage detection,” *Int. Res. J. Eng. Technol.*, vol. 4, no. 5, 2017.

[3] F. Chraim, Y. B. Erol, and K. Pister, “Wireless gas leak detection and localization,” *IEEE Trans. Ind. Informatics*, vol. 12, no. 2, 2016.

[4] X. Huang and L. Du, “Fire detection and recognition optimization based on virtual reality video image,” *IEEE Access*, vol. 8, 2020.

[5] P. Foggia, A. Saggese, and M. Vento, “Real-time fire detection for video-surveillance applications using a combination of experts based on color, shape, and motion,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 9, pp. 1545–1556, 2015.

[6] P. Dvorak, M. Mazanek, and S. Zvanovec, “Fire emissivity detection by a microwave radiometer,” *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 11, 2015.

[7] R. S. Kharisma and A. Setiyansah, “Pembuatan sistem pendeteksi dini kebakaran menggunakan ATmega8,” *Data Manaj. dan Teknol. Inf.*, vol. 18, no. 1, 2017.

[8] S. S. Dewi, D. Satria, E. Yusibani, and D. Sugiyanto, “Prototipe sistem informasi monitoring kebakaran bangunan berbasis Google Maps dan modul GSM,” *J. JTIK (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 1, no. 1, 2017.

[9] EINSTRONIC, “Nodemcu,” *Einstronic*, 2017.

[10] ThingSpeak, “IoT Analytics - ThingSpeak Internet of Things,” *ThingSpeak*. 2020.

[11] Y. Zakaria and K. Michael, “An integrated cloud-based wireless sensor network for monitoring industrial wastewater discharged into water sources,” *Wirel. Sens. Netw.*, vol. 09, no. 08, pp. 290–301, 2017.

[12] L. Li, H. Xiaoguang, C. Ke, and H. Ketai, “The applications of WiFi-based wireless sensor network in internet of things and smart grid,” in *2011 6th IEEE Conference on Industrial Electronics and Applications*, 2011

[13] Z. Yang, Q. Zhou, L. Lei, K. Zheng, and W. Xiang, “An IoT-cloud based wearable ECG monitoring system for smart healthcare,” *J. Med. Syst.*, vol. 40, no. 12, p. 286, 2016.

[14] S. C. Olisa, C. N. Asiegbu, J. E. Olisa, B. O. Ekengwu, A. A. Shittu, and M. C. Eze, “Smart two-tank water quality and level detection system via IoT,” *Heliyon*, vol. 7, no. 8, 2021.

[15] Z. Kaleem, T. M. Yoon, and C. Lee, “Energy efficient outdoor light monitoring and control architecture using embedded system,” *IEEE Embed. Syst. Lett.*, vol. 8, no. 1, pp. 18–21, 2016.

[16] R. Muhendra, Husein, M. Budiman, and Khairurrijal, “Development of digital water meter infrastructure using wireless sensor networks,” *AIP Conf. Proc.*, vol. 1746, pp. 1–7, 2016.

[17] R. Muhendra, “Jaringan sensor nirkabel: studi dan evaluasi kinerja lora transmitter dan long range radio frekuensi (RF) pada luar ruang,” *Jurnal Jaring SainTek*, vol. 3, no. 1, pp. 6–12, 2021.

[18] S. Enayati and H. Saeedi, “Deployment of hybrid FSO/RF links in backhaul of relay-based rural area cellular networks: Advantages and performance analysis,” *IEEE Commun. Lett.*, vol. 20, no. 9, pp. 1824–1827, 2016.

[19] K. Shailandra, “An overview of Technical aspect for WiFi Networks Technology,” *Int. J. Electron. Comput. Sci. Eng.*, 2012.

[20] P. Lech and P. Włodarski, “Analysis of the IoT WiFi mesh network,” in *Advances in Intelligent Systems and Computing*, Cham: Springer International Publishing, 2017, pp. 272–280.

[21] R. Muhendra, A. Rinaldi, M. Budiman, and Khairurrijal, “Development of WiFi mesh infrastructure for internet of things applications,” *Procedia Eng.*, vol. 170, pp. 332–337, 2017.