400-2059-1-ED (1) (1).doc

DESIGN AND IMPLEMENTATION OF IOT BASED WATER PIPE PRESSURE MONITORING INSTRUMENT

Hollanda Arief Kusuma[a], Rady Purbakawaca[b], Irwan Rudy Pamungkas[c], Luthfy Nizarul Fikry[c], Sonny Seftian Maulizar[c]

[a]Departement of Electrical Engineering
Maritime University of Raja Ali Haji
Jl. Polteknik Senggarang
Tanjungpinang, Indonesia
[b]Agricultural Engineering Sciences
IPB University
Gedung Fateta, Kampus IPB Darmaga PO BOX 220, Bogor 16002
Bogor, Indonesia
[c]CV Mantis Indonesia
Jl. Mola No.15, Komplek Cimanggu Perikanan, Kel. Kedung Waringin, Kec. Tanah Sareal
Bogor, Indonesia

Corresponding email : hollandakusuma@umrah.ac.id

Abstract

The water pressure monitoring system in the PDAM pipeline network has been successfully developed for operation and maintenance of water leaks in real-time. This research aims to design a water pressure monitoring system in operational piping networks to identify anomalies as early as possible. The system is built using a microcontroller, 1.2 MPa fluid pressure sensor and a control system equipped with a GSM wireless communication module, Analog to Digital Converter module with 16 Bit resolution, real time clock peripheral, OLED display 128x64, and micro SD card. The developed system was tested in a pressure range of 0.200 - 0.800 Bar with 30 repetitions with a RMSE of 0.058 Bar. This system has a deterministic coefficient of 0.885 against a standard manometer. The system implemented in the field successfully sends data to the server with a success rate of 96.0%. Data is displayed on a monitoring dashboard that can be accessed via a computer or smartphone.

Keywords: water pressure, pressure monitoring, PDAM, water leakage, IoT.

I. INTRODUCTION

Internet of Things (IoT) represents the ability of network devices to sense, collect and sometimes even analyse in loco data from the world around us, and then share that data across the Internet. In other hands, we can say that (IoT) connects cyber, physical and biological worlds via smart sensors and devices immersed in the surrounding environment. The ability to connect to the Internet and between them is a fundamental requirement of IoT devices [1]. Right now, IoT has emerged as a new paradigm aimed at providing solutions for integration, communication, data consumption, and analysis of smart devices. To this end, connectivity, interoperability, and integration are inevitable parts of IoT communication systems. Developing an IoT device is depending on network communication to the server. Several networks used in IoT devices such as Zigbee, Bluetooth Low Energy, LoRa, mobile networks (GPRS, 3G, 4G, CDMA, etc.), WLANs, WSN, and Mobile Adhoc Networks (MANET) ([2], [3]).

Arduino is known as an open-source platform which developing both software and hardware [4]. Arduino provides an integrated development Arduino provides an integrated development environment to write, edit code, and convert the code into instructions that Arduino hardware understands. Arduino IDE based on the Processing language [5]. Arduino board is where the code executed. Generally, Arduino board using Atmel's low-power CMOS 8-bit microcontrollers based on AVR enhanced RISC architecture as the main microcontroller [6].

Arduino has a large online community that providing libraries and stimulates the engagement in developing open-source hardware. Custom Arduino libraries, supporting sensors, and active support from manufacturers for the platform make Arduino familiar and successfully implemented in data collection [7]. The Arduino platform has been used successfully in several other similar data collection efforts including Urban Traffic of Vehicle via GSM [8], Smart Energy Meter ([9], [10]), flood alert system for parking lots [11], monitoring temperature, humidity, and soil moisture for irrigation through GSM [12], and Geohazards Monitoring and Early Warning System [13]. These studies and others shows the ease of use and reliability of arduino platform for open source hardware development.

PDAM Tirta Pakuan Bogor is a local drinking water supply company in Bogor City. Suddenly the fact water supplies are lost due to pipe leakage [14]. This problem cannot be detected real-time. PDAM usually know the problem source from user complains or regular inspection. Usually, PDAM used analogue manometer to monitor the water pressure [15]. After that, the PDAM technician inspects on the field to see the real condition. This approach is not an efficient way to solve the problem. Therefore, a real-time water pressure monitoring instrument need to be build so PDAM can take action as soon as possible when something wrong happens.

In response, we developed the IoT Water Pressure Monitoring System to monitor water pressure in PDAM pipe network. The goal of this system was to build a platfrom for sensor and data logger development that had the following features:

- Designed and constructed using open source hardware and software;
- Long-term onboard data storage on SD cards at user-defined intervals;
- Capable to measure water pressure with acceptable accuracy (±10%).
- Have a real time clock;
- Able to send the data to server periodically.

II. Method

The main objective of this work is to obtain accurate water pressure data from a water pressure sensor. The proposed system divided into three layers: acquisition, transport, and application layer. Acquisition layer consists of a water pressure sensor, ADS1115, Arduino Mega2560, Real-Time Clock (RTC) DS3231, OLED display 128x64 and micro SD card. Data received from the sensor sent to an application layer through the transport layer using GSM module SIM900A.

A. Acquisition Layer

This prototype built based on Atmel ATmega2560 microprocessor. ATmega2560 have adequate internal memory (flash memory 256 KB, SRAM 8 KB, EEPROM 4 KB) to run the program. ATmega2560 have four USART. This instrument just used two serial for serial debugging and GSM communication. Arduino Mega2560 clone named RobotDyn Arduino Mega2560 Pro Mini choosen as the microprocessor board because this board have compact size (38x52 mm) compared to Arduino Mega2560 (53.34x101.6 mm).

Maxim DS3231 used to provide time measurement. RTC used to have consistent timekeeping that doesn't reset when the Arduino power supply dies or reprogrammed. DS3231 use I2C communication to send time data to Arduino Mega2560 [16].

ADC 16 bit ADS1115 used to convert analog value (voltage) from water pressure sensor to digital value. ADS1115 can make high-resolution analogue to digital conversion. This conversion makes the measurement value is more precise. Single ended conversion utilized to get the sensors data because we want to measure the positive voltage. By using this conversion, ADS1115 only get 15 bit resolution.

Water Pressure Sensor G1/4 1.2 MPa (Figure 1) utilized to obtain the pressure value. This sensor working input voltage at 5V, range output voltage 0,5 V – 4,5 V, pressure range between 0 – 1.2 MPa with a maximum pressure of 2.4 MPa. This sensor connect to ADS1115 analog input channel 0. To have pressure value, we used formulation at (1) and (2). Analog value from pressure measurement converted to digital Number (DN). DN multiply with 0.1875 because 1 bit equal to 0.1875 mV and substitute with 1000 to change the unit from mV to V. Equation (2) use a linear regression from minimum value (0 Pa, 0.5 V) and maximum value (1.2 MPa, 4.5 V) with offset define by user. The components diagram and connection of the device shown in Figure 2.

$$V = \frac{DN \times 0.1875}{1000} \quad (1)$$

$$P = (300 \times V - 150) \times 0.01 \quad (2)$$

Where:
V = Voltage (Volt)
P = Pressure (bar)

Figure 1. Water pressure sensor 1.2MPa

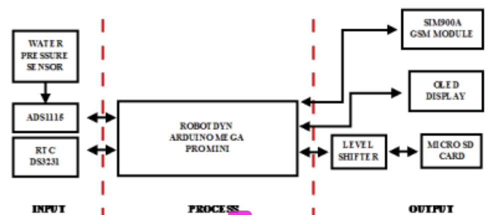

Figure 2. Components diagram of acquisition layer

B. Transport Layer

Transport layer in this work consist of GSM module SIM900A produced by SIMcom. This module (Figure 3) can run with power 5 VDC supply. This module connects with Arduino Mega2560 via serial communication (Serial1) with baudrate 9600 bps. This module used for sending water pressure sensors data to server using AT Commands [17].

Figure 3. SIM900A GSM module

C. Application Layer

Web User Interface (UI) used to display water pressure sensor data sent from the device. This web UI developed to be responsive so can be accessed from smartphone and computer. This web also provides login to secure the user that can access the data. This web UI built using HTML5, CSS3, javascript, MySQL, bootstrap, Highcharts and PHP. In this work, the RESTful application interface with POST method used to send water pressure sensor data to the cloud service. Web UI working principle shown in Figure 4.

Figure 4. Working principle of proposed web user interface

D. Data Analysis

Accuracy is one of the most critical factors to consider the sensor used to detect water pressure. The accuracy of the sensor described how close a measurement to the actual value of water pressure . In this research, manometer analogue used as reference.

Every singular measurement in n samples contain a observation errors (e_i) that influence its accuracy. To assess the accuracy of measurements, the relative error of measurement (ε_r) is used based on ratio of observation error and actual value shown in (3) [18]. Root mean square error (RMSE) in (4) used to obtain total error based on [19].

$$Relative\ error\ (\varepsilon_r) = \frac{Absolute\ error}{True\ value} = \frac{\delta A}{A} \quad (3)$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} e_i^2}. \quad (4)$$

Comparison between the water pressure sensor and manometer done by using linear curve fitting. With this fitting, we get for determining how fit the regression model by using coefficient of determination(r^2). Coefficient of determination can be interpreted as the percent reduction in the total variation in the experiment obtained by using the regression line ([20], [21]).

III. Result and Discussion

A. System Programming Design

This instrument developed for measuring water pressure from water pressure sensor, calculate data, produce water pressure, display it on OLED Display, store in the micro SD card, and send the data to the server. This instrument capable to measure based on interval that saved in micro SD card. This instrument firmware built using Arduino IDE. Figure 7 shows the flowcharts of this system.

Arduino Mega2560, ADS1115, RTC, and OLED Display communicated through the I2C protocol using library Wire.h. Micro SD card communicates with Arduino Mega2560 using SPI protocol. Data received from the water pressure sensor converted using ADS1115. Arduino Mega2560 UART1 connected with SIM900A. In this firmware, we add external library such as RTClib for the RTC, SDFAT for accesing the micro SD card, Adafruit_ADS1015 for communicating with ADS1115, Adafruit_SSD1306 and Adafruit _GFX for OLED display, TimeLib, and TimeAlarm to set sampling time setting.

The program begins with serial communication and OLED Display initialization using I2C address 0x3c. The microcontroller instructs OLED to display the text "USAID- PDAM OPEN SOURCE HARDWARE WATER PRESSURE SENSOR DEVICE 2019". This text became an indicator to make sure that the device is on and work properly. Then microcontroller checks if RTC DS3231 is connected or not using *RTC.begin()*. If microcontroller detects that RTC not connected then the next command will not execute, then the OLED will display text " Error! Contact CS". If RTC initialization complete, then microcontroller get time and date from RTC. These data will be displayed in OLED three seconds. After that, micro SD card initialization begins. If pin Select detect micro SD card absence, then OLED will display text " SD card Error!!!". If micro SD detected, the text " SD Card OK!" will be displayed in OLED.

Next step is checking **config.txt** inside the micro SD card. This text file contains ID Station, data sampling interval, and bursting data configured by user. Sampling interval specified in minutes and burst data is in seconds. If **config.txt** doesn't exist, the microcontroller will instructs OLED to display text "Configuration file Error Reading". After getting data from **config.txt**, ADS1115 initialization performed followed by SIM900A initialization. Microcontroller instructs SIM900A to check operator name using "AT+COPS?". SIM900A will returns the current mode and the currently selected operator. "AT+CSQ" sent to SIM900A to check the signal quality inside the SIM Card. This command returns received signal strength indication from 0 until 30. The last setup is setting sampling time based on interval data. This sampling setting done using function *Alarm.repeat* provided by TimeAlarms library.

The main program begins with taking the time from RTC. After that, the microcontroller will get pressure data in Digital Number (DN) from ADS1115 in 2^{16} resolution using **for** looping based on burst value defined by the user. Figure 5 shows the pseudocode how to get the data. The total data after looping divided with burst interval to get average DN value. The next step is conversion data from digital number to voltage using equation (1) and convert the voltage into pressure in Bar by equation (2). The conversion DN to pressure pseudocode shown in Figure 6. After that, water pressure data displayed in OLED.

Algorithm *Get Mean DN from water pressure sensor*

```
Initialize count = 0, totalDN = 0;
for (count=0; count<burst; count++){
DN = ADS1115.readAnalogChannel(0);
totalDN += DN;
}
meanDN = totalDN/burst;
```

Figure 5. Pseudocode to get average DN from water pressure sensor using ADS1115

Algorithm *Convert DN pressure (bar)*

```
Volt = meanDN * 0.1875 /1000;
Pressure = (300 * Volt - 150) * 0.01
```

Figure 6. Pseudocode to convert DN to bar

The next command saves time, date, and water pressure data into Micro SD card as plain text with a comma (‘,’) as the delimiter in a *.txt file format. yyyymmdd.txt is the name of the file inside Micro SD Card. This file created automatically by getting the date from RTC. After saving data into micro SD card completed, then water pressure data send to the server. This data sent using JSON format. Microcontroller instruct SIM900A to send data. All AT commands used to execute the POST method and send the data shown in Table 1. This commands must execute sequencely. Microcontroller will wait SIM900A give response “OK” in timeout period. The microcontroller will wait until next time interval to get water pressure data again. While waiting, microcontroller gets date and time data from RTC and display it on OLED. The firmware in this system works based on flowchart in Figure 7.

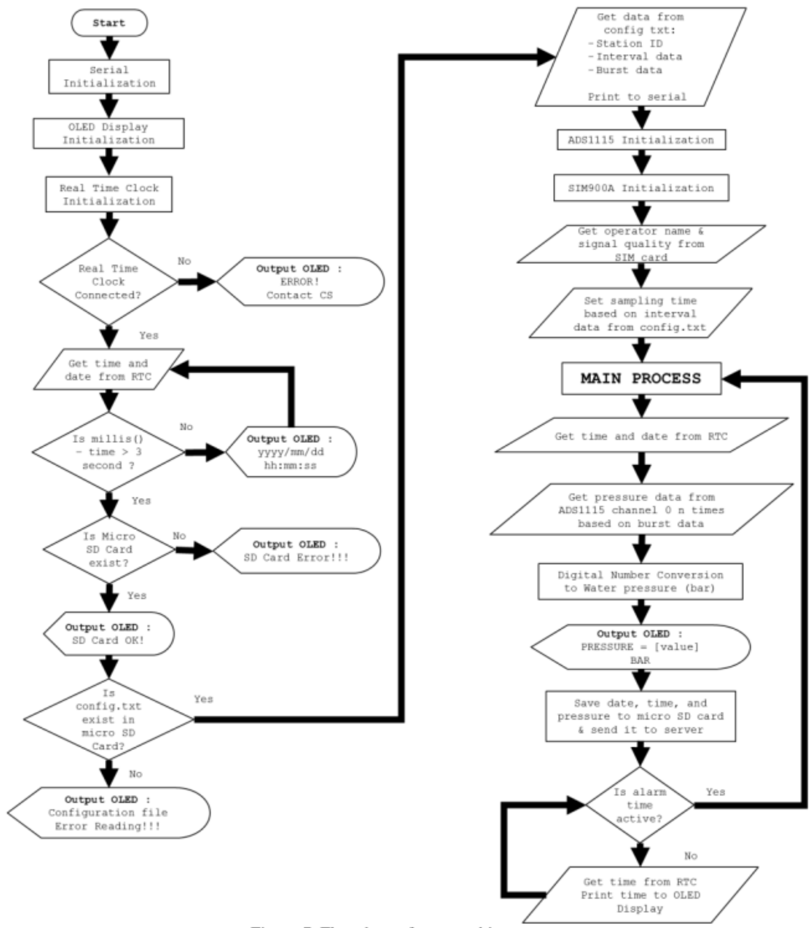

Figure 7. Flowchart of proposed instrument

TABLE 1
LIST AT COMMAND USED TO SEND DATA TO SERVER

AT COMMAND	DESCRIPTION
AT+CGATT=1	Attach GPRS service
AT+CIPSHUT	Deactivate GPRS PDP context
AT+SAPBR=0,1	Close bearer
AT+SAPBR=3,1,"CONTYPE","GPRS"	Set bearer connection type to GPRS
AT+SAPBR=3,1,"APN","Internet”	Set bearer APN to Internet
AT+SAPBR=1,1	Open bearer
AT+SAPBR=2,1	Query bearer to get IP address
AT+HTTPTERM	Terminate HTTP Service
AT+HTTPINIT	Initialize HTTP service
AT+HTTPPARA="CID",1	Set HTTP Bearer profile identifier value into 1
AT+HTTPPARA="URL",”http://www.mantisid.id/api/product/pdam_dt_c.php”	Set HTTP client URL value
{‘Data’:’yyyy-mm-dd hh:mm:ss’,’water pressure’,’burst interval’,’data interval’}	Data in JSON format
AT+HTTPDATA = json.length(), 15000	Input HTTP data with maximum time input data 15000 milliseconds
AT+HTTPACTION=1	HTTP method action set to POST method

B. Water Pressure Sensor Accuracy

Sensor accuracy tested by comparing this prototype with manometer analogue. From 30 repetition of observation, we can see that RMSE was 0.058 bar. Relative error of this sensor was 0.09 or 9%. This value was acceptable. Therefore the accuracy is good. We also find the comparison and linear regression equation based on the water pressure sensor and manometer shown in Figure 8. From this scatter, we can the get linear regression equation. This equation implemented in the code to correct water pressure value. R square (R^2) or the coefficient of determination value from this test was 0.8846. The strength of the relationship between water pressure sensor data and manometer data is 88.46%. It means that about 88.46% of the variation in one of the variables is explained by the other.

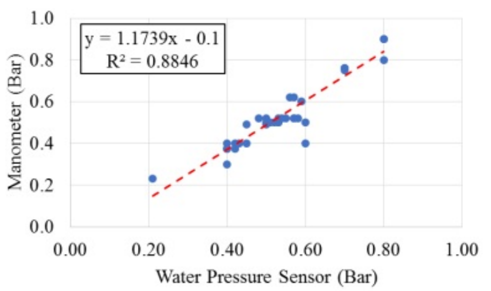

Figure 8. Scatter plot of the Water pressure versus manometer with fitted line and linear regression equation

C. Prototype Deployment and Experimental Result

This prototype developed based on the wiring diagram in Figure 2. All component integrated on fabricated PCB and placed inside a waterproof case (Figure 9). Water pressure sensor connects to the device using CB connector 5 pins. DC adaptor used as main power supply. A fuse used to protects the wiring. Configuration file (config.txt) saved in micro SD card shown in Figure 10.

Figure 9. All components inside waterproof case

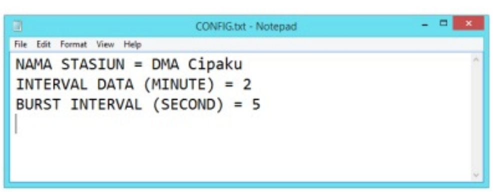

Figure 10. Configuration file

This prototype successfully deployed at Distric Metering Area (DMA) Cipaku PDAM Tirta Pakuan, Bogor, Indonesia. The water pressure sensor installed in a water pipeline (Figure 11) near the primary water tank and connected to an analog manometer as comparison (Figure 12).

Figure 11. Water pipe installation to be measured by the prototype

Figure 12. Sensor installation near analog manometer as comparison

PDAM technician can view the data from http://igauge-logger.mantisid.id. The technician must insert username and password to secure the connection (Figure 13). This prototype send data to server every 5 minutes. this prototype reading and sending data from 10th April 2019 until 31th October 2019. Water pressure sensor displayed in time series using high chart. User can choose the time range as shown in Figure 14. User also can download pressure data from the server using Data Library menu or using high chart menu.

Figure 13. Login form in water pressure dashboard website

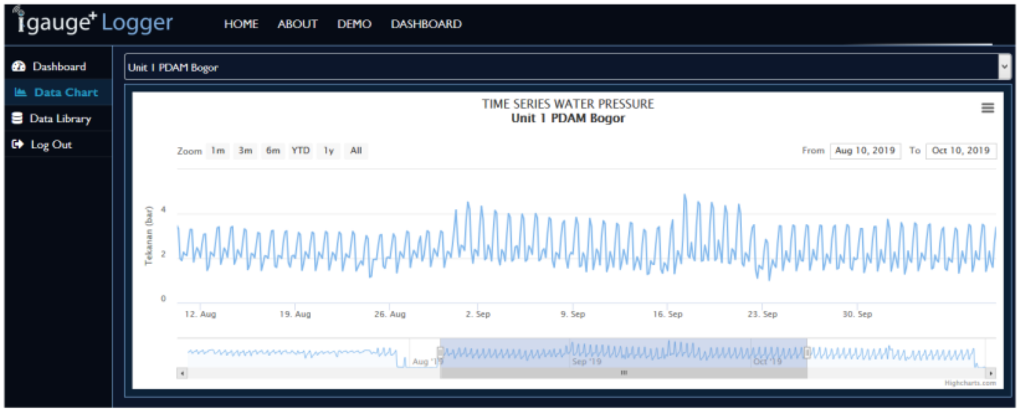

Figure 14. Water pressure data display on real-time dashboard

Based on time stamp from reading sensor and server time stamp, we get average time sending the data around 30 seconds. The amount of sending time is acceptable by PDAM user. We also see that just 96% data were sent to server. We assume that this blank data happen because the operator signal quality not enough high to able to sent the data to the server.

Another interesting feature to be discussed from this experiment was the water supply pattern. From Figure 15 we can see the pattern and fluctuation very clearly. Water pressure drop dramatically in time period 5:29 until 8:04 and17:49 until 18:49. This drop occure because in this period lot of household use the water. It is very different when there no activity, the water pressure going up until reach 3 bar. This pattern can be found from 21:00 until midnight. Based on the data, PDAM team must ensure the water supply debit in the busy time period enough until reach to the user.

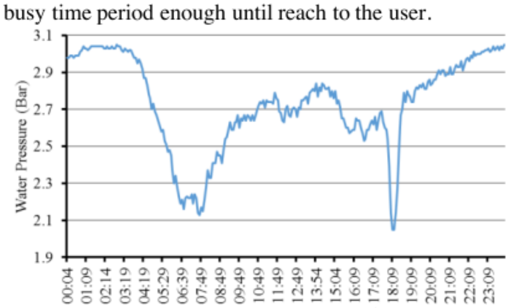

Figure 15. Water pressure data pattern in 11th April 2019

Conclusion

This paper presents the implementation of Open Source Hardware for a real-time water pressure monitoring. This system has been successfully build and tested. Both accuracy and performance tests showed an acceptable result. With the data interval set to 5 minutes, we can get nearly 100% data. Therefore, this prototype can be used widely in PDAM and to help them monitoring pressure in water pipe line. To make it more usable, future work may be required including to develop off grid system to power this prototype so can be installed in remote area where no PLN electricity available.

Acknowledgment

This research funded by USAID – IUWASH PLUS Refinent of Prototype (realtime Water Pressure Sensor) for 3 PDAMs (Kota Bekasi, Kota Bogor and Kota Depok) 2019.

References

[1] B. Di Martino, K.-C. Li, L. T. Yang, and A. Esposito, "Trends and Strategic Researches in Internet of Everything," in *Internet of Everything Algorithms, Methodologies, Technologies and Perspectives*, B. Di Martino, K.-C. Li, L. T. Yang, and A. Esposito, Eds. Singapore: Springe, Singapore, 2018, pp. 1–12.

[2] F. Khodadadi, A. V. Dastjerdi, and R. Buyya, "Internet of Things: An Overview," in *Internet of Things*, Elsevier, 2016, pp. 3–27.

[3] D. Serpanos and M. Wolf, "IoT System Architectures," in *Internet-of-Things (IoT) Systems*, Cham: Springer International Publishing, 2018, pp. 7–15.

[4] M. Margolis, *Arduino Cookbook*, 2nd ed. O'Reilly Media, 2012.

[5] D. Uckelmann, M. Harrison, and F. Michahelles, "An Architectural Approach Towards the Future Internet of Things," in *Architecting the Internet of Things*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 1–24.

[6] M. Banzi and M. Shiloh, *Make: Getting started with Arduino*. 2014.

[7] A. S. Ali, Z. Zanzinger, D. Debose, and B. Stephens, "Open Source Building Science Sensors (OSBSS): A low-cost Arduino-based platform for long-term indoor environmental data collection," *Build. Environ.*, vol. 100, pp. 114–126, 2016.

[8] H. Nugra *et al.*, "A Low-Cost IoT Application for the Urban Traffic of Vehicles, Based on Wireless Sensors Using GSM Technology," in *2016 IEEE/ACM 20th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, 2016, pp. 161–169.

[9] F. Ramdani, S. Rahman, and P. Setiani, "Inexpensive Method to Assess Mangroves Forest through the Use of Open Source Software and Data Available Freely in Public Domain," *J. Geogr. Info Syst.*, vol. 7, no. February, pp. 43–57, 2015.

[10] K. C. Okafor, G. C. Ononiwu, U. Precious, and A. C. Godis, "Development of Arduino Based IoT Metering System for On - Demand Energy Monitoring," vol. 7, no. 23, pp. 3208–3224, 2017.

[11] H. Báez, I. Vergara-Laurens, L. Torres-Molina, L. G. Jaimes, and M. A. Labrador, "A real-time flood alert system for parking lots," *2017 IEEE 7th Annu. Comput. Commun. Work. Conf. CCWC 2017*, pp. 1–5, 2017.

[12] M. N. Rajkumar, S. Abinaya, and V. V. Kumar, "Intelligent irrigation system - An IOT based approach," *IEEE Int. Conf. Innov. Green Energy Healthc. Technol. - 2017, IGEHT 2017*, pp. 1–5, 2017.

[13] W. Honghui *et al.*, "Research of the Hardware Architecture of the Geohazards Monitoring and Early Warning System Based on the Iot," *Procedia Comput. Sci.*, vol. 107, no. ICICT 2017, pp. 111–116, 2017.

[14] PDAM Tirta Pakuan Kota Bogor, "Informasi Gangguan," 2020. [Online]. Available: http://www.pdamkotabogor.go.id/thethme/index.php?m=12&hal=1. [Accessed: 11-Nov-2020].

[15] Kementerian PUPR, *Pedoman Pelaksanaan Efisiensi Energi di PDAM*. 2014.

[16] Maxim Integrated, "DS 3231 RTC General Description," 2015.

[17] Shanghai SIMCom Wireless Solutions, "SIM900 AT Commands Manual V1.11," 2015.

[18] R. S. Sedha, *Electronic Measurements and Instrumentation*. S CHAND & Company Limited, 2013.

[19] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)? - Arguments against avoiding RMSE in the literature," *Geosci. Model Dev.*, vol. 7, no. 3, pp. 1247–1250, 2014.

[20] A. Gilat and V. Subramaniam, *Numerical Methods for Engineers and Scientists, 3rd Edition*. John Wiley & Sons, Inc, 2013.

[21] W. Mendenhall, R. J. Beaver, and B. M. Beaver, *Introduction to Probability and Statistics*, 14th Editi. Boston: Brooks/Cole, 2013.

400-2059-1-ED (1) (1).doc

ORIGINALITY REPORT

17%

SIMILARITY INDEX

PRIMARY SOURCES

1	H A Kusuma, R Anjasmara, T Suhendra, H Yunianto, S Nugraha. "An IoT Based Coastal Weather and Air Quality Monitoring Using GSM Technology", Journal of Physics: Conference Series, 2020 Crossref	202 words — 6%
2	Muhammad Nasar, Novendra Setyawan, Amrul Faruq, Indah Sulistiyowati. "A Simple Real-Time Energy Analytics Model for Smart Building Using Open IoT Platforms", Jurnal Elektronika dan Telekomunikasi, 2019 Crossref	66 words — 2%
3	link.springer.com Internet	62 words — 2%
4	dokumen.pub Internet	49 words — 2%
5	predictabledesigns.com Internet	42 words — 1%
6	silo.pub Internet	39 words — 1%
7	Akram Syed Ali, Zachary Zanzinger, Deion Debose, Brent Stephens. "Open Source Building Science Sensors (OSBSS): A low-cost Arduino-based platform for long-term indoor environmental data collection", Building and Environment, 2016 Crossref	39 words — 1%
8	Homar Baez, Idalides Vergara-Laurens, Luz Torres-Molina, Luis G. Jaimes, Miguel A. Labrador. "A real-	11 words — < 1%

time flood alert system for parking lots", 2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC), 2017
Crossref

9 en.nokisens.com
Internet
9 words — < 1%

10 docplayer.org
Internet
9 words — < 1%

EXCLUDE QUOTES	ON	EXCLUDE MATCHES	OFF
EXCLUDE BIBLIOGRAPHY	ON		