DEVELOPMENT OF ANDROID BASED AIR QUALITY MONITORING AND FORECASTING DEVICE PROTOTYPE AT PT. DIRGANTARA INDONESIA

Kurniansyah Pratama[1], Eko Budi Setiawan[2]

[1,2] Informatics Engineering, Indonesian Computer University
Jl. Dipatiukur 112 Bandung
E-mail : kurniansyah.pratama@gmail.com[1], eko@email.unikom.ac.id[2]

ABSTRACT

Poor air quality can threaten the lives of all living things from humans to plants. WHO says that 92% of the world's human population lives in areas with poor air quality that has exceeded the safe limits. Carbon dioxide is one of the most dangerous gases that human beings have to pay attention to. This gas can cause serious problems to human health. Industries contribute to gas concentration in the air. PT. Dirgantara Indonesia is one of the industries in Indonesia. Painting and some areas that use chemicals can be harmful to air quality in the work area. The purpose of this system is to build a tool to measure air quality around the painting area. This system can inform employees about air quality in their work area using Android smartphones. Forecasting features can be used to prevent harmful air quality in their work areas.

Kata kunci : Air Quality, Carbon dioxide, Android, Forecasting, Industry

1. INTRODUCTION

PT. Dirgantara Indonesia is one of Asia's leading aviation companies. PT. Dirgantara Indonesia has major competencies in aircraft development and design, aircraft structure, assembly, and maintenance for small and medium-sized civil and military aircraft.

Air is important for all living things. Poor air quality can threaten life from all living things starting from humans, animals to plants. The air that at any time we breath contains emissionsfrom various sources such as emissions from industry, motor vehicles, and cigarette smoke. The study results show that air pollution is harmful to human health, especially for elderly and children. As reported in the WHO sitah that 92% of the world's population live in areas that have air quality that exceeds the safe limits set by the WHO organization. [1] [2]

The number of parts painting process done in the painting department resulted in the destruction of air quality around the painting area due to the chemical content used in the work. So this situation is considered as a serious threat to the health of employees working in the department.

With the advancement of technology such as microcontroller and the number of smartphone users [3], PT. Dirgantara Indonesia (Persero) can use the technology to monitor the air quality in its work area. So that the facilities department or employees can anticipate and improve the existing air conditioning system to improve air quality in the area. While the operating system for the smartphone used is Android.

NodeMCU is an IOT device that is open source, eLua based project and built on ESP8266SDK 1.4. This tool uses many open source projects, such as lua-cjson and spiffs. In it there is a firmware that works to run ESP8266 Wi-Fi SoC, and hardware-based ESP-12 module. [4] Using the MQ-135 sensor and NodeMCU devices, a device can be developed to monitor air quality in the painting area. However, to better not only monitor the circumstances that have occurred and are just happening. Therefore it is necessary a method that can predict the next situation. From the research reference has been found that forecasting through the pattern of data on the air quality index can be done using Exponential Smoothing method.

Therefore, the design of air pollution monitoring device using forecasting feature with Exponential Smoothing method can be applied as an indicator of air quality level that exist around the painting area in PT. Dirgantara Indonesia. Thus the system is expected to provide information to employees of PT. Dirgantara Indonesia about the air quality level in the work area.

.

2. METHODOLOGY AND SYSTEM DESIGN

2.1 Metode Pemulusan Eksponensial Musiman

Exponential Smoothing forecasting method is a forecasting method that works by using constants to give weight to the current and previous values to determine a new forecasting. This constant value influences the forecasting response. Therefore, the value of these constants will affect forecasting errors. There have been many efforts focused on finding suitable constants for use. One approach is to use smoothing constants that can minimize forecast errors. [6]

As with the linear exponential smoothing equations that can be used to predict serial data with

trend patterns, other forms of equations can be used when the serial data pattern is seasonally shaped. One of the specific forecast methods for seasonally-patterned data is the Winter and Winter Exponential Smoothing Method from Winter. This method is based on three equations, each for the stationary, trend and seasonal elements, which are formulated as follows:

$$S_t = \alpha\left(\frac{X_t}{I_{t-L}}\right) + (1-\alpha)(S_{t-1} + T_{t-1}) \quad (1)$$
$$T_t = \beta(S_t - S_{t-1}) + (1-\beta)T_{t-1} \quad (2)$$
$$I_t = \gamma\left(\frac{X_t}{S_t}\right) + (1-\gamma)I_{t-L} \quad (3)$$
$$F_{t+m} = (S_t + T_t.m)I_{t-L+m} \quad (4)$$

Figure 1. Seasonal Exponential Smoothing Formulas

Where :
L = The number of periods in one season cycle
I = Season adjustment factor (seasonal index)

Table 1. Provides an example of forecasting calculations from a data series that has a seasonal element. In this example, the seasonal length (L) is four quarters or one year, while the parameter values α = 0.6, β = 0.2 and γ = 0.5.

As in the exponential method discussed earlier, the first St, Tt and It values are initial values (assumed). The initial value of St is equal to the actual value or the average of several values in the same season. In the example Table 1., The initial value for S4 is equal to its actual value (X4) of 441.

The initial value of T can be searched using the following formula:

$$T_L = \frac{1}{L}\left\{\frac{(X_{L+1} - X_1)}{L} + \frac{(X_{L+2} - X_2)}{L} + \cdots + \frac{(X_{L+L} - X_l)}{L}\right\} \quad (6)$$

Figure 2. Initial Value T Formula

Here's an example of forecasting using Winter and seasonal exponential smoothing methods from Winter.

Table 1. Forecast with the Linear and Seasonal Exponential Smoothing Method from Winter

Year	Quarter	Period	Observed X	Forecast Value			
				S (α = 0,6)	T (β = 0,2)	I (γ = 0,5)	F
2001	1	1	460			*0,96*	
	2	2	484			*1,01*	
	3	3	530			*1,11*	
	4	4	441	*441,0*	*10,25*	*0,92*	
2002	1	5	492	487,7	17,55	0,98	
	2	6	509	504,2	17,33	1,01	510,8
	3	7	588	527,3	18,48	1,11	577,4
	4	8	490	537,5	16,82	0,92	502,7
2003	1	9	533	546,5	15,25	0,98	545,9
	2	10	560	557,3	14,37	1,01	567,5
	3	11	532	569,9	14,03	1,11	635,1
	4	12	560	600,2	17,28	0,92	535,2
2004	1	13	604	616,8	17,13	0,98	605,2
	2	14	675	655,5	21,45	1,02	638,7
	3	15	701	649,7	16,00	1,09	751,4
	4	16	607	660,2	14,89	0,92	615,6
2005	1	17	708	703,6	20,61	0,99	661,3
	2	18	787	753,3	26,41	1,03	737,7
	3	19	850	777,8	26,05	1,09	853,3
	4	20	782	830,4	31,35	0,93	741,3
2006	1	21	-				855,7
	2	22	-				921,4
	3	23	-				1011,0
	4	24	-				890,7

Thus, the T4 can be calculated as follow:

$$T_4 = \frac{1}{4}\left\{\frac{492-460}{4} + \frac{509-484}{4} + \frac{588-530}{4} + \frac{490-441}{4}\right\} = 10{,}25$$

The initial value calculation of I_t on one first season cycle (the first four periods, 2001) is done by dividing each observation data (X) with the average observation data in that cycle. Average demand for 2001 :

X_{2001} = (460 + 484 + 530 + 441) / 4 = 478,75

So the initial value of the seasonal index

I_1 = 460 / 478,75 = 0,96
I_2 = 484 / 478,75 = 1,01
I_3 = 530 / 478,75 = 1,11
I_4 = 441 / 478,75 = 0,92

After the initial values S, T and I are obtained, we can calculate St, Tt and It and Ft + m forecasts can be searched. Forecast values are calculated based on the latest data. In the example The forecast value until the 12th period is obtained based on the data of one previous period (m = 1). Forecasts for subsequent periods are obtained using the 20th period data which is the last period that has actual data.

One of the problems that arise in the use of Winter models For forecasts are the penentual values of α, β, γ. The usual approach is to experiment until values of parameters that minimize forecasting errors (MAD or MSE).

2.2 System Design Architecture

The system to be built has several stages starting from the input obtained by using air quality sensor MQ135 to read air quality in the painting area of PT. Dirgantara Indonesia (Persero). Sensory reading data is sent using NodeMCU microcontroller. This process works by converting read data from the sensor in analog form which is then converted into input in digital form by ADC pin found on pin A0 on NodeMCU. Then, the data is forwarded through the WiFi router to be forwarded to the database server by using the internet connection. The data will be processed in the database server to then be accessed by system users, as well as to provide forecasting features on the system. For more details can be seen on Figure 3.

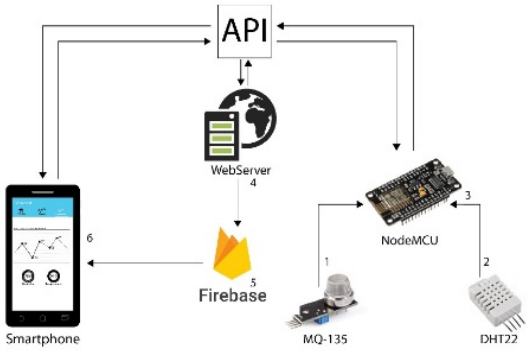


Figure 3. System Architecture

2.3 Carbon dioxide (CO2)

Carbon dioxide is a gas that is colorless and odorless. Carbon dioxide is toxic to the heart and causes a decrease in contractile force. At a concentration of three percent by volume in the air, it is narcotic light and causes an increase in blood pressure and pulse, and leads to a decrease in hearing power. At concentrations of about five percent by volume, it causes respiratory central stimulation, dizziness, confusion and respiratory difficulty followed by headache and shortness of breath. At an eight percent concentration, it causes headaches, sweats, blurred vision, tremor, and loss of consciousness after five to ten minutes of exposure. The global concentration of carbon dioxide (CO2) in the atmosphere has increased since the start of the industrial revolution due to the rapid growth of human activity. Pre-industrial gas concentration of CO2 was 278 ppm while in 2005 it was 379 ppm. [7] Minister of Industry Mohamad Suleman Hidayat stated "Indonesia's greenhouse gas emissions reach 1.3 million gigagrams CO2 equivalent. From that amount, the industrial sector contributes 3.12 percent of the production process, and 9.63 percent of energy use. " [8]

Table 2. Safe Limits of Carbon Dioxide

Air Quality Index	ppm	Symptom
Good	350 < 1000	Average indoor concentration with good air circulation system
Moderate	1000 < 2000	Complaints of drowsiness and poor air.
Unhealthy	2000 < 5000	Headaches, sleepiness and stagnant, stale, stuffy air. Poor concentration, loss of attention, increased heart rate and slight nausea may also be present.
Very Unhealthy	5000	maximum concentration limit of carbon dioxide in the work environment. For 8 hours working hours. According to OSHA (Occupational Safety and Health Administration)
Dangerous	>= 40000	May cause serious oxygen deprivation resulting in permanent brain damage, coma and even death

2.4 MQ135 Sensor

MQ-135 air quality sensor is a sensor that can be used to monitor air quality by detecting the presence of Ammonia gas (NH3), sodium-dioxide (NOx), alcohol / ethanol (C2H5OH), benzene (C6H6), carbon dioxide (CO2), sulfur gas / Sulfur-hydroxide (H2s) or smoke.



Figure 4. MQ135 Sensor

Basically, the workings procedure of the MQ135 air quality sensor are by detecting the presence of harmful gases around this sensor. This sensor will be used as a detector level of CO2 gas around the painting area of PT. Dirgantara Indonesia (Persero). The MQ-135 sensor has high sensitivity to harmful gases such as Ammonia (NH3), sodium-dioxide (NOx), alcohols / ethanol (C2H5OH), benzene (C6H6), carbon dioxide (CO2), sulfur / sulfur-hydroxide gas (H2s) Or smoke. The MQ135 sensor is applied for air pollution detection for domestic, industrial, and portable air pollution measurements.

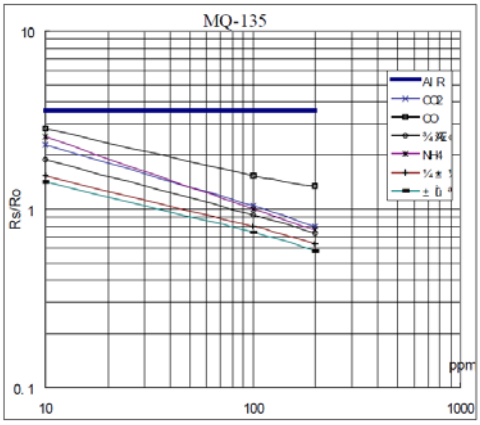


Figure 5. Characteristics Sensitivity of sensor MQ135

Figure 6. Shows the MQ135 sensor workflow from detecting the surrounding air quality level, to analog signal voltage which will be forwarded as microcontroller input to the system.

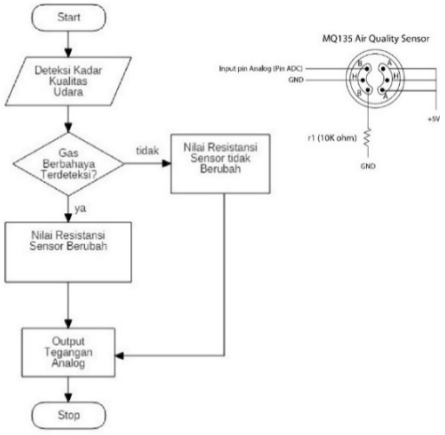


Figure 6. MQ135 Flowchart

Connection between MQ-135 sensor with NodeMCU microcontroller can be seen on Figure 7.

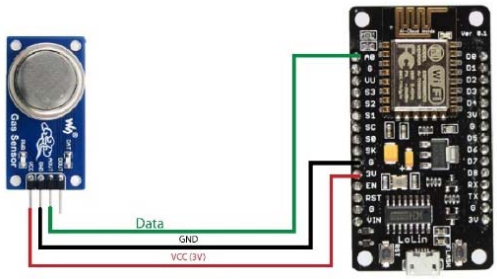


Figure 7. Sensor MQ135 and NodeMCU schematic

2.5 DHT22 Sensor

DHT22 or AM2302 sensor is a humidity and temperature sensor that has a very high precision level. The output of the DHT22 sensor is a digital signal. Inside the sensor component DHT22 consists of humidity sensor components and temperature sensors or thermistor. Both of these sensors have been connected to the 8-bit IC in it that converts the output into digital signals. So the DHT22 sensor can be connected directly with NodeMCU ESP8266 devices or with other microcontoler devices.

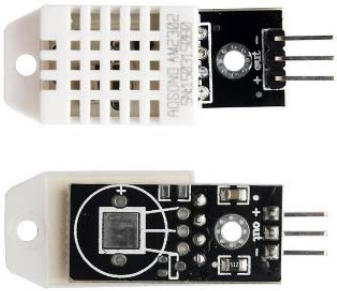


Figure 8. DHT22 Sensor

To measure the humidity, this sensor uses a component consisting of two layers of electrode with a humidity-sensitive layer between the two electrodes as shown in FIG. In principle when the humidity changes, the conductivity of the layers between the two electrodes changes. Then this change is directly measured and processed by IC in it so that data can be directly read by microcontroller.
The following components of the humidity sensor compiler on the DHT22 sensor.

The workflow of the DHT22 sensor can be seen on Figure 9.

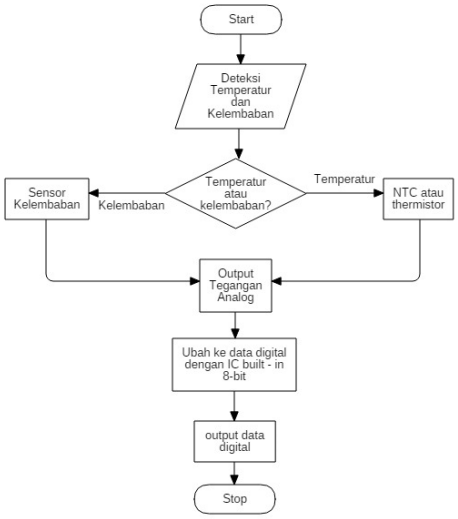


Figure 9. DHT22 Sensor Workflow

For this study, the sensors will be connected via GPIO2 or D4 pins on NodeMCU ESP8266 board. As shown on Figure 10.

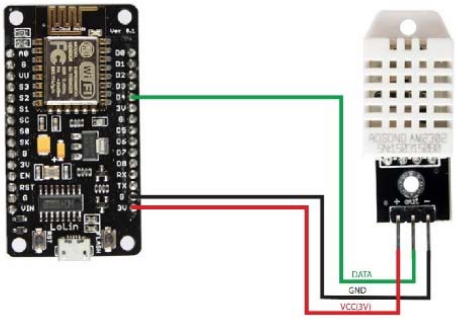


Figure 10. Sensor DHT22 and NodeMCU schematic

2.6 NodeMCU

For microcontroller device used in this research is NodeMCU ESP8266 board. NodeMCU is an open source IoT platform that is used to create IOT products programmed using the Lua programming language or can also use Arduino IDE. The platform is based on the ESP8266 module, which integrates GPIO, PWM (Pulse Width Modulation), and one ADC (Analog to Digital Converter) pin in one board. This board is very small with 4.83cm long, 2.54cm wide. Despite the small size, inside this board is equipped with features WiFi and firmware that is open source. In addition, the price of this board is very cheap compared to other microcontroller devices.



Figure 11. NodeMCU V_3 Lolin

2.7 *Analog to Digital Converter* (ADC)

ADC (Analog to Digital Converter) is an electronic device that serves to convert analog signal (continuous signal) into digital signal. ADC devices (Analog to Digital Converter) can be either a module or a series of electronics and an IC chip. ADC serves to bridge analog signal processing by digital system. In this study, the analog signals obtained from the air quality sensor MQ135 will be processed into digital signals using ADC provided by the NodeMCU ESP8266 board on pin A0. As shown on Figure 12.

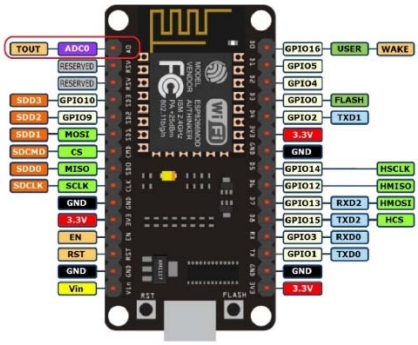


Figure 12. ADC Pin on NodeMCU ESP8266

The working principle of ADC is to convert analog signal into the form of magnitude which is the ratio of input analog voltage and system voltage. For example, an ADC has a 5V system voltage with an 8-bit ADC resolution, then a 5V voltage is assumed to be 255 in the digital value. Value 255 is obtained from 2^n-1 or in this case 2^8-1.

$$\frac{ADC\ Resolution}{System\ Voltage} = \frac{ADC\ read}{Analog\ Voltage} \quad (7)$$

For example if the analog voltage coming into the ADC pin is 3V, then the output of the digital value of the ADC is

$$\frac{255}{5.00V} = \frac{x}{3.00V} \quad (8)$$

$$\frac{255}{5.00V} * 3.00V = x$$

$$x = 153$$

Then, the digital signal to be obtained is 153 (decimal form) or 10011001 (digital form).

The most basic form of communication between digital and analogue forms is the comparator component which is usually an IC. Simply put, this component will compare the two voltages at the two input terminals. The output of the comparator depends on which voltage is greater. If the voltage at terminal V + is greater than V-, then the output will be a digital signal 1 (high) otherwise it will be worth a digital signal 0 (low). Here the concept of how the comparator works on ADC can be seen on **Error! Reference source not found.**

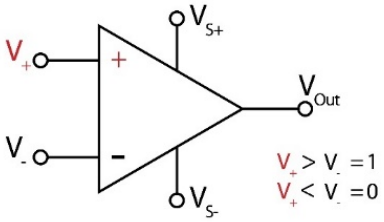


Figure 13. ADC Comparator

ADC consists of several types. One concept of ADC that has the advantage of more rapidly changing the data from analog to digital form is a simultaneous ADC or so-called flash converter. The simultaneous ADC consists of several comparators. The number of comparators can be calculated as if calculating the resolution that is by the formula 2n - 1. In the example Figure 14 The number of comparators in the 3-bit ADC is 2^3 - 1 then the number of comparators in the simultaneous ADC is 7 which is marked by C1 to C7.

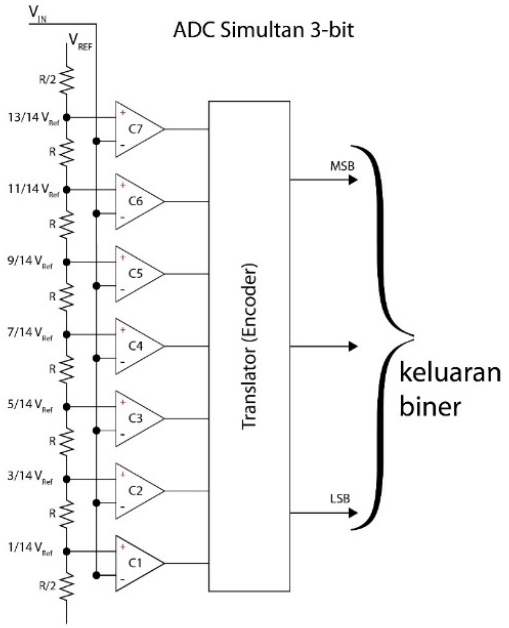


Figure 14. 3-bit Simultaneous ADC Scheme

If V_{REF} or system voltage is set at 5V voltage, then the voltage V- for each comparator is as follows

V- For comparator C7 = V_{REF} * (13/14) = 4,64V
V- For comparator C6 = V_{REF} * (11/14) = 3,93V
V- For comparator C5 = V_{REF} * (9/14) = 3,21V
V- For comparator C4 = V_{REF} * (7/14) = 2,5V
V- For comparator C3 = V_{REF} * (5/14) = 1,78V
V- For comparator C2 = V_{REF} * (3/14) = 1,07V

V. For comparator C1 = V_{REF} * (1/14) = 0,36V

For example, if the VIN gets a 3V input signal, then the result of the comparator to be obtained is

C7 Comparator = 3V < 4,64V = 0 (*low*)
C6 Comparator = 3V < 3,93V = 0 (*low*)
C5 Comparator = 3V < 3,21V = 0 (*low*)
C4 Comparator = 3V > 2,5V = 1 (*low*)
C3 Comparator = 3V > 1,78V = 1 (*low*)
C2 Comparator = 3V > 1,07V = 1 (*low*)
C1 Comparator = 3V > 0,36V = 1 (*low*)

So the output value of ADC after encoder is 100 (binary).

Table 3. Simultaneous ADC Comparator

Comparator Output							*Translator Output (encoder)*		
C7	C6	C5	C4	C3	C2	C1	2^2 (MSB)	2^1	2^0 (LSB)
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	1
0	0	0	0	0	1	1	0	1	0
0	0	0	0	1	1	1	0	1	1
0	0	0	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1

2.8 RESTful API

RESTful API is an implementation of API (Application Programming Interface). REST (Representational State Transfer) is a communication method architecture that uses the HTTP protocol for data exchange and this method is often applied in application development. The goal is to make a system that has a good performance, fast and easy to develop. Especially in the exchange of data communications. The design URL for this research is 'airquality / fire / sensor'. While Verbs used for research is verb GET and POST.

2.9 *Multiplatform* with Ionic

Ionic is a framework can be used to build a mobile application using web technologies such as HTML5, CSS and JavaScript so that the application form is called a mobile hybrid. In addition, Ionic also provides components - components that can be used by developers to access the features of mobile applications like mobile applications (native). Because the hybrid then the application is only made 1 time but can be released on more than 1 platform alias multiplatform. Stages in building multiplatform applications on ionic is by adding a platform that will be used on the project. For the addition of platforms on the Ionic project can use CLI with the following command.

```
ionic cordova platform add $platform
```

`$platform` Replaced with a platform to be used like ios, android, windows or browser. Example to add a browser platform.

```
ionic cordova platform add browser
```

To see the list of platforms that have been cultivated in the project can be seen in ProjectFolder \ platforms.

After the addition of the platfrom is complete, the application can be built in accordance with the required platfrom. But there are several needs that must be met before the project is built into the application for the necessary platfrom. Ionic provides documentation for the requirement in building applications for the desired platfrom. By using CLI, the following commands can be used to build projects into applications that platfrom can use.

```
ionic cordova build $platform
```

Examples of commands to compile web applications are as follows.

```
ionic cordova build Browser
```

The output of the above command has resulted in web applications that can be directly used as a web application. For the results can be retrieved in ProjectFolder \ platforms \ platforms \ browser \ www. These results can be directly hosted into the webserver.

2.10 Use Case Diagram

Use Case Diagram is a diagram that shows the functionality of a system or class and how the system interacts. As well as to explain the system is functionally visible to the user. The following Use Case diagram for the system to be built in this study is shown on Figure 15.

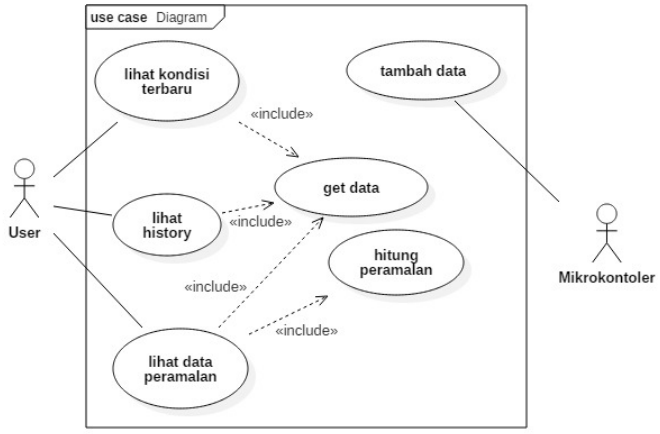


Figure 15. Use Case Diagram

2.10 Class Diagram

Class diagrams are structural diagrams that model the classes and their relationships. The class diagram is depicted with a box that basically includes class names, attributes and operations in the system as a whole, which include classes in the microcontroller, APIs and smartphone apps. For the class diagram in this study can be seen on **Error! Reference source not found.**

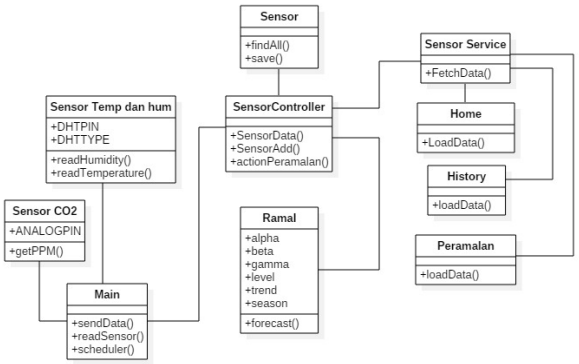


Figure 16. Class Diagram

2.11 Implementations and Testing

The hardware circuits of microcontrollers and sensors used to extract the ppm value of carbon dioxide, temperature and temperature in the case study environment can be seen in Figure 17.

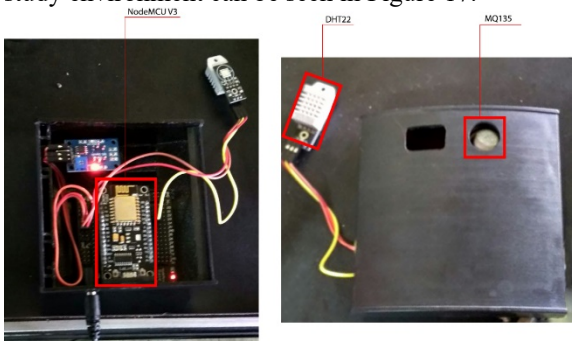


Figure 17. Microcontroller

2.12 API Testing

Testing POST and GET functions on API aims to determine whether the POST function used for the tool succeeds to store data into the database or not. While testing the GET function to find out whether the data retrieval from the server through the API that has been built successfully or not. The GET function in the API is used in smartphone apps to retrieve the data previously sent by the tool. This API function test uses POSTMAN software.

Error! Reference source not found. Shows the results of GET function testing performed using POSTMAN. It can be seen the result that the API managed to provide response data in the form of JSON which means that data can be retrieved and processed for smartphone applications.

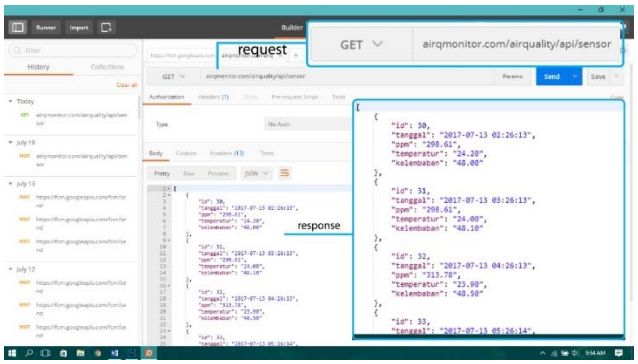


Figure 18. GET Function Test

Figure 19 Shows the result of POST function testing using POSTMAN application. Testing aims to test whether data storage through the API can be done. In its implementation, POST function is used by microcontroller to transmit data into the database.

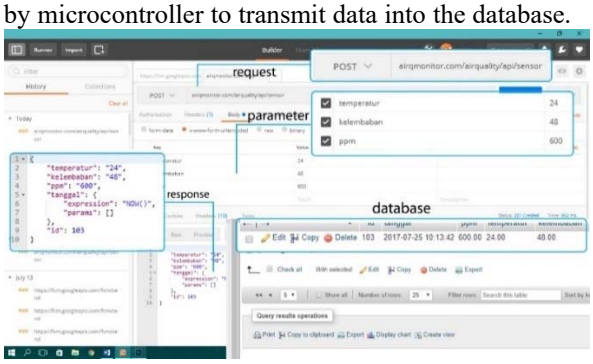


Figure 19. POST Function Test

3. CONCLUSION AND FUTURE SCOPE

Based on the discussion of the analysis, design, implementation to the test phase, can be drawn conclusions and suggestions for the use and development of air pollution monitoring system system with better forecasting features.

3.1 Conclusion

Based on the results obtained from research conducted in the preparation of the final task that mengaju on research purposes, it can be drawn conclusions as follows:

1. The system may provide information on the air quality index at its working area. So it can help succeed K3LH program in PT. Dirgantara Indonesia (Persero) in terms of employee safety.
2. The system can provide air pollution forecasts to anticipate the presence of poor air quality. This can be used as a support decision in improving the work environment or workload done.

3.2 Future Scope

Based on the result of the research, the researcher give suggestion or recommendation of development that can be done to air pollution monitoring system with forecasting feature, to succeed K3LH program at PT. Dirgantara Indonesia (Persero). The suggestions for the development of air pollution monitoring system with forecasting features are as follows:

1. The addition of sensors that are monitored not only gas carbon dioxide (CO2), because there are still many harmful gases produced from the industry.
2. Search values of alpha, beta, and gamma parameters to be searched directly by the application.
3. Automation system to handle the level of bad air pollution.

REFERENCE

[1] H. Riebeek, "Why Monitor Air Quality," 19 May 2009. [Online]. Available: https://terra.nasa.gov/citizen-science/air-quality. [Accessed 03 February 2017].

[2] World Health Organization, "Ambient air pollution," [Online]. Available: http://www.who.int/gho/phe/outdoor_air_pollution/en/. [Accessed 22 March 2017].

[3] A. Wahyudi, "Indonesia Raksasa Teknologi Digital Asia," Tempo.co, 2 October 2015. [Online]. Available: https://www.tempo.co/read/kolom/2015/10/02/2310/indonesia-raksasa-teknologi-digital-asia. [Accessed 22 March 2017].

[4] R. Amrutkar, S. Vikharankar and L. Ahire, "Security: Smart Homes Using Internet of Things (IOT)," *International Engineering Research Journal (IERJ),* vol. 2, no. 2, pp. 558-561, 2016.

[5] R. Jin, T. Liu and S. Wang, "The Application of Holt Exponential Smoothing Model on 2014 Air Quality Index in Miyun County, Beijing, China," *Bulletin of Mathematics,* vol. 3, no. 4, 2015.

[6] H. V. Ravinder, "Forecasting With Exponential Smoothing - What's The Right Smoothing Constant," *Review of Business Information Systems,* vol. 17, no. 3, 2013.

[7] Y. Wicaksono and A. Suismoni, "Deteksi Gas Berbahaya CO, CO2, NOX Dengan Penampil Dot Matriks dan level Bahaya Serta Besarnya," p. 3.

[8] Tempo, "Ini Dia Delapan Industri Penyumbang Emisi Terbesar," Tempo, 07 Mei 2013. [Online]. Available: https://m.tempo.co/read/news/2013/05/07/090478703/ini-dia-delapan-industri-penyumbang-emisi-terbesar.