Design of Micro-Climate Data Monitoring System for Tropical Greenhouse based on Arduino UNO and Raspberry Pi

I Ardiyahsah¹, N Bafdal², E Suryadi², A Bono³

¹Department of Agro-Industrial Technology, Faculty of Agro-Industrial Technology, Universitas Padjadjaran, Bandung, Indonesia.
²Department of Agriculture Engineering and Biosystem, Faculty of Agro-Industrial Technology, Universitas Padjadjaran, Bandung, Indonesia.
³Department of Chemical Engineering, Faculty of Engineering, Universiti Malaysia Sabah, Sabah, Malaysia.

Corresponding author’s e-mail address: irfan@unpad.ac.id

Abstract. The change in global climate has shown us the importance of sustainable environmental monitoring, especially the microclimate in greenhouses. Monitoring is carried out to determine greenhouse data, which differs from outdoor. Internet of Things (IoT) as emergence technology has an important role in collecting microclimate data, IoT is developed using hardware, sensors and software connected to the internet network. This paper implements this by using Arduino UNO, Raspberry Pi, DHT11 microclimate sensor to gain microclimate data and store it in cloud applications. Microclimate changes are recorded every minute and displayed on a web page so it can be analysed without visiting the greenhouse. The test results showed that the data recording was successfully carried out, and the microclimate data analysis showed that in the greenhouse there was a weak association between temperature and relative humidity with an R squared of 0.534.

Keywords: Greenhouse; Microclimate; Data Monitoring System; Cloud Applications

1. Introduction

Agricultural activities, which are mostly carried out by the small and middle level communities, are still not using technology that can assist them in agricultural land and products management. The problem faced by farmers is dependence on season and climate so that yield productivity does not always match expectations and needs adaptation when climate change occurs [1], [2]. Resource management such as changes in policies, technology application in agricultural methods and tools, agricultural information management, and risk management is needed to increase agricultural productivity because of the shrinkage of agricultural land [3]. Greenhouse is a technology approaches that capable to change and manipulate climatic conditions based on crop requirements. Limited land because of industrial and residential activities provides opportunities for greenhouse technology as a solution to meeting the needs for sustainable agricultural products and also function as a cover that can protect crops from climate change, weather, pests and plant diseases, both in four-season countries and two-season countries like Indonesia [2], [4].

Increased use of greenhouses in agriculture occurs because of reduced productive land the agricultural sector can use, reduced irrigation water discharge and global climate change. Greenhouses
provides opportunities for farmers to grow crops by optimising greenhouse area and manipulating ecosystem conditions such as soil, nutrients, irrigation, temperature, humidity, sunlight, wind, and air composition to suit plant needs [4], [5]. This technology is expected to increase the productive life of plants, change the growing season, increase crop yields, improve crop quality, and provide opportunities to plant crops outside their natural habitat [6].

Greenhouses solves the nutritional and food needs toward an increasing human population and makes improvements in post-harvest handling technology to ensure year-round food supplies. In meeting the needs of world food production, greenhouse development can facilitate out-of-season planting of crops and protect crops against varying field conditions. Greenhouses can also be a solution for areas that have extreme soil and climate problems [7], [8]. Global climate change and climate anomalies make planting and harvest times unpredictable, prolonged rains increase the risk of pest and disease strike, long drought causes plants to lose a lot of water and wither so that a system is needed that can maintain environmental conditions for plants one of which is the greenhouse [4].

Greenhouses are widely used in agricultural activities starting from nurseries, maintenance and harvesting, but many of these greenhouses still use manual methods for their management, starting from irrigation without paying attention to soil and plant moisture, fertilisation that is not suitable for plant needs and microclimate regulation and opening windows at inopportune times so that pests and plant diseases can penetrate [9]. Besides that, time management and the lack of tools are also a problem in greenhouse management, especially in several types of plants that require supervision to improve the quality and quantity of their harvest. With proper management of greenhouse data and information, farmers can have a firm foundation in the selection and treatment of crops. This management may not be carried out on agricultural land because of its enormous size requires a lot of equipment and availability of electricity supply at the farm site [10].

A researcher [11] stated that the rapid development of information technology and telecommunications, especially in the industry revolution 4.0 era produce a technology called the Internet of Things (IoT) which provides opportunities for the agricultural industry to implement IoT technology in greenhouses, one of which is the implementation of wireless technology-based greenhouses as was done in this study. This study uses information and telecommunication technology by optimising micro-controller, microcomputer, cloud service, serial and Wi-Fi connections to perform microclimate data monitoring via internet networks to facilitates IoT based greenhouses data collection.

This research is focused on studying the application of a wireless-based monitoring system capable of recording microclimate data and sending it to the cloud so it can be accessed by users via a web application. The proposed system must be reliable, cost-effective, and minimise human intervention. Device development is carried out using Arduino UNO which acts as a monitoring unit and Raspberry Pi which functions as a data processing unit, microclimate data on the Raspberry Pi is sent wirelessly to a cloud database within a certain time span, processed by web application and displays microclimate information system when requested by a web browser.

The basis for choosing the Arduino UNO as a monitoring device is the accessibility of the General Purpose Input Output (GPIO) pin, which is more flexible than the Raspberry Pi, especially the analog pins that the Raspberry Pi does not have. It can be expanded with an abundance of components, sensors, modules, and shields. The drawback is that the more you add, the thicker the size. Arduino UNO is an embedded system that can only perform a single task, meaning that it is only suitable for use as a monitoring device. In contrast, Raspberry Pi as a processing device, because this device is actually a microcomputer that has a processor, memory, storage media and operating system so it can perform data retrieval, data recording, data transmission and calculations simultaneously [12]–[14].

2. Materials and Method
Management of chicken coops using Arduino UNO has been carried out by [15] by implementing microclimate sensors and gas sensors to detect temperature, humidity levels of carbon dioxide and ammonia. Control is done by adding a fan that will turn on when the conditions in the chicken coop
are beyond the threshold. An agricultural land protection system is proposed by [16] to monitor land conditions and detect intruders using the Arduino UNO, PIR sensor, ultrasonic sensor and webcam by detecting any movement that may occur around the sensor. In 2017, [17] examined the application of fuzzy logic in Arduino UNO to monitor soil moisture characteristics in rice and maize crops, the results got in this study were accurate by agricultural experts. The Supervisory Control and Data Acquisition (SCADA) system is used by [18] to monitor the environmental conditions of agricultural crops by implementing the Raspberry Pi network in the zoning system, this research has resulted in a monitoring network that is safe and has minimal signal interference. Another study using Raspberry Pi was conducted by [19] to measure the chlorophyll content in paddy leaves to inform farmers whether their rice plants had disease, lack of nutrients, or too much nitrogen. Monitoring is done by adding a digital camera module to the Raspberry Pi. Another system was built by [7] to control the microclimate in the greenhouse using a Raspberry Pi and fog misting system, although it can monitor microclimate conditions, researchers find it difficult to create a controlling device using a relay module because the Raspberry Pi does not have an analog pin so an analog-to-digital converter is needed as an alternative. Using Arduino UNO and Raspberry Pi has been carried out by [20] by building a smart refrigerator that can detect the availability of products in it, and this device can also display the weight of ingredients and recipes that match the ingredients available in this refrigerator via an LCD attached to the refrigerator door.

2.1 Methods

The hardware used in this research is Arduino UNO, Raspberry Pi, Breadboard and DHT11 Sensor, while the software used is Arduino C, Python3, PHP, Crontab, MariaDB (Raspberry Pi) and MySQL (Cloud Hosting). The data flow on the system can be seen in Figure 1.

![Figure 1. Data flow of the proposed monitoring system.](image)

2.1.1 Monitoring Unit

- **Arduino UNO (UNO)**, an open source micro-controller hardware that has the ATMega328 chipset. UNO functions as a monitoring device by adding a DHT11 sensor module and is not used as a data storage device because it requires physical access to hardware. The power supply comes from the Raspberry Pi via a USB connection [21].

- **DHT11**, a sensor module used to detect microclimates. The sensor is calibrated by comparing sensor and thermometer values at the same location and time. This sensor was chosen because of its high level of accuracy in tropical temperatures and its digital output value [21].
2.1.2 Processing Unit

- **Raspberry Pi (RPI)**, this study uses the Raspberry Pi (RPI) version B microcomputer released in 2012 with ARM processor and 512MB RAM. The RPI specification is reliable enough to be used as a headless server for excellent resource management and electricity usage. RPI has its own operating system called Raspberry Pi OS, one of the Debian derivatives which is easy to use, although many developers also developing operating systems for RPI. The RPI itself already has 5 different versions with increasingly better specifications, but the impact is that power requirements also increase, especially in version 3 and 4, which added a wireless module (Wi-Fi and Bluetooth). Raspberry Pi version 4 is not recommended for automation activities in the field with minimal electricity supply because it is easy to heat, needs a minimum power of 5V 3A and has problems with the USB C chipset being used so it only supports certain types of cables [22].

- **Wi-Fi Dongle**, The RPI used in this study only has an ethernet connector to establish internet connection, therefore a wireless USB dongle is needed to activate the Wi-Fi feature on the RPI by choosing Edimax 7611un which is already supported by Raspberry OS Lite.

- **Raspberry OS Lite**, an operating system optimised for Raspberry Pi devices, the lite version does not have a desktop environment so it can only be accessed using a terminal or remotely via Secure Shell (SSH). The SSH application commonly used on Windows is called Putty and can be downloaded for free on the internet. The lite version is very suitable for use in a server environment because a server does not require complex graphics management so that all resources can be focused on server activities.

- **Python3**, Python3 was chosen because it is a versatile interpreter programming language, easy to understand, has an extensive library and is supported by a wide community. An open source software development application widely used for automation and machine learning [22]. This application is installed by default on the Raspberry Pi OS, requires additional libraries to perform data transactions with UNO and the Cloud Application installed using pip. Python3 is used as a data intermediary between UNO and Cloud Application.

- **MariaDB**, Linux-based database server used on the RPI to store microclimate data, MySQL derivatives developed in an open source manner. The table structure used in this study is as follows (Table 1):

| Column Name | Data Type | Table Declaration |
|-------------|-----------|-------------------|
| Atomic      | int       | Primary Key       |
| Temp        | int       | Not Null          |
| Humid       | int       | Not Null          |

- **Cloud Application (CA)**, a cloud web hosting that supports PHP and MySQL. MySQL database is used as microclimate data storage, while PHP is used to create microclimate information system.
Monitoring system design was carried out using the Fritzing application version 0.9.3b, which is licensed for free, the design was carried out to see the cable requirements and sensor placement layout on monitoring unit (UNO) which can be seen in Figure 2.

2.1.3 Device Development
UNO task is to detect micro-climatic conditions through the DHT11 sensor, which is then parsed through DHT library. Data acquisition is carried out automatically every minute and transmitted via the serial port with a data transfer rate of 9600 bps. Data acquisition is carried out every minute to avoid excessive work on UNO and RPi (Table 2).

| Algorithm |
|-----------|
| load dht library |
| if (serial port) ≠ ready |
| recheck |
| end if |
| while (true) |
| read DHT11 temperature in Celsius |
| read DHT11 humidity in Percent |
| if (temperature and humidity not a number) |
| write “Sensor error” |
| end if |
| send temperature and humidity data using serial connection |
| delay for 60 seconds before continue reading |
| end while |

The RPi connects to the UNO via a USB type B cable and parses the microclimate data every minute using a serial library. These data are inserted into the database using MariaDB connector whose table structure is shown in Table 1. The application is configured as a background process and runs 24 hours nonstop and producing 1440 microclimate data each day (Table 3a).
Table 3. RPi algorithm for (a) reading and saving microclimate data, and (b) sending microclimate data to CA

| Algorithm (a) |
|---------------|
| connect to serial port with 60 seconds timeout |
| connect to greenhouse database |
| while true |
| if (there is data in serial connection) |
| read serial data |
| split temperature and humidity data |
| record unix time |
| insert data to microclimate table |
| delay for 60 seconds before continue reading |
| end if |
| end while |
| Algorithm (b) |
| send json request |
| receive json response |
| read unix time from json response |
| if (rpi unix time > json unix time) |
| select all data whose unix time is greater than json response |
| transmit data to ca through get method |
| ca responds and translate get request |
| insert data into ca database |
| delay for 5 seconds before continue forwarding |
| end if |

Data Synchronization between RPi and CA is done using Python3 and JSON based applications. CA generates a JSON file when there is a request for the latest data from RPi. The latest data in RPi is then compared with the latest data on the CA, when there is a difference, the data on the CA will be updated. Synchronization is carried out every five minutes using the crontab application on the RPi. Algorithm in Table 3b shows a synchronization between RPi and CA.

The data received by the CA is presented in tabular and line chart form (Table 4a), the main page will display the latest 200 microclimate data sorted in descending order. CanvasJS service is used to create Line Charts on CA pages, CanvasJS itself is a JavaScript-based service for converting JSON files into charts.

Another feature of this CA is that it can display daily microclimate data based on the Asia / Jakarta time zone (+7 GMT). Daily data is displayed by selecting the date via the date picker form, which after submission will display tabular data and line charts for all data on the selected day and also display minimum, maximum mean and standard deviation value for air temperature and relative humidity data.

Table 4. CA algorithm for (a) displays the latest 200 microclimate data, and (b) displays daily microclimate data based on the selected date

| Algorithm (a) |
|---------------|
| connect to database |
| select 200 latest microclimate data sort descending |
| send json request for canvasjs graph |
while data fetched from table
  read unix time
  read temperature
  read humidity
  set asia/Jakarta timezone
end while

receive json request for canvasjs graph
display canvasjs graph for 200 latest microclimate data
display a table containing microclimate data

**Algorithm (b)**

connect to database
read date from get request
embed timestart and timeend to date
convert timestart and timeend to unix time
convert timestart and timeend to asia/Jakarta time zone

send json request for canvasjs graph for selected date
select microclimate data for selected data sort by unix time ascending

while data fetched from table
  read unix time and convert to array
  read temperature and convert to array
  read humidity and convert to array
end while

for all data inside array
  search minimum value for microclimate data
  search maximum value for microclimate data
  calculate microclimate data mean
  calculate microclimate data standard deviation
next

receive json request for canvasjs graph for selected date
display canvasjs graph for selected date
display statistics calculation
display a table containing microclimate data

3. Results and Discussion

The proposed greenhouse microclimate monitoring system circuit is assembled, and field tested (Figure 3) to determine if the sensor successfully detects microclimate condition and stream it to RPi and forward it to CA. The test is carried out by turning on the monitoring unit and the control unit simultaneously. Sensor data acquisition results are observed on the RPi remotely via SSH and compare it with CA information system. The device was tested for 20 days by turning it on for 24 hours daily.
Figure 3. Microclimate data monitoring device assembled, and field tested

Observation of microclimate data was carried out through CA installed on cloud hosting; the CA comprises two web pages with similar functions. The main page displays the latest 200 microclimate data in chart (Figure 4a) and tabular (Figure 4b). The second page displays daily microclimate data in chart (Figure 4c) and tabular (Figure 4d) based on the selected date using the date picker component. By default, this page displays the previous day's microclimate data and provides information on minimum and maximum values, mean, and standard deviation for daily air temperature and relative humidity data.

Figure 4. Website interface (a) displaying line chart for the latest 200 micro climate data (b) displaying the latest 200 micro climate data in tabular form in descending order (c) displaying line chart for daily micro climate data (d) displaying daily micro climate data in tabular form
Using air temperature as an independent factor (X) and air relative humidity as a dependent factor (Y) from 28800 rows of data recorded over 20 days, a scatterplot chart is used to determine the response of factor Y to changes in greenhouse temperature [23]. A negative linear pattern in Figure 5 shows that there is a tendency that air relative humidity value inside greenhouse will decrease when the temperature increases, and based on the data distribution and R squared value (0.7369), it proved that the association between two parameters are strong so it is possible that the decrease in greenhouse air temperature will increase air relative humidity.

![Figure 5. Scatterplot of temperature in Celsius (X) and relative humidity in percent (Y) with R^2 = 0.7369 from 28906 microclimate data](image)

### 4. Conclusion

Development and testing of microclimate monitoring tools have been carried out and successfully recorded data for 20 days, recording only failed when UNO or RPi was off. Data update in CA only occurs when RPi is connected to the internet, if the internet connection is lost then RPi will wait until the connection is reconnected and continue synchronization based on the latest CA data.

The device developed is suitable for application in small to medium scale greenhouses when using a single sensor. For large-scale applications, higher costs are required for equipment investment, although monitoring costs will remain the same. The tool is effective in reducing direct field monitoring cost because farmers only need to access the web page for seeing the latest greenhouse micro-climate data. Device application in a greenhouse can eliminate the risk of human error when managing greenhouses, especially in times of rapid micro-climate change. The drawback is the use of a USB cable for the connection between the UNO and RPi which causes these two units to cannot be kept apart, especially if you want to implement them in a large-scale greenhouse.

It is possible to add other types of sensors needed for plant development, such as solar index sensors, wind sensors, carbon dioxide sensors, and soil moisture sensors by changing a few program codes in UNO, RPi and CA, so that the website will be information-rich and those data can be used for data mining or machine learning because recording data every one minute for twenty days yields 28906 rows of data.

### References

[1] A. Calderón, A. Martins da Mota, C. Hopchet, C. Grabulosa Olivé, and M. Roeper, “The GreenHouse: A sustainable solution to grow aromatic plants in small houses.” Universitat Politècnica de Catalunya, 2017.
[2] N. Bafdal and S. Dwiratna, “Water Harvesting System As An Alternative Appropriate Technology To Supply Irrigation On Red Oval Cherry Tomato Production,” Int. J. Adv. Sci. Eng. Inf. Technol., vol. 8, no. 2, pp. 561–566, 2018, doi: 10.18517/ijaseit.8.2.5468.

[3] M. Paustian and L. Theuvsen, “Adoption of precision agriculture technologies by German crop farmers,” Precis. Agric., vol. 18, no. 5, pp. 701–716, 2017, doi: 10.1007/s11119-016-9482-5.

[4] N. Castilla, Greenhouse Technology and Management, 2nd ed. Oxfordshire: CABI, 2013.

[5] I. Ardiansah, N. Bafdal, E. Suryadi, and A. Bono, “Greenhouse Monitoring and Automation Using Arduino: a Review on Precision Farming and Internet of Things (IoT),” Int. J. Adv. Sci. Eng. Inf. Technol., vol. 10, no. 2, 2020.

[6] J. G. Lee, Y. K. Jeong, S. W. Yun, M. K. Choi, H. T. Kim, and Y. C. Yoon, “Field survey on smart greenhouse,” Prot. Hortic. Plant Fact., vol. 27, no. 2, pp. 166–172, 2018.

[7] M. Hafiz, I. Ardiansah, N. Bafdal, A. Info, and M. Control, “Website Based Greenhouse Microclimate Control Automation System Design,” JOIN (Jurnal Online Inform., vol. 5, no. 1, pp. 105–114, 2020, doi: 10.15575/join.v5i1.575.

[8] A. Kumar, G. N. Tiwari, S. Kumar, and M. Pandey, “Role of Greenhouse Technology in Agricultural Engineering,” Int. J. Agric. Res., vol. 1, no. 4, pp. 364–372, 2006, doi: 10.3923/ijar.2006.364.372.

[9] S. V Devika, S. Khamuruddeen, S. Khamurunnisa, J. Thota, and K. Shaik, “Arduino Based Automatic Plant Watering System,” Int. J. Adv. Res. Comput. Sci. Softw. Eng., vol. 4, no. 10, pp. 449–456, 2014.

[10] J. Farfan, A. Lohrmann, and C. Breyer, “Integration of greenhouse agriculture to the energy infrastructure as an alimentary solution,” Renew. Sustain. Energy Rev., vol. 110, pp. 368–377, 2019.

[11] O. Elijah, T. A. Rahman, I. Orikumhi, C. Y. Leow, and M. H. D. N. Hindia, “An overview of Internet of Things (IoT) and data analytics in agriculture: Benefits and challenges,” IEEE Internet Things J., vol. 5, no. 5, pp. 3758–3773, 2018.

[12] V. S. Gunge and P. S. Yalagi, “Smart Home Automation: A Literature Review,” Int. J. Comput. Appl. Natl. Semin. Recent Trends Data Min., pp. 975–8887, 2016.

[13] I. Ardiansah and S. H. Putri, “Perbandingan Analisis SWOT Antara Platform Arduino UNO dan Raspberry Pi,” 2016.

[14] J. Islam et al., “Design and Development of Microcontroller Based Wireless Humidity Monitor,” IOSR J. Electr. Electron. Eng., vol. 13, no. 2, pp. 41–46, 2018, doi: 10.9790/1676-1302034146.

[15] T. S. Gunawan, M. F. Sabar, H. Nasir, M. Kartiwi, and S. M. A. Motakabber, “Development of Smart Chicken Poultry Farm using RTOS on Arduino,” in 2019 IEEE International Conference on Smart Instrumentation, Measurement and Application (ICSIMA), 2019, p. 5 pp., doi: 10.1109/ICSIMA47653.2019.9057310.

[16] S. Yadahalli, A. Parmar, and A. Deshpande, “Smart Intrusion Detection System for Crop Protection by using Arduino,” in 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), 2020, pp. 405–408, doi: 10.1109/ICIRCA48905.2020.9182868.

[17] G. A. Gines, J. G. Bea, and T. D. Palaoag, “Characterization of Soil Moisture Level for Rice and Maize Crops using GSM Shield and Arduino Microcontroller,” in IOP Conference Series: Materials Science and Engineering, 2018, vol. 325, p. 012019 (6 pp.), doi: 10.1088/1757-899X/325/1/012019.

[18] A. J. Moshayedi, A. S. Roy, L. Liao, and S. Li, “Raspberry Pi SCADA Zonal based System for Agricultural Plant Monitoring,” pp. 427–433, 2019, doi: 10.1109/ICISCIE48695.2019.00092.

[19] A. M. T. Nasution, Y. A. Fajrin, and H. Suyanto, “Calibrating of simple and low cost Raspberry-Pi camera-based Chlorophyll meter for accurately determining chlorophyll content in paddy leaves,” in Proceedings of the SPIE, 2019, vol. 11044, p. 1104407 (4 pp.), doi: 10.1117/12.2503830.
[20] C. Manuel, J. Avila, L. Budiman, R. Wijaya, and R. Hedwig, “Customizable smart food cabinet and refrigerator,” Pertanika J. Sci. Technol., vol. 27, no. 1, pp. 143–157, 2019.

[21] N. Dunbar, Arduino Software Internals: A Complete Guide to How Your Arduino Language and Hardware Work Together. Apress, 2020.

[22] A. Pajankar, Raspberry Pi Computer Vision Programming: Design and implement computer vision applications with Raspberry Pi, OpenCV, and Python 3, 2nd Edition. Packt Publishing, 2020.

[23] M. Kubat, An Introduction to Machine Learning. Springer International Publishing, 2017.