Vol. 13, No. 3 (2024): 966 - 977
http://dx.doi.org/10.23960/jtep-l.v13i3.966-977

JURNAL TEKNIK PERTANIAN LAMPUNG

ISSN 2302-559X (print) / 2549-0818 (online)
Journal homepage : https://jurnal.fp.unila.ac.id/index.php/JTP

Automatic Tomato Plant Watering System Using Fuzzy Logic Control with Telegram-Based Monitoring System

Sri Purwiyanti[1,✉], Umi Murdika[1], Pinkga Nata Pratama[1], Ageng Sadnowo Repelianto[1]

[1] Department of Electrical Engineering, University of Lampung. Soemantri Brojonegoro St. #1, Bandar Lampung, INDONESIA. 35145

Article History:

Received : 21 June 2024
Revised : 01 September 2024
Accepted : 08 September 2024

Keywords:

Fuzzy logic,
Humidity,
NodeMcu,
Sensor,
Soil pH.

Corresponding Author:
✉ sri.purwiyanti@eng.unila.ac.id
(Sri Purwiyanti)

ABSTRACT

In tomato plants, one of the decisive factors is the soil humidity factor, where the optimal humidity for tomato plants is 80%. For this reason, the tomato planting process requires regular and measurable irrigation or watering. This research aims to create an automatic watering system for tomato plants with soil humidity and pH detection. In this system, Fuzzy logic is used as the control method and this system is also equipped with a monitoring system using the telegram application. In this system, the humidity sensor and soil pH sensor will provide information about soil conditions and forward the electrical signal to the NodeMcu ESP32 microprocessor, as the control center and signal processing. This input signal is then processed using Fuzzy logic control to decide whether the system will activate the relay on the water pump. If the humidity value is <60%, the pump will activate and sprinkle water until the humidity reaches 80%. The measurement results of soil humidity, soil ph, and pump relay conditions will be sent via Internet of Things technology so that they can be monitored remotely using the Telegram application. From the research results, it is found that the system can work as expected with an accuracy value of 100% with a delay time for sending data to Telegram of 14.1 seconds.

1. INTRODUCTION

Tomato (*Lycopersicum Esculentum* Mill) is a plant of the *Solanaceae* family, native to Central and South America, from Mexico to Peru. Tomato itself has a short life cycle and has a height between 1-3 m (Sulardy & Sani, 2018). Tomato plants can grow well in tropical Indonesia. Temperature factors can affect fruit color because at high temperatures (above 32 °C) the color of tomato fruit tends to be yellow, while at irregular temperatures the fruit color tends to be uneven. The ideal temperature tomato is between 24-28 °C that affecting fruit color to be attractively good which is generally evenly red. High temperature and high humidity conditions have an unfavorable effect on the growth, production and quality of tomato fruit. The relative humidity required for tomato plants is 80%. Tomato plants require sunlight intensity of at least 10-12 hours every day. Tomatoes require regular and measured irrigation or watering (Yahwe *et al*., 2016).

Tomato, as well as other plants in general, require water and fertile soil as one of the requirements in order to grow well. Water functions as a component in the process of photosynthesis and respiration in plants. The growth of plants is influenced by the intensity of the water it contains. Thus, the watering process is a very important process in plant growth and development. The current difficulty is that the watering process is still mostly done by farmers manually, and is carried out regularly twice a day without looking at soil humidity conditions. This makes the watering process not optimal and less efficient, especially when applied to large-scale tomato plantations (Marinus *et al*., 2020).

Several studies have been conducted to be able to produce more efficient automatic watering tools. Starting from automatic watering based on time (Zulfikar, 2018), to more efficient ones by considering soil humidity conditions

(Philander *et al*., 2021; Yudhana & Putra, 2016). Furthermore, automatic watering equipment can also be monitored remotely by utilizing Internet of Things (IoT) technology, namely using a mobile phone with the Blynk application (Sasmoko & Hotman, 2020; Fauzia *et al*., 2021; Alfonsius *et al*., 2024).

In terms of control systems for automatic watering systems, research has been reported on automatic watering systems with conventional controllers, namely using PID control systems (Sirait & Botiwicaksono, 2020) and PLC control (Nadhiroh *et al*., 2024). Even one type of artificial intelligence, namely Artificial Neural Networks, has also been used (Afifah *et al*., 2020). Another type of artificial intelligence, namely Fuzzy Sugeno, has also been used as a control system for plant watering, but applied to pakcoy plants (Yoseph *et al*., 2021). In terms of microcontrollers used, various types of microprocessors have also been used with diverse results. In the research conducted by Iman *et al*. (2022), Wemos D1R1 was used, but other types are also widely used such as Arduino microcontrollers (Nadindra & Chandra, 2022) and Raspberry Phi (Cobantoro *et al*., 2019).

To improve and complement the existing irrigation system, in this research an irrigation system is made that applies an intelligent control system, namely Fuzzy Mamdani logic. Fuzzy logic is used as a control system for activating the water pump relay, which is active or not depending on soil humidity conditions. As for this system, the tool made can do watering automatically when the humidity is less than 60%, and watering will stop when the humidity reaches 80%. The measurement results of soil humidity, pH, and water conditions are sent to the internet and can be monitored remotely via cell phone using the Telegram application.

2. RESEARCH MATERIALS AND METHODS

In this section, the research methods used will be explained, including the working principle of the tool and its supporting components and the creation of software.

2.1. Hardware

Figure 1 below shows the block diagram of the system. The input section consists of three sensors, which are used respectively to measure humidity, soil pH, and water availability. Soil humidity is a major factor in determining plant agronomic characters (Djumali & Mulyaningsih, 2014).

To detect soil humidity, a YL-69 sensor is used. The YL-69 sensor is a sensor that is widely used for automatic plant watering systems (Darmawan *et al*., 2020). This sensor consists of two probes that function to read soil humidity in the form of resistance. The relationship between the probe length and the ADC voltage value with the resistivity value is that the deeper the YL-69 sensor probe is plugged into the ground, the lower the resistance value (Setyowati *et al*., 2020).

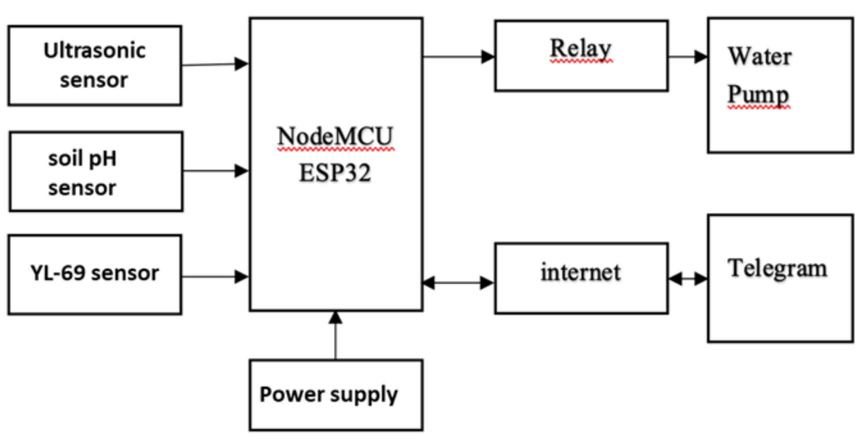

Figure 1. Block diagram of the watering system created using fuzzy logic control with Telegram

The HC-SR04 ultrasonic sensor is used to measure the water level in the reservoir. The data read from the sensor is then sent to the NodeMCU microcontroller for processing. In NodeMCU, the data obtained is then processed using Fuzzy logic to be able to control the relay. If the relay gets a signal, it will activate and make the water pump water the plants. Whether or not the relay is active is influenced by two parameters, namely soil humidity conditions and soil pH levels. The schematic wiring diagram of the device is shown in Figure 2, while the watering system placement scheme is shown in Figure 3.

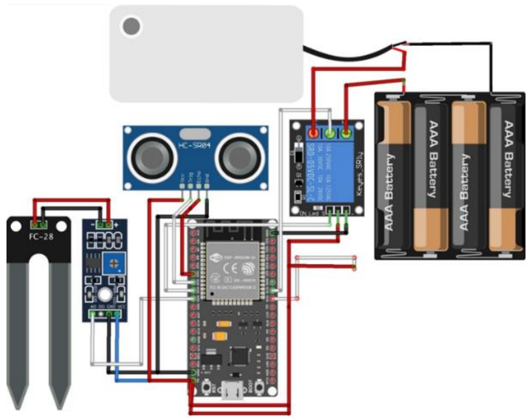

Figure 2. Schematic wiring diagram of the system

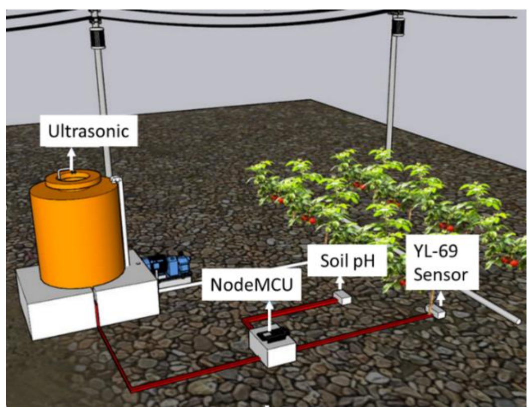

Figure 3. Schematic of watering system

The NodeMCU microcontroller is one of the single-board micro-controllers that has WiFi features, making it useful in making IoT platform products. By using this microcontroller, the results of data processing can also be sent to a mobile phone via the internet network. As a viewer on the cell phone, the Telegram application is used. Integration can be done between messages on Telegram and NodeMCU microcontroller through Telegram Bot (de Oliveira *et al*., 2016). In this system, Telegram functions as a remote monitoring system for the value of soil humidity, soil pH, and water conditions in the reservoir. In addition, Telegram also has a control function, because users can turn off and turn on the water pump remotely using Telegram.

2.2. Flow Chart of the System

The flowchart of the system with Fuzzy logic controller is shown in Figure 4. In the figure, it can be seen that the parameters that determine the water pump to water the plants will be active or not are the measurement results of soil humidity and soil pH. In this system, the tool made can perform watering automatically when the soil pH is less or equal to six and soil humidity is less than 60%, and watering will stop when the humidity has reached 80%. If the soil pH is more than six, the automatic watering system does not work. In this condition, the system will still display the pH measurement results on the Telegram, so that users can perform manual treatment to reduce soil pH.

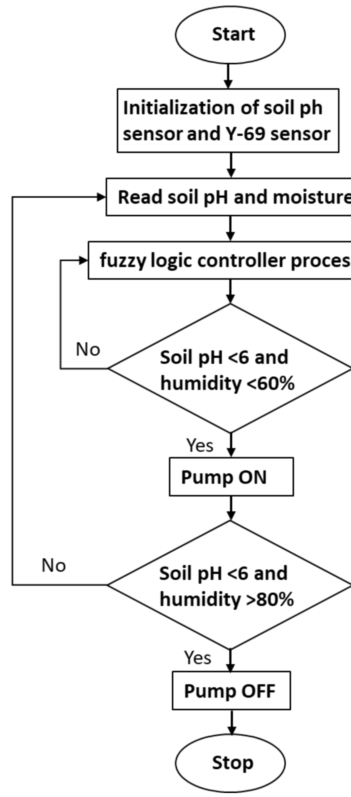

Figure 4. Flowchart of the system with Fuzzy logic controller

2.3. Software

For software creation, programs are created and integrated into the NodeMCU via the Arduino IDE software, including for connection to the telegram application. To be able to start making programs on the NodeMCU ESP32 using the Arduino IDE, the following steps must be taken (Prasetyo, 2020).

1. The first step is to download and install the Arduino IDE software on a laptop or computer.
2. Open Arduino IDE >> then select the File menu >> Preference
3. In the Preference column, enter the link: https://dl.espressif.com/dl/package_esp32_index.json. Then click OK.
4. Then close the Preference window, then open Tools>> Board>> Boards Manager
5. In the Boards Manager window, type ESP>> then select the Highest Version>> then click Install
6. Next, choose the Tools >> Board >> then select ESP32 Dev Module
7. Then connect the NodeMCU ESP 32 to the laptop/computer via the USB port.
8. Type the Program Code, as shown in Figure 5, to test and ensure that the ESP32 Serial Port is working properly.
9. Upload the program code to the NodeMCU ESP32.
10. When the text display appears connecting......, press the BOOT button located on the NodeMCU ESP32 board
11. Finally choose Tools>> Serial Monitor. If successful, it will appear as shown in Figure 5 on the serial monitor.
12. NodeMCU ESP32 is ready to use and can be programmed as needed.

After the NodeMCU is ready to use, programs are made which include programs for reading sensors, connection programs to telegrams, and Fuzzy control programs. All of these programs were made on a laptop using Arduino IDE software and then uploaded to the NodeMCU ESP32.

```
void setup()
{
  Serial.begin(115200);
}

void loop()
{
  Serial.println("ESP32 Serial Test");
  delay(1000);
}
```

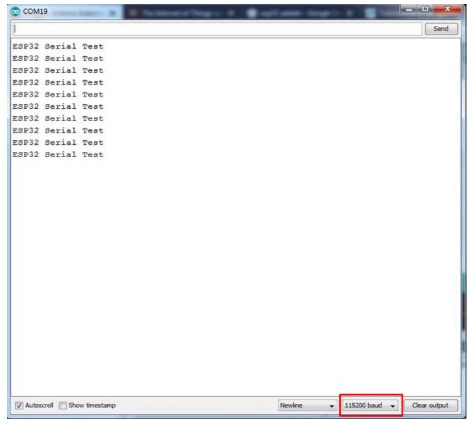

Figure 5. Program code for testing NodeMCU (left) and display on NodeMCU after it is ready to be used (right)

The integration process between the NodeMCU ESP32 microcontroller and the Telegram application is also carried out through the Arduino IDE software, but it requires the help of an application programming interface (API) in the form of BotFather. The first step in this integration is the creation of a Telegram bot through BotFather. Included in this process is giving a name and username for the bot, followed by receiving a unique API token. This token is used as the code uploaded to the NodeMCU ESP32 to authenticate the connection between the device and the Telegram server. With this token, the NodeMCU ESP32 can send and receive messages through the Telegram bot, allowing the device to operate according to the commands given.

Data transmission from the NodeMCU ESP32 to the Telegram bot allows information from sensors connected to the ESP32 to be received in real-time via Telegram. This sending process consists of several stages, from connecting to a Wi-Fi network to using the Telegram API to send data. The first step in this process is connecting the NodeMCU ESP32 to a Wi-Fi network. The NodeMCU ESP32 has a built-in Wi-Fi module that allows connection to the internet. The configuration code on the NodeMCU ESP32 includes setting the SSID and password of the Wi-Fi network being used. Once connected, the NodeMCU ESP32 can access the internet, which is required to communicate with the Telegram server.

After a successful Wi-Fi connection is established, the next step is to authenticate the ESP32 with the Telegram bot using the API token obtained from BotFather. This API token is included in the NodeMCU ESP32 code and is used to send HTTP requests to the Telegram API. For example, to send data from sensors such as soil pH or humidity, the NodeMCU ESP32 will form a text message which is then sent to the Telegram bot. This data transmission is done through the HTTPS protocol to ensure the security of the data sent to the Telegram server.

In the NodeMCU ESP32 code, the use of the HTTPClient library from the Arduino IDE allows sending an HTTP POST request to the Telegram API. This request includes the API token, the bot's chat ID, and a message containing the data to be transmitted. For example, data read from a soil pH sensor (program listing in Figure 6) can be converted to a string and sent as a text message to a Telegram bot. The bot then forwards this message to the connected user, allowing remote monitoring of the sensor data via the Telegram app.

After the manufacture of hardware and software is completed and integrated, then testing of each sub-system is carried out to find out whether each part can function as expected. At the end, the integration of the entire system and the delay time of sending data to the telegram were tested.

3. RESULTS AND DISCUSSION

3.1. Fuzzy Control Execution Steps and Results

In this system, Fuzzy logic is used as a control system in the data processing section using NodeMCU ESP 32. Mamdani Fuzzy logic is one of the artificial intelligence methods that is usually used as a control system and is easily applied to microcontrollers (Sakti, 2014). In classical logic it is stated that everything is binary, which means that it only has two possibilities, "Yes or No". However, in Fuzzy logic the possibility of membership value is between 0 and 1. That is, it is possible that a situation can have two values "Yes and No" simultaneously, but the value depends on the membership weight it has. Development of the Fuzzy control program is carried out on a laptop through the Arduino IDE software, then uploaded to the NodeMCU ESP 32.

The first stage in this Fuzzy control is the fuzzification stage. At this stage, a membership function is created by determining the input and output crisp values. The crisp input used is the value of soil humidity and soil pH while the crisp output used is the relay. The program listing for this process is shown in Figure 6.

```
//FuzzyInput
pH = new FuzzyInput (2) ;
//FuzzySet
asam = new FuzzySet (0, 0, 4, 6) ;
pH->addFuzzySet (asam) ;
//FuzzySet
normal = new FuzzySet (5, 6, 7, 8,) ;
pH->addFuzzySet (normal) ;
//FuzzySet
basa = new FuzzySet (7, 8, 14, 14) ;
pH->addFuzzySet (basa) ;
fuzzy->addFuzzyInput (pH) ;

  //FuzzyInput
humidity = new FuzzyInput (1) ;
//FuzzySet
kering = new FuzzySet (0, 0, 45, 60) ;
Humidity->addFuzzySet (kering) ;
//FuzzySet
lembab = new FuzzySet (45, 60, 70, 80 ;
Humidity->addFuzzySet (lembab) ;
//FuzzySet
basah = new FuzzySet (70, 85, 100, 100) ;
Humidity->addFuzzySet (basah) ;
Fuzzy->addFuzzyInput (humidity) ;
```

```
//FuzzyOutput
relay = new FuzzyOutput (1) ;
//FuzzySet
ON = new FuzzySet (0, 2.5, 5) ;
Relay->addFuzzySet (ON) ;
//FuzzySet
OFF = new FuzzySet (5, 7.5, 10) ;
Relay->addFuzzySet (OFF) ;
Fuzzy->addfuzzyOutput (relay) ;
```

Figure 6. Program listing

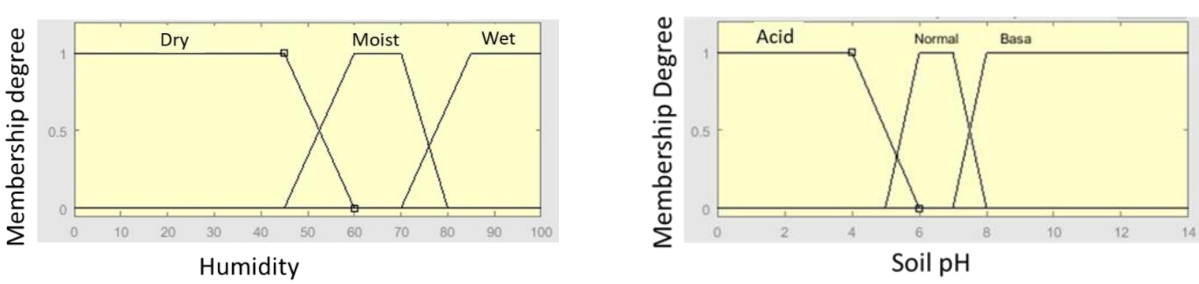

Figure 7. Membership function of soil humidity (left) and soil pH (right)

The soil humidity parameter, the humidity value obtained from the sensor is between 0 and 100 and is classified into three categories: dry, moist, and wet. The membership function graphs of the three categories are shown in Figure 7. Each of humidity value has a membership degree value to each category with a membership degree between 0 and 1. For example, a 40% humidity value has a membership degree of 1 for the dry category and a membership degree of 0 for the moist category. Figure 7 also shows the membership function graph for the soil pH parameter. The soil pH value obtained from the sensor is between 0 and 14. This value is then converted into Fuzzy input sets that are classified into three categories, namely acid, normal and base.

The membership function of the relay output signal also has a graph as shown in Figure 8. The relay output signal shows crisp values from 0 to 10. These crisp values are obtained to control the ON/OFF of the relay. It can be seen that the relay will be ON when the crisp value is from 0 to 5, and will be OFF for crisp values 5 to 10.

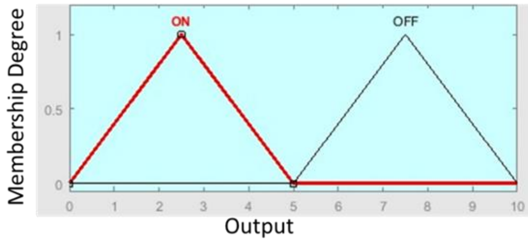

Figure 8. Membership function of relay response as an output

The next stage is the formation of basic Fuzzy rules. In this research, Mamdani Fuzzy logic is used which uses If-Then rules. The Fuzzy rules used include the following:

1. If (soil pH is acid) and (humidity is dry) then (relay is ON)
2. If (soil pH is acid) and (humidity is moist) then (relay is OFF)
3. If (soil pH is acid) and (humidity is wet) then (relay is OFF)
4. If (soil pH is normal) and (humidity is dry) then (relay is ON)
5. If (soil pH is normal) and (humidity is moist) then (relay is OFF)
6. If (soil pH is normal) and (humidity is wet) then (the relay is OFF)
7. If (soil pH is base) and (humidity is dry) then (relay is ON)
8. If (soil pH is base) and (humidity is moist) then (relay is OFF)
9. If (soil pH is base) and (humidity is wet) then (relay is OFF)

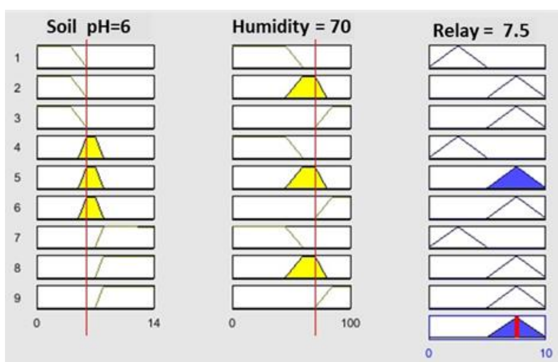

Figure 9. The inference engine

The next step is to convert the Fuzzy input value into a Fuzzy output according to the predetermined rules using an inference engine. The results of the inference engine are shown in Figure 9. The inference engine processes the implication function. For example, in Figure 7, it can be seen that the soil humidity value is normal, which is 60%, and the soil pH value is also normal, which is 6. Based on the value of the degree of membership of both to the normal category, the value is then processed with the Mamdani implication function based on the existing Fuzzy rules. Thus, as the final result, the relay value output will be 7.5, which means the relay is OFF. To analyze the fuzzy control performance data, 20 tests were carried out with different input conditions and then observed the output of the resulting inference engine (as shown in Fig. 9) and calculated the percentage of accuracy. From the test results, it is found that all outputs generated are in accordance with the given input, or the accuracy of the control system is 100%.

3.2. Experimental Results of Hardware Circuits

After all the supporting components are assembled, testing of each sub-system is carried out to determine whether the system is functioning as expected. As explained above, the sensors are connected to the NodeMCU through the Arduino IDE software. One example of the program listing connecting the soil pH sensor with the NodeMCU microprocessor is shown in Figure 10.

```
//ph sensor
sensorValue = analogRead(analogInPin);
outputValue = (-0.0693*sensorValue)+7.3855;
Serial.print("sensor ADC= ");
Serial.print(sensorValue);
Serial.print("  output Ph= ");
Serial.println(outputValue);
```

Figure 10. Program listing of the connection between soil pH sensor and NodeMCU

To determine the performance of the sensor, soil humidity and pH data were collected. Calibration of the sensors is also carried out to validate the measurement results. In this process, the readings of the YL-69 humidity sensor and soil pH sensor were compared with the Soil Tester DM-5 as a reference, and then the difference between the two readings was calculated as shown in Table 1. From 10 test data during the day, an average difference of 0.6% was obtained for humidity and 0.11 for soil pH.

Table 1. Comparison of measurement results between sensor and reference device

No.	Humidity (%)			Soil pH		
	Sensor	Reference device	Difference	Sensor	Reference device	Difference
1	68	67	1	6	6.2	0.2
2	68	68	0	6	6.2	0.2
3	67	67	0	6	6	0
4	68	67	1	6.5	6.5	0
5	67	68	1	6.5	6.2	0.3
6	67	67	0	6	6.2	0.2
7	66	67	1	6.3	6.4	0.1
8	67	68	1	6	6	0
9	68	68	0	6	6	0
10	68	67	1	6	6.1	0.1
	Average		0.6	Average		0.11

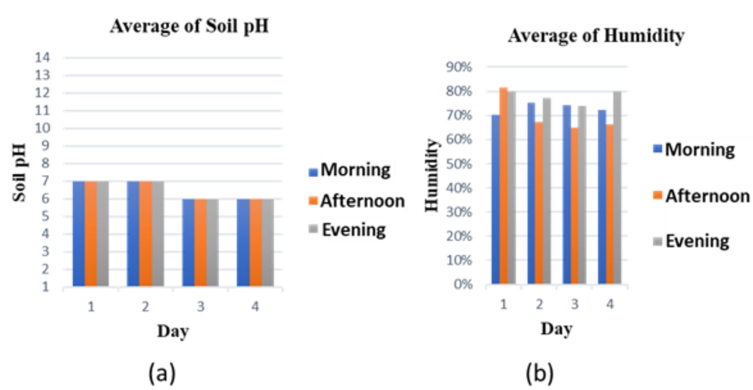

Figure 11. Average of measurement results of (a) soil pH and (b) soil humidity for 4 days

Then data collection was carried out for 4 consecutive days in the morning, afternoon, and evening. Figure 11 below shows the average graph of soil pH and soil humidity measurement results to prove that the sensor can work properly. The sensor sends data continuously every 2 seconds so that the average value of all sensor readings is displayed at a certain time. The soil humidity measurement data shows that during the day the humidity decreases, this is in accordance with the theory that temperature and humidity are inversely proportional, if the temperature is high then the humidity is low, and vice versa. Soil humidity decreases during the day due to the addition of water vapor from evapotranspiration from the surface. The two measurements data above show that the sensor sub-system has worked properly. Furthermore, system testing was carried out which included integration between sensor performance, Fuzzy control system and relay response of the watering section. For this test, data was also collected for four days in the morning, afternoon, and evening. Then the relay response was observed for each measured humidity value. The test results on the fourth day are shown in Table 2.

The values listed in Table 2 are the time and soil humidity during the morning, afternoon, and evening periods. Each period is displayed in a different color (blue for morning, yellow for afternoon, and green for evening. In this system, the relay will turn ON when the soil humidity is less than 60% and the relay will continue to turn ON until the soil humidity reaches 80%, as shown in data no. 25 and 26 in Table 2 above. The accuracy of the system is calculated based on the percentage of error that occurs. Because from the experiment results it is found that the relay provides a response that is in accordance with the soil condition in all data or there is no error, the accuracy of the relay response is 100%. The results of this test show that the sensor circuit, Fuzzy control, and relay watering section have been integrated and function properly.

Table 2. Experiment results of relay response

No.	Time	Humidity	*Relay*	Mark
1	7.50.18	76%	OFF	Morning
2	8.00.34	76%	OFF	
3	8.10.59	76%	OFF	
4	8.20.24	76%	OFF	
5	8.30.02	76%	OFF	
6	8.40.15	76%	OFF	
7	8.50.25	76%	OFF	
8	9.00.33	76%	OFF	
9	9.10.38	76%	OFF	
10	9.20.52	75%	OFF	
11	9.30.53	75%	OFF	
12	9.40.28	75%	OFF	
13	9.50.06	74%	OFF	
14	10.00.17	74%	OFF	
15	10.10.03	74%	OFF	
16	10.20.38	74%	OFF	
17	12.00.31	68%	OFF	Afternoon
18	12.10.05	68%	OFF	
19	12.20.53	68%	OFF	
20	12.30.44	67%	OFF	
21	12.40.42	67%	OFF	
22	12.50.15	67%	OFF	
23	13.00.07	66%	OFF	
24	16.00.40	60%	OFF	Evening
25	16.10.35	59%	ON	
26	16.20.38	80%	ON	
27	16.30.33	82%	OFF	
28	16.40.43	82%	OFF	
29	16.50.47	82%	OFF	
30	17.00.06	82%	OFF	
31	17.10.52	82%	OFF	
32	17.20.08	82%	OFF	
33	17.30.25	82%	OFF	

Note: Success rate of the Relay activity: 100 %

3.3. Experimental Results of Sending Data to Telegram

Telegram is programmed with various commands to carry out a series of instructions given. The display of the

Telegram application is shown in Figure 12. In the figure, it can be seen that the Telegram successfully sends a notification when the module is ready to use and also successfully displays a description of the value of the measurement results in the form of the value of humidity, soil pH, and water capacity. The water capacity referred to here is the water capacity in the reservoir as part of the watering system.

Figure 12. Display on Telegram application

Furthermore, testing was carried out to determine the delay time for sending data from the NodeMCU ESP32 to Telegram. The test results are shown in Table 3. It can be seen that the average delay time is 14.1 seconds. This delay time is influenced by the speed of the provider of the internet network used. In this test, XL provider is used. Furthermore, testing of the system as a whole was carried out, including integration between sensor performance, Fuzzy control system, watering part relays, and data transmission to the internet. The tests were conducted during the day on different days. The data of the integration test results are shown in Table 4, for 10 test data. The resulting data shows that the measurement results of the humidity value and pH value are in accordance with the resulting pump response. Therefore, it can be seen that the percentage of success of this research is 100%. The results of this study are better when compared to similar systems that use Artificial Neural Networks which have a success rate of 90% (Afifah *et al*., 2020).

Table. 3 Experiment results of connection to Telegram

Experiment to:	Delay time (s)	Information
1	18.3	connected
2	13.1	connected
3	11.4	connected
4	16.9	connected
5	18.6	connected
6	12.6	connected
7	10.1	connected
8	12.2	connected
9	14.8	connected
10	15.6	connected
11	14.5	connected
12	11.1	connected
13	12.7	connected
14	16.9	connected
15	12.7	connected
Average	14.1	

Table 4. Experimental results of the integration between hardware parts, fuzzy control, and Telegram response.

Experiment No.	Humidity	Soil pH	Pump Response	Telegram Response	Information
1	58%	6	ON	Module is ready to use	Suitable
2	58%	6	ON	Module is ready to use	Suitable
3	57%	6	ON	Module is ready to use	Suitable
4	59%	6	ON	Module is ready to use	Suitable
5	58%	6	ON	Module is ready to use	Suitable
6	57%	6	ON	Module is ready to use	Suitable
7	59%	6	ON	Module is ready to use	Suitable
8	59%	6	ON	Module is ready to use	Suitable
9	58%	6	ON	Module is ready to use	Suitable
10	54%	6	ON	Module is ready to use	Suitable
Successful integration of sensor, pump response, and telegram response = 100 %					

These experiments have successfully confirmed that the system that has been made and all electronic modules used have been integrated and function properly. The value of the sensor measurement results has also been successfully displayed through the telegram application, with an average delay time of 14.1 seconds, so that users can monitor the condition of the watering system remotely.

4. CONCLUSIONS

From the results of design and testing, it is concluded that this automatic tomato plant watering system is successfully realized and all sub-systems can work with a 100% success presentation. Fuzzy logic was successfully applied as part of the relay control system with 100% success. The system also successfully sends data through the Telegram application with an average delay time of 14.1 seconds. In the future, the overall system will be better if it is equipped with an automatic soil pH control system.

ACKNOWLEDGMENT

The authors would like to thank the Electronics Laboratory, Department of Electrical Engineering, University of Lampung for facilitating this research

References

Afifah, N.N. Pangaribuan, P. & Priramadhi, R.A. (2020). Sistem pengontrolan pengairan budidaya tanaman tomat berdasarkan kelengasan dan suhu tanah berbasis artificial intelligence. *Proceeding of Engineering*, ***7***(3), 8791-8801.

Alfonsius, E., Kalengkongan, W., & Ngangi, S.C.W. (2024). Sistem monitoring dan kontroling prototipe penyiram tanaman otomatis berbasis IoT (Internet of Things). *Jurnal Teknoinfo*, ***18***(1), 44-55

Cobantoro, A.F., Setyawan, M.B., & Wibowo, M.A.B. (2019). Otomasi greenhouse berbasis mikrokomputer raspberry Phi. *Jurnal Ilmiah Teknologi Informasi Asia*, 13(2), 115-124.

Darmawan, I.G.E., Yadie, E., & Subagyo, H. (2020). Rancang bangun alat ukur kelengasan tanah berbasis Arduino Uno. *Jurnal PoliGrid*, ***1***(1), 31-38. https://doi.org/10.46964/poligrid.v1i1.215

Djumali, D., & Mulyaningsih, S. (2014). Pengaruh kelengasan tanah terhadap karakter agronomi, hasil rajangan kering dan kadar nikotin tembakau (*Nicotiana Tabacum* L; *Solanaceae*) Temanggung pada tiga jenis tanah. *Jurnal Berita Biologi*, ***13***(1), 1-11.

Fauzia, N., Kholis, N., & Wardana, H.K. (2021). Otomatisasi penyiraman tanaman cabai dan tomat berbasis IoT. *Jurnal Reaktom*, ***6***(1), 22-28.

Imam, M.K., Permata, E., & Desmira, D. (2022). Sistem kontrol penyiram otomatis tanaman tomat menggunakan Wemos D1R1. *Elkomika*: *Jurnal Teknik Energi Elektrik*, *Teknik Telekomunikasi*, & *Teknik Elektronika*, ***10***(4), 815-829. https://doi.org/10.26760/elkomika.v10i4.815

Marinus, F., Yulianti, B., & Haryanti, M. (2020). Rancang bangun sistem penyiraman tanaman berdasarkan waktu menggunakan RTC berbasis arduino uno pada tanaman tomat. *Jurnal Teknik Industri*, ***9***(1), 78-89.

Nadhiroh, N., Wardhany, A.K., Setiana, H., Renaldy, R., Putri, A.A., & Handayani, M.D. (2024). Penyiram tanaman hidroponik otomatis berbasis IoT dengan PLC Outseal dan ESP 32. *Electrices*: *Jurnal Otomasi Kelistrikan dan Energi Terbarukan*, ***6***(1), 17-26.

Nadindra, D.E., & Chandra, J.C. (2022). Sistem IoT penyiram tanaman otomatis berbasis Arduino dengan kontrol Telegram. *SKANIKA: Sistem Komputer dan Teknik Informatika*, ***5***(1), 104-114. https://doi.org/10.36080/skanika.v5i1.2887

de Oliveira, J.C., Santos, D.H., & Neto, M.P. (2016). Chatting with Arduino platform through Telegram Bot. *Proceedings of the International Symposium on Consumer Electronics*, ***1***(1), 131-132. https://doi.org/10.1109/ISCE.2016.7797406

Philander, K., Suppa, R., & Muhallim, M. (2021). Sistem penyiraman tanaman otomatis berbasis Arduino. *Jurnal Riset Sistem Informasi dan Teknik Informatika (JURASIK)*, ***1***(1), 1-8. http://dx.doi.org/10.30645/jurasik.v6i1.266

Prasetyo, E.A. (2019). Memulai Pemrograman ESP32 Menggunakan Arduino IDE. Accessed on 21 June 2024 from: https://www.arduinoindonesia.id/2019/07/memulai-pemrograman-esp32-menggunakan.html#

Sakti, I. (2014). Methodology of fuzzy logic with mamdani models applied to the microcontroller. *Procc. of the 1st International Conference on Information Technology, Computer and Electrical Engineering (ICITACEE)*, 93-98.

Sasmoko, D., & Horman, R. (2020). Sistem monitoring aliran air dan penyiraman otomatis pada rumah kaca berbasis IoT dengan Esp8266 dan Blynk. *Jurnal Ilmiah Pendidikan Teknik Elektro*, ***1***(1), 1-10. http://dx.doi.org/10.22373/crc.v4i1.6128

Setyowati, I., Novianto, D., & Purnomo, E. (2020). Preliminary design and soil humidity sensor YL-69 calibration for implementation of smart irrigation. *Journal of Physics: Conference Series*, ***1517***, 012078. https://dx.doi.org/10.1088/1742-6596/1517/1/012078

Sirait, R., & Botiwicaksono, C. (2020). Kontrol kelengasan pada tanaman tomat menggunakan PID. *Jurnal Techno*, ***3***(1), 262-273. https://doi.org/10.33633/tc.v19i3.3668

Sulardy, S., & Sany, T.A.M. (2018). Uji pemberian limbah padat pabrik kopi dan urine kambing terhadap pertumbuhan dan produksi tanaman tomat. *Journal of Animal Science and Agronomy Panca Budi*, ***3***(2), 7-13.

Yahwe, C.P., Isnawaty, I., & Aksara, L.M.F. (2016). Rancang bangun prototype system monitoring kelengasan tanah melalui sms berdasarkan hasil penyiraman tanaman - Studi kasus tanaman cabai dan tomat. *Jurnal SemanTIK*, ***2***(1), 97-110.

Yoseph, S., Adi, P.D.P., & Nachrowie, N. (2021). Implementation of fuzzy logic on internet of things-based greenhouse. *IOTA*, ***1***(2), 100-113. https://doi.org/10.31763/iota.v1i2.489

Yudhana, A., & Putra, M.C.F. (2016). Penyiram tanaman otomatis berbasis informasi sinyal sensor kelengasan. *Prosiding Annual Research Seminar*, ***2***(1), 277-280.

Zulfikar, M. (2018). Perancangan sistem penyiraman tanaman otomatis berabasis mikrokontroler Atmega328. *Journal of Informatics and Computer Science*, ***4***(1), 75-90. https://doi.org/10.33143/jics.Vol4.Iss1.533