AN IOT-BASED AUTOMATED WATERING SYSTEM FOR PLANTS USING INTEGRATED FUZZY LOGIC AND TELEGRAM BOT

Muhammad Raihan Anindita[1)], Ary Mazharuddin Shiddiqi[2)], Wahyu Suadi[3)], Rully Soelaiman[4)], Suhadi Lili[5)], and Ilham Gurat Adillion[6)]
[1,2,3,4,5,6)]Department of Informatics. Institut Teknologi Sepuluh Nopember, Surabaya, East Java, Indoensia 60111
e-mail: muhammad.raihan1617@gmail.com[1)], ary.shiddiqi@if.its.ac.id[2)]

ABSTRACT

The development of automatic plant watering systems has recently gained popularity due to the need to conserve water and ensure healthy plant growth. This study focuses on integrating fuzzy logic, sensors, and algorithms to provide an automatic watering system. Fuzzy logic is a powerful tool that allows the system to interpret sensor data and make informed decisions. The sensors measure soil moisture, humidity, temperature, and light intensity. The data collected from these sensors is analyzed using algorithms to determine the appropriate watering schedule. The system's ability to analyze and interpret data ensures that the plants receive the necessary moisture without over-watering or under-watering. Integrating the Telegram Bot is a significant feature of the system, enabling users to monitor and control the system remotely. The Telegram Bot sends users notifications when the system is activated, or the plants require attention. The system can also be controlled remotely through the Bot, enabling users to adjust the watering schedule or turn the system on or off. This research shows that the designed features of the system function effectively and can be used on a daily household scale. The system's automated features reduce the need for constant monitoring and manual watering, making it ideal for those who engage in gardening at home. This innovation is particularly relevant in increasing the productivity of plants. In addition, the system's ability to be controlled remotely through the Telegram Bot is a significant advantage, making it accessible and convenient for users.

***Keywords**: Automatic watering system, Fuzzy Logic, Internet of Things, Telegram Bot.*

I. INTRODUCTION

WATERING plants is one of the tasks that needs to be done routinely and regularly to maintain optimal plant growth [1] [2] [3]. However, this process often poses several challenges, such as determining the quantity of water required for each watering session, which can cause plants to be over or under-watered, leading to plant decay and death. The problem becomes more complex when dealing with plants requiring specific and intensive care, such as grape plants requiring precise and appropriate watering patterns [4]. One potential solution to this problem is implementing an automated watering system, which can replace conventional watering methods and ensure even watering as needed with appropriate duration.

Several studies have been conducted in [5] utilizing fuzzy logic method in a system based on two conditions, namely air (temperature) and soil (humidity) conditions. This research resulted in temperature conditions with an average error of 0.41% and humidity conditions with an average error of 2.3%. Another study was also conducted by applying the fuzzy logic method in a system with inputs of soil moisture and air temperature [6]. The research utilized Raspberry Pi 4 as a means of processing inputs with fuzzy logic, producing output in the form of a water pump and fan duration. The study resulted in an average success rate of 80% in 10 experiments [7]. Another study on plant watering systems utilized the Telegram Bot as a means of control. This research utilized soil moisture and air temperature sensors, resulting in a watering system that sends and receives commands from the Telegram Bot [8]. While the studies have contributed to the development of automated plant watering systems using fuzzy logic and IoT, some gaps in the research still need to be addressed. Although many research had used fuzzy logic to determine watering needs, further studies are needed to fully explore the impact of more integrated critical factors, such as soil humidity and temperature sensors, air humidity and temperature sensors, and light intensity levels, on plant growth and water requirements. We can develop a more comprehensive understanding of plant watering needs by utilizing multiple sensors to collect and analyze data. This method can improve automated watering systems' effectiveness and better plant growth outcomes.

Our study focuses on developing an automatic watering system for grapes using Raspberry Pi and Arduino that incorporates fuzzy logic from the scikit-fuzzy library. To determine the necessary watering duration, we utilize data from three sensors: a soil moisture sensor, a temperature and humidity sensor, and a light intensity sensor. Our system also leverages the pyTelegramBotAPI library and Telegram Bot feature for mobile monitoring and control. In order to make better decisions about plant watering, we integrate knowledge and expertise from farmers with the fuzzy logic system. We aim to design a user-friendly, accurate, and environmentally sustainable automated plant watering system by incorporating multiple sensors, a Telegram bot, and agricultural best practices.

II. Literature Review

We discuss the importance of proper plant watering, the advantages of using fuzzy logic in plant watering automation, and the benefits of using Raspberry Pi as a microcontroller. We will examine the existing literature in these areas, identify the research gaps and challenges, and suggest possible directions for future research.

A. The Characteristic of Grapes

Grapes are one of the plants that grow in lowland areas. Unlike most other plants, grapes actually require a long dry season of about 4-7 months and a sufficiently high intensity of sunlight [9]. The required rainfall for this plant is only 800 mm per year. Therefore, excessive watering can disrupt the fruiting process. The maximum growing temperature is 31^{o}C, and the minimum temperature is 23^{o}C, with air humidity ranging from 75-80% RH [10]. A thin soil surface is also not good for grape growth. Grape roots grow well in soil with high permeability, moderate humidity, and high oxygen content. The water content of the soil is preferably within the range of 60%-80% of the maximum capacity the soil can hold. Water below 30% in the soil can cause grape plants to stop growing.

B. Automated Watering System

Automated watering systems can be developed using hardware and software tools, such as Raspberry Pi [11] and Arduino [14] [15], along with fuzzy logic. These systems can help automate plant watering by monitoring multiple environmental factors, such as soil moisture, air temperature, and light intensity, and adjusting watering schedules accordingly. Raspberry Pi and Arduino are popular single-board computers to control various sensors and actuators, making them ideal for creating automated watering systems applications [6] [12] [13]. With these devices, we can connect sensors that measure environmental factors such as soil moisture, temperature, and light intensity and use actuators to control the watering process. For example, we can program the system to turn on a water pump or open a valve when the soil moisture levels drop below a certain threshold.

Fuzzy logic is an approach to computing that is well-suited to modeling complex systems like plant growth and watering needs due to its ability to handle imprecision and uncertainty [16]. With fuzzy logic, we can develop rules based on data from multiple sensors to determine when and how much to water plants. This approach can also be used to determine each plant's optimal amount and frequency of watering. For instance, some researchers have used fuzzy logic to develop systems that adjust water usage based on real-time soil moisture measurements. Other studies have employed fuzzy logic to develop irrigation systems that account for weather conditions, plant type, and soil type [17].

Telegram Bot is one way to monitor and control an automated watering system next to a web interface or mobile app. These interfaces offer real-time updates on the system's status, enabling users to adjust watering schedules or settings and receive notifications when the system requires attention. Third-party developers can also customize the Telegram Bot feature to fit users' specific needs. Additionally, Telegram Bots can be integrated with various external services [8] [18].

C. State of the art

While plant watering systems have come a long way in recent years, some research gaps still need to be addressed. Limited integration, limited optimization, and limited validation are three areas that require further attention. Some issues that have not been addressed are the need to explore the impact of other factors such as light intensity and humidity levels on plant growth and water requirements, and the need to use multiple sensors for more accurate data collection and analysis. In addition, the use of an automated bot to retrieve instructions and publish information is important to provide interactive interactions between an IoT system and a user. Addressing these gaps helps create more effective and efficient plant watering systems that can contribute to sustainable agriculture practices.

III. Methodology

This section outlines the approach used to develop our automatic watering system using fuzzy logic. We describe the architecture system, the design of the fuzzy logic, the sensors used and the Telegram Bot as the interface to users.

A. System Architecture

The developed automatic plant watering system uses a Raspberry Pi 4 model B as its main processing unit. To retrieve data from the sensors, the system uses an Arduino UNO R3 as an intermediary between the Raspberry Pi and the sensors (Figure 1). Upon receiving a request from the Raspberry Pi, the Arduino turns on the three sensors to retrieve data and then encapsulates the data in JSON format to be sent back to the Raspberry Pi for further processing [19]. By using multiple sensors to monitor various aspects of the plant's environment, the system can provide a comprehensive overview of the plant's condition. This data is then used to automatically adjust the watering schedule, ensuring that the plant receives the appropriate amount of water and nutrients for optimal growth and health. The system uses three sensors with the following functions:

1) The first sensor is a soil moisture sensor that measures the moisture content of the soil. This sensor is essential as it helps the system to determine whether the soil is too dry or too moist and then adjust the watering schedule accordingly. By monitoring soil moisture levels, the system can prevent under or over-watering, which can have detrimental effects on the plant's health.
2) The second sensor used in the system is a temperature and humidity sensor. This sensor measures the temperature and humidity levels of the plant's immediate surroundings. These environmental factors can have a significant impact on the plant's growth and development. By monitoring the temperature and humidity levels, the system can adjust the watering schedule to ensure optimal growing conditions for the plant.
3) The third sensor used in the system is a light sensor that measures the intensity of light. Plants require a certain amount of light to carry out photosynthesis, which is essential for their growth and development. This sensor ensures that the plant receives the appropriate amount of light, and the system can adjust the watering schedule accordingly based on the light levels.

The program is designed to act as a message handler for Telegram through the use of Telegram Bot. The pyTelegramBotAPI library is employed to request sensor data from the connected Arduino, perform calculations using fuzzy logic provided by the Skicit-Fuzzy library, and control the relay through the Raspberry Pi's GPIO. A MySQL database is utilized to store Telegram token data and group IDs and to log data from manual and automatic irrigation modes.

B. *Designing Fuzzy logic*

To determine the necessary watering duration, we first conducted a thorough literature review on grapevine plants to identify relevant factors that could affect their growth and water requirements. We then used this information to develop a set of fuzzy membership functions based on these factors, which were used as inputs and outputs for our automated watering system. The fuzzy membership set functions and fuzzy rules used in fuzzy logic for automatic watering systems are set as follows:

1) Logic Membership Set Fuzzy: Fuzzy logic requires a membership function set so that the data obtained can be processed to get fuzzy output results. In this study, several fuzzy membership sets are used as input and output based on sources in the literature review of the vines [5] [20].
2) Fuzzy Logic rules: The fuzzy rules used in the automatic watering system are developed based on the author’s direct observations in the field. The following is a list of fuzzy rules that are used as rule bases in the fuzzy logic process (Table I) using the relation “AND” and Table V based on literatures in [21].

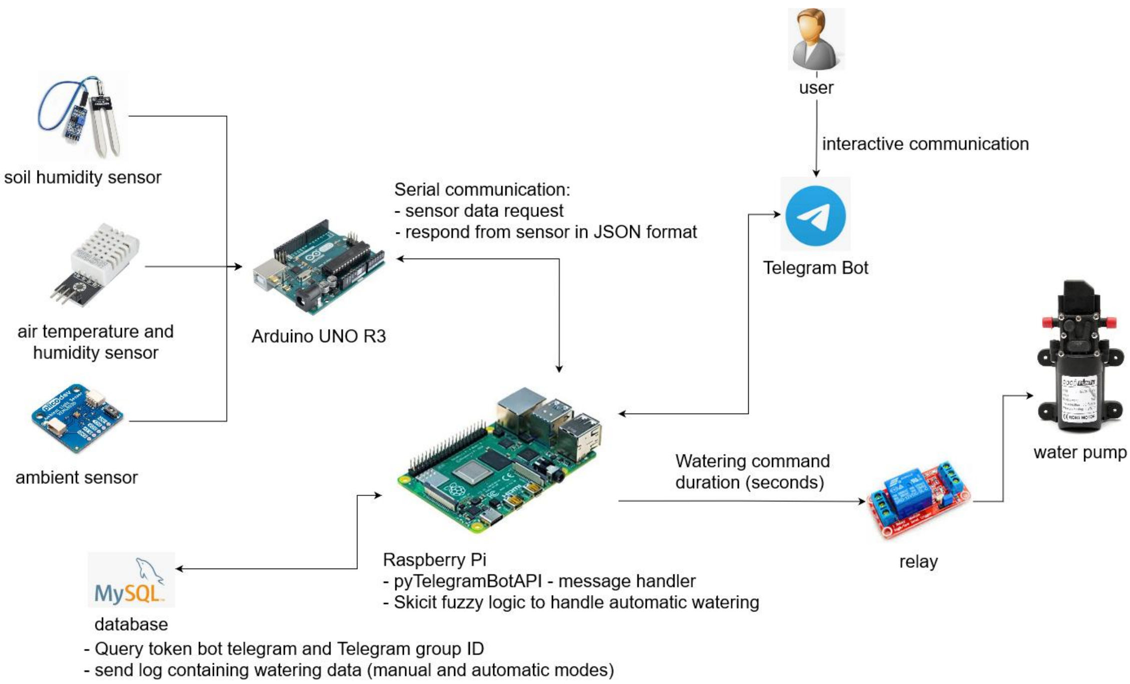

Fig. 1. System architecture

TABLE I
DURATION BASED ON AIR TEMPERATURE AND SOIL HUMIDITY CONDITION.

Air		Soil Humidity	Watering Duration		
Temperature	**Humidity**				
Hot	Wet	Wet	Medium	Medium	Short
Warm	Wet	Wet	Short	Short	Short
Cold	Wet	Wet	Short	Short	Short
Hot	Humid	Wet	Long	Long	Medium
Warm	Humid	Wet	Medium	Medium	Short
Cold	Humid	Wet	Short	Short	Short
Hot	Wet	Humid	Long	Long	Medium
Warm	Wet	Humid	Medium	Medium	Short
Cold	Wet	Humid	Short	Short	Short
Hot	Humid	Humid	Long	Long	Long
Warm	Humid	Humid	Long	Medium	Medium
Cold	Humid	Humid	Medium	Medium	Medium

C. Automatic watering system algorithm

The automatic watering system (Algorithm 1) starts by initializing a database connection and sending a message to subscribers that the system is active. Then, sensor data including soil moisture, temperature, air humidity, and light sensors are fetched. The data is informed to subscribers and used to calculate the watering duration. A control system created using fuzzy rules defined in Table I and Table II. If the calculated duration is greater than zero, the system sends a message to subscribers about the duration of watering, turns on the pump using the relayOn() function, and starts a relayTimer to monitor the watering duration. The watering data is stored into the database to keep track the watering profile. If the calculated duration is zero or negative, the system sends a message to subscribers that the watering is aborted. The loop continues until the system is stopped.

TABLE II. DURATION BASED ON AMBIENCE

Light intensity	Duration
Bright	Long
Dimmed	Medium
Dark	Short
Bright and Dimmed	Medium
Dimmed and Dark	Short

Algorithm 1. Automatic watering based on fuzzy logic

```
Initialize database connection
Send a message to subscribers that automatic watering system is active.
Get the current time and date and store it in the variable sTime.

while TRUE do
    // fetching and storing sensor data
    fetch data from the sensor (soil moisture, temperature, air humidity, and light sensors)
    store the returned data in variables (s1, sn1, s2, sn2, tmp, hmd, and lght)

    // inform sensor data to subscribers
    Send a message (sMsg) containing sensor data and the current time
    Send a message about duration of watering is being calculated

    // calculating watering duration from fuzzy rules
    Create a control system using the fuzzy rules and store it in the variable drtn_ctrl.
    Set the input values from sensor data

    // executing watering
    Get the current time and store it in the variable eTime
    if (rDuration > 0) then
        Send a message to subscribers about duration of watering
        Call the function relayOn() to turn on the pump
        Start a relayTimer
        Store the log of watering data into the database
    else
        Send a message to subscribers watering is aborted
    end if
end while
```

IV. Experiments

The experiment conducted aimed to evaluate the effectiveness and efficiency of an automatic plant watering system designed. We focus on observing the effectiveness of multiple sensors, fuzzy logic calculations, and a Telegram Bot interface to provide a reliable and accurate means of watering plants.

A. Assembling the equipment

This section describes the implementation of the equipment circuitry for the automatic irrigation system. The main equipment circuitry can be seen in Figure 2. The Raspberry Pi is used as the main component of the system. The Arduino is connected to the Raspberry Pi using a USB cable, which acts as the power supply for the Arduino and the communication path between the two devices. Then, modified LAN cables are connected to each sensor's end and to a breadboard.

Jumper cables are used to connect the sensors to the Arduino. LED and 220 Ohm resistors are placed on the breadboard between the power cable for the sensors (at the bottom of the breadboard) and the Arduino jumper cables that provide power. The left part of the relay is connected using a jumper to the GPIO on the Raspberry Pi to control the on or off condition of the relay. The right part of the relay is connected to a DC power cable for the pump, which is connected to the Normally Open (NO) pin and the Common pin, where the electrical circuit is open when the relay is triggered in a low state or no power.

Fig. 2. The equipment assembly

We placed the air temperature and humidity sensor, as well as the light sensor, outside in the grape plant environment to gather data on the surrounding conditions. The sensors were connected to a breadboard via modified CAT5 LAN cables, which allowed for easy transmission of the data to the system (Figure 3). This approach allowed for accurate monitoring of the environmental factors that are important for grape plant growth and development. The two soil moisture sensors are placed 10 cm deep into the 46 cm diameter planter bag containing grape plants. The sensor's placement is 15 cm away from the center of the grape plant stem. Both sensors are connected using modified CAT5 LAN cables, which are then connected to the breadboard. The two soil moisture sensors are used to compare soil moisture values on two sides. The placement of the two sensors can be seen in Figure 4.

Fig. 3. Placement of temperature-humidity sensors and light sensors

Fig. 4. Placement of soil moisture sensor 1 (left) & 2 (right)

B. Implementing the Fuzzy Logic

Fuzzy Logic relies on membership functions to process data and generate fuzzy output results. In this research, we utilized several fuzzy membership functions as inputs and outputs, which were based on relevant sources in the literature review of grapevine plants. Figure 5 to Figure 8 showcase the fuzzy membership set graphs that we used. These membership functions were generated from the rules outlined in Table I.

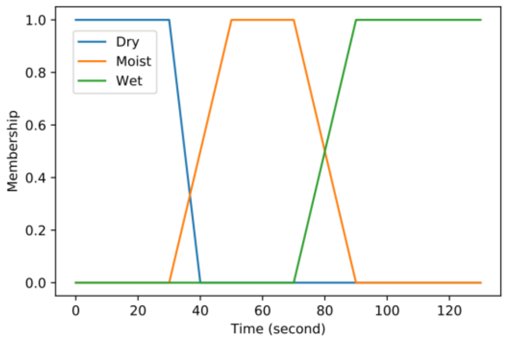

Fig. 5. Fuzzy membership for soil moisture

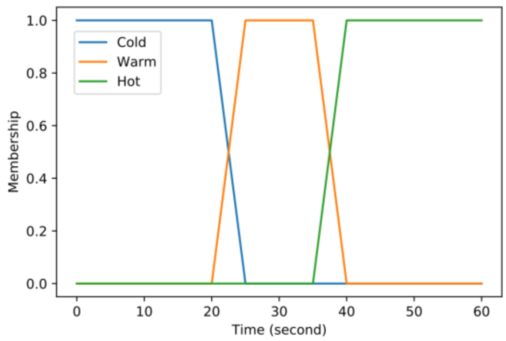

Fig. 6. Fuzzy membership for temperature

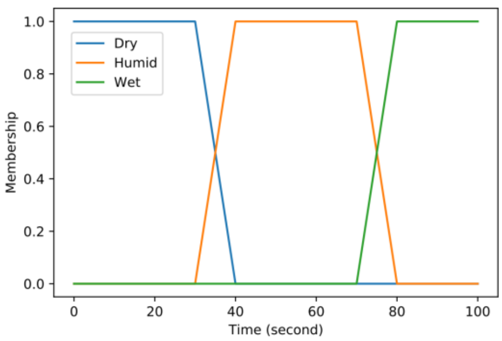

Fig. 7. Fuzzy membership for air moisture

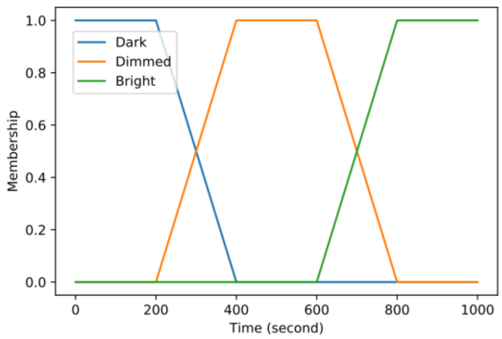

Fig. 8. Fuzzy membership for ambience

The maximum watering duration (Figure 9) is obtained by calculating the volume of water that comes out through the nozzle of the watering hose. The usual amount of water given to grapevine plants in a conventional way is 2.5 liter per day. Experiment was carried out to obtain the maximum watering duration by turning on the pump until the water collected in the container reaches 2.5 liters. The maximum watering duration was found to be approximately 3 minutes (180 seconds).

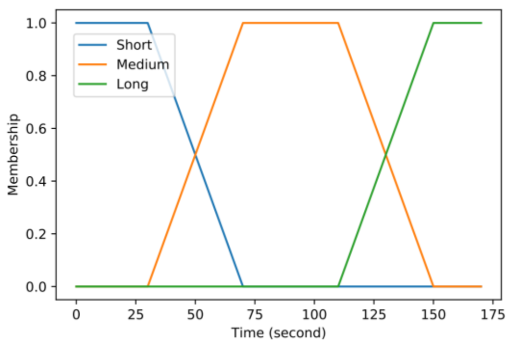

Fig. 9. Fuzzy membership for watering duration

C. *Functionality experiments*

Functional experiments are designed to examine the functionality and reliability of various system components, aiming to improve the efficiency and effectiveness of automated plant watering. Several scenarios are designed for the experiments:

1) The first experiment involves testing the functionality of an automatic watering bot. This bot is designed to water plants automatically, eliminating manual watering.
2) The second experiment involves testing the commands that start or help commonly used in messaging applications to initiate conversations and request assistance.
3) The third experiment involves testing a command and sub-command combination (/pump) that is designed to activate a specific watering pump.
4) The fourth and fifth experiments involve testing the functionality of a sensor command designed to retrieve data from various sensors installed in the system.

5) Finally, the last experiment involves testing the automatic watering function of the system.

Experiment results are summarised in Table III. The results of the experiments suggest that the developed system successfully meets the required functionality and is ready for implementation. The automatic watering bot has been tested and proven effective in automating the watering process, reducing the need for manual intervention. Furthermore, the messaging commands for initiating conversations and requesting assistance were functional, and the sub-command combination successfully activated the desired pump. The sensor commands designed to retrieve data from various sensors installed in the system were found to be functional, adding to the system's reliability. Finally, the automatic watering function of the system was tested and found to be effective. These experiments indicate that the developed system is reliable and meets the requirements for automated plant watering, making it a valuable addition to any plant maintenance system.

Table III. Experiment results

ID	Description	Scenario	Action Performed
UF-001	Activating automatic watering bot	Scenario 1	Yes
		Scenario 2	Yes
UF-002	Executing Command /start and /help	Scenario 3	Yes
		Scenario 4	Yes
UF-003	Executing Command and Sub Command /pump	Scenario 5	Yes
		Scenario 6	Yes
		Scenario 7	Yes
		Scenario 8	Yes
		Scenario 9	Yes
UF-004	Experiment Command /sensor	Scenario 10	Yes
UF-005	Executing Command and Sub Command /mode	Scenario 11	Yes
		Scenario 12	Yes
		Scenario 13	Yes
		Scenario 14	Yes
		Scenario 15	Yes
		Scenario 16	Yes
UF-006	Executing Automatic Watering	Scenario 17	Yes
		Scenario 18	Yes

V. DISCUSSION

Automatic plant watering systems have become increasingly popular in recent years as people look for ways to reduce their water consumption and maintain healthy plants. This research focuses on the development of such a system using multiple sensors, an Arduino Uno microcontroller, a Raspberry Pi mini-computer, and a chatbot for user interaction. The system uses air humidity, soil moisture, and temperature sensors to monitor the plant's environment and determine when watering is necessary. The Arduino Uno microcontroller is responsible for receiving sensor data and controlling the water pump accordingly. The Raspberry Pi minicomputer acts as the brain of the system, providing a platform for data processing and analysis. Additionally, the chatbot allows users to interact with the system, request information about the plant's condition, and receive notifications when watering is needed. Experiment results showed that it effectively maintained optimal soil moisture levels and prevented over or under watering. Furthermore, the use of a chatbot proved to be a convenient way for users to monitor and control the system, making it more user-friendly. With the increasing demand for sustainable and efficient ways of plant care, such a system can offer significant benefits.

We then compare our proposed system to the selected references. We focused on those with the closest scope of work in comparing our proposed system to other references (Table IV). We found that there are seven sensor types used in all the references. All selected references use the soil moisture sensor, indicating that it is an essential component for any watering system, with six using the air temperature sensor. Fuzzy logic was a prevalent feature among the references, with seven projects using it to determine watering logic. Two other research projects with similar systems using four sensors and fuzzy logic for watering were found in references [19] and [21]. However, our system distinguishes itself from these references by using a Telegram bot to manage user engagement, which neither of the other two systems incorporated. Although our proposed system has been proven sound compared to the references, there is still room for improvement. One possible way to enhance it is by incorporating additional sensors our system has yet to utilize. This would provide a more comprehensive understanding of the environment and plants, for instance, by measuring soil temperature, PH level, and nutrient levels.

Table IV. Comparing our proposed system to the selected references

No	Automatic watering system	Sensors							Fuzzy logic	Telegram bot
		Soil moisture	Soil temp.	Air temp.	Air Humidity	Light / ambience	PH Level	Nutrients		
1	Our proposed system	Y	N	Y	Y	Y	N	N	Y	Y
2	An Automated Irrigation and Fertilization management System Using Fuzzy Logic [1]	Y	N	N	N	N	Y	N	Y	N
3	Monitoring and Implementation of Watering System on Farming Robot based on Fuzzy Logic Algorithm [5]	Y	Y	Y	N	N	N	N	Y	N
4	Implementation of Automatic Watering System and Monitoring of Nutrients for Grape Cultivation [10]	Y	Y	N	N	N	N	Y	Y	N
5	Design and Implementation of IoT based Automated Tomato Watering System Using ESP8266 [15]	Y	N	Y	N	N	N	N	N	Y
6	Fuzzy Logic based Smart Irrigation System using Internet of Things [17]	Y	N	Y	N	N	N	N	Y	N
7	Smart Urban Farming (Entrepreneurship through EPICS) [18]	Y	N	N	N	N	N	N	N	Y
8	An Automated Irrigation System: An IoT Application [19]	Y	N	Y	Y	Y	N	N	Y	N
9	Automatic Plant Watering System using IoT [20]	Y	N	N	N	N	N	N	N	N
10	Smart Watering System using Fuzzy Logic [21]	Y	N	Y	Y	Y	N	N	Y	N

VI. Conclusions

We developed an automatic plant watering system by integrating fuzzy logic, sensors, and algorithms implemented in a mini computer. The system was designed to perform watering tasks manually as well as automatically. The fuzzy logic algorithm was utilized to calculate the watering duration based on the data collected from the sensors installed in the soil. The system effectively interprets data from various sensors, such as moisture, humidity, air temperature and humidity, and light intensity, to determine the optimal watering schedule for plants. The integration of the Telegram Bot enables remote monitoring and control of the system, providing users with convenience and accessibility. The results of this research demonstrate that the designed features of the system are effective and suitable for reducing the need for constant monitoring and manual watering. This innovation has the potential to increase plant productivity and offers significant advantages in terms of accessibility and convenience through remote control via the Telegram Bot.

REFERENCES

[1] P. K. Nandi and M. A. Mahmood, "An Automated Irrigation and Fertilization management System Using Fuzzy Logic," in International Conference on Electrical Information and Communication Technology (EICT), Khulna, Bangladesh, 2021.

[2] R. N. Rao and B. Sridhar, "IoT based smart crop-field monitoring and automation irrigation system," in International Conference on Inventive Systems and Control (ICISC), Coimbatore, India, 2018.

[3] G. Nikesh and R. S. Kawitkar, "IoT based smart agriculture," International Journal of advanced research in Computer and Communication Engineering, vol. 5, no. 6, pp. 838-842, 2016.

[4] D. M. Chetan, R. R. Ganesh, S. Jagannathan and R. Priyatharshini, "Smart farming system using sensors for agricultural task automation," in IEEE Technological Innovation in ICT for Agriculture and Rural Development (TIAR), Chennai, India, 2015.

[5] I. G. E. Dirgayussa, Y. Pratama, O. D. Panjaitan and T. A. Prasetyo, "Monitoring and implementation of watering system on farming robot based on fuzzy logic algorithm," in EEE International Conference of Computer Science and Information Technology (ICOSNIKOM), 2022.

[6] J. L. Hatfield and J. H. Prueger, "Temperature extremes: Effect on plant growth and development," Weather and Climate Extremes, vol. 10, pp. 4-10, 2015.

[7] F. Ferdiansyah, Z. Husin, Y. Suprapto and S. Dwijayanti, "Water distribution control using arduino with fuzzy logic algorithm method: A prototype design," in International Conference on Electrical Engineering and Computer Science (ICECOS), 2019.

[8] J. C. de Oliveira, D. H. Santos and M. P. Neto, "Chatting with arduino platform through telegram bot," in IEEE International Symposium on Consumer Electronics (ISCE), 2016.

[9] M. F. Saaid, A. Sanuddin, M. Ali and M. S. A. I. M. Yassin, "Automated pH controller system for hydroponic cultivation," in IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), Langkawi, Malaysia, 2015.

[10] E. P. Wahvu, R. R. Asih, U. R. A. Stania, A. E. Novianti, E. Firmaniar, M. Sarosa and M. Kusumawardani, "Implementation of automatic watering system and monitoring of nutrients for grape cultivation," in 2022 International Conference on Electrical and Information Technology (IEIT), 2022.

[11] J. W. Jolles, "Broad-scale applications of the raspberry pi: A review and guide for biologists," Methods in Ecology and Evolution, vol. 12, no. 9, p. 1562–1579, 2021.

[12] R. Jain, Advanced Home Automation Using Raspberry Pi, Apress Berkeley, CA, 2021.

[13] F. Rozie, I. Syarif and M. U. H. Al Rasyid, "Design and implementation of Intelligent Aquaponics Monitoring System based on IoT," in International Electronics Symposium (IES), Surabaya, Indonesia, 2020.

[14] K. Taneja and S. Bhatia, "Automatic irrigation system using arduino uno," in International Conference on Intelligent Computing and Control Systems (ICICCS), 2017.

[15] R. Nurhasanah, L. Savina, Z. M. Nata and I. Zulkhair, "Design and implementation of iot based automated tomato watering system using esp8266," Journal of Physics: Conference Series, vol. 1898, no. 1, pp. 012 - 041, 2021.

[16] L. Zadeh, "Fuzzy sets," Informationa nd Control, vol. 8, no. 3, p. 338–353, 1965.

[17] R. S. Krishnan, E. G. Julie, Y. H. Robinson, S. Raja, R. Kumar, P. H. Thong and L. H. Son, "Fuzzy Logic based Smart Irrigation System using Internet of Things," Journal of Cleaner Production, vol. 252, p. 119902, 2020.

[18] P. Kullu, S. Majeedullah, P. Pranay and B. Yakub, "Smart Urban Farming (Entrepreneurship through EPICS)," in World Engineering Education Forum, India, 2019.

[19] A. Verma, V. Gupta and A. K. Singh, "An automated irrigation system: An iot application," in International Conference on Intelligent Engineering and Management (ICIEM), 2022.

[20] S. Bhardwaj, S. Dhir and M. Hooda, "Automatic plant watering system using iot," in International Conference on Green Computing and Internet of Things (ICGCIoT), Bangalore, India, 2018.

[21] V. Ramya, A. Kumar, H. Abhishek, A. Kumar and B. M. Anurag, "Smart Watering System using Fuzzy Logic," in International Conference on Design Innovations for 3Cs Compute Communicate Control (ICDI3C), Bangalore, India, 2021.