Jurnal Krisnadana (Jurnal Komputer, Sistem Kendali & Jaringan)
Vol 5, Issue 3, May 2026, pp. 677-688
https://doi.org/10.58982/y56hb388
E-ISSN: 2808-3563
P-ISSN: 2808-3520

# IoT-Based Automatic Watering and Flow Detection System via Telegram

**Ketut Gde Manik Karvana[1*], Ni Ketut Utami Nilawati [2], Evi Dwi Krisna [3]**

[1*,2,3] Institut Bisnis dan Teknologi Indonesia, Denpasar, Indonesia
[1*]*denakarvana@instiki.ac.id* ; [2] *utami.nilawati@instiki.ac.id* ; [3] *evidwikrisna@instiki.ac.id*

ARTICLE INFO

*Article history:*
Received 17 June 2026
Revised 22 July 2026
Accepted 24 July 2026

***Keywords***:
Plants;
Air Watering Systems;
Esp 8266;
Telegram;
Make It Easier

**ABSTRACT**

*Gardens are an important part of the surrounding environment, so they need to be maintained regularly. Manual watering systems are often inefficient. An automatic watering system based on the ESP 8266 was created which offers a more efficient solution, and plants get an adequate water supply. The system will water the plants according to the schedule that has been designed. This system is also equipped with a no-water flow detector to prevent plants from being watered during the watering schedule, and will provide a warning if there is no water flow. This warning is then sent via Telegram Messenger. Of the 20 tests of the automatic watering system, the watering start notification was sent successfully 17 times (80%), while the watering stop notification was successfully sent 18 times (90%). The main obstacle occurs on normal lecture days. Checking via the Wireshark application proved that there were several very high packet loss values (2.8514), which indicates that many data packets were lost during transmission. The water flow meter sensor only detected no water flow on one occasion out of 20 tests. This research shows that an automatic watering system and detection of no water flow can make the work of INSTIKI employees easier.*



## 1. Introduction

Gardens and green spaces are essential components of our surroundings, serving as recreational spaces and crucial elements for improving air and environmental quality [1]. To maintain their beauty and health, regular and efficient plant watering is indispensable [2]. However, daily busy schedules often challenge the consistency of manual watering routines. In this context, the development of automatic watering systems offers a compelling solution, providing significant benefits in terms of ease of garden management and ensuring plants receive adequate water supplies according to their needs [3]. Nevertheless, despite the availability of basic automatic watering systems, several operational aspects still require attention to enhance their efficiency and reliability. One crucial yet often overlooked aspect is the system's capability to detect the absence of water flow. Failure to accurately detect water flow shortages or pipe blockages can ultimately lead to unwatered plants and undetected system failures [4].

Several studies have attempted to address automatic watering with various approaches. Drawing upon previous research by [5], an IoT-based automatic watering monitoring system was developed for ornamental plants, focusing on ideal temperature and humidity control using DHT22 and soil moisture sensors integrated with the ESP8266 microcontroller and Thinger.io platform. Additionally, algorithm optimization for watering duration has been explored, where the Tsukamoto fuzzy inference method achieved 100% accuracy in data compliance, significantly outperforming the Mamdani method which only achieved 45% compliance, resulting in a distinct average watering duration between the two methods [5], [6]. While these related works successfully manage environmental parameters and scheduling, they generally assume a constant water supply and lack mechanisms to detect real-time water flow availability within the piping network.

This technological gap becomes highly problematic when applied to real-world scenarios, such as at the INSTIKI Campus. Based on field interviews and observations with the Head of the General Service and Infrastructure Bureau at the INSTIKI Campus, it was revealed that water usage across the campus buildings, particularly Building 1, is highly unstable and inefficient. This inefficiency is evident from the continuous, uninterrupted operation of the water pump, which triggers a significant increase in the campus's electricity consumption. To mitigate this issue, the campus management requires a structured watering schedule with staggered timing to ensure an equitable and balanced water distribution across the campus buildings, equipped with a failure detection mechanism when the water ceases to flow.

To bridge these gaps, this study develops an IoT-based automatic watering and water flow detection system using the ESP8266 microcontroller at the INSTIKI Campus, utilizing direct electricity as its primary energy source. The system is designed to distribute water according to a predetermined structured schedule and actively monitors the presence of water flow using a water flow sensor [7]. The acquired data and operational status are transmitted in real-time to the Telegram Messenger platform to provide instant notifications to the campus infrastructure team [8]. The primary objective of this research is to design, construct, and rigorously evaluate the operational performance of this automated system. Ultimately, this study aims to provide a reliable solution for mitigating water management inefficiencies, preventing unwatered vegetation due to dry flows, and optimizing utility consumption within the campus environment.

## 2. Method

The research methodology follows a structured engineering design process encompassing data collection, system analysis, hardware and software configuration, architectural design, and evaluation [9]. Fieldwork was initially conducted on February 4, 2024, at the INSTIKI Campus located on Jl. Tukad Pakerisan, Panjer, to observe the specific layout of the greenery infrastructure. Data collection techniques involved both primary and secondary data streams. Primary data was gathered through structured field interviews with the infrastructure and general services management personnel of the campus to understand operational constraints, alongside direct direct observations. The observations confirmed the physical setup near the northern library sector, identifying a single dedicated water supply line supplying eight distinct plant zones. Secondary data was extracted from institutional documentation, operational manuals, and relevant empirical literature focused on IoT-based fluid automation, automated water metering, and micro-controller valve integrations to construct a reliable baseline for the physical prototype [10].

System analysis reveals that the existing manual watering workflow relies entirely on human labor, where operators must manually attach a hose to the local valve, stretch it across the garden plot, irrigate the plants sequentially, shut off the source, and re-spool the equipment. This operational mode leads to massive utility costs due to unmonitored water pump runtimes and excessive electrical consumption. To optimize this pipeline, the proposed automated architecture integrates specialized hardware and software components. The hardware environment comprises an ESP8266-based NodeMCU micro-controller as the central processing unit [11], a digital YF-S201 Water Flow Meter to read fluid volume and verify actual flow rates [12], a mechanical relay board acting as an electronically controlled switch, a step-down switched-mode power supply (SMPS) to convert standard 220V AC utility power to a stable 5V DC line, an electric solenoid valve to open or close the fluid pathway , and a DS3231 Real-Time Clock (RTC) module to maintain time records [13]. On the software spectrum, firmware development was conducted using the Arduino Integrated Development Environment (IDE), circuit schematic simulations were executed in Proteus 8 Professional [14], three-dimensional spatial packaging was designed via SketchUp [15], structural logic mapping was formulated using Draw.io, and the Telegram Messenger application was utilized as the primary end-user monitoring interface.

The architectural framework is organized into three foundational blocks consisting of input, process, and output stages. The input stage incorporates the DS3231 RTC module for real-time temporal referencing, the water flow meter for hydro-metric tracking, and the primary power supply unit. The process stage is managed by the NodeMCU micro-controller, which samples sensor inputs, runs the scheduling algorithms, evaluates flow state criteria, and coordinates output execution. The output stage consists of the relay driver coupled to the 220V AC solenoid valve for physical irrigation control and the Telegram API interface for system status alerts. The detailed interaction between these input, process, and output stages can be observed in the block diagram shown in Figure 1.

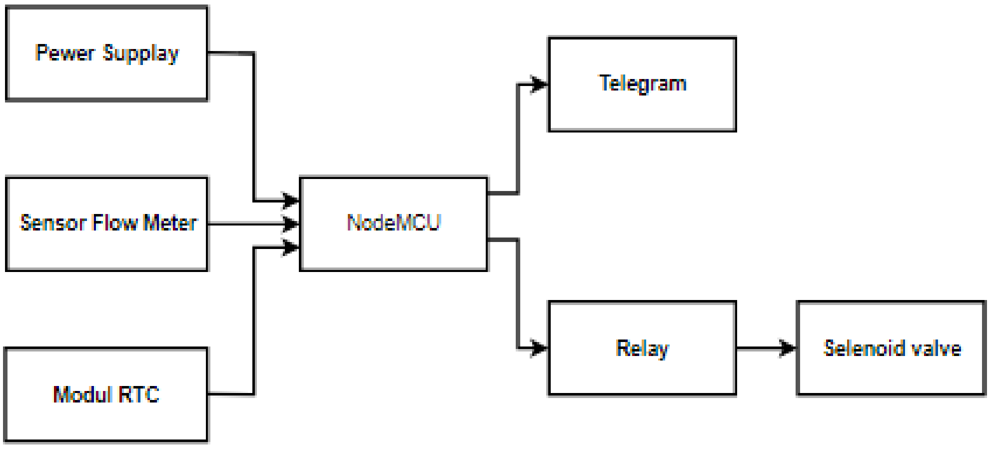


**Figure 1**. Block Diagram of the Proposed IoT-Based Watering System

In terms of physical packaging, the embedded components are contained within a rectangular protective box measuring 22 cm in length, 15 cm in width, and 7.5 cm in height, featuring a specialized cylindrical structural protrusion measuring 2.9 cm by 2 cm with a 3 cm diameter to allow plumbing interfaces [16]. Hardware integration is established through precise terminal mapping. The YF-S201 Water Flow Meter interfaces with the NodeMCU by linking its VCC pin to the 5V (VU) line, GND to common ground, and its digital DATA output to the D5 pin. The DS3231 RTC module maps its VCC pin to the 3.3V pin, GND to ground, and its Serial Clock (SCL) and Serial Data (SDA) lines to digital pins D1 and D2 respectively utilizing the Inter-Integrated Circuit (I2C) communication protocol [17]. The relay board links its VCC pin to the 3.3V supply, GND to ground, and its input signal DATA pin to D3. For the power management block, the AC mains (220V) are connected to the Live (L) and Neutral (N) terminals of the power supply unit, while its DC positive (+V) output routes to the VU pin of the NodeMCU and its negative (-V) output attaches to the common ground (GND) pin. The solenoid valve wiring connects its VCC 1 terminal directly to the Common (COM) port of the relay board, while its VCC 2 terminal loops straight into the 220V AC source grid.

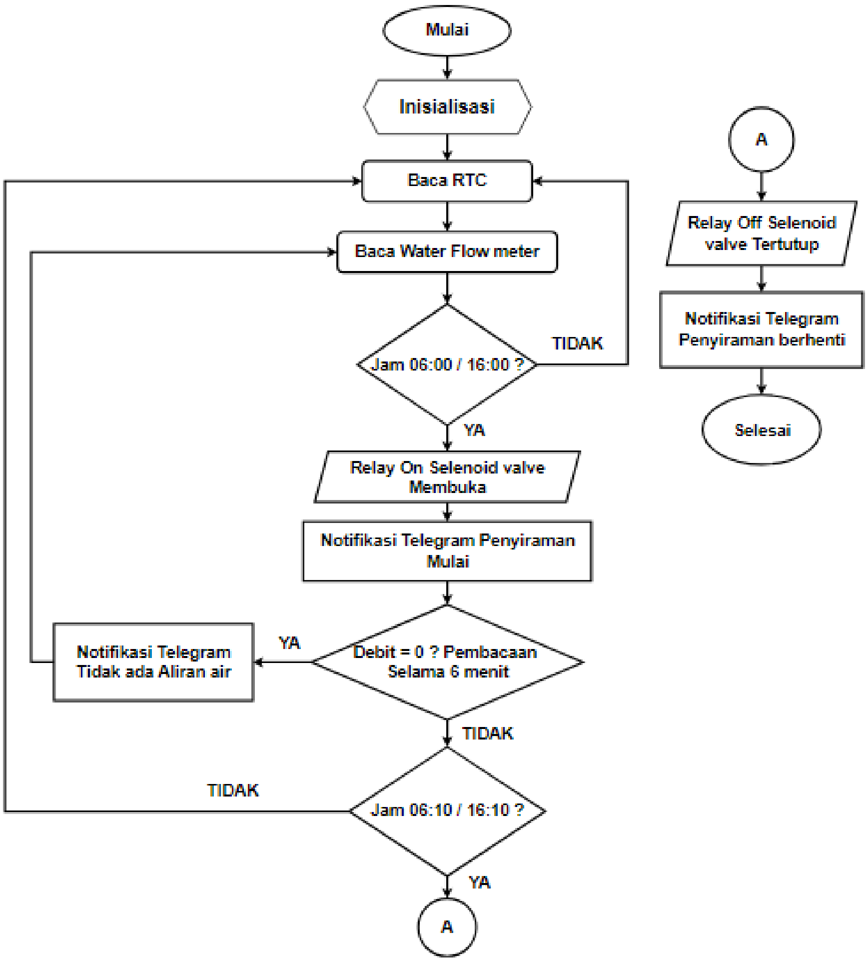


**Figure 2**. System Operational Flowchart

The embedded execution logic operates on a strict sequential algorithm. Upon boot-up, the system initializes internal registers, establishes a connection to the local wireless network, and pulls the current time parameters from the RTC module while reading data from the water flow meter. The algorithm continuously evaluates whether the current clock time matches the pre-programmed irrigation triggers set at 06:00 or 16:00. If the time criteria are met, the NodeMCU pulls the digital D3 pin high to trigger the relay, which opens the solenoid valve, while simultaneously generating a secure HTTP post request to transmit a "Penyiraman tanaman Mulai" alert message to the designated Telegram bot chat via the Telegram Bot API [18]. Once the valve opens, a critical exception-handling sub-routine executes: the micro-controller samples the water flow sensor data continuously; if the read debit value stays at zero for a continuous period of 6 minutes, indicating a dry-run condition or a utility water outage, the system immediately dispatches a critical "Tidak ada Aliran air" error notification via the Telegram channel to request human intervention, preventing potential actuator damage [19]. The irrigation session runs for exactly ten minutes; when the RTC clock reaches 06:10 or 16:10, the micro-controller switches the relay state to close the solenoid valve, cuts off the water supply, and transmits a concluding "Penyiraman tanaman Berhenti" notification text string through the monitoring bot to confirm a successful operation cycle. The logical sequence and conditional branching of this embedded firmware are illustrated in the system flowchart presented in Figure 2.

To validate system reliability, a multi-tier testing framework was instituted across six operational scenarios [20]. The first phase utilizes Wireshark packet analysis to evaluate the local wireless connection stability of the NodeMCU. The second phase assesses the temporal tracking accuracy of the RTC module against real-time standards . The third phase verifies the calibration and pulse-to-volume accuracy of the hydro-metric flow sensor. The fourth phase evaluates the electrical isolation and structural switching thresholds of the micro-controller relay driving the solenoid valve circuit [21]. The fifth phase ensures successful API handshakes and correct message parsing between the NodeMCU and the remote Telegram bot server [22]. Finally, the sixth phase runs an end-to-end operational analysis to ensure the entire system adheres to the staggered schedule, correctly triggers safety flags during simulated supply line cuts, and updates the remote network log without data loss or latency overhead.

## 3. Result and Discussions

The implementation and empirical verification of the IoT-based automatic watering and flow detection system were executed systematically at the INSTIKI Campus to evaluate its physical deployment, network service quality, and functional reliability. The hardware prototyping and physical deployment were split into three operational phases consisting of the plumbing adaptation, components housing encapsulation, and terminal interfacing. For the water supply coupling, a 0.5-inch diameter piping system extending 7 meters was integrated via a dedicated split-Tee junction and a custom 0.5-inch Elbow adapter to link the primary campus storage tank directly to the physical prototype. The fluid distribution grid spans 10 meters, incorporating six thread-type Tee connectors, one thread Elbow, and seven individual mechanical taps deployed across the distinct plant zones to control the volumetric water flow distributed to the vegetation, as shown in the empirical setup across Figure 3 and Figure 4.

**Figure 3**. Physical Installation of the Campus Water Supply Line

**Figure 4**. Physical Layout of the Staggered Piping Distribution Grid

To shield the internal micro-controller architecture from environmental stress, moisture ingress, and thermal degradation, all core electronic modules were organized inside a sealed plastic junction box measuring 220 mm × 150 mm × 75 mm, as illustrated in Figure 5.

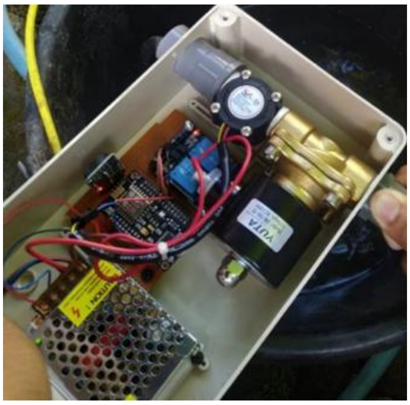

**Figure 5**. Internal Packaging Architecture and Component Placement within the Housing

The internal terminal layout clusters the components into specialized functional sectors: the power supply conversion unit occupies the lower-left sector to safely step down the 220V AC grid line to a stable 5V DC line; the NodeMCU ESP8266 acts as the core processing core; the DS3231 RTC module maintains independent calendar records; and a heavy-duty mechanical relay serves as the high-voltage isolation switch connected to the brass solenoid valve. The operational state of the micro-controller configuration and its sensor-actuator sub-systems during structural deployment are documented in Figure 6 and Figure 7.

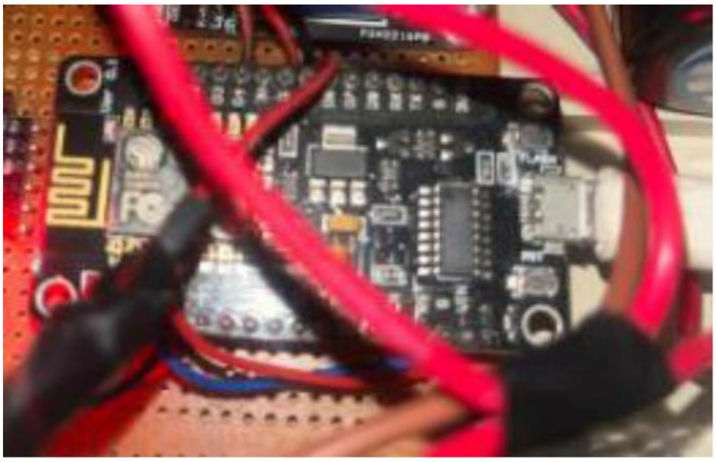

**Figure 6**. Micro-Controller Hardware Integration and Terminal Bus Interface


**Figure 7**. Physical Interfacing of the YF-S201 Water Flow Sensor with the Solenoid

A crucial aspect of this study involved investigating the local wireless transmission efficiency of the NodeMCU ESP8266 when communicating with the remote Telegram API over the institutional network grid. Empirical evaluation was carried out via Wireshark packet capture analysis across multiple tracking cycles to evaluate key Quality of Service

(QoS) parameters, consisting of throughput, packet loss, and connection delay. The quantitative network analysis is detailed in Table 1.

**Table 1**. Wireshark Network Quality of Service (QoS) Test Parameters

| No | Measured Network Metric | Testing Cycle 1 (10/07/2024 at 16:00) | Testing Cycle 2 (11/07/2024 at 16:00) |
| --- | --- | --- | --- |
| 1 | Network Throughput | 2715 kbps | 155 kbps |
| 2 | Packet Loss Rate (%) | 0.00% | 2.85% |
| 3 | Cumulative Delay (s) | 716.6876 s | 720.1170 s |
| 4 | Average Transmission Delay (s) | 3.5350 s | 3.5290 s |

The network telemetry data shows a significant drop in throughput from 2715 kbps down to 155 kbps during the second testing cycle, which correlated with an increase in packet loss up to 2.85%. This fluctuation reflects high network traffic congestion on the campus Wi-Fi grid during peak student activity hours. This empirical finding explains occasional packet dropouts and latency overhead during real-time event alerts, confirming that wireless infrastructure load impacts IoT messaging performance.

Before deploying the automated prototype into continuous operation, a series of preliminary verification trials were carried out to calibrate the DS3231 temporal accuracy and test the initial switching behavior of the YF-S201 flow meter, as summarized in Table 2.

**Table 2**. Preliminary System Calibration and Pre-Testing Performance Matrix

| No | Testing Date and Time | Flow Meter Detection State | Solenoid Open Duration | "Watering Start" Telegram Alert | "Watering Stop" Telegram Alert |
| --- | --- | --- | --- | --- | --- |
| 1 | 22/05/2024 at 15:00 | No Flow Detected | 20 Minutes | Successfully Transmitted | Successfully Transmitted |
| 2 | 28/05/2024 at 06:00 | Active Flow Detected | 10 Minutes | Successfully Transmitted | Successfully Transmitted |
| 3 | 28/05/2024 at 12:00 | Active Flow Detected | 10 Minutes | Successfully Transmitted | Successfully Transmitted |
| 4 | 28/05/2024 at 16:00 | Active Flow Detected | 10 Minutes | Successfully Transmitted | Successfully Transmitted |
| 5 | 29/05/2024 at 06:00 | Active Flow Detected | 10 Minutes | Successfully Transmitted | Successfully Transmitted |
| 6 | 29/05/2024 at 16:00 | Active Flow Detected | 10 Minutes | Successfully Transmitted | Successfully Transmitted |

During the first trial on May 22, 2024, the prototype was initially configured with an extended 20-minute operational window to evaluate structural limits, during which the system successfully verified a simulated "No Flow" alert condition. Following requests from the campus infrastructure team, the firmware logic was updated on May 28 to standard 10-minute irrigation cycles triggered thrice daily, which was subsequently optimized on May 29 to a staggered twice-daily schedule at 06:00 and 16:00 to balance fluid volumes across campus buildings. Following this optimization, the prototype was subjected to an extended 10-day field evaluation to track scheduling precision and sensor exception handling under real campus operational loads, as documented in Table 3.

**Table 3**. Long-Term Field Operational Evaluation and Reliability Matrix

| No | Testing Date and Time | Flow Meter Detection State | Solenoid Open Duration | "Watering Start" Telegram Alert | "Watering Stop" Telegram Alert |
|---|---|---|---|---|---|
| 1 | 30/05/2024 at 06:00 | No Flow (Triggered at Min 7) | 10 Minutes | Successfully Transmitted | Successfully Transmitted |
| 2 | 30/05/2024 at 16:00 | Active Flow Detected | 10 Minutes | Failed to Transmit | Successfully Transmitted |
| 3 | 31/05/2024 at 06:00 | Active Flow Detected | 10 Minutes | Successfully Transmitted | Successfully Transmitted |
| 4 | 31/05/2024 at 16:00 | Active Flow Detected | 10 Minutes | Successfully Transmitted | Successfully Transmitted |
| 5 | 01/06/2024 at 06:00 | Active Flow Detected | 10 Minutes | Successfully Transmitted | Successfully Transmitted |
| 6 | 01/06/2024 at 16:00 | Active Flow Detected | 10 Minutes | Successfully Transmitted | Successfully Transmitted |
| 7 | 02/06/2024 at 06:00 | Active Flow Detected | 10 Minutes | Successfully Transmitted | Successfully Transmitted |
| 8 | 02/06/2024 at 16:00 | Active Flow Detected | 10 Minutes | Successfully Transmitted | Successfully Transmitted |
| 9 | 03/06/2024 at 06:00 | Active Flow Detected | 10 Minutes | Successfully Transmitted | Successfully Transmitted |
| 10 | 03/06/2024 at 16:00 | Active Flow Detected | 10 Minutes | Failed to Transmit | Successfully Transmitted |
| 11 | 04/06/2024 at 06:00 | Active Flow Detected | 10 Minutes | Successfully Transmitted | Successfully Transmitted |
| 12 | 04/06/2024 at 16:00 | Active Flow Detected | 10 Minutes | Failed to Transmit | Successfully Transmitted |
| 13 | 05/06/2024 at 06:00 | Active Flow Detected | 10 Minutes | Successfully Transmitted | Successfully Transmitted |
| 14 | 05/06/2024 at 16:00 | Active Flow Detected | 10 Minutes | Successfully Transmitted | Failed to Transmit |
| 15 | 06/06/2024 at 06:00 | Active Flow Detected | 10 Minutes | Successfully Transmitted | Successfully Transmitted |
| 16 | 06/06/2024 at 16:00 | Active Flow Detected | 10 Minutes | Successfully Transmitted | Successfully Transmitted |
| 17 | 07/06/2024 at 06:00 | Active Flow Detected | 10 Minutes | Successfully Transmitted | Successfully Transmitted |
| 18 | 07/06/2024 at 16:00 | Active Flow Detected | 10 Minutes | Successfully Transmitted | Failed to Transmit |
| 19 | 08/06/2024 at 06:00 | Active Flow Detected | 10 Minutes | Successfully Transmitted | Successfully Transmitted |

| No | Testing Date and Time | Flow Meter Detection State | Solenoid Open Duration | "Watering Start" Telegram Alert | "Watering Stop" Telegram Alert |
|---|---|---|---|---|---|
| 20 | 08/06/2024 at 16:00 | Active Flow Detected | 10 Minutes | Successfully Transmitted | Successfully Transmitted |

The long-term field data demonstrates that the prototype successfully executed its scheduled irrigation cycles on time, guided by the DS3231 RTC tracking module. A critical exception event occurred on May 30, 2024, at 06:00, when the municipal supply line suffered an unexpected fluid drop seven minutes into operation. The YF-S201 water flow sensor immediately registered a flow rate of zero, and the embedded firmware handled this exception correctly by dispatching a real-time notification string ("Tidak ada Aliran air") to the remote Telegram interface. However, the data also highlights transmission dropouts during the evening sessions (16:00) on May 30, June 3, June 4, June 5, and June 7, where specific startup or shutdown notifications failed to appear in the Telegram channel. These communication errors match the high packet loss rates and congested network states identified during the Wireshark telemetry tests.

Scientifically, these findings present a meaningful advancement over the previous frameworks reported by [23] and [24]. While previous works achieved high analytical accuracy in optimizing environmental variables like ambient soil moisture and temperature thresholds, they relied on the assumption of an uninterrupted water supply line. Consequently, those designs lacked failure detection sub-routines to identify infrastructure blockages or dry pump states. In contrast, this study provides a reliable practical improvement by integrating inline hydrometric sensor verification into the control loop. This architectural contribution ensures that any supply failures are immediately flagged, preventing undetected dry runs and reducing excess electrical utility usage caused by unmonitored water pumps.

## 4. Conclusion

This study successfully designed, constructed, and evaluated the operational performance of an IoT-based automatic watering and water flow detection system using the ESP8266 microcontroller at the INSTIKI Campus. Based on the research findings, the system was effectively developed through a structured engineering workflow, starting from algorithm flowchart mapping, block diagram modeling, and comprehensive circuit schematic design using Proteus 8 Professional simulation tools. The integration of core hardware components—comprising the ESP8266 as the data processing unit, an electric solenoid valve for automated fluid control, and a mechanical relay board acting as an isolation switch—met all predetermined scheduling triggers. Furthermore, the system demonstrated robust performance in event-driven monitoring, where the physical apparatus successfully detected fluid supply failures and dispatched real-time operational status alerts to the user through the Telegram Messenger API interface. Despite its successful deployment, several technical limitations were identified during empirical testing, providing avenues for further research and system enhancement. To address the packet loss issues observed during Wireshark network quality analysis, which occasionally disrupted alert transmissions, future development should integrate a cellular SIM module to ensure uninterrupted backup internet connectivity in areas with weak or unavailable Wi-Fi coverage. Additionally, to improve device practicality and deployment flexibility in remote campus zones lacking direct access to the conventional electrical grid, subsequent research should explore alternative off-grid energy harvesting mechanisms, such as integrating a solar panel system or a hydro-generator module to power the micro-controller apparatus autonomously.

## References

[1] A. Addas, "The importance of urban green spaces in the development of smart cities," *Front. Environ. Sci.*, vol. 11, 2023, doi: 10.3389/fenvs.2023.1206372.

[2] J. Tarigan, M. Bukit, D. Siprianus, and N. Yilu, "RANCANG BANGUN SISTEM IRIGASI TETES OTOMATIS UNTUK BUDIDAYA TANAMAN TERONG UNGU (SOLANUM MELONGENA L.) BERBASIS INTERNET OF THINGS (IoT)," 2023.

[3] Dani Rofianto *et al.*, "IoT-Based Smart Irrigation and Fertilization System with Realtime Cloud Integration," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, vol. 15, no. 1, pp. 34–43, Feb. 2026, doi: 10.22146/jnteti.v15i1.23124.

[4] A. Dwi Prasetya *et al.*, "Rancang Bangun Sistem Monitoring dan Pendeteksi Lokasi Kebocoran Pipa Berdasarkan Analisis Debit Air Berbasis IoT," 2020.

[5] "1882-Article Text-7061-1-10-20220609".

[6] M. K. Ardi, B. Darma Setiawan, and M. A. Fauzi, "Optimasi Fuzzy Inference System Tsukamoto Menggunakan Algoritme Genetika Untuk Mengetahui Lama Waktu Siram Pada Tanaman Strawberry," 2018. [Online]. Available: http://j-ptiik.ub.ac.id

[7] L. Guezouli, L. Guezouli, M. B. Eddine Djeghaba, and A. Bentahrour, "IoT and AI for Real-time Water Monitoring and Leak Detection," *Journal of Renewable Energies*, vol. 27, no. 2, pp. 243–281, Dec. 2024, doi: 10.54966/jreen.v27i2.1210.

[8] "125_Laily".

[9] O. Alsamrai, M. D. Redel-Macias, S. Pinzi, and M. P. Dorado, "A Systematic Review for Indoor and Outdoor Air Pollution Monitoring Systems Based on Internet of Things," Jun. 01, 2024, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/su16114353.

[10] E. Elbasi *et al.*, "Artificial Intelligence Technology in the Agricultural Sector: A Systematic Literature Review," 2023, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/ACCESS.2022.3232485.

[11] A. Das and R. K. Jain, "Performance evaluation of mobile robotic system for monitoring and inspection using MQTT protocol and IoT technique," *Int. J. Intell. Robot. Appl.*, vol. 9, no. 4, pp. 1302–1330, 2025, doi: 10.1007/s41315-025-00465-z.

[12] S. Falamba, P. G. Home, S. Ondimu, and E. Bwambale, "Development of a drip irrigation system integrating solar panels, IoT-connected sensors, and a control unit to automate irrigation based on real-time data." [Online]. Available: https://ssrn.com/abstract=6641594

[13] H. Mohammadmoradi, O. Gnawali, and A. Szalay, "Accurately Initializing Real Time Clocks to Provide Synchronized Time in Sensor Networks."

[14] L. Vasetska, "A study of the electric circuit modelling and simulation software efficiency and their accuracy, speed and ease of use comparison," *Вісник Черкаського державного технологічного університету*, vol. 29, no. 2, pp. 32–44, Apr. 2024, doi: 10.62660/bcstu/2.2024.32.

[15] O. Pedgley and B. Şener, "27 TH INTERNATIONAL CONFERENCE ON ENGINEERING AND PRODUCT DESIGN EDUCATION THE ROLE OF PHYSICAL SKETCH MODELLING IN THE DIGITAL DESIGN ERA."

[16] A. E. Elwakeel, "A smart automatic control and monitoring system for environmental control in poultry houses integrated with earlier warning system," *Sci. Rep.*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-17074-2.

[17] H. L. Tsai, L. P. Truong, and W. H. Hsieh, "Design and Evaluation of Wireless Power Monitoring IoT System for AC Appliances †," *Energies (Basel).*, vol. 16, no. 1, Jan. 2023, doi: 10.3390/en16010163.

[18] C. L. Kok, J. B. Heng, Y. Y. Koh, and T. H. Teo, "Energy-, Cost-, and Resource-Efficient IoT Hazard Detection System with Adaptive Monitoring," *Sensors*, vol. 25, no. 6, Mar. 2025, doi: 10.3390/s25061761.

[19] M. A. Mielcioiu, P. Petcu, D. Nedelcu, A. Semenescu, N. Valter, and A. M. Nicolau, "Design and Implementation of a Low-Water-Consumption Robotic System for Cleaning Residential Balcony Glass Walls," *Applied Sciences (Switzerland)*, vol. 16, no. 2, Jan. 2026, doi: 10.3390/app16020945.

[20] M. E. Arzuaga-Ochoa, M. Acosta-Coll, and M. Barrios Barrios, "The Validation–Deployment Gap in Agricultural Information Systems: A Systematic Technology Readiness Assessment," Jan. 01, 2026, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/informatics13010014.

[21] S. M. H. Shaloh *et al.*, "A Novel and Extremely Fast Saturation Less Circuit Breaker Using Non-Mechanical Components," *International Journal of Cloud Applications and Computing*, vol. 16, no. 1, 2026, doi: 10.4018/IJCAC.401331.

[22] H. Zhang, R. Zhang, and J. Sun, "Developing real-time IoT-based public safety alert and emergency response systems," *Sci. Rep.*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-13465-7.

[23] Fhizyel Nazareta, Fauziah, and Gatot Soepriyono, "Smart Agriculture: Pengendalian Kelembapan Dan Suhu Pada Penyiraman Otomatis Tanaman Berbasis IoT," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. Vol. 9, No. 2, pp. 839–854, Jun. 2022.

[24] P. A. Wulandari, P. Rahima, and S. Hadi, "Rancang Bangun Sistem Penyiraman Otomatis Berbasis Internet of Things Pada Tanaman Hias Sirih Gading," *Jurnal Bumigora Information Technology (BITe)*, vol. 2, no. 2, pp. 77–85, Sep. 2020, doi: 10.30812/bite.v2i2.886.