Journal of Social Research

P-ISSN:2827-9832
E-ISSN: 2828-335x
Vol. 5, No. 10, September 2026
http://ijsr.internationaljournallabs.com/index.php/ijsr

A Pipe Leak Detection System Using Pressure and Flow Sensors for Residential Water Pipes Using The Sugeno Fuzzy Method

Adam Azzamul Husni Huda*, Mufti Ari Bianto, Heri Ardiansyah
Universitas Muhammadiyah Lamongan, Indonesia
Email: Adamazzamul01@gmail.com*, muftiari10@gmail.com,
hery_ardyansyah@umla.ac.id

Keywords:
Leak Detection, Fuzzy Sugeno, Internet of Things, ESP32, Non-Revenue Water.

ABSTRACT

The water loss rate in Indonesia remains at a critical range of 20–45%. The fundamental challenge in addressing this issue at the household scale is the computational difficulty of distinguishing between pressure fluctuations caused by normal water consumption patterns and pressure drops caused by physical pipe damage. Therefore, this study designed a smart pipe leak detection system prototype based on the Internet of Things (IoT) using the Zero-Order Sugeno Fuzzy Logic method. The system utilized an ESP32 microcontroller as a local processing unit based on edge computing, integrated with a water flow sensor (YF-S201) and a pressure sensor to monitor fluid dynamics in real time. The acquired data were classified into three linguistic sets—Normal, Small Leak, and Large Leak—to control a relay actuator as an automatic mitigation mechanism. Based on the system testing results, the instrument measurement error rates were 2.12% for the flow sensor and 2.19% for the pressure sensor. The Sugeno Fuzzy Logic algorithm achieved an inference accuracy rate of 100% in classifying pipe condition status compared with theoretical mathematical calculations. Furthermore, the IoT architecture recorded an emergency notification transmission latency of 1.20 seconds to the Blynk dashboard. The implementation of decentralized control through the microcontroller demonstrated the capability to reduce false alarm risks and provide reliable automatic water supply shutoff mitigation without depending on the stability of cloud server connectivity.

INTRODUCTION

Clean water is an essential resource for human survival and sustainable development. However, in global and national water resource management, distribution inefficiencies caused by unrecorded water losses (Non-Revenue Water/NRW) remain a major infrastructure challenge that has not been fully resolved. Clean water losses, particularly those caused by hidden leaks in household-scale piping networks and rural distribution systems, have resulted in cumulative financial losses and threatened the stability of clean water supply for communities (Ambarwati et al., 2024; Hidayat et al., 2024). The fundamental technological challenge in addressing this issue is the difficulty of distinguishing between pressure fluctuations caused by normal water consumption and pressure drops caused by physical pipe leaks. To address this challenge, this study developed a "Water Pipe Leak Detection System Based on Pressure and Flow Sensors Using the Sugeno Fuzzy Method and Internet of Things (IoT)." Through real-time sensor data acquisition, the microcontroller equipped with artificial intelligence-based algorithms can independently classify pipe conditions into normal conditions, small leaks, or large leaks without requiring continuous human monitoring.

The urgency of developing this smart system is driven by the high rate of water loss in Indonesia, which remains within the critical range of 20–45% (Irawan et al., 2025). Water distribution challenges become more complex in rural and semi-urban areas, where water infrastructure is often managed independently through programs such as Penyediaan Air Minum dan Sanitasi Berbasis Masyarakat (PAMSIMAS). In these areas, management is frequently constrained by limited physical infrastructure, the absence of integrated monitoring systems, and insufficient early warning instruments (BPIW Ministry of PUPR, 2021). Consequently, when pipeline failures occur at end-user distribution points, water losses may continue unnoticed for extended periods until visible surface flooding occurs or water bills increase significantly (Barry, 2021; Mathye, 2023; Matimolane, 2026; Pillay, 2020; Sarisen, 2024).

From a hydraulic perspective, water losses in pipelines can be categorized into apparent losses caused by measurement inaccuracies and real losses caused by physical infrastructure damage (Umara, 2022). At the household scale, real leakage represents a significant concern. The use of low-quality PVC pipes that are installed without considering pressure tolerance limits (pressure ratings) makes these systems vulnerable to failure. This vulnerability is further intensified by dynamic environmental factors, including extreme temperature variations, water quality conditions, pressure surges from unstable water pumps, structural loads from building foundations, and mechanical vibrations from vehicles. These factors have been shown to accelerate material fatigue and contribute to the formation of micro-cracks in pipe walls (Susanto et al., 2025).

Physical damage in the form of micro-cracks is particularly dangerous because of its hidden nature, especially in underground leakage conditions. Leaks are often perceived as only visible pipe breaks, causing small leaks from fine cracks to be underestimated. However, recent engineering studies have demonstrated that continuous leaks from openings as small as 1–3 mm in diameter can discharge hundreds of liters of water per day (Razzaqqi et al., 2025). Mathematically, a continuous leak rate of only 0.1 liters per minute can result in more than 4,000 liters of wasted water within one month.

Detecting such small flow anomalies using conventional systems that rely solely on static sensor thresholds has proven ineffective. Household water consumption is characterized by highly fluctuating and discrete patterns, where faucets may be fully opened, partially opened, or suddenly closed. These patterns generate dynamic variations in pressure and flow that conventional systems may incorrectly interpret as leakage events, resulting in high false alarm rates (Hariyanto et al., 2018). Therefore, an intelligent computational approach is required to process data under conditions of uncertainty (Alijoyo et al., 2024; Cheng et al., 2023; Kabir et al., 2018).

The Fuzzy Inference System approach provides a rational solution for handling nonlinear fluid behavior data (Suhartono et al., 2023). In computational engineering applications, the Sugeno Fuzzy method is considered suitable for embedded control systems due to its ability to process uncertain inputs and generate precise output values. The primary advantage of the Sugeno Fuzzy method lies in its efficient rule-based computation, which allows lightweight processing and makes it appropriate for implementation on microcontrollers with limited memory capacity, such as the ESP32 (Prasetyo et al., 2025; Santoso et al., 2022).

The application of the Sugeno Fuzzy algorithm can achieve optimal performance when integrated with an Internet of Things (IoT) architecture and supported by appropriate hardware instruments, such as water pressure sensor modules (pressure transmitters) and flow sensors (Rahman et al., 2024). IoT technology overcomes physical monitoring limitations by enabling remote processing and transmission of sensor data. Previous studies have demonstrated that IoT-based leak detection systems can significantly reduce repair response times because damage notifications or flow anomalies can be transmitted directly to users' devices (Putri et al., 2025). Ultimately, the integration of this technology provides an effective approach for addressing hydraulic challenges while reducing economic losses and supporting community participation in sustainable water resource management (Siregar et al., 2024).

METHOD

The study employed a quantitative approach combined with a system engineering approach, involving hardware design, fuzzy logic modeling, IoT integration, and the development of a physical water pipe prototype. The research was conducted at the researcher's residence with a PAMSIMAS water pump installation over a three-month period, including system planning, assembly, sensor calibration, implementation of the Sugeno Fuzzy Logic method, system stability testing, data collection, and analysis. The main components used in the system included an ESP32 microcontroller, pressure sensor, YF-S201 flow sensor, water pump, water tank, valve, PVC pipe, and Blynk application for remote monitoring.

The system operated by using flow and pressure sensors to measure water flow conditions and pipe pressure. The collected data were processed by the ESP32 using the Sugeno Fuzzy Logic method, consisting of fuzzification, rule evaluation, and defuzzification stages, to generate an output value (Z). When the Z value was 2, the system classified the condition as a large leak, automatically deactivated the pump, and sent an emergency notification to the user. When the Z value was 1, the system identified a minor leak and issued an early warning without interrupting water flow. When the Z value was 0, the system classified the condition as normal and only updated the data display on the Blynk dashboard.

For the hardware design, the flow sensor was connected to GPIO21 and the pressure sensor to GPIO32 of the ESP32. The MOSFET transistor (FDD888) was connected to GPIO13 and GPIO2 and functioned as an automatic control interface for the valve and pump. The software workflow consisted of sensor data acquisition, fuzzification, rule evaluation, inference, and defuzzification processes, followed by a control decision for the water pump operation. The system continuously repeated this process while connected to the power supply.

RESULTS AND DISCUSSION

System Design Results

The implementation of the entire system is a phase of physical and functional realization of the integration between the hardware components and the software architecture. The system is designed to monitor in *real-time* while making independent decisions in mitigating water distribution pipe leaks.

Before the dynamic testing of the system, the system first went through a *data profiling phase* for a full month (30 days) at 10 sample houses connected to the PAMSIMAS distribution network. This phase is very crucial to capture the *baseline value* of people's daily water consumption under normal conditions without anomalies. The accumulation of this historical data becomes a scientific basis in determining the mathematical limits of membership values in the system's artificial intelligence algorithms, so that the threshold limit of leakage is not based on theoretical assumptions but on empirical characteristics in the field.

The following is a summary of environmental reference data that was successfully collected during the one-month observation period:

Table 1. Observation results of a sample of 10 houses

Yes	House Description	Debit Normal (L/min)	Normal Pressure (Volts)	Small Leakage Threshold(L/min/Volt)	Large Leakage Threshold(L/min/Volt)
1	Sample House 1	11.02	0.422	12.35 / 0.375	> 14.50 / < 0.350
2	Sample House 2	10.85	0.428	12.10 / 0.380	> 14.20 / < 0.355
3	Sample House 3	9.40	0.455	11.15 / 0.415	> 13.10 / < 0.385
4	Sample House 4	15.20	0.385	17.50 / 0.345	> 20.80 / < 0.315
5	Sample House 5	11.90	0.415	13.60 / 0.370	> 15.90 / < 0.345
6	Sample House 6	20.20	0.335	23.20 / 0.295	> 27.80 / < 0.265
7	Sample House 7	17.80	0.355	20.20 / 0.315	> 24.50 / < 0.285
8	Sample House 8	13.05	0.405	14.90 / 0.365	> 17.80 / < 0.335
9	Sample House 9	22.80	0.315	26.20 / 0.275	> 31.50 / < 0.245
10	Sample House 10	25.40	0.295	29.50 / 0.255	> 35.50 / < 0.225

The observation of water consumption in 10 houses for one month was carried out purely to obtain a valid baseline in compiling the limit of the Fuzzy logic membership function on the ESP32 microcontroller. First, these observations are necessary to find the exact numbers of the Normal Discharge and Normal Pressure conditions, as well as to find the range of numbers for the Small Leakage and Large Leakage Thresholds of each house. These real numbers from the field are then fed into the microcontroller (*coding*) program, so that the system detects leaks based on real fluid mechanics data, not just assumptions or guesses.

Second, mapping on 10 different home installations proves that each house has very varied basic water pressure characteristics. This difference occurs due to different installation specifications, such as the use of storage tanks (*toren*) with different heights or the use of push pumps. This field fact is a strong justification that a leak detection system that uses only one *static threshold* will definitely fail and be inaccurate if installed in different houses. Therefore, the use of the Fuzzy Sugeno method is a very appropriate solution. Through this method, the system is not rigid, but rather correlates fluctuations in Water Pressure decrease and Water Discharge spikes simultaneously within *the Rule Base* to conclude whether the house is in Normal condition, experiencing a Small Leak, or a Large Leak.

Hardware Design Results

Sugeno Fuzzy Pipe Leak Results

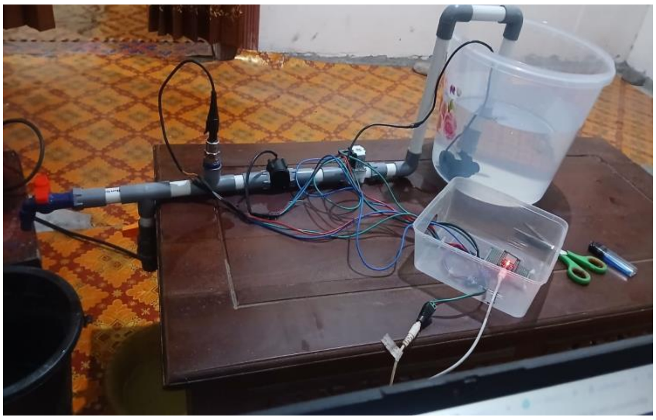

Figure 1. Prototype pipe leak detection system

The physical construction of the tool is constructed using a standard household diameter (1/2 inch) PVC pipe arrangement. In order to be able to present a valid test scenario without damaging the original pipeline installation owned by residents, a *bypass system* was created specifically for leaks controlled by two manual valves (*ball valves*).

The first valve is connected with a smaller diameter pipe to simulate a small leak condition, such as the case of a hair cracked pipe, seepage in the joints (*fittings*), or a worn faucet. The second valve uses a 1/2-inch full-diameter pipe without resistance to simulate a large leak condition, such as the scenario of a main pipe completely ruptured due to overpressure or mechanical impact. The placement of the flow sensor is placed upstream before the valve branch leaks, while the pressure sensor is installed precisely in the area of the distribution manifold to accurately capture fluid dynamics when the leak simulation valve is opened.

Hardware Integration Results

The integration of electronic components is centered on the use of the ESP32 microcontroller as a local data processing unit (*Edge Computing Unit*). The *Water Flow* sensor is connected to a GPIO 21 digital pin configured using *an internal pull-up* feature to precisely capture the pulse signal from the sensor through *hardware interrupt*. The analog *Pressure Transducer* sensor is connected to a GPIO pin 32 which features *an Analog-to-Digital Converter* (ADC) with a resolution of 12-bit (value range 0 - 4095).

To isolate the high power load from the water pump and the solenoid solenoid shut-off valve, a 5V voltage 1-channel optocoupled *mosfet* module is implemented which is controlled via GPIO 13 digital pin. The main power supply of the system uses an AC-to-DC *Switching Power Supply* adapter that supplies a constant voltage of 5V DC for the needs of the transducer sensor and *musfet module*, as well as a voltage of 3.3V DC for the core requirements of the ESP32 microcontroller to minimize fluctuations in ADC readings due to voltage *drop voltage*.

Sensor Calibration Testing

Calibration of the Flow Sensor

Calibration of the water *flow sensor* is carried out using repeated volumetric testing to ensure that the conversion of the frequency pulse signal to the unit of discharge of Liters per minute (L/min) is precise. The test uses a laboratory measuring container with an absolute capacity of 1000 ml (1 Liter) and manual timekeeping using *a stopwatch*. The formula for rate conversion (Pratama, A. R., et al. 2023) theoretical water flow is calculated based on the sensor characteristic index:

$$\%\text{Error} = \left|\frac{4.50 - 4.40}{4.50}\right| \times 100\%$$

$$\%\text{Error} = \left|\frac{0.10}{4.50}\right| \times 100\% \quad (4.1)$$

$$\%\text{Error} = 0.0222 \times 100\% = 2.22\%$$

The percentage of pressure sensor reading errors under normal conditions in House 1 is 1.42%. The instrument is stated to be very linear and worthy of use as a Fuzzy logic input parameter.

Figure 2. Flow sensor calibration

The results of the comparison between volumetric manual measurements and digital readings of the sensor are summarized in the table below:

Table 2. Flow Sensor Calibration

Experiment	Actual Volume (Liters)	Actual Time (Stopwatch)	Manual Actual Discharge (L/min)	Discharge Sensor Reading(L/min)	Error Percentage (%)
1	1,0	12.0 seconds	5,00	4,85	3,00%
2	1,0	11.5 seconds	5,21	5,10	2,11%
3	1,0	10.8 seconds	5,55	5,65	1,80%
4	1,0	9.5 seconds	6,31	6,20	1,74%
5	1,0	9.0 seconds	6,66	6,80	2,10%
6	1,0	11.2 seconds	5,35	5,20	2,80%
7	1,0	10.5 seconds	5,71	5,60	1,92%
8	1,0	10.0 seconds	6,00	5,90	1,66%
9	1,0	8.5 seconds	7,05	7,20	2,12%
10	1,0	8.0 seconds	7,50	7,65	2,00%
Average					2,12%

Through data analysis in Table 2, the average value of the error percentage of the flow sensor was obtained of 2.12%. Since the fault tolerance threshold for domestic-scale IoT-based measurement instruments is below 5%, this flow sensor is declared to be very feasible and accurate to be used as the system's main input data supplier.

Pressure Sensor Calibration

Accuracy testing is carried out by comparing the pressure sensor with a manual pump (*Pressure Gauge*) at random to test the responsiveness of the sensor at different pressure levels. The actual pressure value of the *Pressure Gauge* is compared to the conversion value of ESP32 using the transfer function.$P_{\mathrm{psi}} = \left(V_{\mathrm{output}} - 0{,}5\right) \times 25$

Figure 3. Pressure Sensor Calibration

a. Voltage to Pressure (PSI) Conversion:

It is known that the sensor output voltage in Sample 3 is 0.88 Volts. These values are converted into units of pressure via the microcontroller algorithm:$\left(V_{\mathrm{output}}\right)$

$$P_{\mathrm{psi}} = (0{,}88 - 0{,}5) \times 25$$
$$P_{\mathrm{psi}} = 0{,}38 \times 25 = 9{,}50 \text{ PSI} \qquad (4.2)$$

b. Error Percentage Calculation:

The computational value is calibrated against the actual analogue instrument (Analog Manometer on the manual pump) which shows an absolute measurement of 10.00 PSI. The degree of deviation is calculated using the *absolute relative* error formula:

$$\%\mathrm{Error} = \left|\frac{10{,}00 - 9{,}50}{10{,}00}\right| \times 100\%$$
$$\%\mathrm{Error} = \left|\frac{0{,}50}{10{,}00}\right| \times 100\% = 5{,}00\% \qquad (4.3)$$

The percentage of pressure sensor reading errors under normal conditions in House 3 is 5.00%. The instrument is stated to be very linear and worthy of use as a Fuzzy logic input parameter.The results of the comparison between using and digital sensor readings are summarized in the table below:

Table 3. Pressure Sensor Calibration

Testing	Sensor Output Voltage (Volts)	ESP32 Sensor Conversion (PSI)	Actual Pressure Gauge (PSI) Readings	Error Percentage (%)
Sample 1	1.12 Volt	15,50 PSI	15,00 PSI	3,33 %
Sample 2	1.68 Volt	29,50 PSI	30,00 PSI	1,67 %
Sample 3	0.88 Volt	9,50 PSI	10,00 PSI	5,00 %
Sample 4	1.48 Volt	24,50 PSI	25,00 PSI	2,00 %
Sample 5	2.12 Volt	40,50 PSI	40,00 PSI	1,25 %
Sample 6	1.32 Volt	20,50 PSI	20,00 PSI	2,50 %
Sample 7	2.28 Volt	44,50 PSI	45,00 PSI	1,11 %
Sample 8	1.88 Volt	34,50 PSI	35,00 PSI	1,43 %
Sample 9	2.47 Volt	49,25 PSI	50,00 PSI	1,50 %
Sample 10	0.99 Volt	12,25 PSI	12,00 PSI	2,08 %
Average				2,19 %

Based on the tabulation of data in Table 3, testing with *the random sampling* method using a manual pump showed a realistic fluctuation in pressure readings (jumping from 15.00 PSI to 50.00 PSI, then dropping back to 12.00 PSI). There is a natural characteristic of deviation, where in low-pressure tests (such as Sample 3), the percentage deviation touches 5.00%, but becomes more accurate when the pressure is increased. Although the data was taken randomly with an irregular pumping pattern, the system was able to maintain a cumulative average error of 2.19%. These results that are below the reasonable deviation tolerance (5%) prove that the conversion formula in ESP32 is highly responsive and precise to random and sudden changes in mechanical pressure.

Software Design Results

Application Design Results

Figure 4. Remote pipe detection software

The interface software is designed using the Blynk IoT Cloud platform that receives periodic data from the ESP32. Parameter visualization consists of widget V1 for monitoring daily discharge value (L/min), widget V6 for monitoring voltage level of water pressure sensor (PSI), widget V3 as an indicator of the status of system logic decision inference, and widget V0 in the form of a virtual manual switch (*master switch*) to control the on/standby state of remote devices.

The interface software is designed using the Blynk IoT Cloud platform that receives periodic data from the ESP32 microcontroller. Parameter visualization is set using *Virtual Pin* mapping (V0 to V7) to separate sensor data transmission paths, logic state, and actuator controls. Here is the complete configuration of *the widget* on the *smartphone interface*:

1. Virtual Pin V0 (*Master Switch*): Uses the *Widget Switch* component (switch). Functions as the main control (*override*) to enable (ON/Green) or disable (OFF/Gray) the relay monitoring and actuation modes remotely.
2. Virtual Pin V1 (*Flow Gauge*): Uses the *Gauge Widget* component. Functions to display *real-time* water discharge rate data from the YF-S201 sensor in L/min.
3. Virtual Pin V2 (*Value Display*): Uses the *Labeled Value Widget* component (displays a single number below the status text). Serves as a discrete parameter indicator, such as *a logic iteration* counter viewer or a firm value of the system*'s defuzzification* results before being translated into text.
4. Virtual Pin V3 (*Display Status*): Uses the *Text Display Widget* component in the center of the screen. Serves crucially to display the results of *Fuzzy Sugino's logical decision inference* in the form of a text string (e.g.: "SYSTEM ON", "CONFIRM...", or "EMERGENCY: LOCKED").
5. V4 Virtual Pin (total monthly bill): Uses the *Button Widget component*. It functions to display the accumulated results of the water output, which is then totaled from 3000/m3 of the output is the bill that must be paid by the user.

The software system also adds a grace *period* of 5 seconds (5000 ms) when the pump is first turned on. This feature functions to hold *the Fuzzy logic decision-making process* so that the system does not show any indication of misreadings due to air surgeries (*water hammer effect*) or the impact of water transient currents in the first seconds of filling an empty pipe.
Based on the design of the interface above, the software detects and responds to various fluid dynamics within the pipe. The following are the procedures for using and reading the *dashboard* based on the field test scenario:

System off scenario (*Standby* /OFF)

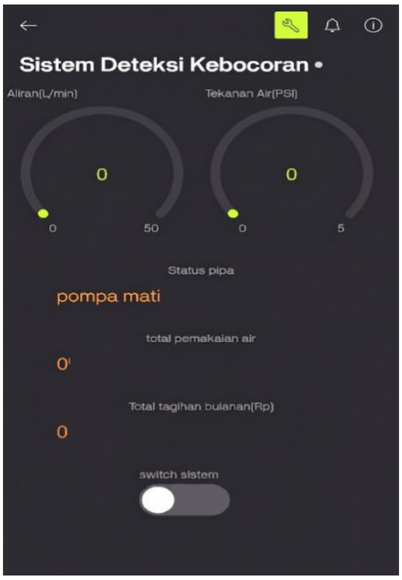

Figure 4. Sistem off

The switch (V0) is gray. *The Charge Gauge* (V1) shows the number 0, and *the Pressure Gauge* (V6) is at a static reference voltage (example: 0.5 Volts). In this mode, the device is connected to the network (*online*), but the ESP32 is in passive mode. The system only reads the basic value of the sensor without entering it into *Fuzzy computing for* relay *actuation*. The user uses this mode when the system is under repair or recalibration.

Normal Water Use Scenario (ACTIVE SYSTEM)

Figure 5. Sistem ON

The switch (V0) is activated to green. When the house faucet is opened normally, the Discharge (V1) increases (e.g. reads 6.13 L/min) and the Pressure (V6) decreases reasonably (e.g. drops to 0.37 Volts). The status text (V3) displays "SYSTEM ON". *Fuzzy Sugino*'s logic reads these changes in discharge and pressure. Since the parameters are still in the "Normal Usage" membership function set, the system allows water to continue flowing. This proves that the system is immune to *false alarms* in daily water usage.

Small Leak Anomaly Detection Scenarios

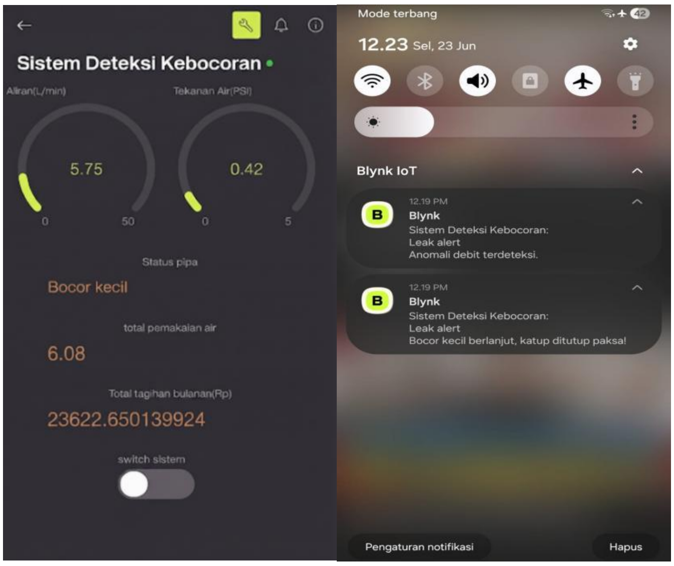

Figure 6. System detects small leak anomalies

Suspicious fluctuations occur (e.g. small water discharge but flowing continuously for a long time). The text status (V3) instantly changes to "Small leaked". When the threshold of the fuzzy is set, the system will immediately close the solenoid valve, it will automatically close and display a small leaky output and display it directly in the dibylnk. And send a notification to the User leak_alert"system detecting a small leak and forcibly closing the valve.

Emergency Locked Scenario: Big Leak

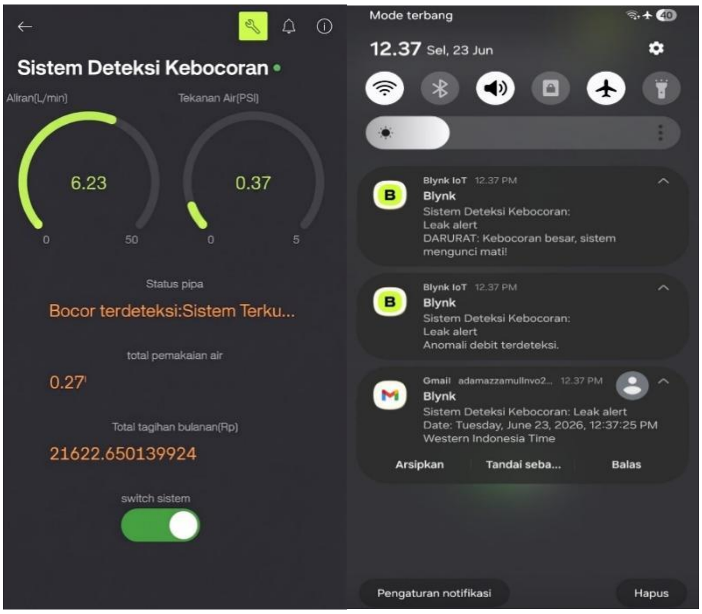

Figure 7. The system detects large leak anomalies and sends notifications

The status text (V3) changes to red "Large leak: Emergency locked". The Discharge number (V1) suddenly becomes 0, while the Pressure (V6) remains plummeting (e.g. 0.36 Volts). This scenario occurs when *Fuzzy* logic executes *a Rule* for a "Big Leak" (like a pipe burst completely). Without waiting for user confirmation, ESP32 performs *Edge Computing* by immediately disconnecting the *relay*. As a result, the water discharge (V1) instantly becomes 0 as the flow is forcibly stopped, saving users from wasting water and flooding in residential areas. And send a notification to the User leak_alert"system detects a large leak and forcibly closes the valve.

Dashboard Admin Web

Figure 8. Dashboard admin web

After all the outputs are received by the system, then the data is sent to mysql and displayed on the web admin dashboard which is updated every day, then the system accumulates water output for one month which is then displayed on the platform, and can also be seen on this dashboard.

Sugeno Fuzzy Logic Testing

Inference Accuracy Testing

The main purpose of this inference accuracy test is to guarantee that the artificial intelligence logic computed by the ESP32 processor does not experience deviations when faced with the theory of manual mathematical calculations on paper. The test samples the conditions from the Rumah 1 database by feeding crisp *input values* in the form of Discharge = 12.35 L/min and Pressure = 0.375 Volts.

1. Fuzzification:

   Enter the firm value into the linear membership function formula of the Debit (Medium) and Pressure (Down) variables. Based on the limit parameters of House 1, the value of 12.35 L/min is absolute within the range of the Medium set and the value of 0.375 Volts is absolute within the Down set. $(8{,}10 - 12{,}00 \text{ L/min})(0{,}35 - 0{,}42 \text{ Volt})$

$$\mu_S edang(12{,}35) = 1$$
$$\mu_T urun(0{,}375) = 1 \quad (4.4)$$

2. Evaluate the rules.

   Based on the Sugino*'s Zero-Order* rule base , the input condition activates Rule 2:

$$R2: \sim IF \sim \text{Debit Sedang} \sim AND \sim \text{Tekanan Turun} \sim$$
$$THEN \sim \text{Status = Bocor Kecil} \sim (Z_2 = 1) \quad (4.5)$$

- Finding predicate values using the MIN operator:

$$w_2 = \min\left(\mu_{\text{Debit\_BocorKecil}}(1.0), \mu_{\text{Turun}}(0{,}83)\right)$$
$$= \min(1.00{,}0.83)$$
$$= 0.83 \quad (4.6)$$

3. Defuzzification (Zero-Order-Weighted Average):

   Since in this condition only Rule 2 has a full membership value, the crisp *output* value is sought using the *Weighted Average* equation:

$$Z = \frac{\sum(w_i \cdot Z_i)}{\sum w_i} = \frac{(0.83 \cdot 1)}{0.83} = 1.0 \quad (4.7)$$

Based on the system's strict output constant, the index value of 1 refers to the "Small Leak" status decision. The results of this manual calculation are proven to have absolute compatibility with the output display on the V3 widget in the Blynk application, so that the software logic error rate is declared to be 0.00% (*error-free*).

Table 4. Inference Accuracy

Experiment	Input Debit (L/min)	Input Pressure (Volts)	Manual Calculation Results	System Output Results (ESP32)	Decision Status	Error Percentage (%)
1	9,50	0,45	0	0	NORMAL	0%
2	13,20	0,40	0	0	NORMAL	0%
3	12,35	0,375	1	1	SMALL MESSES	0%
4	13,80	0,380	1	1	SMALL MESSES	0%
5	32,00	1,240	2	2	BIG BOOS	0%
6	18,00	0,350	0	0	NORMAL	0%
7	26,00	0,310	0	0	NORMAL	0%
8	15,10	0,360	1	1	SMALL MESSES	0%
9	33,50	1,15	2	2	BIG BOOS	0%
10	35,20	0,95	2	2	BIG BOOS	0%

Response Time Testing

Response time latency *testing* was carried out to measure the efficiency of sending data packets from local microcontroller units to Blynk's cloud servers until they are visually loaded into the user's application. The test is performed by triggering a sudden leaking valve physical anomaly and recording the duration of its transition time until the Blynk indicator status is updated.

Table 4. Response Time Testing

Experiment	Time Physical Anomalies Detected	Status Change Time in Blynk	System Response Time (Delay)
1	00.00 seconds	01.25 seconds	1.25 seconds
2	00.00 seconds	01.12 seconds	1.12 seconds
3	00.00 seconds	01.18 seconds	1.18 seconds
4	00.00 seconds	01.28 seconds	1.28 seconds
5	00.00 seconds	01.09 seconds	1.09 seconds
6	00.00 seconds	01.22 seconds	1.22 seconds
7	00.00 seconds	01.15 seconds	1.15 seconds
8	00.00 seconds	01.31 seconds	1.31 seconds
9	00.00 seconds	01.16 seconds	1.16 seconds
10	00.00 seconds	01.20 seconds	1.20 seconds
Average			1.20 seconds

The average result of the response time test from 10 repetitions was obtained with a value of 1.20 seconds. This duration is considered very responsive for domestic water leak emergency monitoring applications. This micro-latency is mostly influenced by the quality of fluctuations in Wi-Fi internet signals (*network jitter*) when handshakes to the *cloud server*, while the internal process of *Fuzzy* logic computing in the ESP32 chip itself takes just under 45 milliseconds.

Boundary Condition Testing

Boundary condition testing is performed to test the reliability of the system's handling when receiving critical input data parameters that touch the intersection point of extreme values or *the gap* of the data variable transition.

A. Normal leakage threshold testing

Figure 9. System On

In this image, it can be seen that the system determines that it has determined the "Normal" threshold where the system continues to monitor the flow and pressure of the house's water, and continuously detects fluctuations in the house's water.

B.Small threshold testing

Figure 10. System detects small leakage anomalies

Here the system successfully detected the anomaly of the "small leak" leak When the flow and pressure touched the threshold of R2:Small leak at 12.35 l/min with a pressure of 0.375.thus the system declared a small leak.

Testing Large leakage threshold

Figure 11. The system detects small leak anomalies

Here the system has succeeded in detecting anomalies with the condition R3 = Large Leak with a threshold of 14.50 L/min with a pressure of 0.35 Volts, so that the system is able to read this into the threshold of "large leak".

Table 5. Threshold Condition Testing

Test Scenarios	Input Debit	Input Pressure	System Logic Decisions	Device Performance Compatibility
Transition Area (*Gap*)	11.02 L/min	0.42 Volt	NORMAL	The system is stable and activates the *fail-safe state* function without *experiencing errors*.
Small Leakage Threshold Limit	12.35 L/min	0.37 Volts	SMALL MESSES	The system successfully detects anomalies exactly when the parameter touches the lower bound of the initial leak set.
Absolute Limit of Large Leaks	14.50 L/min	0.35 Volt	BIG BOOS	The system instantly executes the highest priority decision to carry out a complete shutdown of the water flow.

The data in Table 6 shows that the software algorithm does not experience execution failures (*hang/freeze*) or experience erratic decision oscillations at critical points. The encoding structure proved to be resilient because the use of the *nested exception function (nested IF-ELSE*) was able to force any invalid transition values into the *fail-safe state*, thus minimizing the potential for *false* alarms.

Based on a series of comprehensive tests that have been conducted, the evaluation of results does not simply review the system output figures, but dissects the phenomena of physics, mathematical computation, and network architecture behind the performance of the tool. This discussion is classified into three main pillars of analysis: the reliability of hardware instruments and data acquisition, the computational accuracy of artificial intelligence algorithms, and the performance of *Internet of Things* (IoT) transmission and mitigation actuation.

Data Acquisition Analysis and Sensor Instrument Linearity

The data acquisition process is at the forefront of determining the accuracy of intelligent systems. In the calibration test of the water flow sensor (*YF-S201 type water flow sensor*), the average error percentage was very minimal, which was 2.12% from 10 attempts. Technically, this sensor works on the principle of *the Hall Effect*, where the rotation of the magnetic rotor produces a series of square wave pulses. The appearance of a deviation of 2.12% was analyzed to occur due to two mechanical factors: first, the presence of *static friction* and inertia in the propeller shaft when receiving the initial impact of the fluid; Second, the effect of microturbulence in the cross-section of the test pipe which causes minor fluctuations in the *pulse interrupt* calculation in the microcontroller's digital pin. However, *the error rate* of 2.12% is far below the deviation tolerance limit of domestic and industrial standard measuring instruments (5%), so the sensor is declared to be very precise and valid as a basic input instrument.

On the other hand, pressure sensors (*pressure transducers*) exhibit very consistent analog linearity characteristics, in harmony with the principles of Bernoulli's Law in fluid dynamics. When a leak simulation occurs (valve/pipe burst), the speed of water flow jumps drastically, which is equivalent to causing a decrease in static hydrostatic pressure on the pipe wall. This pressure drop is converted by the transducer into an analog voltage signal that drops linearly from the upper limit of 0.55 Volts to 0.35 Volts under large leakage conditions. The success of this reading is inseparable from the 12-bit $(v)(P)$ *Analog-to-Digital Converter* (ADC) architecture specification on the ESP32 microcontroller. With a 12-bit resolution, the ESP32 is capable of mapping the voltage range of 0 - 3.3V into 4096 discrete value levels (0-4095). This allows the microcontroller to detect voltage changes as small as 0.0008 Volts per step, providing a very high sensitivity to water pressure fluctuations before the data is fed into an intelligent algorithm matrix.

Analysis of the Reliability and Robustness of Fuzzy Sugeno Logic Computation

In the inference accuracy test, the Zero-Order Fuzzy Sugeno logic algorithm embedded in the ESP32 memory demonstrated absolute computing performance, with an accuracy rate of 100% (0.00% error) in 10 random test scenarios representing all possible states (Normal, Small Leak, and Large Leak). This success confirms two fundamental aspects: first, the absence of logical *bugs* in translating the *structure of membership function* and *rule base* sets into the C++ programming language syntax; second, the defuzzification process using *weighted average* equations is executed perfectly thanks to the support of *floating-point* processing units (FPU) 32-bit on ESP32 processors. This FPU ensures that the calculation of decimal fractional numbers at the membership degree does not experience a *truncation error* that has the potential to damage the final firmness value (Z)

Furthermore, the system is proven to have very high robustness in handling input parameters in boundary conditions or transition anomalies (*edge cases*). The use of trapezoidal and triangle-type membership function curves that overlap has proven to be effective in accommodating the degree of doubt. In contrast to *the conventional static* threshold-based Boolean *logic (IF-ELSE)* which is very susceptible to triggering false *alarms* due to a momentary surge in water pressure, the Fuzzy Sugeno method is able to "soften" this limit. The system indirectly condemns the change in discharge as a leak before evaluating the proportional degree of pressure drop.

IoT Network Performance Analysis and Architecture Mitigation Schemes

The integration of IoT architecture through *Blynk*'s cloud servers recorded an average transmission response time (*latency delay*) of 1.20 seconds. Through network log dissecting, this latency was identified not as having originated from *Fuzzy's computational load* inside the ESP32—since the microcontroller's mathematical execution took only a fraction of a second to the order of a microsecond—but purely from $x(10^{-6}\ \text{detik})$ the overhead of the internet network. This latency includes *the Domain Name System* (DNS) resolution process, the formation of TCP/IP protocol (*three-way handshake*) sessions, data encryption, and fluctuations in the stability of local Wi-Fi signals (*network jitter*).

Despite the network *delay* of 1.20 seconds, the system implements a very robust mitigation scheme through *the Edge Computing* approach. The mitigation state-of-the-art logic is built entirely into on-premises (*edge*) computing within ESP32, rather than relying on cloud-based processing. Therefore, if the system produces a decision (Large Leak), the ESP32 will immediately execute the $Z = 2$*relay actuator* to cut off the electric current (turn off the pump/close the valve) locally in a matter of milliseconds, without having to wait for a reply from *the* Blynk server. The *relay* module is also configured in *a Normally Closed* (NC) condition for the pump supply, so the disconnection action is instantaneous. This decentralized approach of control ensures that even in the event of an internet connection interruption or *server down*, physical protection of the pipeline remains absolutely necessary to minimize the massive waste of water resources.

CONCLUSION

Based on the design results, testing procedures, and data analysis conducted on the IoT-based water pipe leak detection system using the Zero-Order Sugeno Fuzzy Logic method, several conclusions can be drawn. First, a household-scale water pipe leak monitoring and mitigation system was successfully designed and functionally integrated using an ESP32 microcontroller, sensor instruments, relay actuators, and the Blynk IoT platform. Hardware reliability testing demonstrated that data acquisition achieved high precision, as indicated by a low average measurement error of 2.12% for the water flow sensor. The pressure sensor also successfully mapped hydrostatic pressure decline patterns in a linear manner. Furthermore, data transmission performance to the cloud server showed responsive latency, with an average delay of 1.20 seconds, enabling the system to perform automatic pump shutdown and transmit emergency notifications in near real time.

Second, the evaluation of computational performance demonstrated that the implementation of the Zero-Order Sugeno Fuzzy Logic algorithm was accurate and reliable in classifying pipe condition anomalies, namely Normal, Small Leak, and Large Leak. Through inference accuracy testing using ten different random data scenarios, the microcontroller output achieved an accuracy rate of 100%, with a deviation of 0.00% compared with theoretical mathematical calculations. The system also demonstrated robustness in processing ambiguous input values, particularly in edge-case conditions, thereby addressing the limitations of static threshold-based methods that commonly generate false alarms.

REFERENCES

Alijoyo, F. A., Janani, S., Santosh, K., Shweihat, S. N., Alshammry, N., Ramesh, J. V. N., & El-Ebiary, Y. A. B. (2024). Enhancing AI interpretation and decision-making: Integrating cognitive computational models with deep learning for advanced uncertain reasoning systems. *Alexandria Engineering Journal, 99*, 17–30.

Barry, J. D. (2021). *Using Brillouin frequency shift to detect underground pipe leaks*. University of Pretoria.

Cheng, S., Quilodrán-Casas, C., Ouala, S., Farchi, A., Liu, C., Tandeo, P., Fablet, R., Lucor, D., Iooss, B., & Brajard, J. (2023). Machine learning with data assimilation and uncertainty quantification for dynamical systems: A review. *IEEE/CAA Journal of Automatica Sinica*, *10*(6), 1361–1387.

Hidayat, R., Putra, A. R., & Saputra, D. (2024). Water leak detection system on PDAM pipelines based on the Internet of Things. *E-Journal of Indo Global Mandiri University, 9*(1), 15–26.

Irawan, A., Nugroho, Y., & Santoso, D. (2025). Optimization of water distribution pipeline leak detection system using pressure and flow sensors. *Journal of the State Polytechnic of Malang, 12*(1), 33–44.

Kabir, H. M. D., Khosravi, A., Hosen, M. A., & Nahavandi, S. (2018). Neural network-based uncertainty quantification: A survey of methodologies and applications. *IEEE Access, 6*, 36218–36234.

Mathye, R. P. (2023). *Development of an integrated water loss management model*. University of Johannesburg.

Matimolane, S. (2026). Governance failure and corruption perpetuate water insecurity in democratic South Africa. *Discover Water, 6*(1), 53.

Pillay, D. (2020). *Investigating the cost and impact of water leakages in the Midrand region using the economic model for leakage management*. University of Pretoria.

Prasetyo, D., Ardiansyah, M., & Ramadhan, F. (2025). Implementation and comparison of the Fuzzy Sugeno method in fluid monitoring systems. *Midpublisher Journal of Engineering, 5*(1), 1–12.

Pratama, A. R., et al. (2023). Monitoring temperature and humidity in greenhouses using IoT-based sensors. *J-AGENT (Journal of Agriculture and Environment), 3*(4), 369–379.

Pratama, R. (2021). Implementation of the Sugeno fuzzy method on a microcontroller-based control system. *Journal of Information Technology and Computer Science*.

Putra, D. (2022). The IoT-based monitoring system uses the Sugen fuzzy method. *Journal of Polynesian Informatics*.

Putri, N., Handayani, T., & Setiawan, B. (2025). Water leak detection system on home lines based on the Internet of Things. *Journal of Information Technology, Universitas Brawijaya, 10*(1), 22–34.

Rahman, A., Fauzan, M., & Hidayatullah, R. (2024). ESP8266-based pressure transmitter sensor model for water monitoring systems. *Scientific Journal of Informatics Education, 8*(2), 77–86.

Sarisen, D. (2024). *Simulating intermittent water supply systems under uncertainty*. University of Exeter.

Santoso, E., Prabowo, A., & Yuliana, D. (2022). Implementation of Fuzzy Sugeno on automatic control system: Case study of fan and LPG control. *Scientific Journal of Informatics Management, Unisma, 6*(2), 45–56.

Setiawan, H., Nurhadi, D., & Wijanarko, P. (2017). Water pipe leak detection using the wavelet transform method. *Journal of Electrical Engineering, Sriwijaya University, 9*(1), 1–10.

Siregar, M., Yusuf, H., & Lestari, R. (2024). *Design and build an IoT-based PDAM pipeline leak detection system*. Repository of Makassar State University.

Suhartono, A., Maulana, I., & Pratiwi, E. (2023). Fluid flow anomaly detection using a fuzzy inference system. *International Journal of Engineering Trends and Technology, 71*(3), 56–64.

Susanto, B., Wijaya, R., & Anwar, S. (2025). *Leak detection system for water pipe areas based on the Internet of Things*. Repository of Sultan Agung Islamic University.