

PROPOSAL CAPSTONE PROJECT
Smart Water Leak Detection Berbasis AloT



Disusun Oleh:

Rifky Akhsanul Hadi	Teknik Informatika	225150207111012
Afifah Maulidiah	Teknik Komputer	225150300111033
Reyhan Rashid Rizqullah	Teknik Komputer	225150300111047
Rihan Hidayat	Teknik Komputer	225150301111009
Muhamad Aditya Sanjaya	Teknik Komputer	225150301111011

FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
2025

LEMBAR PENGESAHAN

PROPOSAL CAPSTONE PROJECT

JUDUL: Smart Water Leak Detection Berbasis AIoT

Anggota Tim:

Rifky Akhsanul Hadi	Teknik Informatika	225150207111012
Afifah Maulidiah	Teknik Komputer	225150300111033
Reyhan Rashid Rizqullah	Teknik Komputer	225150300111047
Rihan Hidayat	Teknik Komputer	225150301111009
Muhamad Aditya Sanjaya	Teknik Komputer	225150301111011

Laporan ini disusun untuk memenuhi nilai pada Mata Kuliah Cloud Computing, Fault Tolerant System, serta Embedded Artificial Intelligence yang merupakan bagian dari Capstone Project Mahasiswa Prodi Ilmu Komputer

Malang, 28 Maret 2025

Menyetujui,

Dosen Koordinator MK Capstone,

Barlian Henryranu Prasetyo, S.T., M.T., Ph.D
NIP/NIK. 2011028210241001

RINGKASAN

Penelitian ini mengembangkan sistem deteksi kebocoran pipa berbasis IoT dengan memanfaatkan dua sensor arus, satu sensor tekanan, dan mikrokontroler ESP32. Model pohon keputusan yang telah dilatih di Google Colab di-embed dalam ESP32 untuk melakukan inferensi secara langsung di edge device. Data sensor dibaca setiap lima detik, diolah oleh model, lalu hasilnya dikirim melalui AWS IoT Core dan divisualisasikan pada dashboard Blynk.

Prototipe pipa simulasi dilengkapi dua titik kebocoran yang diwakili oleh kran air, sehingga pengujian meliputi kondisi aliran normal, bocor pada titik pertama, bocor pada titik kedua, dan bocor pada kedua titik sekaligus.

Hasil pengujian menunjukkan sistem mampu mendeteksi keberadaan dan lokasi kebocoran dengan akurasi tinggi, respons real-time, dan notifikasi instan kepada teknisi lapangan. Keunggulan utama terletak pada kemampuan inferensi lokal tanpa ketergantungan penuh pada koneksi internet serta keamanan data melalui sertifikat TLS di AWS IoT Core.

Rekomendasi untuk pengembangan selanjutnya meliputi peningkatan sensitivitas model menggunakan algoritma ensemble, mekanisme buffering data saat koneksi terputus, dan perancangan enclosure IP-rated untuk ketahanan lapangan.

DAFTAR ISI

BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah	2
1.4. Tujuan	3
BAB II TINJAUAN PUSTAKA	4
2.1. Internet of Things (IoT) dalam Sistem Smart Water Meter	4
2.2. Penerapan AI untuk Deteksi Anomali dalam Konsumsi Air	13
BAB III METODOLOGI	14
3.1. Analisis constrain	14
3.2. Analisis kebutuhan	15
3.3. Analisis solusi	17
3.4. Skenario Pengujian Sistem	17
BAB IV PERANCANGAN DAN IMPLEMENTASI	20
4.1. Perancangan Sistem	20
4.2. Implementasi Sistem	21
ANALISIS HASIL DAN PEMBAHASAN	28
5.1. Pengujian Komunikasi ESP 32 Dengan Server	28
5.2. Pengujian Komunikasi Server dengan Web Blynk	28
5.3. Pengujian Alat Simulasi	28
5.4. Analisis Dan Pembahasan	31
5.4.1. Kelebihan Sistem	32
5.4.2. Kekurangan Sistem	32
KESIMPULAN DAN SARAN	33
6.1. KESIMPULAN	33
6.2. SARAN	33
6.2.1. Dokumentasi (Foto-Foto selama kegiatan)	36
6.2.2. Screen Shoot Upload Video Presentasi berikut link nya	36
6.2.3. Barcode/link Program/sistem	36

DAFTAR TABEL

Tabel 2.1. Spesifikasi Water Pressure Sensor G1/4 1.2 Mpa	7
Tabel 2.2. Spesifikasi Water Flow Sensor YF-S201	9
Tabel 2.3 Spesifikasi Esp32	11
Tabel 3.1 Komponen dan Fungsi Utama Perangkat Keras	16
Tabel 3.2 Sofware dan Fungsi Perangkat Lunak	17
Tabel 5.1 Arus, Tekanan, dan Output ML Ketika Stop Valve ditutup	29
Tabel 5.2 Arus, Tekanan, dan Output ML Ketika Stop Valve dibuka 90°	30
Tabel 5.3 Arus, Tekanan, dan Output ML Ketika Kran Kebocoran 1 dibuka 90°	31
Tabel 5.4 Arus, Tekanan, dan Output ML Ketika Kran Kebocoran 2 dibuka 90°	31
Tabel 5.5 Arus, Tekanan, dan Output ML Ketika Kran Kebocoran 1 dan 2 dibuka 90°	32

DAFTAR GAMBAR

Gambar 2.1. Water Pressure Sensor G1/4 1.2 Mpa	6
Gambar 2.2. Water Flow Sensor YF-S201	7
Gambar 2.3. Tampak dalam Water Flow Sensor YF-S201	8
Gambar 2.4 ESP32 Dev. Board Pinout	10
Sumber: (Ardutech, 2020)	10
Gambar 2.5 Resistor 10k dan resistor 15k	11
Gambar 4.1. Sketsa Diagram Fungsional Alat	20
Gambar 4.2. Implementasi Prototype Simulasi Kebocoran Pipa	22
Gambar 4.3. Koneksi Sensor Arus dan Tekanan ke ESP32	23
Gambar 4.4. Flowchart Program Arduino IDE	25
Gambar 4.5 . Flowchart Program Google Colab	26
Gambar 4.6. Konfigurasi AWS Thing Untuk ESP 32	27
Gambar 4.7. Konfigurasi AWS Policy Thing untuk ESP 32	27
Gambar 4.8. Konfigurasi WiFi, Endpoint, dan Certificate pada ESP 32	28
Gambar 4.9. Fungsi untuk Melakukan Koneksi ke AWS pada ESP 32	28
Gambar 5.1. Komunikasi ESP 32 Dengan Server AWS	30

DAFTAR LAMPIRAN

Dokumentasi 1. Merakit Koneksi Sensor dengan ESP 32	38
Dokumentasi 2. Pembuatan Prototype Simulasi Kebocoran Pipa	39
Dokumentasi 3. Percobaan Alat pada Prototype Simulasi Kebocoran Pipa	39
Dokumentasi 4. Optimasi Prototype Simulasi Kebocoran Pipa	40
Dokumentasi 5. Percobaan Kedua Prototype Simulasi Kebocoran Pipa	41
Dokumentasi 6. Optimasi Prototype Simulasi Kebocoran Pipa	41
Dokumentasi 7. Pembuatan Laporan dan Program	42
Dokumentasi 8. Screenshot Upload Video Presentasi	43

BAB I PENDAHULUAN

1.1. Latar Belakang Masalah

Air bersih diperoleh melalui pengelolaan air dari sumber mata air alami atau waduk buatan di daerah tertentu, yang kemudian mendistribusikannya ke semua rumah. Namun, sering kali terjadi masalah kebocoran pipa. Kebocoran di pipa dapat disebabkan oleh beberapa faktor, seperti umur pipa, tidak tepat instalasi, dan kondisi lingkungan. Berdasarkan data dari situs PDAM Tirta Benteng Tangerang, tingkat kehilangan air sebesar 33% menunjukkan bahwa operasional masih belum efisien. Menurut laporan tahun 2017, angka kehilangan air di seluruh Indonesia mencapai 32,80%, sementara angka kehilangan air PDAM sehat adalah 21,71% dan diperkirakan sebesar 20% (Kusuma, 2021).

Kebocoran yang dibiarkan dalam waktu lama dapat menyebabkan kerugian besar. Semakin lama kebocoran tidak terdeteksi, semakin banyak fluida yang hilang, yang pada akhirnya menimbulkan kerugian finansial. Deteksi kebocoran adalah metode untuk mengidentifikasi adanya kebocoran. Alat ini memanfaatkan mekanika fluida dengan data tekanan yang diperoleh melalui sensor arus dan tekanan, serta menggunakan ESP32 sebagai mikrokontroler. Ketika pipa mengalami kebocoran, terjadi peningkatan arus dan penurunan tekanan dibandingkan dengan kondisi pipa sebelum kebocoran terjadi.

Sensor arus bekerja dengan mendeteksi jumlah aliran air melalui putaran baling-baling yang menghasilkan pulsa. Semakin cepat aliran air, semakin tinggi frekuensi pulsa yang terbaca oleh mikrokontroler. Sementara itu, sensor tekanan menghasilkan sinyal tegangan analog yang berubah sesuai dengan besar tekanan air dalam pipa. Perubahan signifikan pada aliran dan tekanan dapat mengindikasikan adanya kebocoran.

Alat ini dirancang untuk mendeteksi dan menentukan lokasi kebocoran pada pipa PDAM yang lurus, terutama pada pipa utama/primer dan sekunder yang memiliki tekanan tinggi. Alat ini kurang efektif untuk pipa tersier yang langsung melayani pelanggan karena tekanan air yang rendah (di bawah ± 10 Psi). Dengan memanfaatkan dua sensor arus dan satu sensor tekanan, perangkat ini diharapkan dapat memperkecil area pencarian kebocoran. Data dari sensor diproses oleh mikrokontroler dan dikirim secara nirkabel ke cloud

AWS. Selanjutnya, data tersebut diambil oleh aplikasi Blynk untuk ditampilkan.

Alat ini menggunakan mikrokontroler ESP32 karena sudah dilengkapi dengan modul Wi-Fi terintegrasi, sehingga memungkinkan komunikasi nirkabel tanpa perlu modul tambahan. ESP32 berfungsi sebagai pengontrol rangkaian elektronik dan umumnya dapat menyimpan program didalamnya. Mikrokontroler ini umumnya terdiri dari CPU (Central Processing Unit), memori, I/O tertentu dan unit pendukung seperti Analog-to-Digital Converter (ADC) yang sudah terintegrasi di dalamnya.

1.2. Rumusan Masalah

Permasalahan yang dihadapi pada projek *capstone* ini adalah sebagai berikut

- Bagaimana mendesain alat yang dapat mendeteksi kebocoran pipa secara efektif?
- Bagaimana merancang alat yang mampu mengetahui letak kebocoran pipa?

1.3. Batasan Masalah

Batasan masalah projek *capstone* ini antara lain

- Alat ini kami tujuhan untuk digunakan oleh perusahaan PDAM, terutama petugas teknisi lapangan.
- Alat ini menerima input dari dua sensor arus dan satu sensor tekanan, lalu memprosesnya di ESP32 menggunakan algoritma Feedforward Neural Network (FNN) untuk mendeteksi kebocoran. Data sensor dan hasil analisis dikirim via Wi-Fi ke AWS IoT Core dan ditampilkan secara visual di Blynk.
- Alat ini menggunakan pengembangan perangkat lunak agile development dengan pendekatan pemrograman terstruktur.
- Alat ini menggunakan komponen mikrokontroler ESP32, dua sensor arus, dan satu sensor tekanan, serta software pendukung berupa Arduino IDE, Google Colab, AWS, dan Blynk.
- Alat ini diuji dengan mensimulasikan tiga skenario kebocoran, bocor di titik pertama, bocor di titik kedua, dan bocor di kedua titik secara bersamaan. Pengujian dilakukan sesuai dengan rancangan simulasi yang telah disiapkan, di mana terdapat dua titik kebocoran.

1.4. Tujuan

Tujuan yang akan dicapai dar projek *capstone* ini adalah sebagai berikut:

- Mendesain Alat yang mampu mendeteksi kebocoran pipa.
- Merancang Alat yang mampu mengetahui letak kebocoran pipa.

BAB II TINJAUAN PUSTAKA

2.1. Internet of Things (IoT) dalam Sistem Smart Water Meter

Konsep Internet of Things (IoT) mengacu pada konsep yang memungkinkan perangkat untuk terhubung satu sama lain dan berinteraksi satu sama lain melalui jaringan yang dikenal sebagai internet. Dalam sistem Smart Water Meter, IoT memainkan peran utama dalam mengumpulkan data konsumsi air secara real-time dan pengiriman data ke cloud untuk analisis lebih lanjut. Menurut penelitian yang dilakukan oleh Singh dan Ahmed (2020), penggunaan IoT dalam sistem pengelolaan air dapat meningkatkan efisiensi pemantauan dan deteksi kebocoran. Dengan adanya konektivitas IoT, data dari sensor flow meter dan tekanan air dapat dikirim ke cloud, memungkinkan pemantauan jarak jauh oleh pelanggan dan petugas PDAM.

IoT dalam sistem Smart Water Meter juga mendukung otomatisasi dan pengurangan keterlibatan manusia dalam pemantauan konsumsi air, sehingga meningkatkan efisiensi operasional dan mengurangi kesalahan manusia. Studi yang dilakukan oleh Ismail Essamlali, Hasna Nhaila dan Mohamed El Khaili, (2024) menunjukkan bahwa penerapan IoT dalam sistem pemantauan air secara real-time memungkinkan pengambilan keputusan yang lebih cepat dalam menangani kebocoran atau penggunaan air yang tidak wajar. Selain itu, kombinasi IoT dengan teknologi cloud computing memungkinkan penyimpanan dan analisis data dalam skala besar, memberikan wawasan yang lebih baik mengenai pola konsumsi air.

Implementasi AWS IoT Core sebagai platform pengelolaan data IoT juga telah diteliti dalam penelitian terbaru, yang menunjukkan bahwa AWS IoT Core mampu menangani berbagai jenis data sensor secara scalable dengan tingkat latensi yang rendah. Keunggulan AWS IoT Core dalam mengelola koneksi antara ESP32 dan cloud memungkinkan sistem Smart Water Meter tetap berjalan dengan efisien, bahkan dalam kondisi jaringan yang tidak stabil. Menurut penelitian oleh Borra et al., (2024), AWS IoT Core mengelola konektivitas perangkat IoT dan proses ingest data secara aman, sementara AWS Greengrass memungkinkan pemrosesan data lokal, yang sangat penting dalam lingkungan dengan keterbatasan konektivitas.

2.2. Penerapan AI untuk Deteksi Anomali dalam Konsumsi Air

Penerapan kecerdasan buatan (AI) dalam deteksi anomali telah menjadi bidang penelitian yang luas, terutama dalam pemantauan konsumsi air. Berdasarkan penelitian oleh M.N. Kanyama et al., (2024) model AI yang diterapkan dalam sistem pengelolaan air mampu mengenali pola penggunaan air pelanggan dan mendeteksi anomali seperti kebocoran atau penggunaan yang tidak wajar. Model AI dalam sistem Smart Water Meter dapat dilatih menggunakan data historis konsumsi air pelanggan untuk meningkatkan akurasi deteksi anomali.

Dalam proyek ini, digunakan TensorFlow Lite for Microcontrollers (TFLM) untuk memungkinkan pemrosesan AI langsung pada perangkat ESP32. Penelitian oleh Jhaveri, Chi and Wu (2024) menunjukkan bahwa TinyML, yang diimplementasikan menggunakan TFLM, dapat mengurangi latensi pengambilan keputusan dengan melakukan inferensi langsung di edge device. Hal ini mengurangi ketergantungan pada cloud computing, sehingga mempercepat respons sistem dalam mengidentifikasi potensi kebocoran sebelum data dikirim ke AWS SageMaker untuk analisis lebih lanjut.

Studi lain oleh Kanyama et al. (2024) membahas berbagai teknik machine learning dalam deteksi anomali di jaringan Smart Water Metering Networks (SWMNs). Penelitian ini mengungkap bahwa algoritma seperti k-Nearest Neighbors (kNN), Autoencoders, DBSCAN, Random Forest, dan Support Vector Machines (SVM) telah banyak digunakan dalam deteksi anomali dengan hasil yang signifikan. Selain itu, studi yang dilakukan oleh Mashhadi et al. (2021) menyatakan bahwa metode berbasis machine learning dapat secara efektif mengidentifikasi kebocoran dalam jaringan distribusi air, dengan pendekatan berbasis neural network dan regresi logistik menunjukkan kinerja terbaik. Dengan demikian, pemanfaatan AI tidak hanya dapat mendeteksi anomali tetapi juga membantu mengoptimalkan manajemen air dan meningkatkan efisiensi distribusi air.

2.3. Water Pressure Sensor G1/4 1.2MPa

Water pressure sensor merupakan komponen penting dalam sistem pemantauan distribusi air modern. Sensor ini digunakan untuk mengukur tekanan air dalam pipa, yang berguna dalam proses kontrol, pemantauan, dan deteksi anomali sistem perpipaan. Salah satu tipe sensor yang banyak digunakan adalah Water Pressure Sensor G1/4 1.2MPa, yaitu sensor tekanan air dengan rentang

maksimum 1.2 Megapascal (MPa) dan ulir koneksi standar G1/4 yang sesuai untuk sistem pipa kecil hingga menengah.



Gambar 2.1. Water Pressure Sensor G1/4 1.2 Mpa

Sensor ini bekerja dengan prinsip piezoelektrik atau strain gauge, di mana tekanan air akan menyebabkan deformasi mikro pada elemen sensitif di dalam sensor, yang kemudian dikonversi menjadi sinyal listrik analog. Output sinyal ini umumnya berkisar antara 0.5V hingga 4.5V untuk representasi tekanan 0–1.2MPa, menjadikannya ideal untuk diintegrasikan dengan mikrokontroler seperti ESP32 yang mendukung pembacaan ADC (Analog to Digital Converter) secara akurat (Honeywell, 2021).

Berdasarkan spesifikasi, Water Pressure Sensor G1/4 1.2MPa umumnya tahan terhadap tekanan berlebih dan memiliki resistansi internal tinggi terhadap kelembaban dan korosi, menjadikannya cocok untuk lingkungan tropis seperti Indonesia. Konsumsi dayanya rendah dan mampu bekerja dalam tegangan 5V, sehingga tidak membebani sistem ESP32 secara signifikan.

<i>Working Voltage</i>	DC 5±0.5V
<i>Output Voltage</i>	DC 0,5-4,5 Volt
<i>Working Current</i>	≤10mA (DC 5V)
<i>Working Pressure Rate Range</i>	0~1.2Mpa
<i>Max. Pressure</i>	2.4Mpa
<i>Destructive pressure</i>	3.0Mpa
<i>Working Temperature</i>	0 - 85°C

<i>Storage Temperature</i>	0~+ 105°C
<i>Measurement Accuracy</i>	±1.5%FS
<i>Response Time</i>	≤2.0ms
<i>Cycle Life</i>	1,000,000 pcs

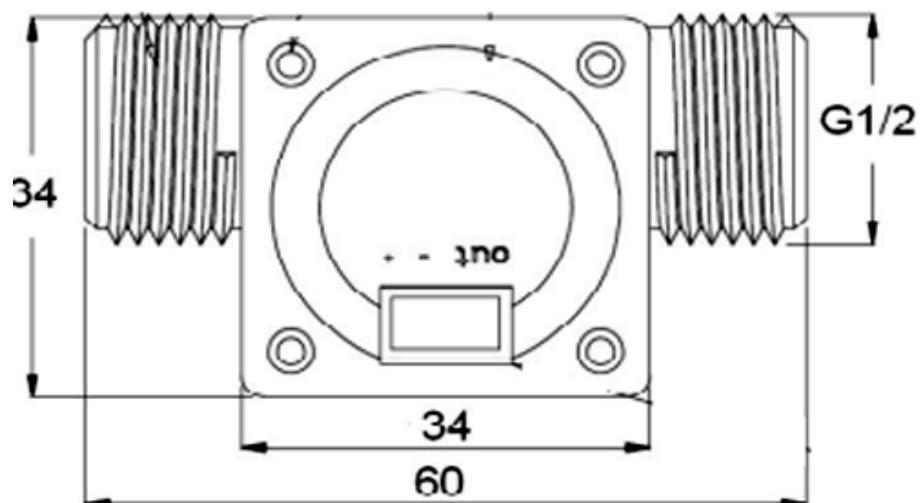
Tabel 2.1. Spesifikasi Water Pressure Sensor G1/4 1.2 Mpa

Water Pressure Sensor G1/4 1.2MPa memiliki sejumlah fitur unggulan yang menjadikannya cocok untuk digunakan dalam sistem Smart Water Meter berbasis IoT. Desainnya yang kompak dan mudah dipasang memungkinkan integrasi cepat ke dalam jaringan pipa tanpa perlu modifikasi tambahan. Sensor ini juga memiliki *high sealing performance* sehingga mampu mencegah kebocoran pada sambungan pipa, menjaga akurasi pembacaan tekanan.

Selain itu, penggunaan *Hall Effect* Sensor berkualitas tinggi menjamin sensitivitas dan stabilitas dalam pengukuran tekanan air, serta menghasilkan sinyal analog yang linier dan presisi. Sensor ini juga telah memenuhi standar RoHS (Restriction of Hazardous Substances), menunjukkan bahwa komponennya bebas dari bahan berbahaya dan aman untuk lingkungan

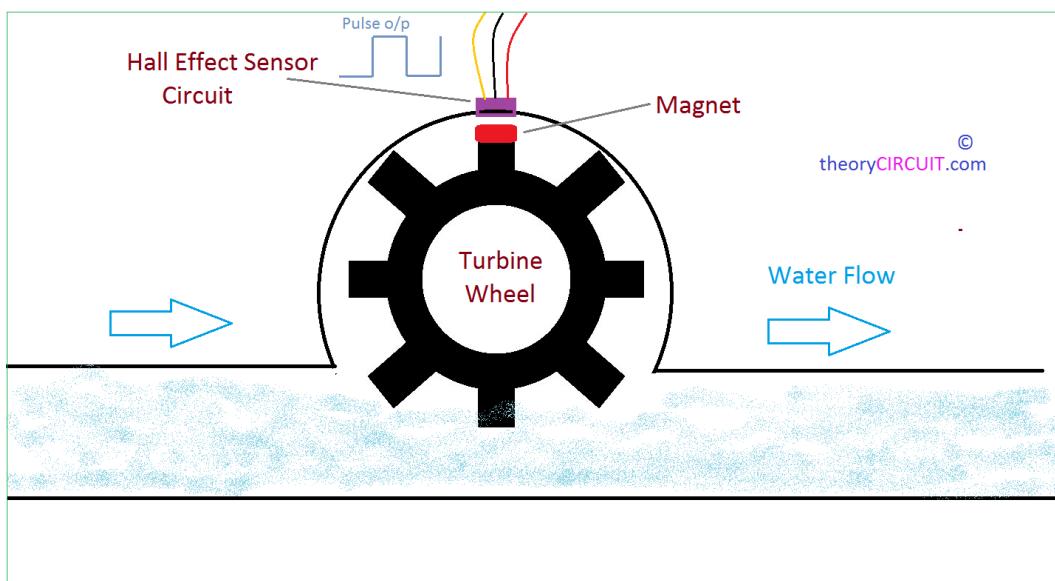
2.4. Water Flow Sensor YF-S201

Flow sensor YF-S201 merupakan salah satu jenis sensor pengukur aliran air berbasis efek Hall, yang banyak digunakan dalam aplikasi pemantauan air, termasuk sistem Smart Water Meter berbasis IoT.



Gambar 2.2. Water Flow Sensor YF-S201

Sensor ini bekerja dengan prinsip dasar rotasi baling-baling kecil yang terletak di dalam ruang aliran air. Ketika air mengalir melalui sensor, baling-baling akan berputar, dan putaran ini akan dipantau oleh sensor Hall Effect untuk menghasilkan sinyal pulsa. Jumlah pulsa yang dihitung dalam satuan waktu tertentu berkorelasi langsung dengan kecepatan aliran air, sehingga dapat dihitung sebagai flow rate dalam satuan liter per menit (L/min).



Gambar 2.3. Tampak dalam Water Flow Sensor YF-S201

Sensor YF-S201 dapat membaca rentang aliran air dari 1 hingga 30 liter per menit, dengan tegangan kerja sekitar 5V DC, dan menghasilkan output sinyal digital berbentuk pulsa, yang mudah dibaca menggunakan mikrokontroler seperti ESP32. Sensor ini sangat sesuai untuk pengukuran aliran air dalam pipa berdiameter 1/2 inch, yang juga digunakan dalam proyek ini, karena tingkat sensitivitas dan jangkauannya yang sesuai untuk kebutuhan rumah tangga maupun aplikasi skala kecil hingga menengah.

Dalam sistem smart water meter sensor YF-S201 digunakan pada dua titik strategis, yaitu sebelum titik simulasi bocor satu, dan di antara titik bocor satu dan dua. Perbedaan data antara kedua titik ini dapat dianalisis untuk mendeteksi adanya kebocoran, pencurian air, atau inefisiensi distribusi. Penelitian oleh Ayyasamy & Baskaran (2021) menyatakan bahwa penggunaan dua sensor aliran dalam sistem tertutup memungkinkan peningkatan akurasi dalam deteksi kebocoran hingga lebih dari **90%**, jika dipadukan dengan sensor tekanan dan model kecerdasan buatan.

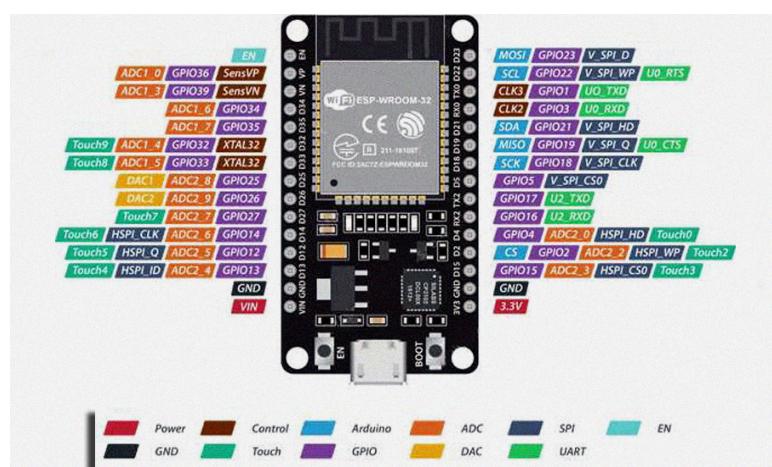
Sensor ini bekerja pada tegangan operasi 5V DC, dengan arus kerja rata-rata 15 mA dan menghasilkan output pulsa digital yang bisa langsung dihubungkan ke pin interrupt pada mikrokontroler seperti ESP32. Rentang aliran yang dapat dideteksi berada di kisaran 1 hingga 30 liter per menit (L/min), dengan frekuensi output pulsa sekitar $7,5 \times Q$, di mana Q adalah laju aliran dalam liter per menit. Tingkat akurasi sensor ini mencapai $\pm 3\%$ dalam kondisi ideal, dan mampu bekerja pada suhu cairan antara -25°C hingga 80°C, dengan tekanan maksimum operasi sekitar 1.75 MPa.

<i>Working Voltage</i>	DC 4.5V~24V
<i>Normal Voltage</i>	DC 5V~18V
<i>Max. Working Current</i>	15mA (DC 5V)
<i>Load capacity</i>	≤ 10 mA (DC 5V)
<i>Flow Rate Range</i>	1~30L/min
<i>Load Capacity</i>	≤ 10 mA (DC 5V)
<i>Operating Temperature</i>	$\leq 80^\circ\text{C}$
<i>Liquid Temperature</i>	$\leq 120^\circ\text{C}$
<i>Operating Humidity</i>	35%~90%RH
<i>Allowing Pressure</i>	≤ 1.75 MPa
<i>Storage Temperature</i>	-25~+ 80°C
<i>Storage Humidity</i>	25%~95%RH
<i>Electric strength</i>	1250V/min
<i>Insulation resistance</i>	≥ 100 MΩ
<i>External threads</i>	1/2"
<i>Outer diameter</i>	20mm
<i>Intake diameter</i>	9mm
<i>Outlet diameter</i>	12mm

Tabel 2.2. Spesifikasi Water Flow Sensor YF-S201

2.5. ESP32

ESP 32 dibuat dan dikembangkan oleh sebuah perusahaan China yang berbasis di Shanghai, yaitu Espressif Systems. Kemudian diproduksi oleh TSCM menggunakan proses 40nm. ESP 32 menggunakan mikroprosesor Tensilica Xtensa LX6 dan tersedia dalam dual-core atau single-core. Modul ESP 32 memiliki fitur Analog to Digital Converter (ADC), dimana ESP 32 memiliki kemampuan merubah data analog menjadi data digital. ESP32 memiliki prosesor dual-core Tensilica Xtensa LX6, dengan kecepatan hingga 240 MHz dan didukung dengan berbagai fitur seperti Wi-Fi, Bluetooth, GPIO, ADC, DAC, PWM, dan lainnya dalam satu chip. Terdapat 15 Pin Input atau Output yang dapat merubah data atau sinyal analog menjadi digital yang digunakan untuk menghubungkan sensor dan aktuator. ESP 32 menggunakan 11b/g/n wifi transceiver dan 48 pin GPIO.



Gambar 2.4 ESP32 Dev. Board Pinout

Sumber: (Ardutech, 2020)

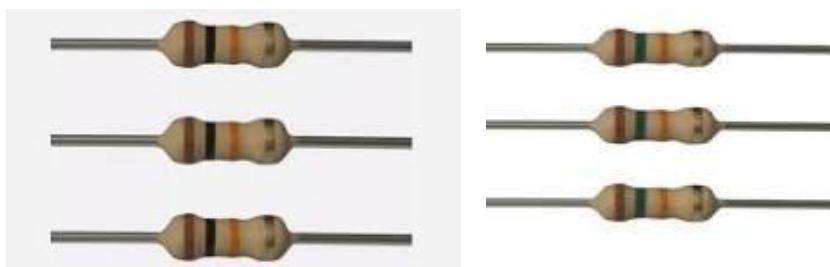
<i>Processor</i>	Dual-core Tensilica LX6, 32-bit, up to 240 MHz
<i>Flash Memory</i>	4MB (varies by board version)
<i>SRAM</i>	520 KB
<i>Wi-Fi</i>	IEEE 802.11 b/g/n
<i>Bluetooth</i>	Bluetooth v4.2 BR/EDR and BLE
<i>Operating Voltage</i>	5V via micro USB

<i>GPIO Pins</i>	Up to 36 GPIOs
<i>ADC</i>	12-bit, up to 18 channels
<i>DAC</i>	2 channels, 8-bit each
<i>Interfaces</i>	SPI, I2C, UART, CAN, PWM, Capacitive Touch, Hall Sensor
<i>Operating Temperature</i>	-40°C to +125°C

Tabel 2.3 Spesifikasi Esp32

2.6. Resistor

Resistor merupakan komponen pasif elektronik yang memiliki fungsi utama untuk membatasi arus dan menurunkan tegangan dalam suatu rangkaian listrik. Salah satu penerapan umum resistor adalah dalam konfigurasi pembagi tegangan (*voltage divider*), yang digunakan untuk menurunkan tegangan dari suatu sumber agar sesuai dengan kebutuhan komponen lainnya dalam rangkaian.



Gambar 2.5 Resistor 10k dan resistor 15k

Dalam proyek ini, digunakan dua buah resistor dengan nilai resistansi $15\text{k}\Omega$ dan $10\text{k}\Omega$ yang disusun secara seri sebagai pembagi tegangan. Konfigurasi ini berfungsi untuk menurunkan tegangan output dari sensor tekanan air (water pressure sensor) yang beroperasi pada level tegangan 5V menjadi tegangan yang aman untuk dibaca oleh mikrokontroler ESP, yang hanya mampu menangani tegangan maksimum sebesar 3.3V pada pin input analognya.

Rangkaian pembagi tegangan bekerja berdasarkan hukum Ohm dan prinsip pembagian tegangan, di mana tegangan input V_{in} dibagi secara

proporsional terhadap nilai resistansi dua resistor yang tersusun seri. Rumus dasar dari pembagi tegangan adalah sebagai berikut:

$$V_{out} = V_{in} \times \frac{R_2}{R_1 + R_2}$$

Dengan $R_1 = 15k\Omega$ dan $R_2 = 10k\Omega$, maka tegangan output yang masuk ke ESP adalah:

$$V_{out} = 5V \times \frac{10k\Omega}{15k\Omega + 10k\Omega} = 5V \times \frac{10}{25} = 2V$$

Tegangan sebesar 2V ini berada dalam rentang aman untuk dibaca oleh ESP, sehingga dapat menghindari kerusakan pada mikrokontroler akibat overvoltage.

Penggunaan pembagi tegangan ini merupakan teknik yang efisien, sederhana, dan murah dalam menyesuaikan level tegangan antar komponen yang memiliki batasan tegangan kerja berbeda. Namun, perlu diperhatikan bahwa konfigurasi ini hanya cocok untuk aplikasi dengan arus yang sangat kecil, seperti input ADC (Analog to Digital Converter) pada ESP, dan tidak direkomendasikan untuk mengatur tegangan dengan beban arus besar.

BAB III METODOLOGI

3.1. Analisis *constraint*

Dalam merancang sistem Smart Water Meter Prediksi Kebocoran Air Berbasis AloT, terdapat beberapa batasan teknis dan desain yang perlu diperhatikan agar sistem dapat berfungsi secara optimal. Batasan ini tidak berkaitan dengan aspek ekonomi, manufacturability, maupun sustainability, melainkan lebih pada kendala yang dapat mempengaruhi performa dan fungsionalitas sistem secara langsung saat diimplementasikan. Meskipun masih dalam bentuk prototipe. Berikut ini adalah analisis beberapa aspek constraint :

1. Keterbatasan Daya Komputasi Mikrokontroler

Sistem ini menggunakan algoritma Feedforward Neural Network yang ditanam langsung ke dalam mikrokontroler ESP32. Namun, ESP32 memiliki keterbatasan dalam hal kapasitas memori dan pemrosesan. Hal ini menjadi kendala teknis karena hanya model machine learning sederhana dapat dijalankan secara lokal. Jika kompleksitas model ditingkatkan untuk akurasi lebih tinggi, maka pemrosesan harus dipindah ke cloud, yang menyebabkan ketergantungan pada koneksi internet.

2. Konektivitas Internet di Lokasi Uji

Sistem ini membutuhkan koneksi internet stabil untuk mengirim data ke AWS IoT Core dan menampilkan informasi pada dashboard Blynk. Dalam pengujian prototipe di kampus Wi-Fi tersedia dan stabil. Namun, sistem dirancang untuk mengatasi permasalahan distribusi air PDAM, yang tidak selalu berada di lokasi dengan koneksi internet yang andal. Hal ini menjadi kendala karena dapat menyebabkan keterlambatan pengiriman data atau bahkan kegagalan sistem monitoring secara real-time.

3. Ketahanan Terhadap Kondisi Lingkungan

Walaupun prototipe diuji di lingkungan terkontrol, desain akhir dari sistem perlu memperhitungkan kondisi lapangan seperti kelembaban tinggi, cipratan air, atau suhu ekstrem. Tanpa pelindung tambahan seperti casing tahan air (IP-rated), komponen elektronik seperti ESP32 dan sensor akan rentan mengalami gangguan atau kerusakan. Hal ini menjadi batasan teknis penting yang harus diselesaikan jika sistem akan digunakan dalam jangka panjang di lingkungan luar ruang.

3.2. Analisis kebutuhan

- Kebutuhan Fungsional

Dalam pengembangan sistem Smart Water Meter berbasis IoT ini, analisis kebutuhan dilakukan untuk memastikan bahwa sistem yang dibangun dapat mengatasi permasalahan yang telah diidentifikasi

1. Pemantauan Aliran dan Tekanan Air

Sistem harus mampu membaca data dari:

- 2 sensor flow YF-S201
- 1 sensor tekanan G1/4 1.2MPa

2. Deteksi Kebocoran

Sistem harus mampu mendeteksi adanya anomali/kebocoran pada pipa berdasarkan

- Perbedaan data aliran dari dua titik sensor flow
- Perubahan signifikan tekanan air

3. Pemrosesan AI

Sistem harus menggunakan model machine learning (Feedforward Neural Network) untuk mengenali pola normal dan anomali dan Model AI harus bisa dijalankan di mikrokontroler (Edge AI via ESP32 dengan TinyML/TFLM).

4. Pengiriman dan Penyimpanan Data

Data harus dikirim ke AWS IoT Core untuk penyimpanan dan analisis lanjutan dan Sistem harus mendukung pengiriman data secara real-time melalui koneksi Wi-Fi.

5. Visualisasi dan Notifikasi

Aplikasi berbasis Blynk digunakan untuk menampilkan data dan notifikasi kebocoran secara visual kepada pengguna

- Kebutuhan Non-Fungsional

Kebutuhan non-fungsional yang harus dipenuhi meliputi skalabilitas sistem, keandalan dalam menghadapi gangguan teknis, efisiensi dalam penggunaan daya dan bandwidth, serta keamanan data yang dikirim melalui jaringan.

1. Reliabilitas Sistem

Sistem harus tetap bisa melakukan deteksi meskipun sebagian komponen tidak aktif (Fault Tolerance).

2. Keamanan Data

Komunikasi data dari ESP32 ke AWS harus aman, misalnya menggunakan protokol MQTT dengan enkripsi TLS.

3. Efisiensi Energi

Sistem harus hemat energi karena bisa saja digunakan di lokasi outdoor dengan keterbatasan daya.

4. Ketahanan Lingkungan

Perangkat harus tahan terhadap:

- Air
- Suhu ekstrem
- Kelembaban tinggi

- Kebutuhan Perangkat Keras

Komponen	Fungsi Utama
ESP32	Mikrokontroler utama dengan Wi-Fi terintegrasi
2x YF-S201 Flow Sensor	Mendeteksi laju aliran air
1x Water Pressure Sensor G1/4 1.2MPa	Mendeteksi tekanan air dalam pipa
Resistor 15k dan 10k	Pembagi tegangan untuk ADC ESP32
Catu daya (Power Bank/Adaptor)	Memberikan suplai tegangan ke ESP32
Modul koneksi ke AWS	Melalui Wi-Fi bawaan ESP32

Tabel 3.1 Komponen dan Fungsi Utama Perangkat Keras

- Kebutuhan Perangkat Lunak

Software	Fungsi
Arduino IDE	Pemrograman Esp32
TensorFlow Lite (TFLM)	Implementasi model AI di ESP32
Google Colab	Training model AI
AWS IoT Core	Penyimpanan dan pengolahan data cloud

Blynk	Visualisasi data dan notifikasi
-------	---------------------------------

Tabel 3.2 Software dan Fungsi Perangkat Lunak

3.3. Analisis solusi

Pendekatan yang digunakan dalam sistem ini adalah pendekatan hybrid computing, yaitu kombinasi antara edge computing dengan cloud computing untuk mendukung mendukung kebutuhan pemrosesan real-time, efisiensi data, dan pengelolaan sistem yang terpusat. Pendekatan hybrid ini menggabungkan dua lapisan utama dalam arsitektur komputasi modern, yaitu edge computing dan cloud computing

1. Pendekatan Edge Computing (ESP32)

Mikrokontroller ESP 32 bertindak sebagai pusat pemrosesan lokal. Sensor arus (flow meter) dan sensor tekanan (pressure) akan mengirimkan data ke ESP32, yang kemudian diproses secara real-time menggunakan model Feedforward Neural Network yang telah dideployed ke dalam perangkat. Model ini digunakan untuk mendeteksi adanya anomali atau kebocoran berdasarkan pola tekanan dan aliran air. Pendekatan ini memungkinkan sistem tetap dapat menjalankan proses deteksi awal meskipun terjadi gangguan koneksi internet, karena pemrosesan dilakukan langsung di perangkat (*on-device inference*).

2. Pendekatan Cloud Computing (AWS)

Setelah data diproses secara lokal, baik data mentah dari sensor maupun hasil klasifikasi oleh Feedforward Neural Network kemudian dikirimkan ke AWS IoT Core. Di sisi cloud, data akan disimpan dalam database AWS untuk kebutuhan pencatatan dan analisis lebih lanjut. Platform cloud juga memungkinkan integrasi dengan layanan seperti AWS Analytics atau Google Colab apabila di kemudian hari dibutuhkan pengembangan model yang lebih kompleks atau pelatihan ulang model berbasis data historis. Dashboard Blynk kemudian mengambil data dari cloud untuk ditampilkan kepada pengguna secara real-time.

3.4. Skenario Pengujian Sistem

Pengujian sistem bertujuan untuk memastikan bahwa setiap komponen dari sistem Smart Water Meter dapat berjalan dengan baik, mulai

dari pembacaan data sensor, inferensi kebocoran oleh AI, hingga pengiriman dan visualisasi data melalui cloud dan dashboard. Berikut adalah skenario pengujian sistem yang dirancang:

1. Pengujian Sensor Arus dan Tekanan

Tujuan: Mengetes akurasi pembacaan sensor flow YF-S201 dan sensor tekanan G1/4 1.2MPa.

Langkah Pengujian:

- Mengalirkan air pada sistem simulasi tanpa kebocoran.
- Membandingkan data yang dibaca sensor dengan standar output datasheet.
- Memastikan data yang diterima ESP32 sesuai dengan kondisi fisik.

2. Pengujian Deteksi Kebocoran

Tujuan: Mengetes efektivitas model Feedforward Neural Network dalam mendeteksi adanya kebocoran.

Langkah Pengujian:

- Menjalankan sistem dengan aliran air konstan.
- Membuka salah satu kran pada titik kebocoran simulasi.
- Mengecek apakah inferensi pada ESP32 mendeteksi status kebocoran dan titik lokasinya sesuai.

3. Pengujian Pengiriman Data ke AWS IoT Core

Tujuan: Memverifikasi bahwa data dari ESP32 terkirim secara real-time ke AWS Cloud.

Langkah Pengujian:

- Menjalankan sistem dengan koneksi Wi-Fi stabil.
- Melihat data terkirim ke AWS IoT Core dan tersimpan di database.
- Memeriksa apakah data juga dapat diambil oleh aplikasi Blynk.

4. Pengujian Visualisasi Dashboard Blynk

Tujuan: Mengetes apakah pengguna dapat melihat status sistem dan notifikasi kebocoran melalui aplikasi.

Langkah Pengujian:

- Menyimulasikan aliran normal dan aliran dengan kebocoran.
- Mengamati perubahan status di dashboard Blynk.
- Menilai keterlambatan (delay) dan kejelasan tampilan.

5. Uji Ketahanan Sistem

Tujuan: Mengetes stabilitas sistem dalam kondisi berjalan terus-menerus.

Langkah Pengujian:

- Menyalakan sistem selama 3–5 jam tanpa henti.
- Mengamati apakah terjadi error, reboot ESP32, atau gangguan komunikasi.
- Menilai kestabilan suhu perangkat dan konsistensi data sensor.

6. Uji Kelayakan Mitra (User Acceptance Test)

Tujuan: Memastikan sistem dapat digunakan oleh calon pengguna (misalnya teknisi PDAM).

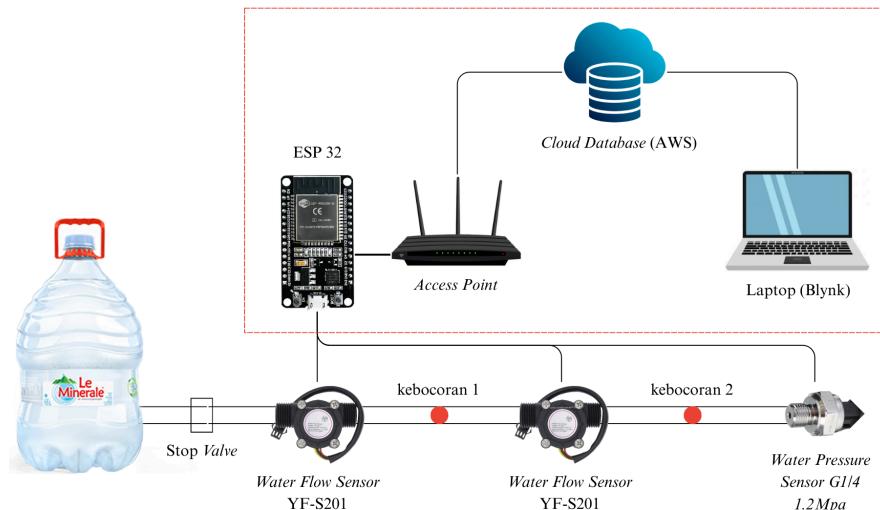
Langkah Pengujian:

- Menyediakan alat dan simulasi untuk diuji oleh mitra atau dosen
- Menggunakan lembar observasi dan form umpan balik.
- Mencatat kejelasan informasi, kemudahan penggunaan, dan potensi implementasi di lapangan.

BAB IV PERANCANGAN DAN IMPLEMENTASI

Pada bab ini berisi tahapan mengenai tahapan yang dilakukan dalam perancangan dan implementasi *project capstone*. Penjelasan diawali dengan blok fungsional sistem secara keseluruhan yang meliputi proses kerja alat dalam bentuk alur diagram sebagai rencana untuk penerapan. Implementasi perangkat keras membahas mengenai desain dan pembuatan alat simulasi. Implementasi perangkat lunak membahas mengenai pembuatan software IDE Arduino untuk membaca sensor pada ESP32 dan pembuatan machine learning untuk deteksi kebocoran beserta letaknya. Konfigurasi server database membahas pemanfaatan layanan AWS IoT Core untuk menerima dan menyimpan data sensor yang dikirim secara nirkabel dari ESP32, sehingga data dapat diakses dan divisualisasikan melalui platform Blynk.

4.1. Perancangan Sistem



Gambar 4.1. Sketsa Diagram Fungsional Alat

Sebelum melakukan implementasi perangkat keras, implementasi perangkat lunak dan konfigurasi server database, diperlukan sebuah perancangan blok fungsional sistem berupa blok diagram yang menjelaskan kerja secara keseluruhan *project capstone* ini.

Pada Gambar 4.1, aliran air dimulai dari sumber (galon atau reservoir), lalu melewati stop valve yang berfungsi mengatur tekanan awal dalam sistem. Setelah stop valve, air mengalir melalui dua water flow sensor

(YF-S201) yang masing-masing ditempatkan sebelum dan sesudah daerah pengujian kebocoran. Perbedaan frekuensi pulsa dari kedua sensor inilah yang digunakan untuk mendeteksi pola aliran tidak normal.

Selanjutnya, sebelum mencapai titik akhir, air juga melewati water pressure sensor G1/4 (1.2 MPa) untuk memantau tekanan absolut dalam pipa. Sinyal analog dari sensor tekanan dan pulsa digital dari kedua flow sensor dibaca oleh ESP32. Di sinilah seluruh data sensor dikumpulkan dan diproses.

ESP32 menjalankan algoritma Feedforward Neural Network untuk menentukan ada atau tidaknya kebocoran, serta memetakan letak kebocoran. Hasil klasifikasi dan data mentah kemudian dikirim secara nirkabel ke AWS IoT Core, di mana data disimpan dalam cloud database.

Untuk keperluan monitoring dan visualisasi real-time, sebuah aplikasi Blynk diakses melalui laptop/PC. Aplikasi ini mengambil data dari AWS dan menampilkan hasil inferensi model machine learning yang dijalankan di ESP32. Hasil ini menunjukkan apakah terjadi kebocoran atau tidak, sehingga teknisi lapangan dapat segera melakukan penanganan.

4.2. Implementasi Sistem

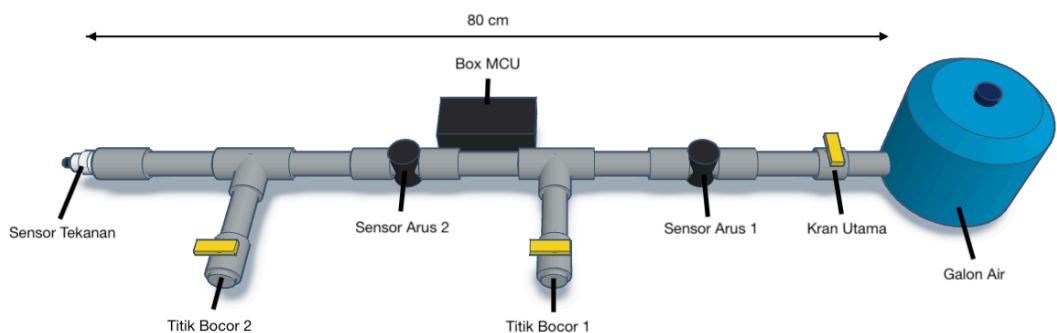
Bagian Implementasi Sistem mencakup seluruh aspek, yaitu pemasangan perangkat keras, pengembangan perangkat lunak, dan konfigurasi server beserta database.

4.2.1. Implementasi Perangkat Keras

Pada implementasi hardware yang dibahas terdiri dari implementasi prototype simulasi kebocoran pipa dan implementasi elektrik berupa rangkaian sensor beserta komunikasinya.

4.2.1.1. Implementasi Prototype Simulasi Kebocoran Pipa

Prototype simulasi kebocoran pipa ini dimaksudkan untuk jarak yang jauh dan dengan kondisi lingkungan yang seperti pada saluran distribusi air.



Gambar 4.2. Implementasi Prototype Simulasi Kebocoran Pipa

Bahan bahan yang digunakan untuk pembuatan alat simulasi kebocoran di atas antara lain:

1. Galon Air
2. Pipa PVC $\frac{1}{2}$ dim 0,7 meter
3. Lem PVC
4. Tee drat dalam 2 buah
5. Sok drat dalam 5 buah
6. Kran untuk kebocoran 2 buah
7. Kran untuk *stop valve* 1 buah
8. Nepel $\frac{5}{8}$ 1 buah
9. Box hitam plastik 1 buah

Dimensi alat simulasi:

1. Panjang : 80 cm
2. Lebar : 30 cm

Rancangan prototype di atas menggunakan pipa PVC diameter $\frac{1}{2}$ " sepanjang 80 cm, dua buah sensor arus YF-S201, satu buah sensor tekanan G1/4 (1,2 MPa), galon air sebagai reservoir, kran utama, dua kran simulasi kebocoran, serta kotak MCU yang menampung modul ESP32 dan breadboard.

Lalu sebagai alat simulasi, maka akan diibaratkan terdapat 2 titik kebocoran. Dimana letak kebocoran tersebut ditempatkan setelah sensor arus. Pada implementasi tersebut, kebocoran pipa digantikan dengan kran air. Jadi ketika salah satu kran air dibuka, maka akan diketahui kebocoran pipa yang dideteksi dari berkurangnya debit air. Titik kebocoran ditentukan

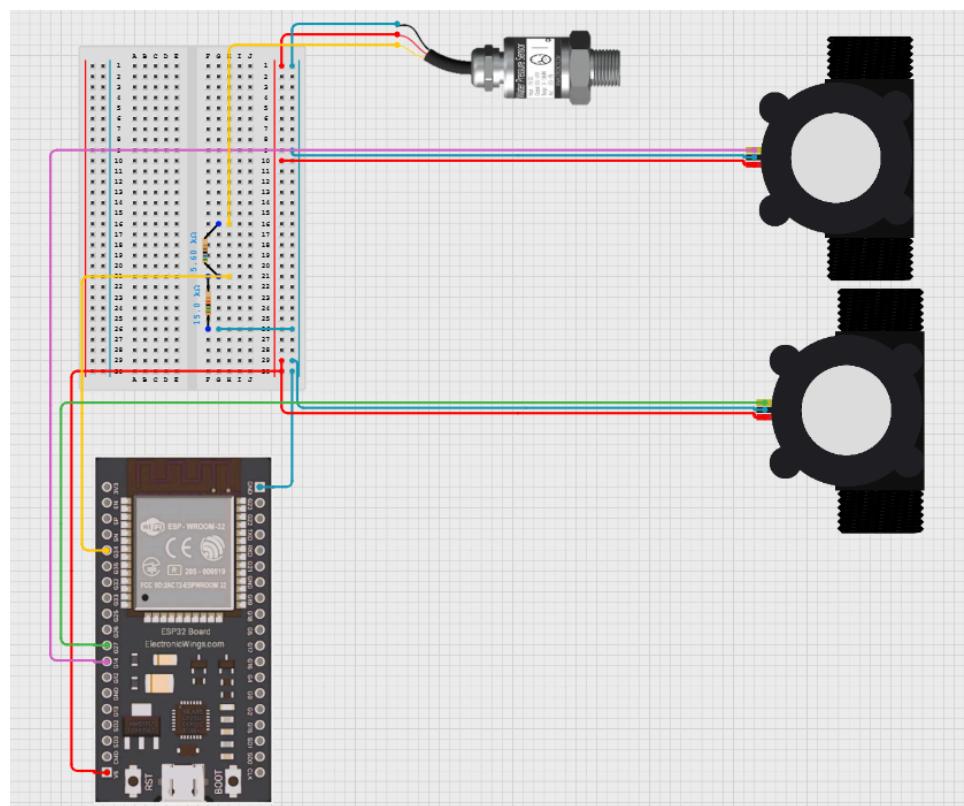
sebanyak 2 titik dikarenakan untuk menguji seberapa akurat prototype yang akan dibuat.

4.2.1.2. Implementasi Rangkaian Sensor Arus dan Tekanan



Gambar 4.3. Koneksi Sensor Arus dan Tekanan Ke ESP 32

Rangkaian Skematik Sensor Tekanan Pin Vout dari sensor dihubungkan dengan pin data pada ESP32 yang dikonfigurasi sebagai input yang mana pin data tersebut akan memberikan nilai ADC ketika terjadi perubahan nilai tekanan.



Gambar 4.3. Koneksi Sensor Arus dan Tekanan ke ESP32

Pada Gambar 4.3 mikrokontroler ESP32 dihubungkan pada sensor arus dan tekanan. Untuk sensor tekanan kabel merah dihubungkan pada pin 5 Volt, kabel hitam dihubungkan pada pin Gnd dan kabel kuning pada setiap sensor dihubungkan pada pin data yang berbeda di mikrokontroler ESP32

Pada konfigurasi pin yang digunakan pada mikrokontroler ESP32, terdapat tiga pin data yang berfungsi untuk mentransfer data dari sensor sebagai input. Dari ketiga pin tersebut, satu di antaranya merupakan pin analog, yaitu pin 34, yang digunakan untuk menghubungkan keluaran dari sensor tekanan air (water pressure sensor) ke input ESP32 melalui rangkaian pembagi tegangan (voltage divider) dengan nilai resistansi sebesar $20.600\ \Omega$. Selain itu, terdapat dua pin digital, yaitu pin 27 dan pin 14, yang masing-masing digunakan untuk menghubungkan dua sensor aliran air (flow water sensor) dengan mikrokontroler ESP32.

4.2.2. Implementasi Perancangan Perangkat Lunak

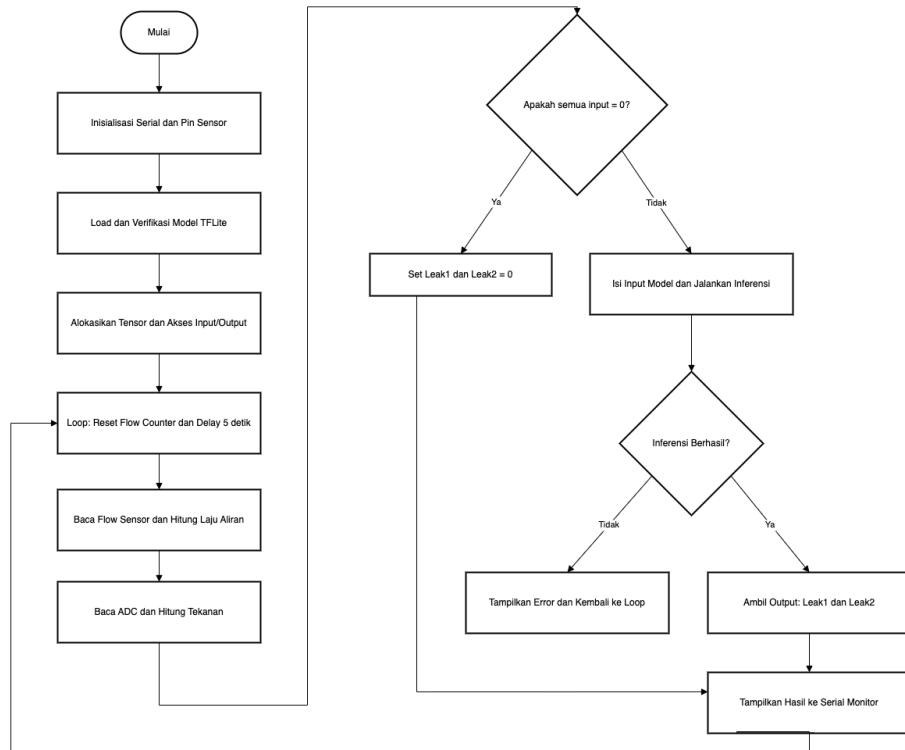
Pada implementasi perangkat software yang dibahas terdiri dari pemrograman pembacaan arus, tekanan, dan status kebocoran oleh sensor pada software Arduino IDE, dan implementasi pemrograman machine learning pada software Google Colab.

4.2.2.1. Implementasi Pemrograman Pembacaan Sensor Arus dan Tekanan

Pemrograman sensor arus dan tekanan dilakukan menggunakan aplikasi Arduino IDE. Sistem ini dirancang untuk membaca data dari dua sensor arus dan satu sensor tekanan.

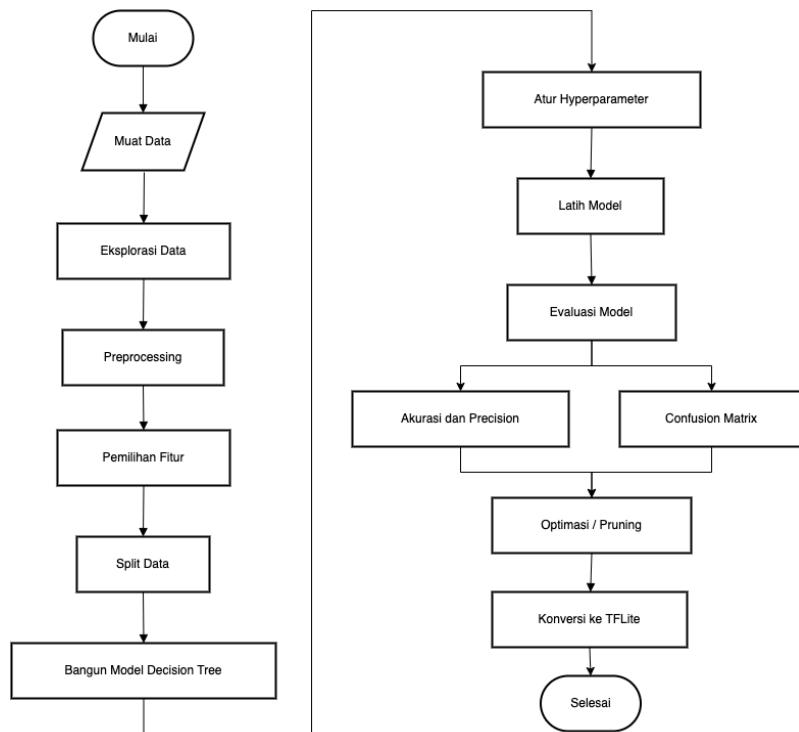
Agar sistem tidak menjalankan proses inferensi saat semua input sensor bernilai nol, maka ESP32 secara otomatis menetapkan nilai kebocoran pada kondisi tersebut sebagai nol. Namun apabila sensor mendekksi adanya aliran atau tekanan, maka data tersebut dimasukkan ke dalam model machine learning yang telah ditanamkan dalam ESP32 untuk dilakukan inferensi.

Hasil dari inferensi berupa status kebocoran pada dua titik ditampilkan secara langsung melalui Serial Monitor. Proses pembacaan sensor hingga inferensi dilakukan secara berulang setiap lima detik. Alur keseluruhan proses pemrograman ini ditunjukkan pada gambar 4.4.



Gambar 4.4. Flowchart Program Arduino IDE

4.2.2.2. Implementasi Pemrograman Machine Learning pada Google Colab



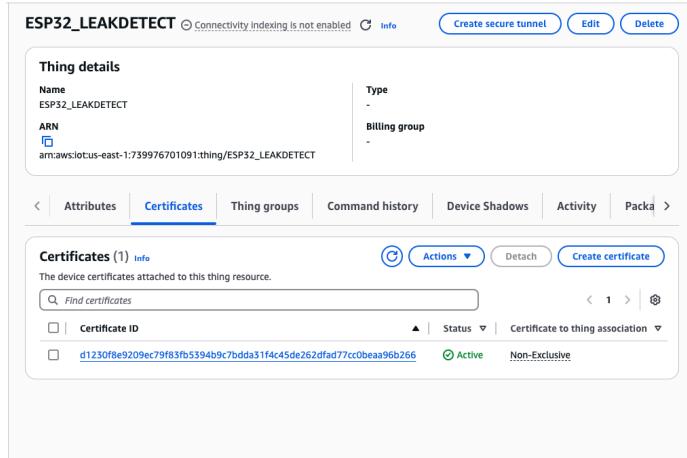
Gambar 4.5 . Flowchart Program Google Colab

Sesuai dengan gambar 4.5 yang ditampilkan. Pertama, *dataset training* diunggah ke Google Colab dan diperiksa untuk menghapus nilai duplikat atau anomali. Setelah bersih, data dibagi menjadi dua bagian—80 % untuk melatih model dan 20 % untuk menguji kinerja—agar bisa menilai keakuratan secara objektif. Proses pelatihan mengubah struktur pohon keputusan dengan menyesuaikan kedalaman cabang hingga diperoleh titik seimbang antara akurasi tinggi dan kerampingan model.

Setelah pelatihan, kinerja model dievaluasi menggunakan metrik akurasi dan matriks kebingungan untuk melihat seberapa baik model membedakan antara kondisi normal dan kebocoran. Cabang pohon yang kontribusinya rendah kemudian dipangkas (pruning) untuk mengurangi kompleksitas tanpa menurunkan ketepatan. Terakhir, struktur pohon yang sudah optimal dikonversi menjadi format TensorFlow Lite dan disimpan sebagai berkas ringkas, siap diunduh untuk dijalankan di perangkat ESP32.

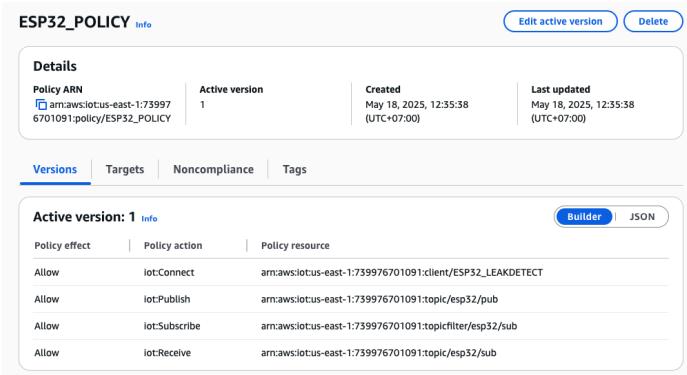
4.2.3. Konfigurasi Server Database

Konfigurasi server menggunakan layanan AWS IoT Core untuk menerima dan menyimpan data dari ESP32. Proses dimulai dengan membuat thing sebagai representasi perangkat seperti yang ditunjukkan pada gambar 4.6, diikuti oleh pembuatan sertifikat keamanan yang terdiri dari device certificate, kunci publik, kunci privat, dan root CA. Sertifikat ini digunakan untuk mengamankan koneksi antara ESP32 dan server.



Gambar 4.6. Konfigurasi AWS Thing Untuk ESP 32

Selanjutnya dibuat policy yang memberikan izin perangkat untuk melakukan operasi dasar seperti connect, publish, subscribe, dan receive seperti pada gambar 4.7. Kebijakan ini dihubungkan ke sertifikat, lalu dikaitkan dengan thing yang telah dibuat.



Gambar 4.7. Konfigurasi AWS Policy Thing untuk ESP 32

Setelah semua konfigurasi selesai, endpoint AWS dan data sertifikat dimasukkan ke dalam program ESP32 seperti yang terlihat pada gambar 4.8 dan gambar 4.9. Dengan pengaturan ini, perangkat dapat mengirim data sensor ke AWS IoT Core melalui protokol MQTT.

```
iolaws.iar     bloomfahh     Secretch
1  #include <ppm-pace.h>
2
3  #define SECRET
4  #define THINGNAME "ESP32_CAPSTONE"           //change this
5
6  const char WiFi_SSID[] = "medit_10_2022";      //change this
7  const char WiFi_PASSWORD[] = "japyyppajah";      //change this
8  const char AWS_TOT_ENDPOINT[] = "aqizfrudefe12n-ats.iot.us-east-1.amazonaws.com";    //change this
9
10 // Amazon Root CA
11 static const char AWS_CERT_CA[] = PROGEND R"EOF(
12 -----BEGIN CERTIFICATE-----
13 MIIDTC1CAQAwDQIBTBImYzSwJASd841khdJzByA0nkgkh1Gw0B042f
14 ASADyKQWQYDQGqXVUZLPMgAULCUEHgNqHeoRkaiw7QOQD0Bm46
15 b2g4oqMhCUDQsA0Wx0X1ELP0u9JwAfPmEh-NQDPMelwLwvQDQH0f0L
16 b3g4oqMhCUDQsA0Wx0X1ELP0u9JwAfPmEh-NQDPMelwLwvQDQH0f0L
17 b3g4oqMhCUDQsA0Wx0X1ELP0u9JwAfPmEh-NQDPMelwLwvQDQH0f0L
18 c4g4oqMhCUDQsA0Wx0X1ELP0u9JwAfPmEh-NQDPMelwLwvQDQH0f0L
19 c4g4oqMhCUDQsA0Wx0X1ELP0u9JwAfPmEh-NQDPMelwLwvQDQH0f0L
20 TjD9H0QDc0f1t8b2wIwet0tjghzKq1zpwV0t4f1nf0bdq1ca25L7/q
21 90C1184c0f1t8b2wIwet0tjghzKq1zpwV0t4f1nf0bdq1ca25L7/q
22 93700Wf0f0y1kr1A8p9V14tbcceWor346f7D5H7WV0C8/TDXef+0y0R
23 lgS1Subt3hAuFAMfCfAcwvDwBT3751/8Luwef7/ABRMw0B84f8EBfNC
```

Gambar 4.8. Konfigurasi WiFi, Endpoint, dan Certificate pada ESP 32

```
ESP32 Dev Module
leakaws.ino
1 // bermacam macam
2 #include <WiFi.h>
3
4 void connectAWS() {
5   WiFi.mode(WIFI_STA);
6   WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
7   Serial.println("Menghubungkan ke WiFi..."); 
8   while (WiFi.status() != WL_CONNECTED) {
9     delay(500);
10    Serial.print(".");
11  }
12
13 net.setCACert(AWS_CERT_CA);
14 net.setSSLCertificate(AWS_CERT_CRT);
15 net.setSSLPinVerify(AWS_CERT_PRIVATE);
16
17 client.setServer(AWS_IOT_ENDPOINT, 8883);
18 client.setCallback(messageHandler);
19
20 Serial.println("Menghubungkan ke AWS IoT..."); 
21 while (!client.connect(IOTHNGNAME)) {
22   Serial.print(".");
23   delay(100);
24 }
25 }
```

Gambar 4.9. Fungsi untuk Melakukan Koneksi ke AWS pada ESP 32

BAB V ANALISIS HASIL DAN PEMBAHASAN

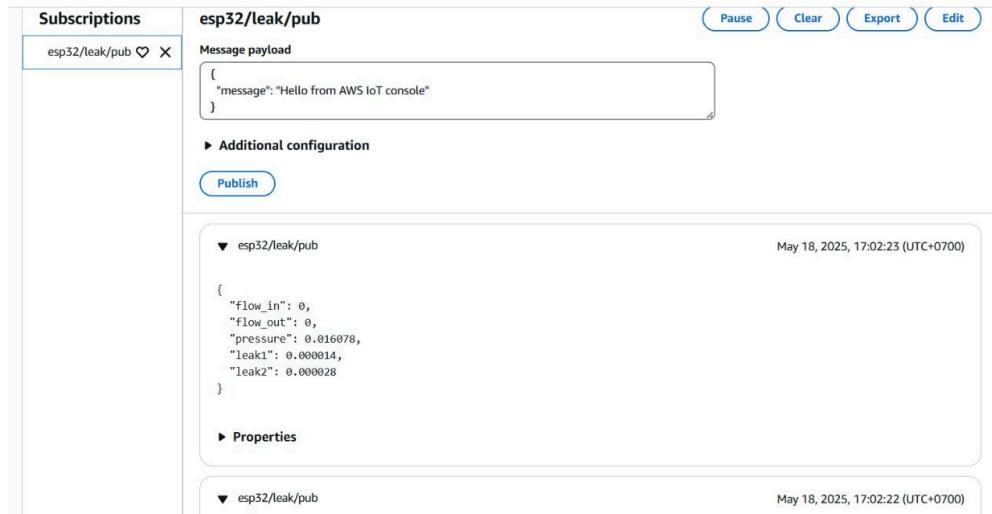
Dalam pengembangan sistem, pengujian dan analisis memegang peranan penting. Tujuan dari pengujian dan analisis adalah untuk memastikan apakah sistem berfungsi sesuai dengan rencana. Melalui hasil pengujian dan analisis, kelemahan-kelemahan sistem dapat diidentifikasi, sehingga memungkinkan untuk dilakukan perbaikan, pengembangan, dan penyempurnaan sistem.

Bab ini membahas pengujian dan analisis dari sistem yang telah dirancang dalam proyek capstone. Pengujian awal dilakukan secara parsial untuk mengevaluasi kinerja perangkat keras setiap unit. Setelah kinerja perangkat keras setiap unit diketahui, pengujian integrasi sistem secara keseluruhan akan dilaksanakan. Beberapa pengujian yang telah dilakukan adalah sebagai berikut

5.1. Pengujian Komunikasi ESP 32 Dengan Server AWS

Pengujian ini bertujuan untuk memastikan bahwa perangkat ESP32 dapat terhubung dan berkomunikasi dengan AWS IoT Core menggunakan protokol MQTT secara aman. ESP32 dikonfigurasi untuk mengirim data sensor dalam format JSON ke topik MQTT yang telah ditentukan, seperti esp32/leak/pub. AWS IoT Core, yang telah dikonfigurasi untuk berlangganan (subscribe) ke topik tersebut, menerima data yang dipublikasikan oleh ESP32.

Selama pengujian, data yang dipublikasikan oleh ESP32 berhasil diterima oleh AWS IoT Core dan dapat diamati melalui fitur MQTT Test Client pada dashboard AWS. Hal ini menunjukkan bahwa komunikasi antara ESP32 dan AWS IoT Core berjalan dengan baik, memungkinkan integrasi sistem untuk monitoring dan pengendalian secara real-time seperti yang terlihat pada gambar 5.1.



Gambar 5.1. Komunikasi ESP 32 Dengan Server AWS

5.2. Pengujian Alat Simulasi

Pengujian secara keseluruhan adalah kombinasi dari berbagai pengujian yang telah dilakukan serta pengujian kebocoran pipa pada prototipe simulasi distribusi air.

5.2.1. Pengujian Tanpa Aliran Air

Letak titik kebocoran dapat dilihat pada Gambar 4.1. Untuk pengujian tanpa aliran air, langkah langkah dari pengujian ini antara lain:

1. Memasang semua sensor pada titik yang sudah ada di pipa.
2. Menyalakan ESP 32 dan sambungkan sensor pada ESP 32.
3. *Stop Valve* ditutup rapat.
4. Mencatat nilai arus, tekanan, dan status kebocoran.

Hasil pengambilan data adalah sebagai berikut

No.	Sensor Arus 1 (l/min)	Sensor Arus 2 (l/min)	Sensor Tekanan (MPa)	Leak 1 (Yes/No)	Leak 2 (Yes/No)
Percobaan 1	0.00	0.00	0.00	No	No
Percobaan 2	0.00	0.00	0.00	No	No

Percobaan n 3	0.00	0.00	0.00	No	No
---------------	------	------	------	----	----

Tabel 5.1 Arus, Tekanan, dan Output ML Ketika Stop Valve ditutup

Dari Tabel 5.1 dapat disimpulkan bahwa pada saat stop valve ditutup rapat, tidak terjadi perubahan pada sensor dalam pipa. Dari tabel juga dapat disimpulkan bahwa model machine learning berhasil mendeteksi bahwa tidak ada kebocoran.

5.2.2. Pengujian Dengan Aliran Air

Letak titik kebocoran dapat dilihat pada Gambar 4.1. Untuk pengujian dengan aliran air, Langkah langkah dari pengujian ini antara lain:

1. Memasang semua sensor pada titik yang sudah ada di pipa.
2. Menyalakan ESP 32 dan sambungkan sensor pada ESP 32.
3. *Stop Valve* dibuka 90° (dibuka penuh).
4. Mencatat nilai nilai arus, tekanan, dan status kebocoran.

Hasil pengambilan data adalah sebagai berikut

No.	Sensor Arus 1 (l/min)	Sensor Arus 2 (l/min)	Sensor Tekanan (MPa)	Leak 1 (Yes/No)	Leak 2 (Yes/No)
Percobaan n 1	0.00	0.00	0.10	No	No
Percobaan n 2	0.00	0.00	0.05	No	No
Percobaan n 3	0.00	0.00	0.08	No	No

Tabel 5.2 Arus, Tekanan, dan Output ML Ketika Stop Valve dibuka 90°

Dari Tabel 5.2 dapat disimpulkan bahwa pada saat stop valve dibuka 90° , sensor tekanan saja yang mengalami perubahan dimana sensor tekanan naik ± 0.08 MPa. Dari tabel juga dapat disimpulkan bahwa model machine learning berhasil mendeteksi bahwa tidak ada kebocoran.

5.2.3. Pengujian Kebocoran dan Penentuan Lokasi

Letak titik kebocoran dapat dilihat pada Gambar 4.1. Untuk pengujian tanpa aliran air, langkah langkah dari pengujian ini antara lain:

1. Memasang semua sensor pada titik yang sudah ada di pipa.
2. Menyalakan Arduino dan sambungkan sensor pada ArduinoUno
3. Menyalakan pompa air dan atur stop valve pada ± 54 tertutup
4. Mencatat tekanan yang ditampilkan pada LCD dengan kondisi kran kebocoran 1 dibuka 90° , kondisi kran kebocoran 2 dibuka 90° , kondisi kran kebocoran 1 dan kran kebocoran 2 dibuka 90° .
5. Mencatat nilai nilai arus, tekanan, dan status kebocoran.

Hasil pengambilan data adalah sebagai berikut

No.	Sensor Arus 1 (l/min)	Sensor Arus 2 (l/min)	Sensor Tekanan (MPa)	Leak 1 (Yes/No)	Leak 2 (Yes/No)
Percobaan 1	6.00	0.53	0.03	Yes	No
Percobaan 2	5.87	0.40	0.02	Yes	No
Percobaan 3	4.53	1.00	0.03	Yes	No

Tabel 5.3 Arus, Tekanan, dan Output ML Ketika Kran Kebocoran 1 dibuka 90°

Dari Tabel 5.3 dapat dilihat bahwa pada saat kran kebocoran 1 dibuka 90° maka sensor arus 1 naik ± 5 l/min, sensor arus 2 naik ± 0.05 l/min dan sensor tekanan menurun ± 0.05 MPa. Dari tabel juga dapat disimpulkan bahwa model *machine learning* berhasil mendeteksi kebocoran pada letak yang sesuai.

No.	Sensor Arus 1 (l/min)	Sensor Arus 2 (l/min)	Sensor Tekanan (MPa)	Leak 1 (Yes/No)	Leak 2 (Yes/No)
Percobaan 1	4.21	5.01	0.02	No	Yes

Percobaan n 2	5.71	4.90	0.03	No	Yes
Percobaan n 3	4.82	5.25	0.03	No	Yes

Tabel 5.4 Arus, Tekanan, dan Output ML Ketika Kran Kebocoran 2 dibuka 90°

Dari Tabel 5.4 dapat dilihat bahwa pada saat kran kebocoran 1 dibuka 90° maka sensor arus 1 naik ±4.5 l/min, sensor arus 2 naik ±5 l/min dan sensor tekanan menurun ±0.05 MPa. Dari tabel juga dapat disimpulkan bahwa model *machine learning* berhasil mendeteksi kebocoran pada letak yang sesuai.

No.	Sensor Arus 1 (l/min)	Sensor Arus 2 (l/min)	Sensor Tekanan (MPa)	Leak 1 (Yes/No)	Leak 2 (Yes/No)
Percobaan n 1	4.30	1.07	0.03	Yes	Yes
Percobaan n 2	5.15	1.47	0.03	Yes	Yes
Percobaan n 3	6.00	1.20	0.02	Yes	Yes

Tabel 5.5 Arus, Tekanan, dan Output ML Ketika Kran Kebocoran 1 dan 2 dibuka 90°

Dari Tabel 5.5 dapat dilihat bahwa pada saat kran kebocoran 1 dan 2 dibuka 90° maka sensor arus 1 naik ±5 l/min, sensor arus 2 naik ±1 l/min dan sensor tekanan menurun ±0.05 MPa. Dari tabel juga dapat disimpulkan bahwa model *machine learning* berhasil mendeteksi kebocoran pada letak yang sesuai.

5.3. Analisis Dan Pembahasan

Berdasarkan serangkaian pengujian, sistem mampu membedakan kondisi pipa dengan tingkat akurasi tinggi. Pada pengujian tanpa kebocoran, model menandai “tidak bocor” dengan konsisten. Saat aliran normal, sensor tekanan meningkat sesuai ekspektasi dan model tetap mengkonfirmasi “tidak bocor”.

Ketika salah satu kran simulasi dibuka, kombinasi turunnya tekanan dan perbedaan aliran pada dua titik memicu model untuk mendeteksi kebocoran pada lokasi yang tepat. Pada skenario kedua dan ketiga, pola perubahan sensor juga diinterpretasi dengan benar oleh Feedforward Neural Network. Hasil ini menunjukkan bahwa pendekatan pemrosesan di edge device dan model yang di-prune memadai untuk deteksi cepat dan akurat.

5.3.1. Kelebihan Sistem

Kelebihan sistem adalah sebagai berikut:

1. Menggunakan dua sensor arus dan satu sensor tekanan memungkinkan sistem membedakan antara kebocoran di titik pertama, kedua, atau keduanya.
2. Model Feedforward Neural Network (FNN) berjalan langsung di ESP32 sehingga tidak bergantung pada latensi jaringan untuk inferensi awal.
3. Pemanfaatan AWS IoT Core dengan sertifikat TLS menjamin keamanan data, sekaligus mudah diintegrasikan ke skala lebih besar.
4. Dashboard Blynk menyediakan tampilan interaktif dan notifikasi segera, memudahkan teknisi lapangan mengambil tindakan.

5.3.2. Kekurangan Sistem

Kekurangan sistem adalah sebagai berikut:

1. Meski inferensi lokal berjalan offline, pengiriman dan penyimpanan data real-time tetap memerlukan koneksi internet stabil.
2. Feedforward Neural Network (FNN) sederhana efektif untuk simulasi, namun mungkin kurang sensitif pada variasi kebocoran kecil atau kondisi aliran sangat dinamis.
3. Prototipe belum dilengkapi casing tahan air atau debu (IP-rated), sehingga komponen elektronik rentan di lingkungan lapangan.

BAB VI KESIMPULAN DAN SARAN

6.1. KESIMPULAN

Pada rumusan masalah dijabarkan dua hal utama yaitu efektivitas deteksi kebocoran pipa dan kemampuan menentukan lokasi kebocoran. Berdasarkan hasil pengujian, sistem prototype berhasil memenuhi kedua tujuan tersebut. Sistem mampu mendeteksi keberadaan kebocoran pada skenario titik pertama, titik kedua, maupun keduanya sekaligus.

Produk penelitian ini terdiri dari perangkat keras prototipe pipa dengan dua sensor arus dan satu sensor tekanan, perangkat lunak pada ESP32 yang telah ditanami model pohon keputusan, serta infrastruktur AWS IoT Core dan dashboard Blynk untuk visualisasi real time.

Kelebihan sistem terletak pada kemampuannya melakukan inferensi secara langsung di ESP32 tanpa ketergantungan penuh pada jaringan, serta menyajikan notifikasi dan grafik deteksi kebocoran secara instan untuk memudahkan teknisi lapangan.

6.2. SARAN

Sistem saat ini masih memiliki beberapa keterbatasan yang perlu diperbaiki pada pengembangan berikutnya. Pertama, model pohon keputusan sederhana kurang sensitif terhadap variasi kebocoran kecil atau perubahan aliran yang dinamis sehingga disarankan menguji algoritma ensemble atau metode deteksi anomali untuk meningkatkan akurasi.

Kedua, kestabilan koneksi internet mempengaruhi pengiriman data sehingga perlu ditambahkan mekanisme penyimpanan sementara di ESP32 agar data tidak hilang saat jaringan terganggu.

Terakhir, karena prototipe belum dilengkapi pelindung, disarankan merancang enclosure dengan sertifikasi IP untuk melindungi komponen elektronik dari debu, cipratan air, dan kondisi lingkungan lapangan. Dengan langkah-langkah tersebut, sistem akan menjadi lebih andal dan siap diterapkan pada jaringan distribusi pipa sesungguhnya.

DAFTAR PUSTAKA

- Kumar, P., Gupta, G. and Buyya, R., 2021. Reliable and Fault-Tolerant IoT-Edge Architecture. *IEEE Internet of Things Journal*, 8(10), pp. 8057-8071.
- Kusuma, H. et al. (2021) 'Judul prototype Pendekripsi Kebocoran Pipa Berbasis IOT menggunakan nodemcu esp8266 Melalui dashboard Adafruit.io', Jurnal Teknologi Dan Sistem Informasi Bisnis, 3(2), pp. 327–333. doi:10.47233/jteksis.v3i2.253.
- Rodríguez, J., Pérez, D. and García, M., 2022. *CEFIoT: A Fault-Tolerant IoT Architecture for Edge and Cloud. Future Generation Computer Systems*, 127, pp. 112-126.
- Zhao, L., Chen, X. and Wang, Y., 2020. *CoMP: Efficient Monitoring for Constrained Embedded Devices. ACM Transactions on Embedded Computing Systems*, 19(5), pp. 1-21.
- Atanane, K., Garcia, F., Hounkonnou, N.M., Santi, E. and Hounkpevi, H.A. (2023) 'Smart Buildings: Water Leakage Detection Using TinyML', Sensors, 23(9), p. 9210.
- Borra, P., Mullapudi, M., Nerella, H. and Prakashchand, L.K. (2024). Analyzing AWS Edge Computing Solutions to Enhance IoT Deployments. International Journal of Engineering and Advanced Technology, 13(6), pp.8–12. doi:<https://doi.org/10.35940/ijeat.f4519.13060824>.
- Ismail Essamlali, Hasna Nhaila and Mohamed El Khaili (2024). Advances in machine learning and IoT for water quality monitoring: A comprehensive review. *Heliyon*, 10(6), pp.e27920–e27920. doi:<https://doi.org/10.1016/j.heliyon.2024.e27920>.
- Jhaveri, R.H., Chi, H.R. and Wu, H. (2024). TinyML for Empowering Low-Power IoT Edge Consumer Devices. *IEEE Transactions on Consumer Electronics*, 70(4), pp.7318–7321. doi:<https://doi.org/10.1109/tce.2024.3482353>.
- M.N. Kanyama, F. Bhunu Shava, A.M. Gamundani and Hartmann, A. (2024). Machine learning applications for anomaly detection in Smart Water Metering Networks: A systematic review. *Physics and Chemistry of the Earth, Parts A/B/C*, pp.103558–103558. doi:<https://doi.org/10.1016/j.pce.2024.103558>.
- Mashhadi, N., Shahrour, I., Attoue, N., El Khattabi, J. and Aljer, A. (2021). Use of Machine Learning for Leak Detection and Localization in Water

- Distribution Systems. Smart Cities, 4(4), pp.1293–1315.
doi:<https://doi.org/10.3390/smartcities4040069>.
- Singh, M. and Ahmed, S. (2020). IoT based smart water management systems: A systematic review. Materials Today: Proceedings, 46.
doi:<https://doi.org/10.1016/j.matpr.2020.08.588>.
- Honeywell. (2021). *Board Mount Pressure Sensors: Basic & Advanced Performance*. Datasheet.
- Datasheet YF-S201 – Flow Meter Sensor. [SeTech & OEM Modules]

LAMPIRAN

6.2.1. Dokumentasi (Foto-Foto selama kegiatan)



Dokumentasi 1. Merakit Koneksi Sensor dengan ESP 32



Dokumentasi 2. Pembuatan *Prototype* Simulasi Kebocoran Pipa



Dokumentasi 3. Percobaan Alat pada *Prototype* Simulasi Kebocoran Pipa



Dokumentasi 4. Optimasi Prototype Simulasi Kebocoran Pipa



Dokumentasi 5. Percobaan Kedua *Prototype Simulasi Kebocoran Pipa*



Dokumentasi 6. Optimasi *Prototype Simulasi Kebocoran Pipa*



Dokumentasi 7. Pembuatan Laporan dan Program

6.2.2. Screen Shoot Upload Video Presentasi Beserta Link

Link Video :

<https://drive.google.com/drive/folders/16KtZJi1k1vpOeyyocWCLOcva98m1H49q?usp=sharing>

Screen Shoot :



Dokumentasi 8. Screenshot Upload Video Presentasi

6.2.3. Barcode/link Program/sistem

Link Program Sistem:

[Setup Program Smart Water Leak Detection](#)