

**IMPLEMENTASI WSN PADA DETEKSI KEBOCORAN PIPA AIR
RUMAH MENGGUNAKAN KOMUNIKASI DATA ESP-NOW
BERBASIS IOT (*Internet of Things*)**

SKRIPSI

Untuk memenuhi sebagian persyaratan
Memperoleh gelar Sarjana Teknik

Disusun oleh:
Aisha Putri Kinanti
NIM: 205150301111040



PROGRAM STUDI TEKNIK KOMPUTER
DEPARTEMEN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2024

PENGESAHAN

IMPLEMENTASI WSN PADA DETEKSI KEBOCORAN PIPA AIR RUMAH
MENGUNAKAN KOMUNIKASI DATA ESP-NOW BERBASIS
IOT (*Internet of Things*)

SKRIPSI

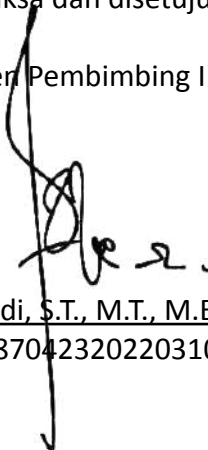
Diajukan untuk memenuhi sebagian persyaratan
Memperoleh gelar Sarjana Teknik

Disusun oleh:
Aisha Putri Kinanti
NIM: 205150301111040

Skripsi ini telah diuji dan dinyatakan lulus pada
12 Januari 2024

Telah diperiksa dan disetujui oleh:

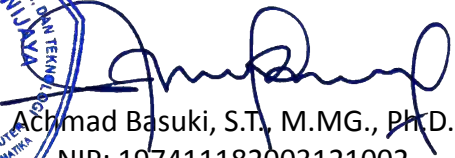
Dosen Pembimbing I



Agung Setia Budi, S.T., M.T., M.Eng., Ph.D.
NIP: 198704232022031003

Mengetahui
Ketua Departemen Teknik Informatika,




Achmad Basuki, S.T., M.MG., Ph.D.
NIP: 197411182003121002

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata di dalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).



Malang, 12 Januari 2024

A handwritten signature in black ink, appearing to read 'Aisha'.

Aisha Putri Kinanti

NIM: 205150301111040

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa sehingga peneliti dapat menyusun skripsi berjudul “Implementasi WSN pada Deteksi Kebocoran Pipa Air Rumah Menggunakan Komunikasi Data ESP-Now Berbasis IoT (*Internet of Things*)” hingga selesai. Selama peneliti melakukan penelitian dan menyusun skripsi ini, peneliti mendapatkan bantuan dan dukungan dari beberapa pihak. Maka dari itu, dalam kesempatan kali ini peneliti ingin mengucapkan terima kasih kepada pihak-pihak tersebut, diantaranya

1. Tuhan Yang Maha Esa.
2. Ibu, Bapak, Adik, Sepupu, dan segenap keluarga peneliti yang memberikan dukungan dan kasih sayang kepada peneliti, sehingga peneliti senantiasa termotivasi untuk menyelesaikan skripsi ini.
3. Bapak Agung Setia Budi, S.T., M.T., M.Eng., Ph.D, selaku dosen pembimbing yang selalu meluangkan waktu dan memberikan arahan kepada peneliti selama mengerjakan skripsi.
4. Bapak Achmad Basuki, S.T., M.MG., Ph.D. selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
5. Bapak Barlian Henryranu Prasetyo, S.T., M.T., Ph.D selaku Ketua Program Studi Sarjana Teknik Komputer Fakultas Ilmu Komputer Universitas Brawijaya.
6. Teman-teman yang senantiasa memberi dukungan dan menemani peneliti ketika mengerjakan skripsi, seperti Mas Bayu, Mbak Rina, Sasa, Fikriya dan beberapa teman lainnya yang tidak bisa disebutkan.
7. Segenap musisi seperti Laufey, Raisa, Stevie Wonder, GFRIEND, dan lainnya, yang karyanya selalu menjadi penyemangat peneliti saat mengerjakan skripsi ini.

Peneliti menyadari bahwa dalam penyusunan skripsi ini masih memiliki banyak kekurangan. Untuk itu, peneliti dengan lapang dada sangat mengharapkan dan menerima saran serta kritik yang membangun. Akhir kata, penulis berharap agar kedepannya skripsi ini dapat memberikan manfaat bagi banyak orang.

Malang, 22 Desember 2023

Penulis,
ochakinan3103@student.ub.ac.id

ABSTRAK

Aisha Putri Kinanti, Implementasi WSN pada Deteksi Kebocoran Pipa Air Rumah Menggunakan Komunikasi Data ESP-Now Berbasis IoT (*Internet of Things*)

Pembimbing: Agung Setia Budi, S.T., M.T., M.Eng., Ph.D

Air merupakan kebutuhan dasar manusia. Setiap rumah memiliki saluran pipa air untuk penggunaan air sehari-hari. Namun, kebocoran pipa air seringkali menjadi permasalahan karena merugikan secara ekonomi dan lingkungan. Sayangnya hal ini sulit untuk dideteksi oleh manusia. Maka dari itu, perlu adanya sebuah sistem yang dapat mendeteksi adanya kebocoran di saluran pipa air. Pada penelitian ini, dibuat sebuah sistem untuk pemantauan kondisi air secara real-time menggunakan sensor YF-S201. Dalam konteks ini, Wireless Sensor Network (WSN) dapat menjadi solusi karena WSN memungkinkan penggunaan beberapa sensor dan hemat energi yang dapat ditempatkan pada sistem perairan rumah. Namun, untuk menggunakan WSN perlu adanya sebuah protokol komunikasi data untuk mengatur pertukaran data yang didapatkan dari sensor. ESP-Now dapat digunakan sebagai protokol komunikasi data karena pengimplementasiannya yang mudah. Arsitektur WSN digunakan sebagai pemetaan sistem terhadap sensor dan mikrokontroler (ESP8266) untuk mendapatkan data laju alir air dalam saluran air rumah. Node sensor yang terdiri atas sensor YF-S201 dan ESP8266, akan dipasang pada jalur pipa air. Kemudian, node tersebut akan mengirimkan hasil baca sensor ke sebuah node master untuk dideteksi kebocorannya. Sistem yang dibuat berhasil menerapkan pengimplementasian WSN, di mana keandalan pengiriman oleh sistem sebesar 100% untuk mengirimkan data antar node. Lalu, otomatisasi sistem yang dibuat untuk dapat dilakukan pemantauan menggunakan telepon genggam, menjadikan sistem ini dapat dikategorikan sebagai produk Internet of Things (IoT).

Kata kunci: deteksi kebocoran air, ESP-Now, Internet of Things, Wireless Sensor Network, YF-S201

ABSTRACT

Aisha Putri Kinanti, *Implementation of WSN in House Water Pipe for Water Leak Detection Using ESP-Now Data Communication with IoT (Internet of Things) Based*

Supervisor: Agung Setia Budi, S.T., M.T., M.Eng., Ph.D

Water is a basic human need. Every house has a water pipe for daily needs. However, water pipe leaks often become a problem because they are economically and environmentally detrimental. Unfortunately, this is difficult for humans to detect. Therefore, it is necessary to have a system that can detect leaks in water pipes. In this research, a system was created to do real-time monitoring water conditions using the YF-S201 sensor. In this context, Wireless Sensor Network can be a solution because it allows several sensors and energy efficiency that can be placed in the house. Data communication protocol is needed to regulate the transfer of data from sensors. ESP-Now can be used as a protocol because of its easy implementation. The WSN architecture is used as a mapping system for sensors and microcontrollers (ESP8266) to get water flow rate data. The sensor node, which consists of YF-S201 and ESP8266 sensors, will be installed on the water pipe line. Then, the node will send the sensor reading results to a master node for leak detection. The system has successfully implemented WSN, where the reliability of delivery by the system is 100% for sending data between nodes. Then, the automation of the system was created so that monitoring can run in smartphones, making this system can be categorized as an Internet of Things (IoT).

Keywords: *ESP-Now, Internet of Things, water leak detection, Wireless Sensor Network, YF-S20*

DAFTAR ISI

PENGESAHAN.....	i
PERNYATAAN ORISINALITAS.....	ii
KATA PENGANTAR.....	iii
ABSTRAK.....	iv
ABSTRACT.....	v
DAFTAR ISI.....	vi
DAFTAR TABEL.....	ix
DAFTAR GAMBAR.....	xi
DAFTAR LAMPIRAN.....	xii
BAB 1 PENDAHULUAN.....	13
1.1 Latar Belakang.....	13
1.2 Rumusan Masalah.....	14
1.3 Tujuan.....	14
1.4 Manfaat.....	14
1.5 Batasan Masalah.....	15
1.6 Sistematika Pembahasan.....	15
BAB 2 LANDASAN KEPUSTAKAAN.....	16
2.1 Tinjauan Pustaka.....	16
2.2 Dasar Teori.....	18
2.2.1 Saluran Air Bersih di Rumah.....	18
2.2.2 Internet of Things (IOT).....	18
2.2.3 Wireless Sensor Network (WSN).....	19
2.2.4 Protokol ESP-Now.....	19
2.2.5 ESP8266.....	20
2.2.6 Sensor Flow Meter YF-S201.....	20
2.2.7 Blynk.....	20
BAB 3 METODOLOGI.....	22
3.1 Tipe Penelitian.....	22
3.2 Strategi dan Rancangan Penelitian.....	22
3.2.1 Studi Literatur.....	22
3.2.2 Rekayasa Kebutuhan Sistem.....	22
3.2.3 Perancangan Sistem.....	23
3.2.4 Implementasi Sistem.....	23
3.2.5 Pengujian Sistem.....	23
3.2.6 Analisis Sistem.....	23
3.2.7 Kesimpulan dan Saran.....	23
3.3 Subjek Penelitian.....	24

3.4 Objek Penelitian.....	24
3.5 Lokasi Penelitian.....	24
3.6 Metode Pengambilan Data.....	24
BAB 4 REKAYASA KEBUTUHAN.....	25
4.1 Kajian Masalah.....	25
4.2 Identifikasi Stakeholder.....	25
4.3 Kebutuhan Fungsional.....	25
4.4 Spesifikasi Sistem.....	26
4.5 Analisis Kebutuhan Perangkat.....	26
4.5.1 Analisis Kebutuhan Perangkat Keras.....	26
4.5.2 Analisis Kebutuhan Perangkat Lunak.....	28
BAB 5 PERANCANGAN DAN IMPLEMENTASI.....	30
5.1 Perancangan Sistem.....	30
5.1.1 Perancangan Prototipe.....	31
5.1.2 Perancangan Diagram Skematik Keseluruhan.....	31
5.1.3 Perancangan Perangkat Keras.....	32
5.1.4 Perancangan Perangkat Lunak.....	32
5.2 Implementasi Sistem.....	33
5.2.1 Implementasi Prototipe.....	33
5.2.2 Implementasi Perangkat Keras.....	34
5.2.2.1 Implementasi Perangkat Keras Keseluruhan.....	35
5.2.2.2 Implementasi Perangkat Keras Tiap Node.....	35
5.2.3 Implementasi Perangkat Lunak.....	35
5.2.3.1 Implementasi Perangkat Lunak Node Sensor.....	35
5.2.3.2 Implementasi Perangkat Lunak Node Master.....	38
5.3 Pengujian Awal.....	41
5.3.1 Prosedur Pengambilan Data.....	42
5.3.2 Perhitungan Persamaan Kalibrasi.....	42
BAB 6 PENGUJIAN DAN ANALISIS.....	43
6.1 Pengujian Sensor Water Flow YF-S201.....	43
6.1.1 Prosedur Pengujian.....	43
6.1.2 Hasil dan Analisis Pengujian.....	43
6.2 Pengujian ESP-NOW untuk Pengiriman Data dari Node Sensor ke Node Master.....	44
6.2.1 Prosedur Pengujian.....	44
6.2.2 Hasil dan Analisis Pengujian.....	44
6.3 Pengujian Pengiriman Data Node Master ke Blynk.....	44
6.3.1 Prosedur Pengujian.....	45
6.3.2 Hasil dan Analisis Pengujian.....	45
6.4 Pengujian Keseluruhan Sistem.....	45

6.4.1 Prosedur Pengujian.....	45
6.4.2 Hasil dan Analisis Pengujian.....	46
6.4.2.1 Pengujian Penerimaan Data Node Master.....	46
6.4.2.2 Pengujian Penerimaan Data pada Blynk.....	46
6.4.2.3 Pengujian Deteksi Kebocoran Air.....	47
BAB 7 PENUTUP.....	50
7.1 Kesimpulan.....	50
7.2 Saran.....	50
DAFTAR REFERENSI.....	51

DAFTAR TABEL

Tabel 2.1 Perbandingan Jurnal oleh Penelitian Terdahulu.....	18
Tabel 4.1 Spesifikasi ESP8266 yang digunakan.....	29
Tabel 4.2 Spesifikasi Sensor Water Flow YF-S201 yang digunakan.....	29
Tabel 4.2 Spesifikasi Power Adaptor yang digunakan.....	30
Tabel 4.2 Spesifikasi Kabel Jumper yang digunakan.....	30
Tabel 5.1 Koneksi Sensor Water Flow YF-S201 dengan ESP8266.....	34
Tabel 5.1 Kode Program Library Sistem.....	37
Tabel 5.2 Penjelasan Kode Program Library Sistem.....	38
Tabel 5.3 Kode Program Membaca Nilai Sensor.....	38
Tabel 5.4 Penjelasan Kode Program Membaca Nilai Sensor.....	39
Tabel 5.5 Kode Program ESP-Now.....	39
Tabel 5.6 Penjelasan Kode Program ESP-Now.....	40
Tabel 5.7 Kode Program Library Sistem.....	40
Tabel 5.8 Penjelasan Kode Program Library Sistem.....	41
Tabel 5.9 Kode Program Membaca Nilai Sensor.....	41
Tabel 5.10 Penjelasan Kode Program Membaca Nilai Sensor.....	42
Tabel 5.11 Kode Program ESP-Now.....	42
Tabel 5.12 Penjelasan Kode Program ESP-Now.....	42
Tabel 5.11 Kode Program Blynk dan Klasifikasi Kebocoran.....	43
Tabel 5.12 Penjelasan Kode Program Blynk dan Klasifikasi Kebocoran.....	43
Tabel 5.13 Tabel Pengujian Awal.....	44
Tabel 5.14 Tabel Menghitung Nilai Kalibrasi Sensor.....	44
Tabel 6.1 Pengujian Sensor Water Flow YF-S201.....	46
Tabel 6.2 Pengujian ESP-NOW Dari Node Sensor ke Node Master.....	47
Tabel 6.3 Pengujian Penerimaan Data Node Master.....	46
Tabel 6.4 Pengujian Deteksi Kebocoran Air.....	47

DAFTAR GAMBAR

Gambar 2.1 Instalasi Saluran Air di Rumah.....	18
Gambar 2.2 Model Autentifikasi pada WSN.....	19
Gambar 2.3 Gambar Ilustrasi Arsitektur ESP-Now.....	19
Gambar 2.4 ESP8266.....	20
Gambar 2.5 Sensor Flow Meter.....	20
Gambar 2.6 Platform Blynk.....	21
Gambar 5.1 Perancangan Sistem WSN.....	30
Gambar 5.2 Perancangan Prototipe Sistem.....	31
Gambar 5.3 Diagram Skematik Keseluruhan.....	31
Gambar 5.4 Perancangan Perangkat Keras.....	32
Gambar 5.5 Perancangan Perangkat Lunak.....	33
Gambar 5.6 Implementasi Prototype Keseluruhan Sistem.....	34
Gambar 5.7 Implementasi Prototype Tiap Node.....	34
Gambar 5.8 Implementasi Perangkat Keras Keseluruhan.....	35
Gambar 5.9 Implementasi Perangkat Keras Pada Node.....	35
Gambar 6.1 Pengujian Pengiriman Data ke Blynk.....	47
Gambar 6.2 Penerimaan Data di Blynk.....	49
Gambar 6.3 Notifikasi Adanya Kebocoran Air di Smartphone.....	51

DAFTAR LAMPIRAN

LAMPIRAN A KODE SUMBER PROGRAM.....	55
A.1 Kode Program Node Sensor.....	55
A.2 Kode Program Node Master.....	57
LAMPIRAN B.....	58
B.1 Pengujian Sensor YF-S201.....	58
B.2 Pengujian Komunikasi Data ESP-NOW.....	58
B.3 Konfigurasi Blynk.....	59
B.4 Dokumentasi Pengujian.....	61

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Air merupakan sebuah kebutuhan yang mendasar bagi seluruh manusia. Manusia menggunakan air untuk macam-macam keperluan, seperti minum, mandi, mencuci pakaian, dan sebagainya (Mansur, 2022). Dilansir dari survei yang dilakukan Direktorat Pengembangan Air Minum, Ditjen Cipta karya, Departemen PU tahun 2006, didapatkan data bahwa setiap harinya seseorang membutuhkan setidaknya 70 liter air untuk memenuhi kebutuhan hidupnya. Maka dari itu, demi terpenuhinya kebutuhan air bersih, masyarakat di zaman sekarang menggunakan berbagai cara untuk mendapatkan air bersih.

Terdapat beberapa sumber untuk mendapatkan air bersih, diantaranya sumur, mata air, air hujan, dan air PDAM (Perusahaan Daerah Air Minum). Masing-masing dari sumber air tersebut memiliki kelebihan serta kekurangan. Seperti contoh air PDAM, air dari sumber ini mudah didapatkan karena air PDAM merupakan sebuah program dari pemerintah untuk memberikan hak kepada masyarakat agar bisa mendapatkan air bersih. Meskipun demikian, air yang disediakan belum tentu bersih dan masyarakat harus membayar sesuai dengan banyak air yang digunakan.

Banyak masyarakat yang mengeluhkan melonjaknya tagihan air meskipun penggunaannya masih dalam batas normal. Penyebab membengkaknya tagihan air secara tidak wajar dapat diakibatkan oleh menunggaknya pembayaran air bulan sebelumnya, rusaknya meter air, hingga adanya kebocoran yang tidak diketahui pada saluran air yang digunakan.

Dari informasi yang diunggah oleh Badan Pusat Statistik, didapatkan data mengenai kebocoran saluran air di beberapa daerah di Indonesia. Pada tahun 2018 di Provinsi Nusa Tenggara Timur, dari 24.696 ribu m³ air bersih yang dialirkan terdeteksi 1.547.000 m³ air bocor. Sedangkan di tahun 2017, Provinsi Jawa Tengah mendapati 17,01 % kebocoran air dari jumlah total 316.345.655 m³ air bersih yang diproduksi. Selain kerugian biaya yang dialami oleh masyarakat akibat bocornya air bersih, hal ini juga menyebabkan terbuangnya air bersih secara sia-sia. Padahal jumlah air bersih yang bocor dapat digunakan untuk hal lain yang dapat memberikan manfaat bagi manusia dan alam.

Seiring berkembangnya teknologi, manusia menciptakan banyak macam cara untuk hidup dengan berbagai kemudahan. Kemudahan ini memberikan dampak positif bagi manusia dari berbagai aspek kehidupan, termasuk untuk memaksimalkan penggunaan air bersih. Teknologi saat ini memungkinkan sebuah alat dapat bekerja secara otomatis tanpa manusia. Teknologi tersebut merupakan *Internet of Things* (IoT). IoT merupakan teknologi canggih yang merujuk pada banyaknya perangkat dan sistem yang saling terhubung satu sama lain dengan menggunakan internet di seluruh dunia untuk dapat saling berbagi data (Selay, A.

dkk, 2022). Dengan adanya perpaduan antara teknologi IoT dan sistem saluran air ini dapat memaksimalkan penggunaan air bersih, sehingga meminimalisir adanya kebocoran pada saluran air di dalam rumah.

Dalam perkembangannya, IoT memiliki berbagai fungsi sesuai dengan kegunaannya. Pada penggunaan air PDAM misalnya, baru-baru ini dapat ditemukan beberapa teknologi untuk melakukan *monitoring* penggunaan air secara online. Sehingga memungkinkan kedepannya untuk dapat mengetahui banyak penggunaan air serta deteksi kebocoran air menggunakan teknologi IoT.

Dari uraian sebelumnya, dapat dibuat sebuah sistem yang berguna untuk mendeteksi kebocoran air. Untuk menunjang sistem tersebut, dibutuhkan teknologi seperti ESP8266 dan sensor *water flow*. Sensor dan ESP8266 (atau yang kemudian disebut node) dipasang pada keran-keran yang terpasang dalam instalasi pipa air di rumah. Data dari tiap node kemudian dikirim ke sebuah node master yang kemudian dapat diolah untuk diklasifikasi adanya kebocoran atau tidak. Dalam proses pengiriman data tersebut dibutuhkan protokol komunikasi data untuk masing-masing node-nya. Pada perangkat ESP terdapat protokol ESP-Now yang dapat mengirimkan data secara efektif hingga 50-100 meter. Hasil dari klasifikasi tersebut dapat dilihat oleh pengguna pada *gadget* pengguna.

1.2 Rumusan Masalah

Melalui latar belakang yang sudah terpaparkan sebelumnya, rumusan masalah dapat disusun seperti berikut

1. Bagaimana keberhasilan pembacaan sensor YF-S201?
2. Bagaimana proses pengiriman data menggunakan protokol ESP-Now?
3. Bagaimana hasil performa secara keseluruhan dari sistem yang dibuat untuk mendeteksi kebocoran?

1.3 Tujuan

Dari rumusan masalah yang telah didapatkan, diperoleh beberapa tujuan, sebagai berikut

1. Mengetahui hasil pembacaan sensor YF-S201.
2. Mendapatkan hasil pengiriman data yang baik untuk ditampilkan ke pengguna.
3. Mendapatkan performa sistem yang baik secara keseluruhan untuk mendeteksi kebocoran.

1.4 Manfaat

Penelitian ini diharapkan mampu berguna kepada masyarakat, khususnya yang menggunakan layanan air bersih di rumahnya. Dari penelitian yang dilakukan, diharapkan dapat membantu untuk mendeteksi adanya kebocoran air dalam saluran air di rumah. Sehingga meminimalisir adanya lonjakan pada

pembayaran air dan mengurangi terbuangnya air bersih karena kebocoran tersebut.

1.5 Batasan Masalah

Agar penelitian ini berfokus terhadap permasalahan yang ada, perlu adanya beberapa batasan masalah, yaitu

1. Penelitian merupakan prototipe menggunakan tiga node sensor dengan satu node master.
2. Penelitian yang dilakukan hanya untuk mendeteksi adanya kebocoran, belum hingga menentukan lokasi kebocorannya.
3. Penelitian menggunakan sensor YF-S201 yang memiliki akurasi pembacaan yang kurang baik.

1.6 Sistematika Pembahasan

BAB 1 Pendahuluan

Bab ini merupakan bagian pertama kajian yang memuat beberapa unsur. Unsur tersebut adalah latar belakang, rumusan masalah, tujuan dari penelitian, kepentingan penelitian, batasan masalah, dan sistematika pembahasan.

BAB 2 Landasan Pustaka

Dalam bab ini merupakan bagian di mana terdapat kajian atau tinjauan literatur terhadap penelitian yang sebelumnya pernah dilakukan. Tinjauan ini dilakukan sebagai sumber untuk pengembangan penelitian.

BAB 3 Metodologi Penelitian

Bab ketiga, berisi penjelasan atas bagaimana penelitian dilakukan. Di dalamnya terdapat studi literatur, analisis kebutuhan, perancangan sistem, implementasi sistem, pengujian, analisis, kesimpulan, dan metode pengumpulan data.

BAB 4 Rekayasa Kebutuhan

Pada bab ini akan dijelaskan mengenai kebutuhan-kebutuhan oleh sistem yang dibuat, baik dari segi perangkat keras dan perangkat lunak.

BAB 5 Perancangan dan Implementasi

Dalam bab ini terdapat penjelasan mengenai bagaimana perancangan dan implementasi akan sistem dari penelitian yang dilakukan.

BAB 6 Pengujian dan Analisis

Dalam bab keenam, akan membahas mengenai hasil dari pengujian terhadap sistem. Selain itu juga terdapat analisis mengenai hasil yang didapatkan.

BAB 7 Penutup

Bab ini berisikan kesimpulan dari penelitian yang dilakukan dan kemudian saran mengenai apabila ada pengembangan untuk penelitian selanjutnya.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Tinjauan Pustaka

Tinjauan pustaka diperlukan sebagai penunjang berjalannya penelitian yang dilakukan. Proses ini dilakukan guna mendapatkan informasi dan pengetahuan yang dapat digunakan untuk menyusun penelitian ini. Berikut merupakan landasan kepastakaan dari penelitian berjudul “Implementasi WSN pada Deteksi Kebocoran Pipa Air Rumah Menggunakan Komunikasi Data ESP-Now Berbasis Iot (*Internet Of Things*)”.

Tabel 2.1 Perbandingan Jurnal oleh Penelitian Terdahulu

No.	Judul Penelitian	Kesamaan	Perbedaan
1	Sistem Monitoring Kondisi Aliran Pada Gorong-gorong Saluran Air Menggunakan Metode Fuzzy (Moch. Alfian Zainullah, dkk, 2018)	Mengukur kecepatan debit air pada saluran air dengan <i>flow meter</i> .	Menggunakan NRF24L01 dan metode fuzzy.
2	Implementasi Sistem Monitoring Sungai berbasis LoRa-MQTT Gateway (Bimo Wismoyo Aji, dkk, 2023)	Menggunakan ESP8266 sebagai mikrokontroler dan terdapat komunikasi data antar node.	Sistem digunakan untuk mendeteksi ketinggian air sungai. Sistem ini juga menggunakan sensor yang berbeda, yaitu HC-SR04.
3	Implementasi WSN Pada Kandang Ayam Menggunakan Topologi Tree dan Protokol Komunikasi Data ESP-Now (Yusuf Hamdani, dkk, 2023)	Implementasi WSN dengan ESP-Now. Menggunakan beberapa node sensor yang terhubung ke sebuah node master.	Sistem digunakan untuk monitoring suhu dan kelembaban kandang ayam.
4	Sistem Monitoring Kadar Gas Berbahaya Pada Lokasi Parkiran Bawah Tanah Menggunakan Protokol MQTT (Loki Sudiarta Mongin, dkk, 2019)	Menggunakan protokol komunikasi antar node dan ESP8266.	Sistem digunakan untuk mendeteksi gas berbahaya.

Penelitian yang dilakukan oleh (Moch. Alfian Zainullah, dkk, 2018) membahas mengenai monitoring kondisi aliran air pada gorong-gorong menggunakan metode fuzzy. Fokus penelitian ada pada pengecekan lancar atau tidaknya saluran air dalam gorong-gorong tersebut. Faktor kelancaran debit air dipengaruhi oleh tersumbatnya aliran air yang dapat menyebabkan banjir. Peneliti menggunakan mikrokontroler berjenis arduino nano yang kemudian terkoneksi dengan beberapa sensor kecepatan air G1/2. Komunikasi jarak jauh didukung oleh adanya *wireless nrf24l01*. Kemudian data yang didapat diolah menggunakan metode fuzzy. Keluaran dari sistem ini menggunakan LED RGB (*Light Emitting Diode Red, Green, Blue*).

Selanjutnya, pada penelitian yang dilakukan oleh (Bimo Wismoyo Aji, dkk, 2023), merupakan penelitian untuk mendeteksi tinggi air sungai. Hal ini dilakukan untuk deteksi dini banjir yang diakibatkan oleh meningginya air sungai. Peneliti menggunakan perangkat keras ESP8266 untuk mengirimkan data yang diperoleh dari sensor HC-SR04. Kemudian data yang telah diambil akan masuk ke sebuah server Raspberry pi di mana kemudian data tersebut diolah untuk menghasilkan data berupa angka ketinggian air (dalam cm) pada sungai dan klasifikasi banjir atau tidak. Data yang sudah diolah tersebut kemudian akan dikirimkan oleh Raspberry pi ke *smartphone* yang terkoneksi melalui aplikasi Discord.

Kemudian untuk penelitian yang telah dilakukan oleh (Yusuf Hamdani, dkk, 2023), merupakan penelitian yang digunakan untuk monitoring kandang ayam. Penelitian ini menggunakan mikrokontroler ESP8266 yang diletakkan pada setiap kandang bersamaan dengan sensor DHT11 untuk melakukan deteksi terhadap suhu serta kelembaban kandang ayam. Tiap-tiap node akan saling terkoneksi dengan sebuah node master melalui protokol komunikasi MQTT dan menggunakan topologi tree. Hasil akhir dari penelitian ini adalah menampilkan data pembacaan sensor DHT11 yang kemudian dikirimkan ke *smartphone* melalui aplikasi yang dibuat sendiri.

Pada tinjauan pustaka terakhir untuk penelitian yang dilakukan oleh (Loki Sudiarta Mongin, dkk, 2019), membahas mengenai deteksi ada atau tidaknya gas berbahaya di parkiran bawah tanah. Gas yang disebutkan, yaitu karbon monoksida dan nitrogen oksida. Penelitian menggunakan sensor MQ-7 dan MQ-135 untuk mendeteksi gas yang ditempatkan di parkiran bawah tanah. Kemudian data yang didapatkan oleh sensor akan dikirimkan oleh sebuah mikrokontroler arduino uno ke sebuah ESP8266 berformat json. Selanjutnya, data yang didapatkan oleh ESP8266 akan dikirimkan ke sebuah server thingsboard.io melalui protokol komunikasi MQTT. Hasil akhir dari penelitian ini berupa nilai besarnya gas berbahaya yang terdeteksi. Dari hasil tersebut juga didapatkan rata-rata waktu *delay* sebesar 631425,55 mikrosekon untuk *delay* eksekusi dan 1,92 detik untuk *delay* thingsboard.io.

2.2 Dasar Teori

Dalam mendukung penelitian ini, perlu adanya dasar teori untuk mendalami penelitian yang dilakukan. Bab ini akan menjelaskan teori-teori yang dibutuhkan agar menunjang penelitian yang dilakukan dapat berhasil.

2.2.1 Saluran Air Bersih di Rumah

Saluran air bersih berbayar merupakan sebuah sistem yang menggunakan pipa atau saluran yang digunakan untuk mendistribusikan air layak digunakan ke rumah masyarakat. Sumber air dapat diperoleh dari sumur, air tanah, maupun PAM. Masing-masing dari sumber air tersebut memiliki kelebihan dan kekurangannya. Untuk air sumur, masyarakat tidak perlu membayar untuk kebutuhan air bersih ke pihak pengelola air bersih, tetapi harus memberikan uang ekstra untuk membayar listrik karena mesin pompa air. Sedangkan air bersih yang bersumber dari PAM dinilai mudah didapatkan karena terdapat pihak pengelola yang bekerja untuk menyediakan air bersih, tetapi masyarakat harus membayar tagihan air sesuai dengan banyaknya penggunaan air perbulannya.



Gambar 2.1 Instalasi Saluran Air di Rumah

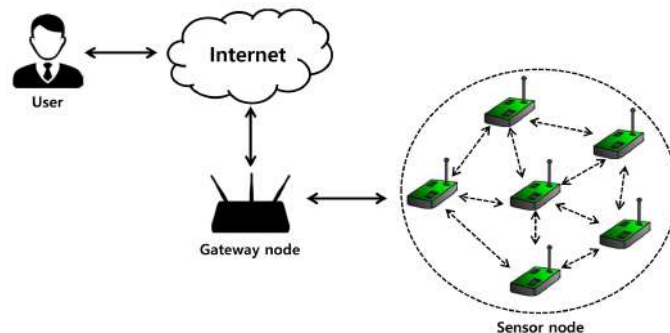
Sumber: jatim.tribunnews.com

2.2.2 *Internet of Things (IoT)*

Internet of Things adalah sebuah sistem di mana terdapat benda atau objek yang dihubungkan dengan teknologi seperti aplikasi atau sensor, yang kemudian dapat melakukan komunikasi pertukaran data, menghubungkan sesuatu, hingga mengendalikan sesuatu dengan koneksi internet. IoT dapat dijumpai dalam berbagai bidang, seperti pertanian, kesehatan, transportasi, hingga otomatisasi rumah.

2.2.3 Wireless Sensor Network (WSN)

Wireless Sensor Network adalah sebuah jaringan nirkabel yang tidak memerlukan infrastruktur dan terdiri dari beberapa sensor nirkabel yang terhubung secara ad-hoc untuk digunakan sebagai pemantau sistem, kondisi fisik, hingga lingkungan. WSN terdiri dari beberapa node: sensor node, *actuator node*, *router*, dan *gateway*. WSN biasanya digunakan untuk mengukur kondisi lingkungan, seperti kelembaban, suhu, angin, tingkat polusi, dan lain-lain. Untuk mengaplikasikan WSN diperlukan node sensor yang terhubung dengan baterai sebagai sumber tenaganya. Penggunaan baterai perlu diperhatikan supaya sistem dapat bekerja dengan baik tanpa sering mengganti baterai.



Gambar 2.2 Model Autentifikasi pada WSN

Sumber: mdpi.com

2.2.4 Protokol ESP-Now

Protokol ESP-Now merupakan sebuah protokol komunikasi yang dikembangkan oleh Espressif eksklusif untuk perangkat ESP. Protokol nirkabel ini berjalan pada layer data-link. Koneksi antar perangkat yang disusun dengan protokol ini dapat dikatakan meski tidak melakukan *handshake*.



Gambar 2.3 Gambar Ilustrasi Arsitektur ESP-Now

Sumber: randomnerdtutorials.com

2.2.5 ESP8266

ESP8266 merupakan sebuah *System on Chip* (SoC) Wi-Fi microchip yang digunakan untuk implementasi sistem IoT yang diproduksi oleh Espressif Systems. Modul ESP8266 dapat diprogram menggunakan bahasa pemrograman bernama Lua. Untuk terhubung dan berkomunikasi dengan perangkat yang diinginkan, ESP8266 memiliki komunikasi serial USB dan komunikasi Wi-Fi.



Gambar 2.4 ESP8266

Sumber: iotstudio.labs.telkomuniversity.ac.id

2.2.6 Sensor Flow Meter YF-S201

Sensor *flow meter* merupakan sebuah sensor atau alat ukur yang digunakan untuk mengukur debit alir dari sebuah cairan atau gas yang berada di dalam pipa atau tabung. Fungsi dari sensor ini adalah untuk mengetahui data dari volume maupun kecepatan aliran dari sebuah aliran fluida. Cara kerja dari sensor ini adalah dengan mengukur volume atau kecepatan aliran yang mengalir melalui pipa. Sensor ini terdiri oleh katup dan rotor air yang bekerja menggunakan prinsip *hall effect* (Suharjono et al., 2015).



Gambar 2.5 Sensor Flow Meter

Sumber: how2electronics.com

2.2.7 Blynk

Blynk adalah sebuah platform IoT yang dapat diakses melalui website dan aplikasi (Android dan iOS) untuk kendali beberapa perangkat IoT, seperti Arduino, ESP8266, Raspberry Pi, dan modul lainnya melewati sebuah

prototype dan internet. Koneksi antara perangkat IoT dengan Blynk dapat dilakukan secara nirkabel, yaitu dengan menggunakan kode otentikasi yang disediakan oleh Blynk untuk masing-masing *template*. Pengguna Blynk dapat mengatur kendali perangkat IoT melalui pengaturan *interface*, sehingga tidak perlu pendalaman ilmu khusus untuk menggunakan platform ini.



Gambar 2.6 Platform Blynk

sumber: docs.blynk.io

BAB 3 METODOLOGI

3.1 Tipe Penelitian

Pada saat melakukan penelitian ini, jenis penelitian yang digunakan adalah jenis implementasi (*Development*). Tipe penelitian ini merupakan jenis penelitian yang bertujuan untuk melakukan pengembangan berdasarkan hasil riset atau penelitian ilmiah terdahulu yang sudah dilakukan.

3.2 Strategi dan Rancangan Penelitian

Pada sub bab ini berisikan rencana dan struktur yang akan dibuat untuk memastikan bahwa data yang didapatkan setelah penelitian merupakan data yang valid dan sesuai dengan tujuan penelitian.

3.2.1 Studi Literatur

Dalam menyusun penelitian, studi literatur merupakan hal yang sangat penting dilakukan. Studi literatur digunakan untuk mencari referensi, menjadi dasar teori dari penelitian, memilih metode hingga memilih variabel. Studi literatur dapat diperoleh dengan membaca jurnal, buku, atau sumber lainnya yang kredibel dan terpercaya. Berikut adalah literatur yang digunakan dalam penelitian ini:

1. Saluran air bersih di rumah
2. *Internet of Things* (IoT)
3. *Wireless Sensor Network* (WSN)
4. Protokol ESP-Now
5. Sensor flow meter YF-S201

3.2.2 Rekayasa Kebutuhan Sistem

Rekayasa kebutuhan sistem merupakan kebutuhan atau persyaratan yang diperlukan agar penelitian dapat dilakukan. Untuk melaksanakan penelitian ini diperlukan dua kebutuhan sistem yaitu perangkat keras dan perangkat lunak.

Kebutuhan perangkat keras terdiri dari

1. ESP8266
2. Sensor YF-S201
3. *Power Adaptor*
4. Kabel USB
5. Kabel jumper
6. Pipa
7. Keran

Sedangkan kebutuhan perangkat lunak terdiri dari

1. Arduino IDE
2. Python
3. *Library* untuk penggunaan ESP-Now

3.2.3 Perancangan Sistem

Pada bagian ini terdapat perancangan sistem sesuai dengan rekayasa dari kebutuhan sistem yang sudah diuraikan sebelumnya. Perancangan sistem merupakan jembatan antara rekayasa kebutuhan dan implementasi sistem.

3.2.4 Implementasi Sistem

Pada bagian ini terdapat implementasi dari sistem sesuai dengan perencanaan yang telah dilakukan sebelumnya. Implementasi dimulai dari tahap pemasangan alat, kalibrasi sensor, hingga *testing*.

3.2.5 Pengujian Sistem

Pada bagian ini terdapat pengujian sistem yang telah dibuat. Pengujian ini berfungsi untuk melihat bagaimana sistem berhasil untuk memenuhi tujuan. Pengujian dilakukan dari pembacaan sensor yang kemudian akan dilanjutkan ke sebuah node master yang nantinya akan menampilkan data yang didapatkan ke pengguna. Pengujian sistem yang dilakukan, meliputi

1. Pengujian fungsionalitas perangkat, baik perangkat keras maupun lunak;
2. Pengujian pengiriman data yang dihasilkan oleh perangkat.

3.2.6 Analisis Sistem

Pada bagian ini terdapat analisis mengenai sistem yang telah dibuat. Analisis dilakukan untuk memahami, menggambarkan, serta mengevaluasi sebuah sistem. Selain itu, pada tahap analisis sistem juga dapat melihat akurasi dari sistem yang telah dibuat.

3.2.7 Kesimpulan dan Saran

Pada bagian ini akan berisikan kesimpulan dan saran atas penelitian yang telah dilakukan. Kesimpulan memuat ringkasan hasil yang diperoleh selama penelitian, sedangkan saran memuat rekomendasi untuk penelitian selanjutnya jika penelitian dikembangkan.

3.3 Subjek Penelitian

Subjek yang dijadikan pokok dari penelitian yang dilakukan adalah kemampuan dari sistem yang dibuat untuk mengaplikasikan *Internet of Things* sebagai pendeteksi kebocoran saluran air di dalam rumah.

3.4 Objek Penelitian

Objek dari penelitian yang dilakukan adalah nilai data yang didapatkan melalui sensor *flow meter* untuk aliran air.

3.5 Lokasi Penelitian

Lokasi yang digunakan dalam penelitian ini bertempat di rumah. Lokasi tersebut dipilih karena sesuai dengan penelitian yang dilakukan, yaitu deteksi kebocoran air pada saluran air di rumah. Saluran air yang digunakan untuk penelitian bersumber dari air PAM yang didistribusikan ke masing-masing rumah warga.

3.6 Metode Pengambilan Data

Metode pengambilan data yang diperoleh dalam penelitian ini merupakan metode pengambilan data primer. Metode ini merupakan sebuah metode untuk mengumpulkan data yang secara langsung didapatkan dari sumber yang asli tanpa menggunakan data yang sudah didapatkan sebelumnya. Data yang diambil bersumber dari pembacaan sensor yang kemudian diproses oleh mikrokontroler dan selanjutnya akan dikirimkan ke telepon genggam untuk dapat dibaca oleh manusia.

BAB 4 REKAYASA KEBUTUHAN

4.1 Kajian Masalah

Kajian masalah dapat mencakup beberapa aspek penting yang relevan dengan penelitian yang dilakukan. Kajian masalah ini dapat membentuk dasar yang kokoh untuk merinci lebih lanjut dalam penelitian mengenai implementasi WSN pada deteksi kebocoran pipa air dengan menggunakan komunikasi data ESP-NOW berbasis IoT. Beberapa kajian masalah tersebut, diantaranya

- a. Urgensi deteksi dini kebocoran pipa air untuk mencegah kerugian air yang signifikan, serta dampak positifnya terhadap konservasi sumber daya air.
- b. Konsep WSN dan relevansinya dalam deteksi kebocoran pipa air rumah, termasuk kelebihan dan kemungkinan tantangan yang dihadapi.
- c. Kemampuan protokol komunikasi ESP-Now dalam mendukung komunikasi efisien antara node-node sensor dalam jaringan.
- d. Pemahaman mengenai peran sensor dalam deteksi kebocoran pipa air.

4.2 Identifikasi Stakeholder

Identifikasi stakeholder dari penelitian yang dilakukan melibatkan berbagai pihak untuk penggunaan, pengembangan, hingga melakukan penelitian lebih lanjut. Beberapa stakeholder dari penelitian ini, yaitu

a. Pengguna Rumah Tangga

Pengguna rumah tangga merupakan pengguna akhir sistem yang akan menggunakan sistem yang dihasilkan penelitian ini untuk mendeteksi ada/tidaknya kebocoran dalam pipa air di rumahnya.

b. Lembaga Pengelola Air Bersih

Pihak yang memiliki kepentingan dalam pengelolaan sumber air yang dapat menggunakan sistem yang dibuat untuk peningkatan efektivitas penggunaan air.

c. Pihak Teknisi atau Instalator Air Bersih

Pihak yang melakukan penyediaan jasa untuk pemasangan saluran air, pemeliharaan sistem yang dibuat, hingga penanganan permasalahan teknik lainnya.

4.3 Kebutuhan Fungsional

Kebutuhan fungsional adalah komponen yang harus ada dalam sistem, sehingga dapat memastikan bahwa sistem mampu berjalan secara baik. Kebutuhan tersebut, yaitu

1. Sistem dapat melakukan monitoring laju alir air.

Sistem harus dapat memonitor laju kecepatan alir air yang ada dalam instalasi pipa di rumah secara real time menggunakan sensor yang sudah terpasang pada beberapa bagian pipa.

2. Sistem dapat mendeteksi data adanya kebocoran.

Sistem harus dapat mendeteksi ada atau tidak adanya kebocoran menggunakan perhitungan data yang dibaca oleh sensor.

3. Sistem dapat mengirim dan menerima data sensor ke antar node.

Sistem harus dapat mengirimkan serta menerima data yang dibaca oleh sensor dari beberapa node ke sebuah node master. Penggunaan protokol komunikasi ESP-Now digunakan untuk melakukan pertukaran data ini.

4. Sistem dapat melakukan pemberitahuan kebocoran.

Sistem yang dibuat harus dapat melakukan pemberitahuan mengenai adanya kebocoran dalam saluran air rumah pengguna. Pemberitahuan ini akan muncul melalui telepon genggam pengguna dengan perantara aplikasi layanan Blynk.

4.4 Spesifikasi Sistem

Persyaratan non-fungsional dibutuhkan agar sistem dapat berfungsi sebagaimana mestinya. Persyaratan non-fungsional ini merupakan karakteristik yang bukan merupakan persyaratan fungsional, namun dibutuhkan agar sistem yang dibuat memastikan keamanan, performa sistem, dan pengalaman pengguna yang baik.

1. Sistem yang dibuat merupakan sistem berbasis mikrokontroler, sehingga membutuhkan daya berkisar 5 – 12 volt.
2. Sistem yang dibuat kemudian akan diletakkan dalam box agar sistem lebih tertata dan terhindar dari intervensi yang tidak diinginkan (komponen listrik terkena air hingga korslet, paparan panas berlebih, dan lainnya).
3. Sistem yang dibuat membutuhkan layanan untuk menampilkan data kebocoran. Platform Blynk dapat digunakan untuk hal tersebut karena platform ini dapat mengakses data sensor dan monitoring sistem yang dibuat melalui telepon genggam.

4.5 Analisis Kebutuhan Perangkat

Untuk membuat sistem yang diinginkan, dibutuhkan adanya analisis kebutuhan perangkat. Dalam sistem yang dibuat, kebutuhan perangkat digolongkan ke dalam dua jenis, yaitu perangkat keras dan lunak.

4.5.1 Analisis Kebutuhan Perangkat Keras

1. ESP 8266

Tabel 4.1 Spesifikasi ESP8266 yang digunakan

Spesifikasi	Keterangan
Mikrocontroller 10 Kanal	ESP8266
Tegangan Input	3.3 – 5 V
Ukuran Board	57 mm x 30 mm
GPIO	13 PIN
Kanal PWM	10 Kanal
10 bit ADC Pin	1 Pin
Flash Memory	4 MB
Clock Speed	40/26/24 MHz
WiFi	IEEE 802.11 b/g/n
Frekuensi	2.4 GHz – 22.5 GHz
USB Port	Micro USB
USB to Serial Converter	CH3 40G

Dalam penelitian ini, peneliti menggunakan ESP8266 sebagai mikrokontroler karena perangkat ini mendukung layanan yang dibutuhkan dari sistem yang akan dibuat. Sistem membutuhkan perangkat yang berfungsi sebagai mikrokontroler, menyediakan Wi-Fi *module*, dapat menjalankan ESP-Now, dan dapat berfungsi menjadi Node Sensor. Perangkat lain yang memungkinkan adalah ESP32, tetapi perangkat tersebut memiliki layanan yang tidak dibutuhkan sistem dan penggunaan daya yang lebih besar dibandingkan ESP 8266.

2. Sensor *Water Flow* YF-S201

Tabel 4.2 Spesifikasi Sensor Water Flow YF-S201 yang digunakan

Spesifikasi	Keterangan
Working Flow Rate	1 – 30 liter/menit
Maximum water pressure	2.0 MPa
Pulses per Liter	450

Accuracy	$\pm 10 \%$
Operating Temperature	-25 – 80 °C
Dimensi	62.5 x 36 x 35 mm

3. Power Adaptor

Tabel 4.2 Spesifikasi Power Adaptor yang digunakan

Spesifikasi	Keterangan
Voltase Input	100 – 240 V
Frekuensi	50 – 60 Hz
Voltase Output	12 V 3 A
Ukuran Pin	5.5 x 2.5 mm
Dimensi	80 x 44 x 30 mm

4. Kabel Jumper

Tabel 4.2 Spesifikasi Kabel Jumper yang digunakan

Spesifikasi	Keterangan
Panjang Kabel	10 cm
Jenis Kabel	Tembaga
Konektor	Female-Male

4.5.2 Analisis Kebutuhan Perangkat Lunak

1. Arduino IDE

Arduino IDE adalah perangkat lunak untuk menuliskan program hingga upload hasil kompilasi sebuah mikrokontroler. Perangkat lunak ini digunakan karena penggunaannya yang mudah dan mendukung perangkat keras yang digunakan.

2. *Library* ESP-Now

Library ini digunakan untuk dapat mengakses program dengan protokol komunikasi ESP-Now yang sudah dibuat untuk ESP8266. Penggunaan *library* ini mengakibatkan komunikasi antar ESP8266 terjadi, sehingga data dapat terkirim.

3. Aplikasi Blynk IoT

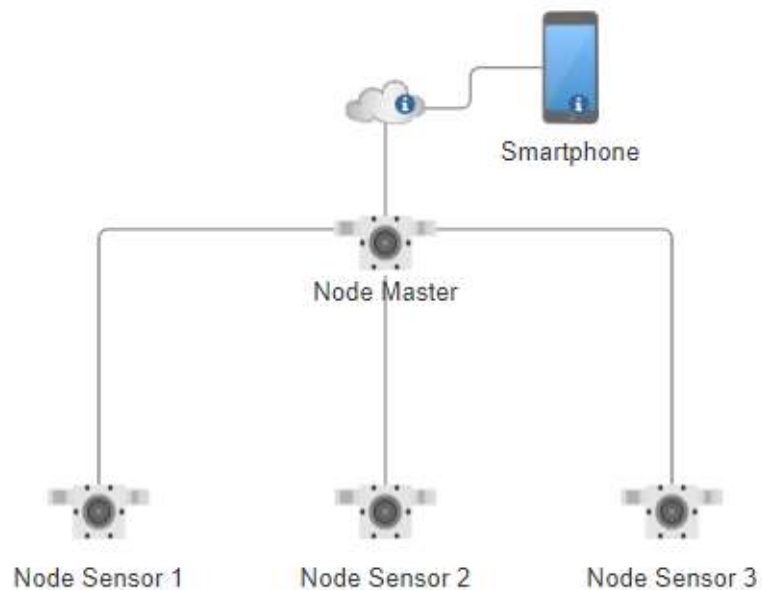
Blynk IoT merupakan sebuah layanan *Internet of Things* (IoT) yang sering digunakan untuk memantau perangkat elektronik berbasis IoT menggunakan telepon genggam. Platform ini digunakan karena dapat menampilkan hasil penelitian dengan mudah ke telepon genggam.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Dalam bab ini, akan fokus pada penggambaran lengkap mengenai perancangan dan implementasi dari penelitian ini. Pembahasan akan terbagi dua, yaitu mengenai perancangan sistem dan implementasi sistem.

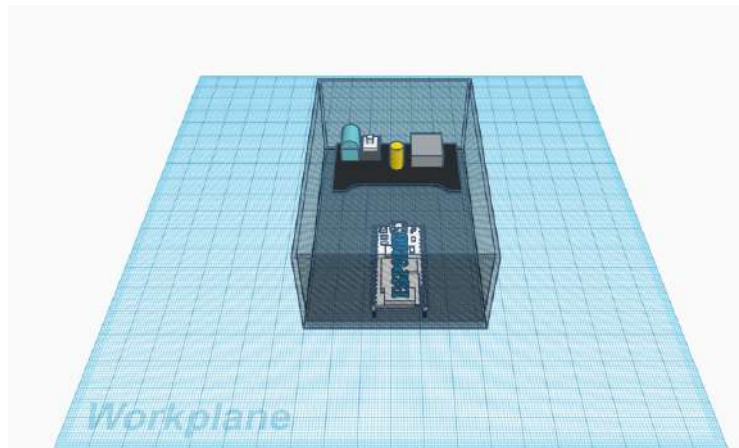
5.1 Perancangan Sistem

Perancangan sistem adalah proses perencanaan bagaimana penelitian ini dilakukan. Dalam sub bab ini akan menjelaskan bagaimana perancangan sistem dalam implementasinya untuk WSN. Node sensor akan disusun ke dalam sistem menggunakan topologi star bersama dengan node master, lalu ESP-Now digunakan sebagai protokol komunikasi data antar node sensor dengan node master. Ketika node master sudah menerima nilai data dari node sensor, kemudian nilai tersebut akan diolah untuk mengklasifikasi ada atau tidaknya kebocoran. Setelah diklasifikasi, informasi mengenai kebocoran akan diteruskan oleh node master ke *smartphone* melalui platform Blynk.



Gambar 5.1 Perancangan Sistem WSN

5.1.1 Perancangan *Prototype*

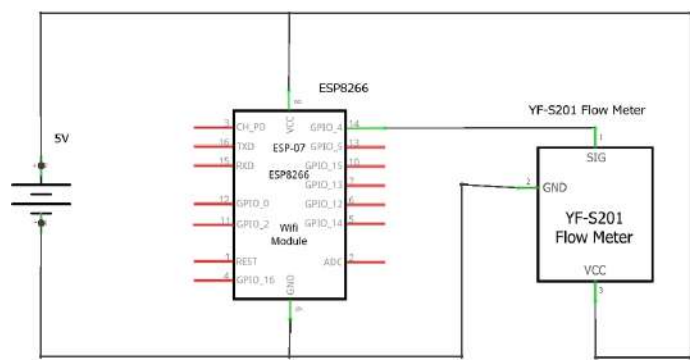


Gambar 5.2 Perancangan Prototipe Sistem

Prototipe dirancang seperti pada gambar yang ditunjukkan oleh Gambar 5.2. Bentuk dari prototipe untuk node dalam sistem ini adalah sebuah kotak (*box project*) di mana di dalamnya terdapat perangkat keras yang digunakan dalam penelitian. Pada bagian samping kotak prototipe, diberi lubang sebagai akses kabel ke sensor dan tegangan. Dibutuhkan empat buah kotak untuk menyusun prototipe keseluruhan yang terdiri atas Node Master dan tiga Node Sensor.

5.1.2 Perancangan Diagram Skematik Keseluruhan

Dalam melakukan perancangan diagram skematik, diperlukan perangkat berupa ESP8266 dan sensor Water Flow YF-S201, yang keduanya kemudian terhubung ke sumber tegangan sebesar 5V. Node Master dan Node Sensor memiliki diagram skematik yang sama. Berikut merupakan gambar rancangan diagram skematik dari sistem.



Gambar 5.3 Diagram Skematik Keseluruhan

Dari gambar yang ditunjukkan pada Gambar 5.3, ditampilkan hubungan antara ESP8266 dengan sensor Water Flow YF-S201 yang difungsikan untuk mendapatkan nilai data debit alir air. Kedua perangkat tersebut saling terkoneksi menggunakan kabel yang menghubungkan antara pin dari ESP8266 dan sensor

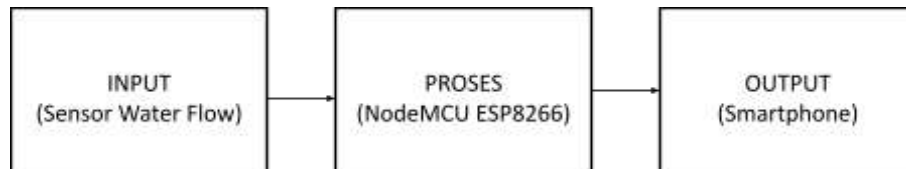
YF-S201. Berikut merupakan tabel yang menunjukkan koneksi pin dari perangkat yang digunakan.

Tabel 5.1 Koneksi Sensor Water Flow YF-S201 dengan ESP8266

ESP8266	Sensor Water Flor YF-S201	Sumber Tegangan
Pin 4	SIG	
VCC	VCC	5V
GND	GND	GND

5.1.3 Perancangan Perangkat Keras

Dalam melakukan penelitian ini, perlu dilibatkan beberapa perangkat keras. Berikut penjelasan mengenai perancangan perangkat keras tersebut.

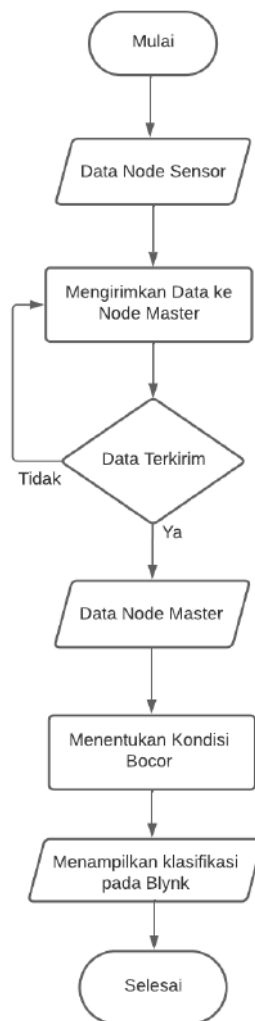


Gambar 5.4 Perancangan Perangkat Keras

Dari gambar yang ditampilkan oleh Gambar 5.2 dijelaskan bahwa perancangan perangkat keras terdiri dari tiga bagian: input, proses, dan output. Pada bagian input, terdapat perangkat keras Sensor Water Flow yang digunakan untuk mengambil nilai flowmeter dari air yang mengalir pada sistem. Kemudian nilai tersebut akan dikirim dan diolah melalui perangkat keras NodeMCU ESP8266, yang merupakan termasuk bagian Proses. Kemudian pada bagian Output menggunakan perangkat keras *Smartphone* untuk menampilkan data yang sudah dikirim dan diolah pada langkah sebelumnya.

5.1.4 Perancangan Perangkat Lunak

Dalam sub bab ini, akan dijelaskan terkait *flowchart* perancangan perangkat lunak. Berikut adalah penjelasan sistem yang akan dibuat.



Gambar 5.5 Perancangan Perangkat Lunak

5.2 Implementasi Sistem

Sub bab ini merupakan bagian yang akan menjelaskan terkait tahapan pengembangan sistem, di mana rancangan yang telah dibuat sebelumnya akan diwujudkan menjadi sistem yang berfungsi secara penuh. Pengimplementasian sistem akan terbagi menjadi tiga, yaitu Implementasi Prototipe, Implementasi Perangkat Keras, dan Implementasi Perangkat Lunak.

5.2.1 Implementasi Prototipe

Untuk mengimplementasikan *prototype* yang telah dibuat sebelumnya, maka dalam penelitian ini dibutuhkan empat buah *project box* sebagai wadah perangkat keras yang digunakan. Kotak tersebut memiliki ukuran volume sebesar panjang 150 mm x lebar 95 mm x tinggi 50 mm.



Gambar 5.6 Implementasi *Prototype* Keseluruhan Sistem



Gambar 5.7 Implementasi *Prototype* Tiap Node

Dalam pengimplementasian sistem, perangkat keras dari sistem adalah ESP8266, Sensor Water Flow YF-S201, dan catu daya. Untuk menghubungkan perangkat keras tersebut membutuhkan beberapa kabel jumper. Gambar 5.6 menunjukkan *prototype* yang telah dibuat untuk keseluruhan sistem, meliputi Node 1, Node 2, Node 3, dan Node Master. Sedangkan Gambar 5.7 menunjukkan hasil implementasi *prototype* sebuah node.

5.2.2 Implementasi Perangkat Keras

Setelah dilakukan implementasi pada prototype, hal selanjutnya yang dilakukan adalah implementasi perangkat keras. Implementasi perangkat keras ini dibagi menjadi dua yaitu Implementasi Perangkat Keras secara Keseluruhan dan implementasi perangkat keras pada node.

5.2.2.1 Implementasi Perangkat Keras Keseluruhan



Gambar 5.8 Implementasi Perangkat Keras Keseluruhan

Gambar 5.8 tersebut menunjukkan implementasi perangkat keras secara keseluruhan, di mana Node 1, Node 2, dan Node 3, hingga Node Master terhubung ke dalam sebuah saluran pipa air. Seluruh node memiliki perangkat keras dan koneksi yang sama, yaitu ESP8266, Sensor Flow Meter YF-S201, dan catu daya.

5.2.2.2 Implementasi Perangkat Keras Tiap Node



Gambar 5.9 Implementasi Perangkat Keras Pada Node

Dari Gambar 5.9, dapat dijelaskan bahwa beberapa perangkat keras yang terhubung adalah ESP8266, Sensor YF-S201, catu daya, dan modul tegangan listrik. Pin D4 pada ESP8266 terhubung ke kabel sinyal atau pulsa output dari sensor, sedangkan pin 5V dan Ground baik milik ESP8266 dan Sensor YF-S201 terhubung ke modul tegangan. Modul

tegangan kemudian diberi tegangan oleh power adaptor yang nantinya akan terhubung ke aliran listrik.

5.2.3 Implementasi Perangkat Lunak

Di sub bagian ini akan menjelaskan implementasi perangkat lunak. Setelah perangkat keras diterapkan, perangkat lunak diterapkan pada perangkat. Penerapan perangkat lunak kemudian dibagi menjadi dua bagian, yaitu penerapan perangkat lunak pada node sensor dan penerapan perangkat lunak pada node master.

5.2.3.1 Implementasi Perangkat Lunak Node Sensor

Tabel 5.1 Kode Program Library Sistem

No.	Kode Program
1	#include <ESP8266WiFi.h>
2	#include <espnw.h>

Tabel 5.2 Penjelasan Kode Program Library Sistem

No.	Penjelasan Kode Program
1	Menyertakan <i>library</i> bernama WiFi ESP8266
2	Menyertakan <i>library</i> untuk penggunaan ESP-Now

Tabel 5.3 Kode Program Membaca Nilai Sensor

No.	Kode Program
1	#define FLOW_1 D4
2	
3	float flowRate;
4	long currentMillis = 0;
5	long previousMillis = 0;
6	int interval = 1000;
7	float calibrationFactor = 4.5;
8	volatile byte pulseCount;
9	byte pulse1Sec = 0;
10	unsigned long flowMilliLitres;
11	unsigned int totalMilliLitres;
12	float flowLitres;
13	float totalLitres;
14	
15	void IRAM_ATTR pulseCounter() {
16	pulseCount++;
17	}
18	

19	void setup() {
20	Serial.begin(115200);
21	
22	pinMode(FLOW_1, INPUT_PULLUP);
23	pulseCount = 0;
24	flowRate = 0.0;
25	flowMilliLitres = 0;
26	totalMilliLitres = 0;
27	previousMillis = 0;
28	
29	attachInterrupt(digitalPinToInterrupt(FLOW_1),
	pulseCounter, FALLING);
30	
31	float readFlowMeter() {
32	currentMillis = millis();
	if (currentMillis - previousMillis > interval){
33	pulse1Sec = pulseCount;
34	pulseCount = 0;
35	
36	flowRate = ((1000.0 / (millis() - previousMillis)) * pulse1Sec) / calibrationFactor;
37	previousMillis = millis();
38	
39	flowMilliLitres = (flowRate / 60) * 1000;
40	flowLitres = (flowRate / 60);
41	totalMilliLitres += flowMilliLitres;
42	totalLitres += flowLitres;
43	
44	Serial.print("Sensor 1\n");
45	Serial.print("Flow rate: ");
46	Serial.print(float(flowRate));
47	Serial.print("L/min");
48	Serial.print("\t");
49	}
50	return 0.0;
51	}

Tabel 5.4 Penjelasan Kode Program Membaca Nilai Sensor

No.	Penjelasan Kode Program
1	Mendefinisikan pin D4 sebagai FLOW_1
3-13	Mengukur flow meter melalui pulsa dari aliran yang melewati sensor
15-17	Membuat fungsi pulseCounter untuk menghitung pulsa
19-29	Fungsi setup program untuk inisialisasi variabel dan parameter sensor
31-42	Membuat fungsi readFlowMeter untuk mengukur flow meter dalam satuan waktu (liter/menit)
44-48	Menampilkan data yang dibaca oleh sensor

Tabel 5.5 Kode Program ESP-Now

No.	Kode Program
1	<code>uint8_t masterNodeMAC[] = {0x68, 0xC6, 0x3A, 0xF2, 0xEB, 0xCF};</code>
2	
3	<code>struct struct_message {</code>
4	<code>int nodeId = 1;</code>
5	<code>float HASIL_BACA_SENSOR;</code>
6	<code>};</code>
7	<code>struct_message flowMeterData;</code>
8	
9	<code>void onDataSent(uint8_t *mac_addr, uint8_t sendStatus) {</code>
10	<code>Serial.print("Last Packet Send Status: ");</code>
11	<code>if (sendStatus == 0) {</code>
12	<code>Serial.println("Delivery Success!");</code>
13	<code>} else {</code>
14	<code>Serial.println("Delivery Fail!");</code>
15	<code>}</code>
16	<code>}</code>
17	
18	<code>void setup() {</code>
19	<code>Serial.begin(115200);</code>
20	<code>if (esp_now_init() != 0) {</code>
21	<code>Serial.println("ESP-NOW initialization failed");</code>
22	<code>return;</code>
23	<code>}</code>
24	
25	<code>esp_now_set_self_role(ESP_NOW_ROLE_CONTROLLER);</code>
26	<code>esp_now_register_send_cb(onDataSent);</code>
27	
28	<code>esp_now_add_peer(masterNodeMAC, ESP_NOW_ROLE_SLAVE, 1, NULL, 0);</code>
29	<code>}</code>
30	<code>void loop() {</code>
31	<code>float HASIL_BACA_SENSOR = readFlowMeter();</code>
32	
33	<code>flowMeterData.HASIL_BACA_SENSOR = flowRate;</code>
34	
35	<code>esp_now_send(masterNodeMAC, (uint8_t *)&flowMeterData, sizeof(flowMeterData));</code>
36	
37	<code>Serial.print("Flow Rate sent: ");</code>
38	<code>Serial.println(HASIL_BACA_SENSOR);</code>
39	
40	<code>delay(5000);</code>
41	<code>}</code>

Tabel 5.6 Penjelasan Kode Program ESP-Now

No.	Penjelasan Kode Program
1	Mendeklarasikan alamat MAC dari Node Master (penerima data)
3-7	Membuat struct dan variabel untuk menyimpan data bacaan sensor
9-16	Membuat fungsi onDataSent untuk memeriksa pengiriman data
18-23	Fungsi setup untuk menginisialisasi ESP-Now
25-26	Mengatur perangkat sebagai role Controller untuk pengiriman data
28	Menambah peer Master Node sebagai penerima data
30-40	Fungsi loop untuk mengirimkan data bacaan sensor dan menampilkan data yang dikirimkan pada serial monitor

5.2.3.2 Implementasi Perangkat Lunak Node Master

Tabel 5.7 Kode Program Library Sistem

No.	Kode Program
1	#include <ESP8266WiFi.h>
2	#include <espnow.h>
3	#include <BlynkSimpleEsp8266.h>

Tabel 5.8 Penjelasan Kode Program Library Sistem

No.	Penjelasan Kode Program
1	Menyertakan <i>library</i> bernama WiFi ESP8266
2	Menyertakan <i>library</i> untuk penggunaan ESP-Now
3	Menyertakan <i>library</i> untuk penggunaan Blynk

Tabel 5.9 Kode Program Membaca Nilai Sensor

No.	Kode Program
1	#define FLOW_1 D4
2	
3	float flowRate;
4	long currentMillis = 0;
5	long previousMillis = 0;
6	int interval = 1000;
7	float calibrationFactor = 7.5;
8	volatile byte pulseCount;
9	byte pulse1Sec = 0;
10	unsigned long flowMilliLitres;
11	unsigned int totalMilliLitres;
12	float flowLitres;

13	float totalLitres;
14	
15	void IRAM_ATTR pulseCounter(){
16	pulseCount++;
17	}
18	
19	void setup() {
20	Serial.begin(115200);
21	
22	pinMode(FLOW_1, INPUT_PULLUP);
23	pulseCount = 0;
24	flowRate = 0.0;
25	flowMilliLitres = 0;
26	totalMilliLitres = 0;
27	previousMillis = 0;
28	
29	attachInterrupt(digitalPinToInterrupt(FLOW_1),
30	pulseCounter, FALLING);
31	float readFlowMeter() {
32	currentMillis = millis();
33	if (currentMillis - previousMillis > interval){
34	pulse1Sec = pulseCount;
35	pulseCount = 0;
36	flowRate = ((1000.0 / (millis() - previousMillis)) * pulse1Sec) / calibrationFactor;
37	previousMillis = millis();
38	
39	flowMilliLitres = (flowRate / 60) * 1000;
40	flowLitres = (flowRate / 60);
41	totalMilliLitres += flowMilliLitres;
42	totalLitres += flowLitres;
43	
44	Serial.print("Sensor Master\n");
45	Serial.print("Flow rate: ");
46	Serial.print(float(flowRate));
47	Serial.print("L/min");
48	Serial.print("\t");
49	}
50	return 0.0;
51	}

Tabel 5.10 Penjelasan Kode Program Membaca Nilai Sensor

No.	Penjelasan Kode Program
1	Mendefinisikan pin D4 sebagai FLOW_1
3-13	Mengukur flow meter melalui pulsa dari aliran yang melewati sensor
15-17	Membuat fungsi pulseCounter untuk menghitung pulsa
19-29	Fungsi setup program untuk inisialisasi variabel dan parameter sensor
31-42	Membuat fungsi readFlowMeter untuk mengukur flow meter dalam

44-48	satuan waktu (liter/menit) Menampilkan data yang dibaca oleh sensor
-------	--

Tabel 5.11 Kode Program ESP-Now

No.	Kode Program
1	<code>struct struct_message {</code>
2	<code> int nodeId;</code>
3	<code> float HASIL_BACA_SENSOR;</code>
4	<code>};</code>
5	<code>struct_message flowMeterData;</code>
6	<code>struct_message data1;</code>
7	<code>struct_message data2;</code>
8	<code>struct_message data3;</code>
9	
10	<code>struct_message dataStruct[3] = {data1, data2, data3};</code>
11	
12	<code>void onDataReceived(uint8_t *mac_addr, uint8_t *data,</code>
13	<code>uint8_t dataLen) {</code>
14	
15	<code> memcpy(&flowMeterData, data, sizeof(flowMeterData));</code>
16	<code> Serial.printf("Node Sensor %u: ", flowMeterData.nodeId,</code>
17	<code>dataLen);</code>
18	
19	<code> dataStruct[flowMeterData.nodeId-1].HASIL_BACA_SENSOR =</code>
20	<code>flowMeterData.HASIL_BACA_SENSOR;</code>
21	<code> Serial.printf("%f L/m\n",</code>
22	<code>dataStruct[flowMeterData.nodeId-1].HASIL_BACA_SENSOR);</code>
23	<code> Serial.println();</code>
24	<code>}</code>
25	<code>void setup() {</code>
26	<code> if (esp_now_init() != 0) {</code>
27	<code> Serial.println("ESP-NOW initialization failed");</code>
28	<code> return;</code>
29	<code> }</code>
30	<code> esp_now_register_recv_cb(onDataReceived);</code>
31	<code>}</code>

Tabel 5.12 Penjelasan Kode Program ESP-Now

No.	Penjelasan Kode Program
1-4	Struktur data yang diterima oleh ESP-Now berisikan NodeId dan HASIL_BACA_SENSOR
5-10	Deklarasi variabel untuk menyimpan data dari node sensor
12-13	Fungsi untuk menerima data yang dikirimkan
15-20	Menyalin data yang dikirim ke variabel flowMeterData
21-23	Menampilkan data yang sudah disalin ke serial monitor
25-28	Inisialisasi kode ESP-Now
30	Menjalankan <i>callback</i> untuk menangani data yang diterima

Tabel 5.11 Kode Program Blynk dan Klasifikasi Kebocoran

No.	Kode Program
1	#define BLYNK_TEMPLATE_ID "TMPL64yyM66vT"
2	#define BLYNK_TEMPLATE_NAME "NodeMCUWaterflow"
3	
4	char auth [] = "yGKr7ZQuXZZD2Ugloht3_x4GzF-xzTwa";
5	
6	BlynkTimer timer;
7	
8	void setup() {
9	Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
10	timer.setInterval(5000L, sendSensor);
11	}
12	
13	void loop() {
14	Blynk.run();
15	timer.run();
16	}
17	
18	void sendSensor() {
19	float valueData1 = dataStruct[0].HASIL_BACA_SENSOR;
20	float valueData2 = dataStruct[1].HASIL_BACA_SENSOR;
21	float valueData3 = dataStruct[2].HASIL_BACA_SENSOR;
22	float toleransi = 0.1;
23	
24	Blynk.virtualWrite(V0, valueData1);
25	Blynk.virtualWrite(V1, valueData2);
26	Blynk.virtualWrite(V2, valueData3);
27	Blynk.virtualWrite(V3, sensorMaster);
28	
29	if (sensorMaster - toleransi > valueData1 + valueData2 +
30	valueData3) {
31	Blynk.logEvent("notifications", "Sistem Mendeteksi
32	Kebocoran");
33	} else {
34	//No water leaks detected
35	}
36	}

Tabel 5.12 Penjelasan Kode Program Blynk dan Klasifikasi Kebocoran

No.	Penjelasan Kode Program
1-2	Mendefinisikan ID dan nama template yang digunakan pada Blynk
4	Mendeklarasikan token otentikasi untuk menghubungkan perangkat ke Blynk
6	Bagian dari <i>library</i> Blynk untuk mengelola waktu
8-11	Memulai koneksi Blynk dengan interval 5 detik
13-16	Bagian loop untuk terus menjalankan komunikasi ke Blynk
18-27	Membuat fungsi sendSensor untuk mengirim hasil baca sensor yang

29-36	diterima dari node sensor 1, 2, dan 3 Membuat notifikasi pada Blynk akan adanya deteksi kebocoran
-------	--

5.3 Pengujian Awal

Dalam penelitian ini, diperlukan adanya pengujian awal. Hal ini dikarenakan sensor yang digunakan pada tiap node membutuhkan kalibrasi. Pengujian awal ini merupakan langkah untuk memastikan akurasi dan keandalan data yang dihasilkan oleh sensor. Untuk mengetahui nilai kalibrasi sensor tersebut, digunakan persamaan tiga variabel dari tiga percobaan pengambilan data sensor yang berbeda. Proses pengujian awal ini terinci ke dalam sub bab berikut:

5.3.1 Prosedur Pengambilan Data

Pengambilan data untuk tahap pengujian awal dilakukan sebanyak tiga kali. Proses pengambilan data tersebut dibedakan berdasarkan perbedaan laju air pada tiap keran di saluran air. Tabel berikut adalah data yang diambil untuk pengujian awal. Nilai pada node sensor merupakan nilai tanpa kalibrasi, sedangkan nilai pada Node Master merupakan nilai dengan nilai kalibrasi awal sensor (0,1333).

Tabel 5.13 Tabel Pengujian Awal

No	Sensor Master	Node Sensor 1	Node Sensor 2	Node Sensor 3
1	11,83	10.8	23.85	18.585
2	11,53	14.4	17.235	20.25
3	11,37	10.575	24.84	15.75

5.3.2 Perhitungan Persamaan Kalibrasi

Dari data yang dihasilkan pada sub bab sebelumnya, kemudian dibuat persamaan untuk menghitung nilai kalibrasi. Persamaan tersebut merupakan persamaan linear empat variabel. Dari perhitungan persamaan tersebut, kemudian akan menghasilkan nilai akhir berupa nilai dari masing-masing variabel. Persamaan tersebut adalah sebagai berikut:

$$10,8x + 23,85y + 18,585z = 11,83$$

$$14,4x + 17,235y + 20,25z = 11,53$$

$$10,575x + 24,84y + 15,75z = 11,37$$

Dari persamaan yang didapatkan, kemudian nilai masing-masing variabel dapat dihasilkan menggunakan array sebagai berikut.

Tabel 5.14 Tabel Menghitung Nilai Kalibrasi Sensor

No.	Kode Program
1	import numpy as np
2	
3	# Koefisien matriks (A)
4	coefficients = np.array([[10.8, 23.85, 18.585],
5	[14.4, 17.235, 20.25],
6	[10.575, 24.84, 15.75]])
7	
8	# Konstanta (b)
9	constants = np.array([11.83, 11.53, 11.37])
10	
11	# Menghitung solusi
12	solution, residuals, _, _ = np.linalg.lstsq(coefficients,
13	constants, rcond=None)
14	
15	# Menampilkan solusi
16	print("Solusi:")
17	print("x =", solution[0])
18	print("y =", solution[1])
	print("z =", solution[2])

Dari pengujian awal, akan dihasilkan nilai variabel sebagai berikut:

x = 0.2222222222222214

y = 0.2222222222222227

z = 0.22222222222222326

Selanjutnya, dari nilai yang didapatkan kemudian dimasukkan ke dalam implementasi perangkat lunak sebagai nilai pengkalibrasian sensor. Sehingga pembacaan sensor dapat selaras antara pembacaan di tiap node sistem.

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini akan dibahas perihal pengujian dan analisis oleh sistem. Setelah sistem selesai dibuat baik dari perangkat keras dan perangkat lunaknya, maka selanjutnya sistem tersebut akan diuji untuk memeriksa terpenuhi atau tidaknya kebutuhan sistem. Dalam tahap ini, pengujian akan dilakukan untuk mengetahui fungsionalitas dari sistem, baik dari Sensor Water Flow YF-S201, implementasi *Wireless Sensor Network* (WSN) yang berjalan menggunakan ESP-Now, hingga pengiriman data yang telah diolah ke smartphone.

6.1 Pengujian Sensor Water Flow YF-S201

Pengujian ini dilakukan untuk mendapatkan hasil dari kinerja Sensor YF-S201 dalam mendeteksi kecepatan arus air yang melewati sensor tersebut. Untuk melakukan pengujian ini, dibutuhkan ESP8266 sebagai mikrokontroler yang digunakan untuk mengkonversi data analog ke digital, penyimpanan dan pemrosesan data, hingga mengirimkan data ke Node Master nantinya.

6.1.1 Prosedur Pengujian

Prosedur melakukan tahapan pengujian Sensor *Water Flow* YF-S201 adalah sebagai berikut:

1. Menghubungkan kabel dari sensor ke pin ESP8266.
2. Menjalankan program untuk pembacaan sensor melalui ESP8266.
3. Melakukan pengujian pada kondisi yang berbeda.

6.1.2 Hasil dan Analisis Pengujian

Hasil yang didapatkan oleh pengujian sensor ditampilkan pada Tabel 6.1. Pengujian diterapkan terhadap tiga node sensor dan Node Master. Masing-masing sensor akan diuji untuk menampilkan data pembacaan hasil sensor.

Tabel 6.1 Pengujian Sensor YF-S201

No	Nama Sensor	Output Sensor (L/m)
1	Node Sensor 1	9.20 L/m
2	Node Sensor 2	3.24 L/m
3	Node Sensor 3	9.20 L/m

Setelah dilakukan pengujian pada keempat sensor, didapatkan hasil bahwa seluruh sensor tersebut dapat berjalan dengan baik. Nilai yang dihasilkan dari keempat sensor terbaca pada serial monitor.

6.2 Pengujian ESP-NOW untuk Pengiriman Data dari Node Sensor ke Node Master

Pengujian dilakukan untuk mengetahui keberhasilan pengiriman data dari node sensor ke node master berhasil. Untuk melakukan pengujian ini, digunakan ESP-Now sebagai protokol komunikasi data yang menghubungkan antara node sensor dan node master.

6.2.1 Prosedur Pengujian

Prosedur melakukan pengujian ESP-Now untuk mengirimkan data Node Sensor ke sebuah Node Master adalah sebagai berikut:

1. Menjalankan program mengirimkan data menggunakan ESP-Now.
2. Melihat data yang terkirim pada node sensor di serial monitor.
3. Memeriksa data yang terkirim pada node master di serial monitor.

6.2.2 Hasil dan Analisis Pengujian

Tabel 6.2 Pengujian ESP-NOW Dari Node Sensor ke Node Master

No.	Node Sensor	Data yang Dikirimkan	Pembacaan Node Master
1	Node 1	2,710569 L/m	2,710569 L/m
2	Node 2	3,865894 L/m	3,865894 L/m
3	Node 3	3,954765 L/m	3,954765 L/m

6.3 Pengujian Pengiriman Data Node Master ke Blynk

Pengujian ini dibuat agar mendapatkan keberhasilan data yang terkirim di node master melalui platform Blynk. Data yang ditampilkan berupa *gauge chart* yang berisikan nilai dari data yang terkirim dari node sensor ke node master.

6.3.1 Prosedur Pengujian

Prosedur untuk melakukan pengujian Pengiriman Data Node Master ke Blynk adalah sebagai berikut:

1. Menjalankan program untuk menghubungkan node master ke Blynk.
2. Memeriksa koneksi antara node master dengan Blynk.
3. Menampilkan hasil baca Sensor YF-S201 dari node di Blynk.

6.3.2 Hasil dan Analisis Pengujian



Gambar 6.1 Pengujian Pengiriman Data ke Blynk

Dari Gambar 6.1, dapat disimpulkan bahwa sistem mampu melakukan pengiriman data ke Blynk. Data yang dikirimkan ditampilkan dalam bentuk *gauge chart*. Masing-masing node memiliki *gauge chart*-nya sendiri yang dibedakan berdasarkan penamaan dan warnanya pada Blynk. Nilai data yang ditampilkan pada Blynk memiliki satuan liter/menit. Data tersebut akan mengalami *update* setiap 5 detik.

6.4 Pengujian Keseluruhan Sistem

Pengujian keseluruhan sistem dilakukan untuk memastikan sistem yang sudah dibuat dapat mengirimkan data dengan baik. Selain itu, dalam pengujian keseluruhan sistem ini juga akan diperoleh hasil akhir berupa informasi mengenai ada atau tidaknya deteksi kebocoran air. Informasi tersebut berupa peringatan (*alert*) yang ditampilkan di telepon genggam. Jika terjadi kebocoran air, maka telepon genggam akan menerima notifikasi dari platform Blynk. Namun apabila kebocoran tidak terdeteksi, maka telepon genggam tidak akan menerima notifikasi dari Blynk (keadaan normal).

6.4.1 Prosedur Pengujian

Prosedur pengujian keseluruhan sistem adalah sebagai berikut:

1. Menghubungkan semua perangkat yang membutuhkan daya pada semua node ke power supply.
2. Menjalankan program pada masing-masing node sensor dan node master.
3. Memeriksa data yang dikirimkan oleh node sensor sudah diterima pada node master dengan membuka serial monitor pada program node master.
4. Memeriksa keberhasilan pengiriman data dari node master ke platform Blynk untuk tiap node sensor.

5. Mengatur kondisi kebocoran pada pipa sehingga sistem akan memunculkan notifikasi terdeteksinya kebocoran pada sistem.
6. Mengamati keandalan pengiriman data dan keberhasilan penggolongan kondisi kebocoran pipa air.
7. Membuat kesimpulan.

6.4.2 Hasil dan Analisis Pengujian

Berikut merupakan hasil dari sistem setelah dilakukan pengujian keseluruhan sistem.

6.4.2.1 Pengujian Pengiriman Data

Tabel 6.5 Pengujian Node Master

No.	Data Node 1	Data Node 2	Data Node 3	Status Pengiriman
1	2,13	4,889	4,67	Data Diterima
2	2,09	4,932	4,71	Data Diterima
3	2,44	4,31	5,11	Data Diterima
4	2,44	5,599	3,87	Data Diterima
5	2,8	4,977	4,62	Data Diterima
6	1,73	3,954	4,71	Data Diterima
7	2,18	1,599	5,02	Data Diterima
8	2,31	5,332	3,73	Data Diterima
9	2,71	3,911	3,96	Data Diterima
10	1,38	4,222	5,02	Data Diterima
11	3,02	3,155	4,04	Data Diterima
12	2,71	6,043	2,49	Data Diterima
13	2,31	5,242	4,04	Data Diterima
14	0	0	0	Data Diterima
15	0	0	0	Data Diterima
16	0	0	0	Data Diterima

17	0	0	8,44	Data Diterima
18	0	9,954	1,69	Data Diterima
19	3,47	0	8	Data Diterima
20	3,69	7,998	0	Data Diterima

6.4.2.3 Pengujian Penerimaan Data pada Blynk



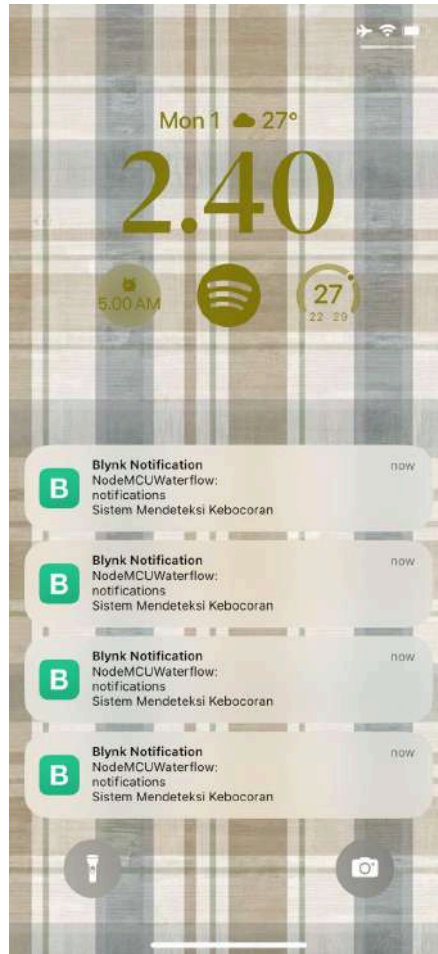
Gambar 6.2 Penerimaan Data di Blynk

Dari Gambar 6.3, diperoleh bahwa data dari ketiga node berhasil dikirimkan ke Blynk melalui master node yang terkoneksi ESP-Now. Implementasi ESP-Now pada penelitian ini memungkinkan visualisasi data dengan efektif melalui antarmuka Blynk dan *gauge chart*.

6.4.2.4 Pengujian Deteksi Kebocoran Air

No	Sensor1	Sensor2	Sensor3	Master	Kondisi Nyata	Notifikasi Bocor	Benar /Salah
1	2,13	4,889	4,67	11,53	Tidak	Tidak	Benar
2	2,09	4,932	4,71	11,68	Tidak	Tidak	Benar
3	2,44	4,31	5,11	11,68	Tidak	Tidak	Benar
4	2,44	5,599	3,87	11,83	Tidak	Tidak	Benar
5	2,8	4,977	4,62	11,83	Tidak	Tidak	Benar
6	1,73	3,954	4,71	11,68	Bocor	Ada	Benar
7	2,18	1,599	5,02	11,68	Bocor	Ada	Benar

8	2,31	5,332	3,73	12,14	Bocor	Ada	Benar
9	2,71	3,911	3,96	11,53	Bocor	Ada	Benar
10	1,38	4,222	5,02	12,91	Bocor	Tidak	Salah
11	3,02	3,155	4,04	11,83	Bocor	Ada	Benar
12	2,71	6,043	2,49	11,83	Bocor	Ada	Benar
13	2,31	5,242	4,04	11,99	Bocor	Ada	Benar
14	0	0	0	0,31	Bocor	Ada	Benar
15	0	0	0	0,77	Bocor	Ada	Benar
16	0	0	0	0,46	Bocor	Tidak	Salah
17	0	0	8,44	9,84	Bocor	Tidak	Salah
18	0	9,954	1,69	11,68	Tidak	Tidak	Benar
19	3,47	0	8	11,37	Tidak	Tidak	Benar
20	3,69	7,998	0	11,53	Tidak	Tidak	Benar



Gambar 6.3 Notifikasi Adanya Kebocoran Air di *Smartphone*

Setelah dilakukan pengujian dengan membuka kebocoran, maka sensor akan mendeteksi nilai *flowrate* yang berbeda antara node sensor dan node master. Perbedaan nilai inilah yang kemudian menjadi acuan deteksi kebocoran air pada sistem. Ketika sistem mendeteksi perbedaan nilai pada sensor, maka Blynk akan mengirimkan notifikasi ke *smartphone* pengguna. Notifikasi tersebut ditampilkan pada Gambar 6.3.

BAB 7 PENUTUP

Setelah dilakukan pengujian dan analisis, maka dalam bab ini akan memberikan kesimpulan mengenai keberhasilan sistem yang dibuat. Keberhasilan tersebut dapat diukur melalui sesuai dan tidaknya sistem tersebut dengan sistem yang diharapkan sebelumnya. Kesimpulan yang didapatkan kemudian diharapkan menjadi pertimbangan sebagai tolak ukur untuk mengembangkan penelitian ini apabila ingin dikembangkan kedepannya.

7.1 Kesimpulan

Setelah perangkaian perangkat dan pengujian sistem telah dilakukan, maka peneliti dapat menarik beberapa kesimpulan. Berikut merupakan kesimpulan dari penelitian ini berdasarkan rumusan masalah yang terdapat dalam Bab 1:

1. Proses pembacaan data berhasil dilakukan oleh sensor, baik sensor YF-S201 yang berada pada Node 1, Node 2, Node 3, hingga Node Master. Pembacaan sensor menunjukkan laju air dengan satuan liter/menit. Sehingga pembacaan sensor sudah sesuai dengan apa yang diharapkan.
2. Proses pengiriman data dari masing-masing node sensor ke node master menggunakan ESP-Now berjalan dengan baik. Semua data dapat terkirim, terbaca, dan diolah di master node. Data yang dikirimkan kemudian juga dapat dilihat dengan mudah oleh pengguna menggunakan aplikasi Blynk pada *smartphone*.
3. Performa sistem menunjukkan hasil yang cukup baik. Dari proses pengujian, didapatkan hasil bahwa sistem dapat mendeteksi kebocoran yang ada.

7.2 Saran

Berikut merupakan saran yang dapat dipertimbangkan ketika ingin melanjutkan penelitian ini:

1. Apabila akan dilakukan penelitian pengembangan, diharapkan dapat melakukan pemilihan metode deteksi yang lebih akurat dan praktis, sehingga akan meminimalisir adanya kesalahan deteksi pada sistem.
2. Sistem yang dibuat hanya mendeteksi kebocoran, belum mendeteksi lokasi bocornya. Perlu dilakukan olah data agar sistem dapat berjalan untuk dapat mendeteksi lokasi kebocorannya.
3. Mempertimbangkan sensor yang digunakan untuk memperbaiki sistem yang ada. Sensor pada penelitian ini memiliki nilai akurasi yang kurang baik, sehingga perlu waktu untuk menyelaraskan nilai yang dibaca sensor untuk diolah di dalam program.

DAFTAR REFERENSI

- Aji, B. W., & Nurwasito, H. . (2023). Implementasi Sistem Monitoring Sungai berbasis LoRa-MQTT Gateway. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 7(4), 1689–1698. Diambil dari <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/12562>
- Badan Pusat Statistik. (2020, December 29). Volume Air yang Disalurkan dan yang Bocor pada Perusahaan Air Bersih di Provinsi Nusa Tenggara Timur Menurut Kategori Pelanggan 2012-2015 dan 2017-2018 (ribu m3). BPS Provinsi NTT. Retrieved September 1, 2023, from <https://www.google.com/url?q=https://ntt.bps.go.id/statictable/2020/01/27/748/volume-air-yang-disalurkan-dan-yang-bocor-pada-perusahaan-air-bersih-menurut-kategori-pelanggan-2012-2015-dan-2017-2018-ribu-m3-.html&sa=D&source=docs&ust=1693580255028738&usg=AO>
- Badan Pusat Statistik. (2022, October 2). Statistik Air Bersih Provinsi Jawa Tengah, 2017. Retrieved September 1, 2023, from <https://www.google.com/url?q=https://jateng.bps.go.id/backend/images/Statistik-Air-Bersih-Provinsi-Jawa-Tengah-2017-ind.jpg&sa=D&source=docs&ust=1693580480985283&usg=AOvVaw0O95Q2vrEeiqO8aBa-ZaS1>
- Hamdani, Y. (2023). Implementasi WSN Pada Kandang Ayam Menggunakan Topologi Tree dan Protokol Komunikasi Data ESP-Now. Diakses pada tanggal 1 September 2023.
- Hübschmann, I. (2020, August 10). *ESP8266 for IoT: A Complete Guide*. Nabto. Retrieved September 11, 2023, from <https://www.nabto.com/esp8266-for-iot-complete-guide/>
- Jain, S. (2023, March 17). *Wireless Sensor Network (WSN)*. GeeksforGeeks. Retrieved September 11, 2023, from <https://www.geeksforgeeks.org/wireless-sensor-network-wsn/>
- Jost, D. (2019, October 15). *What is a Flow Sensor?* Fierce Electronics. Retrieved September 11, 2023, from <https://www.fierceelectronics.com/sensors/what-a-flow-sensor>
- Kementerian Pekerjaan Umum dan Rumah Tangga. (2007, March 6). PEMAKAIAN AIR RUMAH TANGGA PERKOTAAN 144 LITER PERHARI. Retrieved September 1, 2023, from <https://pu.go.id/berita/pemakaian-air-rumah-tangga-perkotaan-144-liter-perhari#:~:text=Masih%20berdasarkan%20survei%20tersebut%2C%20menurut,70%20liter%2Forang%2Fhari>
- Mongin, L. S., Kurniawan, W., & Ichsan, M. H. H. (2018). Sistem Monitoring Kadar Gas Berbahaya Pada Lokasi Parkiran Bawah Tanah Menggunakan Protokol MQTT. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 3(1), 68–74. Diambil dari <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/4038>. Diakses pada tanggal 1 September 2023.

- PT. Ferindo Energi Instrumen. (2019, May 3). *Apa fungsi flow meter sensor? Bagaimana sih cara operasinya? | Artikel | Ferindo Energi Instrumen*. Flow Meter Tokico. Retrieved September 11, 2023, from <https://www.feitech.id/Artikel/Apa-fungsi-flow-meter-sensor-Bagaimana-sih-cara-operasinya.html>
- Senouci, M. R., & Mellouk, A. (2016). *Deploying Wireless Sensor Network*. Wireless Sensor Network. Retrieved September 11, 2023, from <https://www.sciencedirect.com/topics/engineering/wireless-sensor-network>
- Setiawan, R. (2021, September 8). *Memahami Apa Itu Internet of Things*. Dicoding. Retrieved September 11, 2023, from <https://www.dicoding.com/blog/apa-itu-internet-of-things/>
- Zainullah, M. A., Syauqy, D., & Ichsan, M. H. H. (2018). Sistem Monitoring Kondisi Aliran Pada Gorong-gorong Saluran Air Menggunakan Metode Fuzzy. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 3(2), 1435–1443. Diambil dari <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/4401>. Diakses pada tanggal 1 September 2023.

LAMPIRAN A KODE SUMBER PROGRAM

A.1 Kode Program Node Sensor

No.	Kode Program
1	#include <ESP8266WiFi.h>
2	#include <espnw.h>
3	
4	// MAC address of the master node 68:C6:3A:F2:EB:CF
5	uint8_t masterNodeMAC[] = {0x68, 0xC6, 0x3A, 0xF2, 0xEB, 0xCF}; // Change this MAC address
6	
7	const char *ssid = "HUAWEI-HeYW";
8	const char *pass = "12345678";
9	
10	#define FLOW_1 D4
11	
12	float flowRate;
13	long currentMillis = 0;
14	long previousMillis = 0;
15	int interval = 1000;
16	float calibrationFactor = 4.5;
17	volatile byte pulseCount;
18	byte pulse1Sec = 0;
19	unsigned long flowMilliLitres;
20	unsigned int totalMilliLitres;
21	float flowLitres;
22	float totalLitres;
23	
24	int wifiStatus;
25	WiFiClient client;
26	
27	void IRAM_ATTR pulseCounter() {
28	pulseCount++;
29	}
30	
31	// Data structure to hold flow meter data
32	struct struct_message {
33	int nodeId = 1;
34	float HASIL_BACA_SENSOR;
35	};
36	struct_message flowMeterData;
37	
38	// Callback function when data is sent
39	void onDataSent(uint8_t *mac_addr, uint8_t sendStatus) {
40	Serial.print("Last Packet Send Status: ");
41	if (sendStatus == 0) {
42	Serial.println("Delivery Success!");
43	} else {
44	Serial.println("Delivery Fail!");
45	}
46	}
47	
48	void setup() {


```

49   Serial.begin(115200);
50
51   WiFi.begin(ssid,pass);
52   while (WiFi.status() != WL_CONNECTED) {
53       delay(500);
54       Serial.print(".");
55   }
56
57   pinMode(FLOW_1, INPUT_PULLUP);
58   pulseCount = 0;
59   flowRate = 0.0;
60   flowMilliLitres = 0;
61   totalMilliLitres = 0;
62   previousMillis = 0;
63
64   attachInterrupt(digitalPinToInterrupt(FLOW_1),
65   pulseCounter, FALLING);
66
67   // Initialize ESP-NOW
68   if (esp_now_init() != 0) {
69       Serial.println("ESP-NOW initialization failed");
70       return;
71   }
72
73   // Set the callback function to handle sent data
74   esp_now_set_self_role(ESP_NOW_ROLE_CONTROLLER);
75   esp_now_register_send_cb(onDataSent);
76
77   // Add the receiver MAC address
78   esp_now_add_peer(masterNodeMAC, ESP_NOW_ROLE_SLAVE, 1,
79   NULL, 0);
80   }
81
82   void loop() {
83       float HASIL_BACA_SENSOR = readFlowMeter();
84
85       // Update flow meter data
86       flowMeterData.HASIL_BACA_SENSOR = flowRate;
87
88       // Send data via ESP-NOW
89       esp_now_send(masterNodeMAC, (uint8_t *)&flowMeterData,
90   sizeof(flowMeterData));
91
92       Serial.print("Flow Rate sent: ");
93       Serial.println(HASIL_BACA_SENSOR);
94
95       delay(5000); // Delay for 5 seconds before sending the
96   next data
97   }
98
99   // Read data from the flow meter
100  float readFlowMeter() {
101      currentMillis = millis();
102      if (currentMillis - previousMillis > interval){
103          pulse1Sec = pulseCount;
104          pulseCount = 0;

```

105	
106	flowRate = ((1000.0 / (millis() - previousMillis)) * pulse1Sec) / calibrationFactor;
107	previousMillis = millis();
108	
109	
110	flowMilliLitres = (flowRate / 60) * 1000;
111	flowLitres = (flowRate / 60);
112	
113	totalMilliLitres += flowMilliLitres;
114	totalLitres += flowLitres;
115	
116	Serial.print("Sensor 1\n");
117	Serial.print("Flow rate: ");
118	Serial.print(float(flowRate));
119	Serial.print("L/min");
120	Serial.print("\t");
121	}
122	return 0.0;
123	}

A.2 Kode Program Node Master

No.	Kode Program
1	#include <ESP8266WiFi.h>
2	#include <espnow.h>
3	#define BLYNK_TEMPLATE_ID "TMPL64yyM66vT"
4	#define BLYNK_TEMPLATE_NAME "NodeMCUWaterflow"
5	
6	#include <BlynkSimpleEsp8266.h>
7	
8	uint8_t node1MAC[] = {0x48, 0x55, 0x19, 0xE4, 0x81, 0xC7};
9	uint8_t node2MAC[] = {0x48, 0x55, 0x19, 0xe4, 0xfe, 0x05};
10	uint8_t node3MAC[] = {0xa4, 0xcf, 0x12, 0xfd, 0x9e, 0xae};
11	
12	char auth [] = "yGKr7ZQuXZZD2Ugloht3_x4GzF-xZTwa";
13	const char *ssid = "HUAWEI-HeYW";
14	const char *pass = "12345678";
15	
16	#define FLOW_1 D4
17	
18	float flowRate;
19	long currentMillis = 0;
20	long previousMillis = 0;
21	int interval = 1000;
22	float calibrationFactor = 7.5 ;
23	volatile byte pulseCount;
24	byte pulse1Sec = 0;
25	unsigned long flowMilliLitres;
26	unsigned int totalMilliLitres;
27	float flowLitres;
28	float totalLitres;
29	

```

30 int wifiStatus;
31 WiFiClient client;
32
33 void IRAM_ATTR pulseCounter() {
34     pulseCount++;
35 }
36
37 BlynkTimer timer;
38
39 struct struct_message {
40     int nodeId;
41     float HASIL_BACA_SENSOR;
42 };
43
44 struct_message flowMeterData;
45 struct_message data1;
46 struct_message data2;
47 struct_message data3;
48
49 struct_message dataStruct[3] = {data1, data2, data3};
50
51 float valueData1;
52 float valueData2;
53 float valueData3;
54 float sensorMaster;
55
56 // Callback function when data is received
57 void onDataReceived(uint8_t *mac_addr, uint8_t *data,
58 uint8_t dataLen) {
59
60     memcpy(&flowMeterData, data, sizeof(flowMeterData));
61     Serial.printf("Node Sensor %u: ", flowMeterData.nodeId,
62 dataLen);
63     // Update data yang diterima
64     dataStruct[flowMeterData.nodeId-1].HASIL_BACA_SENSOR =
65 flowMeterData.HASIL_BACA_SENSOR;
66     Serial.printf("%f L/m\n",
67 dataStruct[flowMeterData.nodeId-1].HASIL_BACA_SENSOR);
68     Serial.println();
69 }
70
71 void setup() {
72     Serial.begin(115200);
73
74     WiFi.begin(ssid,pass);
75     while (WiFi.status() != WL_CONNECTED) {
76         delay(500);
77         Serial.print("Connecting to Wi-Fi\n");
78     }
79
80     pinMode(FLOW_1, INPUT_PULLUP);
81     pulseCount = 0;
82     flowRate = 0.0;
83     flowMilliLitres = 0;
84     totalMilliLitres = 0;
85     previousMillis = 0;

```

```

86
87     attachInterrupt(digitalPinToInterrupt(FLOW_1),
88 pulseCounter, FALLING);
89
90     // Initialize ESP-NOW
91     if (esp_now_init() != 0) {
92         Serial.println("ESP-NOW initialization failed");
93         return;
94     }
95     // Register the callback function to handle received data
96     esp_now_register_recv_cb(onDataReceived);
97
98     // Add this device's MAC address
99     esp_now_add_peer(node1MAC, ESP_NOW_ROLE_CONTROLLER, 1,
100 NULL, 0);
101     esp_now_add_peer(node2MAC, ESP_NOW_ROLE_CONTROLLER, 1,
102 NULL, 0);
103     esp_now_add_peer(node3MAC, ESP_NOW_ROLE_CONTROLLER, 1,
104 NULL, 0);
105
106     Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
107     timer.setInterval(5000L, sendSensor);
108 }
109
110 float readFlowMeter() {
111     currentMillis = millis();
112     if (currentMillis - previousMillis > interval){
113         pulse1Sec = pulseCount;
114         pulseCount = 0;
115
116         flowRate = ((1000.0 / (millis() - previousMillis)) *
117 pulse1Sec) / calibrationFactor;
118         previousMillis = millis();
119
120         flowMilliLitres = (flowRate / 60) * 1000;
121         flowLitres = (flowRate / 60);
122
123         // Add the millilitres passed in this second to the
124 cumulative total
125         totalMilliLitres += flowMilliLitres;
126         totalLitres += flowLitres;
127
128         // Print the flow rate for this second in litres /
129 minute
130         Serial.print("Sensor Master\n");
131         Serial.print("Flow rate: ");
132         Serial.print(float(flowRate)); // Print the integer
133 part of the variable
134         Serial.print("L/min \n");
135         Serial.println();
136     }
137     return flowRate; // Placeholder value, replace with
138 actual reading
139 }
140
141 void loop() {

```

```

142     float flowRate = readFlowMeter();
143     sensorMaster = flowRate;
144
145     Blynk.run();
146     timer.run();
147 }
148
149 void sendSensor() {
150     float valueData1 = dataStruct[0].HASIL_BACA_SENSOR;
151     float valueData2 = dataStruct[1].HASIL_BACA_SENSOR;
152     float valueData3 = dataStruct[2].HASIL_BACA_SENSOR;
153     float toleransi = 0.1;
154
155     Blynk.virtualWrite(V0, valueData1);
156     Blynk.virtualWrite(V1, valueData2);
157     Blynk.virtualWrite(V2, valueData3);
158     Blynk.virtualWrite(V3, sensorMaster);
159
160     if (sensorMaster - toleransi > valueData1 + valueData2 +
161 valueData3) {
162         Blynk.logEvent("notifications", "Sistem Mendeteksi
163 Kebocoran");
164     } else {
165         //Kondisi pipa aman
166     }
167 }
168

```

LAMPIRAN B

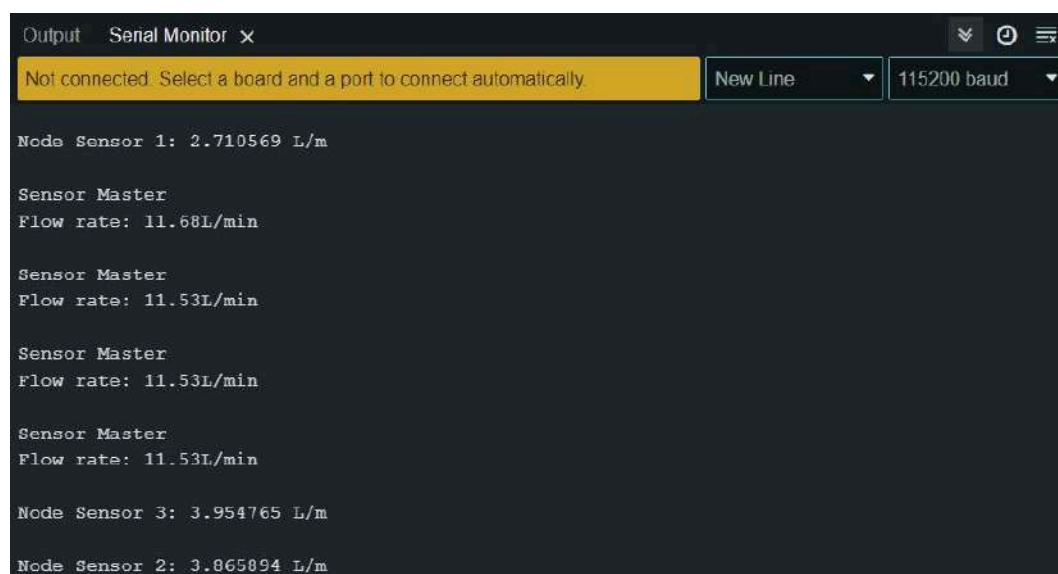
B.1 Pengujian Sensor YF-S201

```
Sensor 1
Flow rate: 9.20L/min    Flow Rate sent: 0.00
Last Packet Send Status: Delivery Fail!
Sensor 1
Flow rate: 9.86L/min    Flow Rate sent: 0.00
Last Packet Send Status: Delivery Fail!
```

```
Sensor 2
Flow rate: 3.24L/min    Flow Rate sent: 0.00
Last Packet Send Status: Delivery Fail!
```

```
Sensor 3
Flow rate: 9.20L/min    Flow Rate sent: 0.00
Last Packet Send Status: Delivery Fail!
```

B.2 Pengujian Komunikasi Data ESP-NOW



The screenshot shows a Serial Monitor window with a title bar 'Output Serial Monitor x'. Below the title bar is a yellow status bar that reads 'Not connected. Select a board and a port to connect automatically.' To the right of this bar are two dropdown menus: 'New Line' and '115200 baud'. The main area of the window displays the following text:

```
Node Sensor 1: 2.710569 L/m

Sensor Master
Flow rate: 11.68L/min

Sensor Master
Flow rate: 11.53L/min

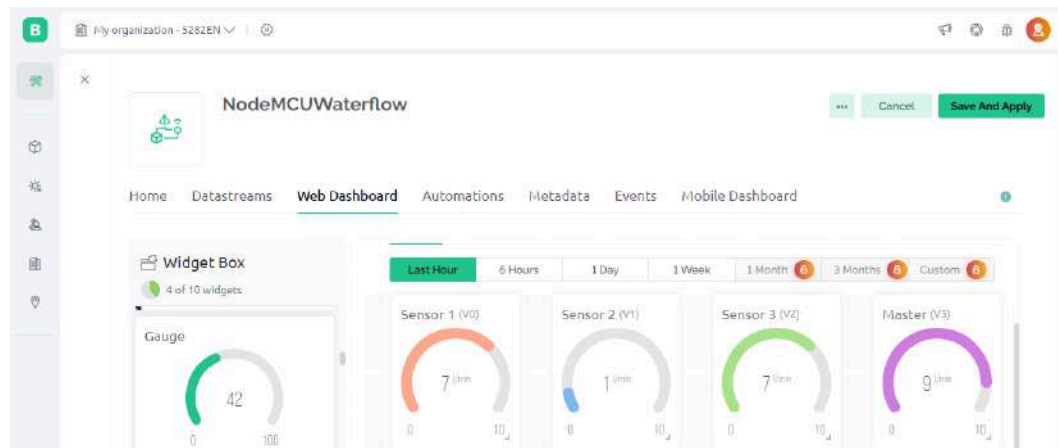
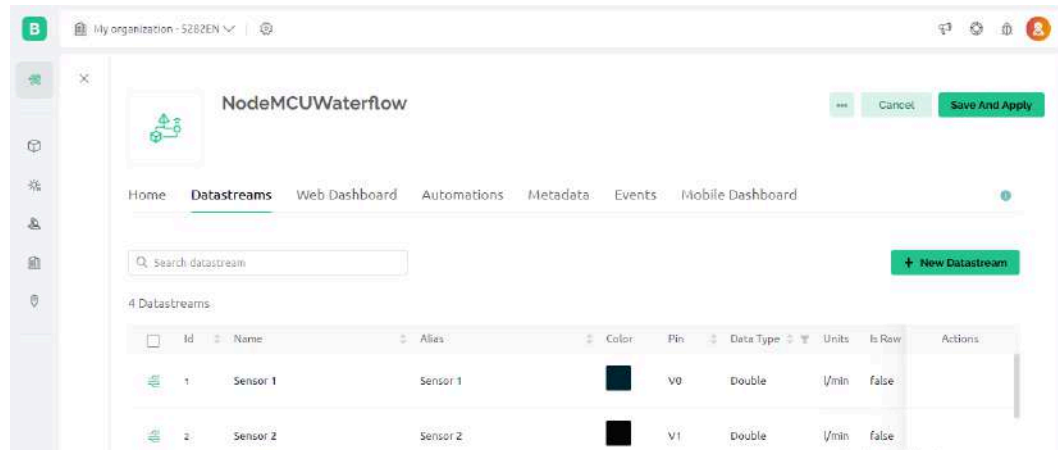
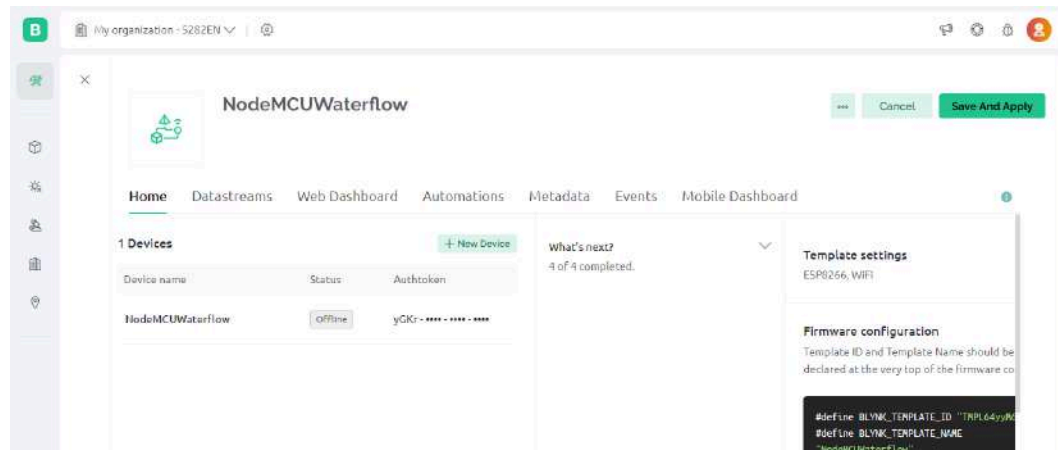
Sensor Master
Flow rate: 11.53L/min

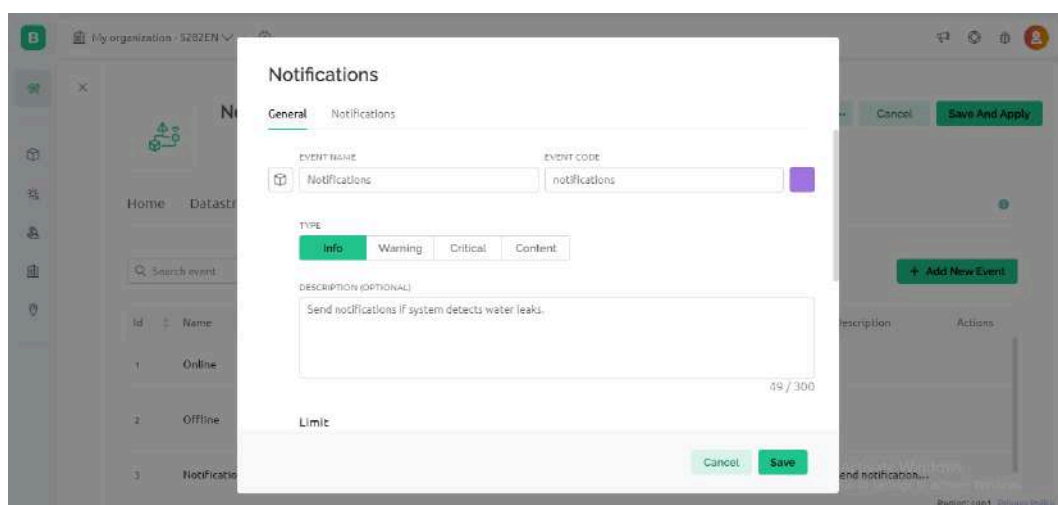
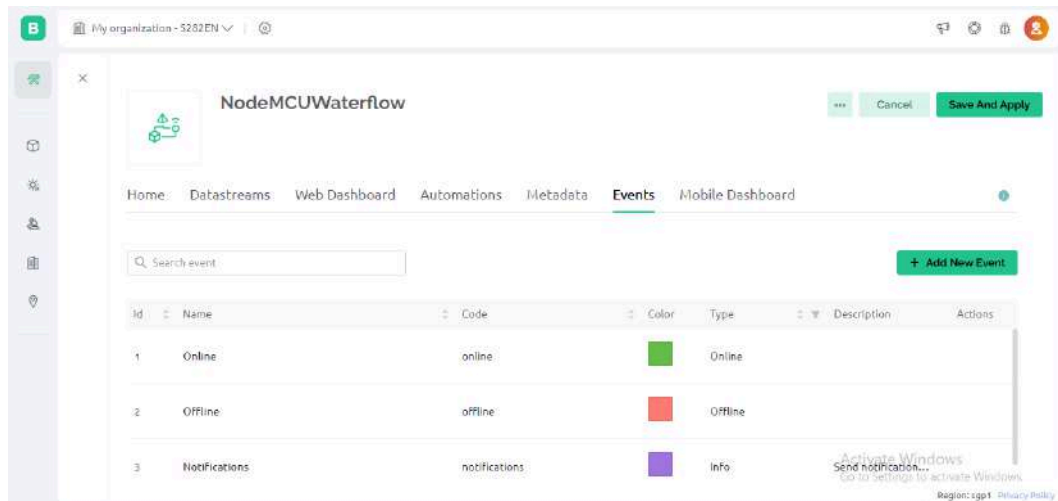
Sensor Master
Flow rate: 11.53L/min

Node Sensor 3: 3.954765 L/m

Node Sensor 2: 3.865894 L/m
```

B.3 Konfigurasi Blynk





B.4 Dokumentasi Pengujian

