

# Rancang Bangun *Prototype* Sistem *Monitoring* Tekanan Air Pada *Header* Berbasis *Internet of Things*

Nofi Cahyono<sup>1)</sup>

Teknologi Informasi, Politeknik Gajah Tunggal  
[nofi@poltek-gt.ac.id](mailto:nofi@poltek-gt.ac.id)

Henry Prasetyo<sup>3)</sup>

Teknik Elektronika, Politeknik Gajah Tunggal  
[henry@poltek-gt.ac.id](mailto:henry@poltek-gt.ac.id)

Faris Fadhil Rosyid<sup>2)</sup>

Teknik Elektronika, Politeknik Gajah Tunggal  
[farisfadhilr@gmail.com](mailto:farisfadhilr@gmail.com)

Maulana Yazid Ramadhan<sup>4)</sup>

Teknologi Informasi, Politeknik Gajah Tunggal  
[maulanayzid@gmail.com](mailto:maulanayzid@gmail.com)

## ABSTRAK

Mesin pompa air digunakan untuk menghasilkan air bertekanan dalam proses produksi di PT. BIN dengan tekanan standar 0,45 MPa sampai 0,5 MPa. Rata-rata digunakan 3 dari 6 mesin pompa untuk menghasilkan tekanan tersebut. Untuk memastikan tekanan air sesuai dengan standar, dilakukan *monitoring* dan pencatatan secara manual setiap satu jam sekali. Pada saat melakukan *monitoring* tekanan air, operator harus berjalan  $\pm 75$ m dari *office power house* sehingga mengakibatkan lamanya proses. Kemudian, ketika terjadi penurunan tekanan di bawah 0,45 MPa, operator harus menyalakan mesin cadangan secara manual. Oleh karena itu, akan dilakukan rancang bangun sistem *monitoring* tekanan air untuk mempermudah operator dalam melakukan *monitoring* dan pencatatan. Penelitian dilakukan dengan menggunakan sensor *pressure transducer* untuk mengukur tekanan air dan sensor *water flow* untuk mengukur debit air. Data dari kedua sensor tersebut akan diproses menggunakan *microcontroller* Arduino Nano, yang kemudian akan dikirim ke NodeMCU ESP8266 untuk diintegrasikan dengan platform Blynk untuk melakukan *monitoring*, dan Firebase sebagai tempat transit data yang nantinya data tersebut akan disimpan dalam Google Spreadsheets. Hasil menunjukkan akurasi pembacaan sensor *pressure transducer* sebesar 99,27% dan sensor *water flow* YF-S201 sebesar 97,31%.

Kata Kunci: *Monitoring, Pressure Transducer, Water Flow, NodeMCU ESP8266, IoT*

## ABSTRACT

A water pump machine is used to generate pressurized water in the production process at PT. BIN with a standard pressure of 0.45 MPa to 0.5 MPa. On average, 3 out of 6 pump machines are used to achieve this pressure. To ensure the water pressure meets the standard, manual monitoring and recording are conducted every hour. During water pressure monitoring, the operator has to walk approximately 75 meters from the powerhouse office, resulting in a lengthy process. Furthermore, when the pressure drops below 0.45 MPa, the operator has to manually activate the backup machine. Therefore, a water pressure monitoring system will be designed and built to facilitate the operator in monitoring and recording. The study is conducted using a pressure transducer sensor to measure water pressure and a water flow sensor to measure water flow rate. Data from these sensors will be processed using an Arduino Nano microcontroller, then sent to a NodeMCU ESP8266 to be integrated with the Blynk platform for monitoring, and Firebase as a data transit point, where the data will eventually be stored in Google Spreadsheets. The results show an accuracy of 99.27% for the pressure transducer sensor readings and 97.31% for the YF-S201 water flow sensor readings.

Keywords: *Monitoring, Pressure Transducer, Water Flow, NodeMCU ESP8266, IoT*

## I. PENDAHULUAN

PT. BIN merupakan salah satu perusahaan di Indonesia yang memproduksi komponen otomotif dan industri. Perusahaan ini menghasilkan tiga jenis produk utama, yaitu *v-belt/fanbelt*, *conveyor-belt*, dan *weight roller*. Dalam pembuatan produk, terdapat beberapa departemen yang bertugas untuk melaksanakan dan mendukung proses produksi. salah satu dari departemen tersebut adalah Departemen *Maintenance*.

Departemen *Maintenance* merupakan departemen yang bertanggung jawab untuk memelihara dan memperbaiki peralatan, fasilitas, dan mesin agar tetap berfungsi dengan baik. Departemen *Maintenance* terdiri dari dua bagian, yaitu *power house* dan *engineering*. *Power house* bertugas menyediakan energi, sementara *engineering* bertanggung jawab merawat, menjaga, dan memperbaiki mesin-mesin produksi.

*Power house* adalah bagian penting dalam proses produksi karena tempat dimana energi dipersiapkan. Terdapat mesin-mesin seperti *boiler* yang menghasilkan uap panas, kompresor yang menghasilkan udara bertekanan, dan pompa air yang menghasilkan air bertekanan. Selain itu, terdapat rumah panel listrik sebagai pusat kontrol distribusi listrik di PT. BIN. Untuk memastikan kelancaran produksi, pemantauan dan pencatatan pasokan energi dilakukan setiap jam dan hasilnya dicatat dalam buku jurnal. Terdapat tiga buku jurnal, yaitu jurnal 1 untuk panel induk, jurnal 2 untuk kompresor dan tekanan air, dan jurnal 3 untuk mesin *Boiler*.

Secara umum, pemantauan dan pencatatan yang dilakukan oleh operator masih dilakukan secara manual, yang mengakibatkan lamanya proses *monitoring*. Proses yang memakan waktu terbanyak terjadi pada pengecekan tekanan air pada jurnal 2, yang memakan waktu 215 detik, karena operator harus menempuh jarak  $\pm 75\text{m}$  dari *office power house* ke *pressure gauge* yang berada di *header* dan masih perlu membaca *pressure gauge* yang berjenis analog. Jarak yang cukup jauh dan jenis sensor yang masih analog ini menjadi faktor utama yang menyebabkan lamanya pengecekan pada jurnal 2. Untuk menganalisis penyebab terjadinya masalah terkait lamanya durasi pemantauan dan pencatatan tekanan air, maka dibuat analisis *fishbone* seperti pada Gambar 1.



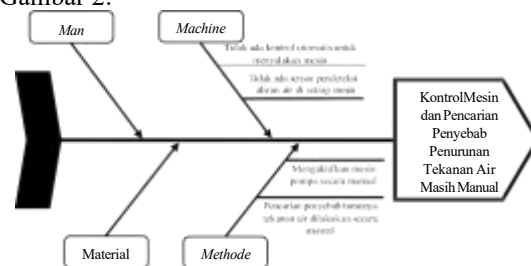
Gambar 1. Fishbone 1

Selain masalah di atas, terdapat juga masalah pada area mesin pompa air. Pada bagian *power house* terdapat enam mesin pompa yang digunakan untuk menghasilkan air bertekanan ke area produksi. Setiap hari, rata-rata tiga mesin pompa utama digunakan untuk mencapai tekanan 0,45 MPa sampai 0,5 MPa, sementara tiga mesin lainnya sebagai cadangan.

Berdasarkan observasi dari November 2023 hingga Januari 2024, terjadi delapan kali penurunan tekanan dengan rata-rata waktu 10 menit untuk menemukan penyebabnya. Penurunan tekanan air pada *header* dapat disebabkan oleh beberapa faktor, seperti meningkatnya penggunaan air pada proses produksi atau *strainer* pada mesin pompa yang kotor. Penurunan tekanan air dapat diketahui melalui lampu indikator (Andon) yang menyala ketika tekanan air kurang dari 0,45 MPa.

Jika Andon menyala, operator akan mengaktifkan mesin pompa cadangan secara manual karena belum ada sistem kontrol otomatis untuk menghidupkan mesin cadangan saat tekanan turun. Operator kemudian memeriksa mesin pompa untuk mencari penyebab turunnya tekanan air dengan mengaktifkan dan menonaktifkan mesin pompa utama secara bergantian hingga penyebabnya diketahui.

Jika penurunan tekanan disebabkan oleh meningkatnya penggunaan air pada proses produksi, pompa cadangan dibiarkan aktif agar tekanan tetap stabil. Namun, jika penurunan tekanan disebabkan oleh masalah pada mesin, maka perbaikan akan dilakukan pada mesin tersebut. Untuk menganalisis penyebab terjadinya masalah terkait belum adanya sistem kontrol otomatis untuk menyalakan mesin pompa dan pencarian penyebab penurunan tekanan air, maka dibuat analisis *fishbone* seperti pada Gambar 2.



Gambar 2. Fishbone 2

Berdasarkan analisis *fishbone* Gambar 1 dan Gambar 2, penelitian ini berfokus pada perbaikan pada titik mesin pada poin *machine* dengan merancang bangun *prototype* sistem *monitoring* tekanan air pada *header* berbasis IoT serta menambahkan sensor pada mesin pompa air untuk mendeteksi debit air. Dalam proses rancang bangun digunakan sensor *pressure transducer* untuk membaca tekanan air pada *header* dan sensor *water flow* untuk mengukur debit air pada masing-masing

mesin pompa air. Nilai dari kedua sensor tersebut akan diproses menggunakan mikrokontroler Arduino Nano, yang kemudian akan dikirim ke NodeMCU ESP8266 melalui komunikasi serial. Selanjutnya, NodeMCU ESP8266 diintegrasikan dengan sistem IoT menggunakan platform Blynk sebagai antarmuka untuk melakukan *monitoring*, dan Firebase sebagai tempat transit data yang nantinya data tersebut akan disimpan dalam Google Spreadsheets.

Berikut merupakan kajian pustaka yang digunakan pada penelitian:

Tabel 1. Kajian Sebelumnya

Sumber	Hasil Kajian
[1]	NodeMCU ESP8266 digunakan untuk mengintegrasikan data sensor yang diperoleh dari Arduino Uno ke dalam sistem IoT.
[2]	Dalam penelitian ini, Firebase dimanfaatkan sebagai basis data <i>cloud realtime</i> untuk menyimpan nilai dari <i>flex</i> sensor, dan nilai tersebut digunakan untuk menginisiasi pengoperasian <i>relay</i> .
[3]	Pada penelitian ini, digunakan <i>pressure</i> Sensor G1/4 untuk membaca tekanan <i>upstream</i> (sebelum meter air) dan <i>downstream</i> (setelah meter air).
[4]	Penelitian ini menggunakan sensor <i>water flow</i> YF-S201 sebagai pembaca laju aliran air.
[5]	Pada penelitian ini, Spreadsheets digunakan untuk menyimpan data penggunaan daya listrik.
[6]	Arduino Nano digunakan sebagai mikrokontroler untuk membaca data dari sensor ultrasonik dan mengontrol gerakan <i>servo</i> .

Arduino Nano adalah salah satu board mikrokontroler yang berukuran kecil, lengkap, dan mendukung penggunaan *breadboard*. Arduino Nano didesain dengan basis mikrokontroler ATmega328 (untuk versi 3.x) atau Atmega 16 (untuk versi 2.x). Meskipun fungsinya mirip dengan Arduino Duemilanove, Arduino Nano hadir dalam paket yang berbeda. Arduino Nano dapat diaktifkan melalui koneksi USB Mini-B, atau melalui catu daya eksternal dengan tegangan antara 6-20 Volt yang dihubungkan melalui pin 30 atau pin VIN, atau menggunakan catu daya eksternal dengan tegangan regulasi 5 volt melalui pin 27 atau pin 5V. Sumber daya akan dipilih secara otomatis dari sumber tegangan yang lebih tinggi [6].

*Internet of Things* (IoT) adalah konsep perangkat yang mampu mentransfer data tanpa perlu campur tangan manusia secara langsung, tetapi

menggunakan internet sebagai medianya. Dengan IoT, data bisa dipindahkan melalui jaringan tanpa perlu interaksi dua arah antara manusia, seperti dari sumber ke tujuan atau interaksi manusia ke komputer. IoT adalah perkembangan teknologi yang menjanjikan untuk meningkatkan kehidupan dengan menggunakan sensor-sensor pintar dan objek yang terhubung ke internet, serta beroperasi secara terkoordinasi melalui jaringan tersebut [7].

NodeMCU ESP8266 adalah modul WIFI yang digunakan untuk mengirim atau menerima data melalui jaringan internet. Dalam era Industri 4.0, modul ini sangat berguna untuk menghubungkan semua sistem ke internet atau *Internet of Things* (IoT) karena harganya yang terjangkau dan kualitas yang memadai. NodeMCU ESP8266 dilengkapi dengan IC CH340, sehingga dapat langsung diprogram dari komputer menggunakan port serial. Modul ini mendukung bahasa pemrograman Python dan bahasa pemrograman yang digunakan oleh Arduino IDE (*Integrated Development Environment*). *Interface* yang digunakan adalah serial pada pin RX dan TX untuk berkomunikasi dengan PC atau perangkat lainnya [8].

*Pressure transducer* adalah jenis alat yang mengubah tekanan menjadi format digital. Dalam konteks industri, peran *transducer* sangat penting dalam mengontrol dan memantau sistem tanpa harus berinteraksi langsung dengan perangkat pengukur. *Output* sinyal dari *pressure transducer* berkisar antara 4 hingga 20 mA agar dapat diolah oleh mikrokontroler [9].

Sensor aliran YF-S201 adalah alat yang dipakai untuk mengukur laju aliran fluida. Cara kerja sensor aliran YF-S201 yaitu saat air mengalir melalui rotor kincir, rotor tersebut akan berputar. Kecepatan rotasi ini bergantung pada laju aliran air. Sensor akan menghasilkan sinyal pulsa sebagai respons terhadap jumlah aliran air yang melewati [9].

Blynk adalah platform aplikasi untuk sistem operasi *mobile* seperti iOS dan Android yang dirancang untuk mengontrol modul Arduino, Raspberry Pi, ESP8266, WEMOS D1, dan modul serupa melalui internet. Aplikasi ini memungkinkan pengguna untuk membuat antarmuka grafis untuk proyek mereka dengan metode *drag and drop widget*. Blynk tidak terbatas pada satu papan atau modul tertentu, sehingga pengguna dapat mengontrol berbagai perangkat dari jarak jauh dengan koneksi internet yang stabil, sesuai dengan konsep *Internet of Things* (IoT) [10].

Database Firebase secara otomatis menyinkronkan aplikasi klien yang terhubung. Aplikasi yang mendukung berbagai platform, seperti Android, iOS, dan JavaScript, akan secara otomatis menerima pembaruan data terbaru ketika terhubung ke server Firebase. Firebase *realtime database* merupakan platform database yang digunakan oleh

aplikasi *realtime*. Ketika terjadi perubahan data, aplikasi yang terhubung ke database akan secara otomatis diperbarui di berbagai perangkat, termasuk situs web dan perangkat seluler [11].

Google Spreadsheets adalah program atau aplikasi web yang dikembangkan oleh Google untuk membuat tabel, melakukan perhitungan sederhana, dan memanipulasi data. Yang membedakannya dari perangkat lunak serupa adalah kemudahan kolaborasi pengguna. Program ini berbasis *cloud*, yang berarti bergantung pada koneksi internet untuk mendukung komunikasi dan kolaborasi antar pengguna, membuatnya lebih mudah digunakan. Keunggulan utamanya adalah dapat digunakan secara gratis. Google Spreadsheets juga sering digunakan dalam beberapa pekerjaan pemrograman sebagai basis data karena keunggulannya [12].

Regresi linear sederhana adalah teknik statistik yang digunakan untuk menentukan hubungan antara dua variabel atau lebih, atau untuk mengukur dampak variabel independen (X) terhadap variabel dependen (Y). Variabel *predictor*, sering dilambangkan sebagai X atau *predictor*, dan variabel respons, dilambangkan sebagai Y atau respons. Ketika terdapat hubungan antara X dan Y, nilai X yang diketahui dapat digunakan untuk memperkirakan atau meramalkan nilai Y menggunakan rumus (1). Regresi linear sederhana (*Simple Linear Regression*) atau SLR merupakan metode statistik yang digunakan untuk prediksi atau peramalan dalam produksi, terutama terkait karakteristik kualitas atau kuantitas. Model persamaan regresi linear sederhana dapat dilihat pada rumus (1), dengan konstanta dan variabel yang didefinisikan pada rumus (2) dan (3) [13].

$$= + \quad (1)$$

$$= \frac{(\sum)(\sum^2) - (\sum)(\sum)}{(\sum^2) - (\sum)^2} \quad (2)$$

$$= \frac{(\sum) - (\sum)(\sum)}{(\sum^2) - (\sum)^2} \quad (3)$$

Debit aliran atau laju aliran (*flow rate*) adalah jumlah volume air yang mengalir melalui pipa atau penampung air dalam satu satuan waktu. Satuan laju aliran dapat dinyatakan dalam liter / menit, meter kubik / menit, atau liter / detik. Persamaan Debit dinyatakan dalam persamaan (4) [14].

$$Debit = V/t \quad (4)$$

Keterangan:

Q = Laju aliran atau debit (liter / menit)

V = Volume fluida (liter)

t = Waktu (menit)

Penentuan penggunaan sensor dapat ditentukan dengan menghitung persentase nilai *error* dari perbandingan pembacaan sensor dengan instrumen pengujian. Dalam proses menentukan persentase nilai *error*, digunakan rumus (5) [15].

$$\% = \frac{|\sum|}{\sum} \times 100\% \quad (5)$$

Keterangan:

Y = nilai sebenarnya (Master)

X = nilai yang terukur (Sensor)

MAPE adalah pengujian yang mengevaluasi hasil dari model prediksi yang telah dibuat dengan menggunakan data masa lampau. Untuk menilai tingkat keakuratan, dapat digunakan metode MAPE dengan menghitung selisih antara data aktual dan data prediksi. Semakin kecil nilai MAPE, semakin tinggi tingkat keakuratannya, yang mengindikasikan bahwa metode yang digunakan mampu memprediksi dengan baik. Perhitungan MAPE dapat menggunakan persamaan (6) [16].

$$AE = \sum |x| \times 100\% \quad (6)$$

Keterangan:

Y = nilai sebenarnya (Master)

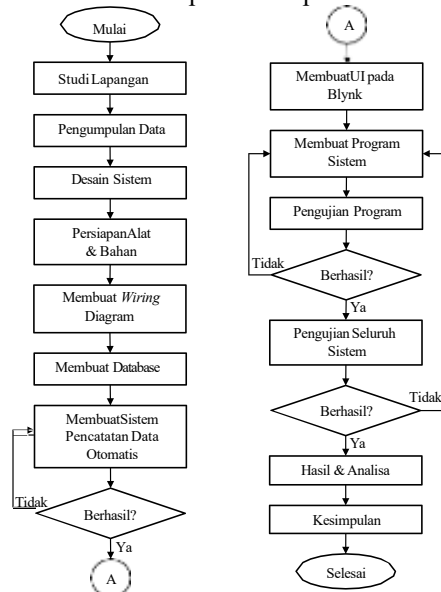
X = nilai yang terukur (Sensor)

Dalam menentukan akurasi suatu alat, digunakan rumus matematis khusus yang telah dirancang. Penghitungan akurasi ini tidak hanya bertujuan untuk memastikan apakah alat tersebut memberikan hasil yang mendekati nilai sebenarnya, tetapi juga untuk menilai seberapa dekat hasil yang diperoleh dengan nilai yang seharusnya. Rumus akurasi dihitung menggunakan persamaan (7) [17].  
Akurasi =  $100\% - (aa - aa Eo \%)$  (7)

## II. METODE PENELITIAN

### Alur Penelitian

Alur penelitian mencakup tahapan-tahapan yang dilaksanakan dalam penelitian ini. Alur penelitian tersebut dapat dilihat pada Gambar 3.

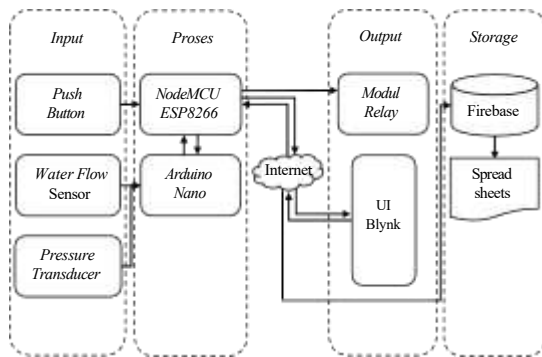


Gambar 3. Alur Penelitian

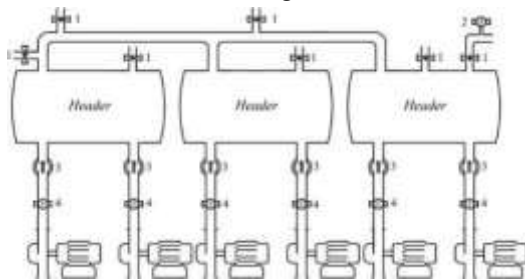
### Desain Sistem

Pada tahap ini, dibuat blok diagram dan perencanaan penempatan komponen yang akan digunakan untuk memantau tekanan dan kecepatan

aliran air. Berikut ini adalah blok diagram dan perencanaan penempatan komponen yang telah disusun:



Gambar 4. Diagram Sistem



Gambar 5. Rencana Penempatan komponen

Keterangan:

1. *Gate Valve* : Berfungsi untuk mengatur aliran fluida dalam sistem dengan cara membuka dan menutup aliran.
2. *Pressure Transducer / Pressure Gauge* : Berfungsi untuk memberikan informasi tekanan air pada pipa.
3. *Check Valve* : Berfungsi untuk mencegah aliran balik pada sistem dan memastikan bahwa air hanya mengalir ke satu arah.
4. *Water Flow Sensor* : Berfungsi untuk memberikan informasi debit air yang mengalir pada pipa.

### Desain UI Blynk

Pada tahap ini dilakukan pembuatan *User Interface* pada aplikasi Blynk. Sketsa tampilan dapat dilihat melalui 6.



Gambar 6. Desain UI Blynk

Gambar 6 merupakan rencana tampilan antarmuka yang berisi beberapa perangkat, seperti tombol untuk mengatur mode sistem dan

mengaktifkan mesin pompa secara manual. Selain itu, terdapat juga perangkat yang berfungsi untuk menampilkan mode yang sedang aktif, mesin yang sedang aktif, dan nilai sensor.

### Alat dan Bahan

Alat dan Bahan yang digunakan dalam pembuatan prototipe dapat dilihat melalui Tabel 2 dan Tabel 3.

Tabel 2. Alat

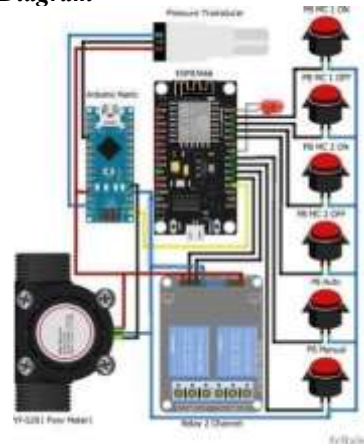
No.	Alat	Jumlah
1	Tang Potong	1 pcs
2	Obeng +	1 pcs
3	Laptop	1 pcs
4	Smartphone	1 pcs
5	Kunci Inggris	1 pcs
6	Pressure Gauge ¼" 1.0 MPa	1 pcs

Tabel 3. Bahan

No	Bahan	Jumlah
1	Kabel Jumper (M to M)	40 pcs
2	Project Board (170 point)	2 pcs
3	NodeMCU ESP8266	1 pcs
4	Arduino Nano	1 pcs
5	Kabel Mini USB	1 pcs
6	Kabel USB tipe B	1 pcs
7	Sensor Pressure Transducer	1 pcs
8	Sensor YF-S201	1 pcs
9	Modul Relay 2 Channel	1 pcs
10	Push button	6 pcs
11	Seal tape (12mm)	1 pcs
12	Ball Valve (Drat dalam ¼")	1 pcs
13	Fitting Elbow (Drat luar ¼")	1 pcs
14	Double Nepple (Drat luar ¼")	3 pcs
15	Nepple sok ring (¼"x ½")	4 pcs
16	Double sok (Drat dalam ¼")	2 pcs
17	Tee Female (Drat ¼")	1 pcs
18	Tee Female (Drat ½")	1 pcs
19	Nepple Slang (½"x12)	1 pcs
20	Klem Slang	2 pcs

## III. HASIL DAN PEMBAHASAN

### Wiring Diagram

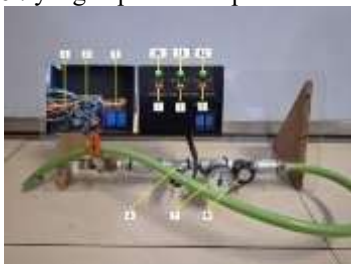


Gambar 7. Wiring Diagram

Gambar 7 menunjukkan hubungan komponen yang digunakan dalam penelitian, seperti Arduino Nano, NodeMCU ESP8266, modul *relay* 2-channel, sensor *pressure transducer*, sensor *water flow*, dan *push button*. Modul *relay* dan *push button* berfungsi sebagai pengendali *output* dalam sistem, sementara Arduino Nano dan NodeMCU ESP8266 berfungsi sebagai mikrokontroler yang mengolah data dari sensor *pressure transducer* dan *water flow* YF-S201.

### Gambaran Umum Alat

Dalam penelitian ini dilakukan pembuatan sistem kontrol dan *monitoring* tekanan dan debit air dalam bentuk *prototype*. Dalam pembuatan *prototype* digunakan komponen seperti Arduino Nano, NodeMCU ESP8266, modul *relay* 2-channel, sensor *pressure transducer*, sensor *water flow*, dan *push button* yang dapat dilihat pada Gambar 8.



Gambar 8. *Prototype*

Penjelasan Gambar 8 dapat dilihat melalui Tabel 4.

Tabel 4. Keterangan Gambaran *Prototype*

No.	Nama	Fungsi
1	NodeMCU ESP8266	Pengendali Sistem
2	Arduino Nano	Membaca nilai pada sensor
3	Modul <i>Relay</i> 2-Channel	Pengendali Mesin pompa
4	Sensor <i>Pressure Transducer</i>	Membaca Tekanan Air
5	<i>Pressure Gauge</i>	Pembandingan Sensor <i>Pressure Transducer</i>
6	Sensor <i>Water Flow</i> YF-S201	Membaca Debit Air
7	<i>Push button</i> Manual	Mengaktifkan Mode Manual
8	<i>Push button</i> MC 2 OFF	Menonaktifkan Mesin Pompa 2
9	<i>Push button</i> MC 1 OFF	Menonaktifkan Mesin Pompa 1
10	<i>Push button</i> Auto	Mengaktifkan Mode Otomatis
11	<i>Push button</i> MC 2 ON	Mengaktifkan Mesin Pompa 2
12	<i>Push button</i> MC 1 ON	Mengaktifkan Mesin Pompa 1

### Pembuatan Database

Dalam perancangan sistem *monitoring*, database digunakan untuk menyimpan nilai-nilai sensor yang dikirim oleh NodeMCU. Dalam perancangan yang dilakukan, database yang digunakan adalah Firebase. Firebase dipilih karena mudah diintegrasikan integrasi platform IoT seperti NodeMCU, serta menyediakan layanan *realtime database* yang dapat diakses dan diperbarui secara langsung. Berikut adalah hasil dari *realtime database* yang sudah dibuat pada Firebase, yang dapat dilihat melalui Gambar 9.



Gambar 9. Tampilan *Realtime Database*

### Pembuatan Sistem Pencatatan Data Otomatis

Dalam membuat sistem pencatatan data otomatis digunakan *Spreadsheets* sebagai tempat penyimpanan data sensor. Tampilan *Spreadsheet* dapat dilihat pada Gambar 10.



Gambar 10. Tampilan *Spreadsheets*

Gambar 10 merupakan tampilan dari *Spreadsheet* untuk menyimpan data dari sensor. Untuk melakukan pencatatan data otomatis harus dilakukan kustom fungsi *Spreadsheet* agar dapat mengambil data yang ada pada *realtime database* ke dalam *Spreadsheet*. Contoh dari kustom fungsi tersebut dapat dilihat pada Gambar 11.

```

1 function getAJData() {
2   var ss = SpreadsheetApp.getActiveSheet();
3   var sheet = ss.getActiveSheet();
4
5   var FirebaseURL = "https://firebasedemo.firebaseio.com/monitoring";
6   var data = FirebaseApp.getDatabase().getReference();
7
8   var data = data.getData();

```

Gambar 11. Kustom Fungsi *Spreadsheet*

### Tampilan File Pencatatan Data

Sistem pencatatan data otomatis mencatat tekanan dan debit air setiap jam ke Google *Spreadsheet* dan mengirimkan data harian melalui email. Dengan sistem ini, operator tidak perlu mencatat secara manual, dan dapat mempercepat proses *monitoring*. Tampilan sistem dan data harian yang dikirimkan melalui email dapat dilihat pada Gambar 12 dan Gambar 13.



Gambar 12. File Penyimpanan Data

Gambar 12 adalah tampilan dari lembar Spreadsheet yang digunakan untuk menyimpan data tekanan dan debit air per harinya. Lembar tersebut akan kosong secara otomatis ketika data tersebut sudah terkirim melalui email.

Gambar 13. Data yang Terkirim ke Email

Gambar 13 adalah tampilan dari file PDF yang merupakan salinan dari lembar penyimpanan harian yang sudah terkirim melalui email.

### Tampilan Aplikasi Blynk

Dalam memantau tekanan dan aliran air serta memberikan peringatan saat terjadi penurunan aliran, diperlukan antarmuka pengguna yang memudahkan *monitoring* melalui *smartphone* Android. Antarmuka ini dibuat dengan menggunakan aplikasi Blynk dan memanfaatkan beberapa perangkat dari *widget box*. Tampilan antarmuka yang sudah dibuat dapat dilihat pada Gambar 14.



Gambar 14. Tampilan Aplikasi Blynk

Pada Gambar 14 terdapat beberapa perangkat yang digunakan untuk melakukan *monitoring* dan kontrol. Selain itu, telah dilakukan pengaturan Virtual Pin sebagai alamat virtual untuk semua perangkat yang digunakan dalam tampilan Blynk. Setiap perangkat memiliki Virtual Pin masing-

masing yang memiliki fungsi yang berbeda. Detail konfigurasi perangkat dan fungsi Virtual Blynk dapat dilihat pada Tabel 5 dan Tabel 6.

Tabel 5. Konfigurasi Perangkat Tampilan Blynk

No.	Device	Virtual Pin	No.	Device	Virtual Pin
1	Labeled Value "Mesin On"	V0	7	Push Button "Mesin 2 On"	V4
2	Labeled Value "Mode On"	V1	8	Push Button "Mesin 2 Off"	V5
3	Value Display "Pressure"	V8	9	Push Button "Auto"	V6
4	Push Button "Mesin 1 Off"	V3	10	SuperChart "Flow"	V9
5	Value Display "Flow"	V9	11	Push Button "Manual"	V4
6	Push Button "Mesin 1 On"	V2	12	SuperChart "Pressure"	V8

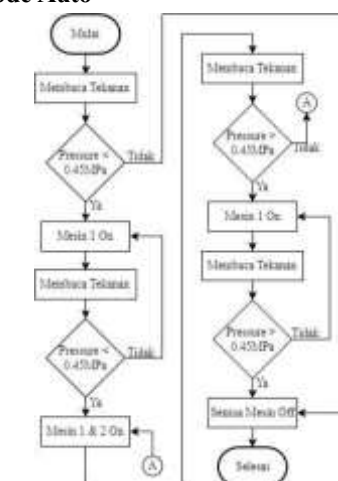
Tabel 6. Fungsi Virtual Pin Blynk

No.	Virtual Pin	Fungsi
1	V0	Menyimpan jumlah mesin yang aktif dan kemudian ditampilkan pada Labeled Value "Machine On"
2	V1	Menyimpan mode yang aktif dan kemudian ditampilkan pada Labeled Value "Mode On"
3	V2	Mengirim nilai ke NodeMCU untuk mengaktifkan mesin 1 ketika Push Button "MC1 ON" ditekan
4	V3	Mengirim nilai ke NodeMCU untuk menonaktifkan mesin 1 ketika Push Button "MC1 OFF" ditekan
5	V4	Mengirim nilai ke NodeMCU untuk mengaktifkan mesin 2 ketika Push Button "MC2 ON" ditekan
6	V5	Mengirim nilai ke NodeMCU untuk menonaktifkan mesin 2 ketika Push Button "MC2 OFF" ditekan
7	V6	Mengirim nilai ke NodeMCU untuk mengaktifkan mode Auto ketika Push Button "Auto" ditekan
8	V7	Mengirim nilai ke NodeMCU untuk mengaktifkan mode Manual ketika Push Button "Manual" ditekan
9	V8	Menyimpan nilai tekanan air
10	V9	Menyimpan nilai debit air

### Flowchart Sistem

Terdapat dua mode sistem yang dibuat dalam penelitian, yaitu Mode Auto dan Mode Manual. Untuk mempermudah dalam melakukan pemrograman, dibuatlah *flowchart* untuk kedua mode tersebut.

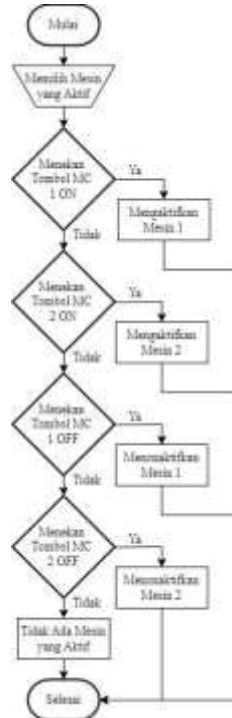
#### 1. Mode Auto



Gambar 15. Flowchart Mode Auto

Pada Gambar 15 dijelaskan bahwa mesin pompa air bekerja berdasarkan nilai pembacaan sensor *pressure transducer*. Mesin pompa akan aktif secara otomatis apabila tekanan air kurang dari 0,45 MPa dan akan nonaktif secara otomatis ketika tekanan air lebih dari 0,5 MPa.

## 2. Mode Manual



Gambar 16. Flowchart Mode Manual

Pada Gambar 16 dijelaskan bahwa mesin pompa air bekerja berdasarkan kontrol yang dilakukan oleh operator. Aktif dan nonaktifnya mesin pompa tergantung pada tombol yang ditekan oleh operator.

## Program Sistem

Program sistem dalam penelitian ini terbagi menjadi dua bagian. Bagian pertama adalah membuat program untuk membaca sensor pada Arduino Nano. Bagian kedua adalah membuat program kontrol manual dan otomatis pada NodeMCU ESP8266, di mana mode otomatisnya beroperasi berdasarkan nilai-nilai sensor yang diukur.

## Kalibrasi Sensor Pressure Transducer

Pengukuran tekanan air menggunakan *prototype* yang telah dibuat dilakukan dengan menghubungkan *input prototype* ke *output* tekanan air yang dapat mencapai 0,5 MPa, di mana tekanan ini diatur oleh *ball valve*. Selama proses pengukuran, digunakan *pressure gauge* (master) untuk membandingkan hasil tekanan air yang dibaca oleh *pressure transducer*. *Pressure gauge* berfungsi sebagai acuan utama untuk memastikan bahwa data yang dihasilkan oleh *pressure transducer* akurat. Kemudian, dilakukan pengambilan data sebanyak

20 sampel yang dapat dilihat melalui Tabel 7.

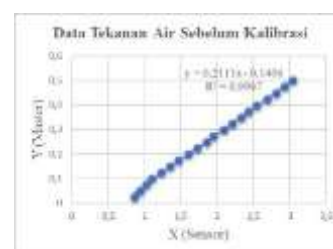
Tabel 7. Hasil Pengukuran Awal Tekanan Air Menggunakan *Pressure Transducer*

No	Tanggal	Sensor (X) (MPa)	Master (Y) (MPa)	X-Y	Error (%)	Akurasi (%)
1	15/05/24	0,861	0,025	0,836	3344	-3244
2	15/05/24	0,939	0,05	0,889	1778	-1678
3	15/05/24	1,022	0,075	0,947	1263	-1163
4	15/05/24	1,1	0,1	1	1000	-900
5	15/05/24	1,232	0,125	1,107	886	-786
6	15/05/24	1,35	0,15	1,2	800	-700
7	15/05/24	1,469	0,175	1,294	739	-639
8	15/05/24	1,602	0,2	1,402	701	-601
9	15/05/24	1,73	0,225	1,505	669	-569
10	15/05/24	1,854	0,25	1,604	642	-542
11	15/05/24	1,949	0,275	1,674	609	-509
12	15/05/24	2,092	0,3	1,792	597	-497
13	15/05/24	2,211	0,325	1,886	580	-480
14	15/05/24	2,329	0,35	1,979	565	-465
15	15/05/24	2,435	0,375	2,06	549	-449
16	15/05/24	2,546	0,4	2,146	537	-437
17	15/05/24	2,69	0,425	2,265	533	-433
18	15/05/24	2,812	0,45	2,362	525	-425
19	15/05/24	2,928	0,475	2,453	516	-416
20	15/05/24	3,047	0,5	2,547	509	-409
		Rata-rata			867,1	-767,1

Berdasarkan Tabel 7, ditemukan bahwa persentase *error* pada *pressure transducer* cukup besar, dengan rata-rata mencapai 867,1%. Persentase *error* ini dihitung menggunakan persamaan (5). Untuk itu akan dilakukan kalibrasi untuk mengurangi tingkat *error* dari pembacaan sensor tersebut. Hasil kalibrasi yang diharapkan memiliki rata-rata akurasi sebesar  $\pm 1,5\%$  dari pembacaan *pressure gauge*. Untuk menghitung nilai akurasi digunakan persamaan (7).

Kalibrasi dilakukan dengan membuat model persamaan regresi dari data yang telah diambil dengan menggunakan persamaan (1).

$$= -0,1406 + 0,2111$$



Gambar 17. Grafik Linearitas Sensor *Pressure Transducer* Pertama

Setelah didapatkan persamaan regresi linear seperti Gambar 17 berdasarkan perhitungan yang sudah dilakukan, maka rumus tersebut akan dimasukan ke dalam program Arduino agar pembacaan dari sensor tersebut memiliki tingkat *error* yang rendah dan memiliki akurasi yang tinggi.

Setelah program untuk kalibrasi sensor *pressure transducer* ditambahkan, langkah



berikutnya adalah melakukan pengambilan data sampel untuk mengevaluasi keberhasilan kalibrasi tersebut. Hasil pengambilan data sampel dapat dilihat melalui

Tabel 8. Hasil Pengukuran Tekanan Air Menggunakan *Pressure Transducer* Setelah Kalibrasi Pertama

No	Tanggal	Sensor (X) (MPa)	Master (Y) (MPa)	X-Y	Error (%)	Akurasi (%)
1	15/05/24	0,282	0,025	0,26	1028,0	-928,0
2	15/05/24	0,338	0,05	0,29	576,0	-476,0
3	15/05/24	0,362	0,075	0,29	382,7	-282,7
4	15/05/24	0,377	0,1	0,28	277,0	-177,0
5	15/05/24	0,402	0,125	0,28	221,6	-121,6
6	15/05/24	0,431	0,15	0,28	187,3	-87,3
7	15/05/24	0,451	0,175	0,28	157,7	-57,7
8	15/05/24	0,479	0,2	0,28	139,5	-39,5
9	15/05/24	0,513	0,225	0,29	128,0	-28,0
10	15/05/24	0,542	0,25	0,29	116,8	-16,8
11	15/05/24	0,559	0,275	0,28	103,3	-3,3
12	15/05/24	0,591	0,3	0,29	97,0	3,0
13	15/05/24	0,608	0,325	0,28	87,1	12,9
14	15/05/24	0,631	0,35	0,28	80,3	19,7
15	15/05/24	0,663	0,375	0,29	76,8	23,2
16	15/05/24	0,688	0,4	0,29	72,0	28,0
17	15/05/24	0,705	0,425	0,28	65,9	34,1
18	15/05/24	0,740	0,45	0,29	64,4	35,6
19	15/05/24	0,769	0,475	0,29	61,9	38,1
20	15/05/24	0,799	0,5	0,30	59,8	40,2
Rata-rata					199,2	-99,2

Berdasarkan Tabel 8 terlihat bahwa persentase *error* telah mengalami penurunan dari 867,1% menjadi 199,2%. Meskipun demikian, tingkat *error* tersebut masih tergolong tinggi. Selain itu, tingkat akurasi juga mengalami peningkatan dari sebelumnya -767,1% menjadi -99,2%. Namun, tingkat akurasi masih jauh dari toleransi yang tertera pada *datasheet* sensor *pressure transducer*, yaitu  $\pm 1,5\%$ . Oleh karena itu, akan dilakukan kalibrasi kedua agar tingkat akurasi dapat mencapai tingkat yang sesuai dengan toleransi yang diharapkan.

Kalibrasi dilakukan dengan membuat model persamaan regresi dari data hasil kalibrasi pertama menggunakan persamaan (1).

$$= -0,2635 + 0,9625$$



Gambar 18. Grafik Linearitas Sensor *Pressure Transducer* Kedua

Setelah didapatkan persamaan regresi linear dari hasil perhitungan kalibrasi kedua yang dapat dilihat pada Gambar 18, maka rumus tersebut akan dimasukkan ke dalam program Arduino sama seperti

kalibrasi pertama dan lakukan kembali pengambilan data sampel untuk mengevaluasi hasil kalibrasi kedua yang telah dilakukan. Hasil pengambilan data sampel dapat dilihat melalui Tabel 9.

Tabel 9. Data Pengukuran Tekanan Air Menggunakan *Pressure Transducer* Setelah Kalibrasi Kedua

No	Tanggal	Sensor (X) (MPa)	Master (Y) (MPa)	X-Y	Error (%)	Akurasi (%)
1	15/05/24	0,024	0,025	0,001	4,000	96,000
2	15/05/24	0,051	0,05	0,001	2,000	98,000
3	15/05/24	0,076	0,075	0,001	1,333	98,667
4	15/05/24	0,099	0,1	0,001	1,000	99,000
5	15/05/24	0,124	0,125	0,001	0,800	99,200
6	15/05/24	0,149	0,15	0,001	0,667	99,333
7	15/05/24	0,176	0,175	0,001	0,571	99,429
8	15/05/24	0,202	0,2	0,002	1,000	99,000
9	15/05/24	0,225	0,225	0,000	0,000	100,000
10	15/05/24	0,251	0,25	0,001	0,400	99,600
11	15/05/24	0,273	0,275	0,002	0,727	99,273
12	15/05/24	0,299	0,3	0,001	0,333	99,667
13	15/05/24	0,326	0,325	0,001	0,308	99,692
14	15/05/24	0,35	0,35	0,000	0,000	100,000
15	15/05/24	0,377	0,375	0,002	0,533	99,467
16	15/05/24	0,401	0,4	0,001	0,250	99,750
17	15/05/24	0,424	0,425	0,001	0,235	99,765
18	15/05/24	0,449	0,45	0,001	0,222	99,778
19	15/05/24	0,475	0,475	0,000	0,000	100,000
20	15/05/24	0,501	0,5	0,001	0,200	99,800
Rata-rata					0,729029	99,27

Berdasarkan Tabel 9, terlihat bahwa persentase *error* telah mengalami penurunan dari 199,2% menjadi 0,73%. Dengan hasil tersebut, dapat dikatakan bahwa *error* yang terjadi setelah kalibrasi kedua memiliki tingkat *error* yang sangat rendah, yaitu di bawah 1%. Selain itu, tingkat akurasi juga mengalami peningkatan dari -99,2% menjadi 99,27%. Nilai akurasi tersebut sudah memenuhi syarat karena tidak melebihi toleransi yang tertera pada *datasheet* sensor *pressure transducer*, yaitu  $\pm 1,5\%$ .

### Kalibrasi Sensor *Water Flow* YF-S201

Pengukuran debit air menggunakan *prototype* yang telah dibuat dilakukan dengan menghubungkan *input prototype* ke keran. Proses pengukuran debit air dilakukan dengan membandingkan nilai yang dibaca oleh sensor *water flow* YF-S201 dengan hasil pengukuran menggunakan gelas ukur berukuran 1liter dan *stopwatch*. Gelas ukur digunakan untuk menampung air yang keluar dari sensor *water flow*, sementara *stopwatch* digunakan untuk menghitung waktu yang dibutuhkan untuk mengisi gelas ukur tersebut. Untuk menghasilkan nilai debit (l/m) dalam pengukuran menggunakan gelas ukur dan *stopwatch*, dilakukan perhitungan menggunakan persamaan (4).

Pengambilan data dilakukan dengan mengambil sampel sebanyak 15 kali yang dapat

dilihat melalui Tabel 10.

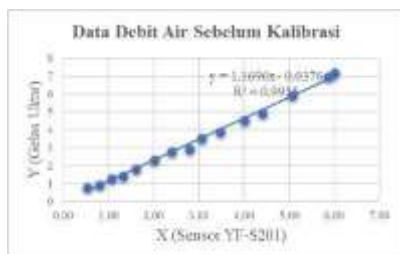
Tabel 10. Data Pengukuran Debit Air Menggunakan *Water Flow* YF-S201

No	Tanggal	YF-S201 (X) (l/m)	Gelas (Y) (l/m)	X-Y	Error (%)	Akurasi (%)
1	15/05/24	0,53	0,80	0,2658	33,40	66,60
2	15/05/24	0,80	0,94	0,1362	14,55	85,45
3	15/05/24	1,07	1,31	0,2438	18,56	81,44
4	15/05/24	1,33	1,44	0,1106	7,68	92,32
5	15/05/24	1,60	1,83	0,2343	12,77	87,23
6	15/05/24	2,00	2,33	0,3328	14,27	85,73
7	15/05/24	2,40	2,79	0,3894	13,96	86,04
8	15/05/24	2,80	2,93	0,1340	4,57	95,43
9	15/05/24	3,07	3,55	0,4761	13,43	86,57
10	15/05/24	3,47	3,91	0,4413	11,28	88,72
11	15/05/24	4,00	4,56	0,5558	12,20	87,80
12	15/05/24	4,40	4,93	0,5302	10,75	89,25
13	15/05/24	5,07	5,92	0,8472	14,32	85,68
14	15/05/24	5,87	6,96	1,0906	15,67	84,33
15	15/05/24	6,00	7,19	1,1856	16,50	83,50
Rata-rata					14,26	85,74

Tabel 10 menunjukkan hasil uji coba sensor *water flow* YF-S201 tanpa dilakukan kalibrasi, dengan rentang ukur 1-6 liter per menit. Ditemukan bahwa persentase *error* pada sensor *water flow* cukup besar, dengan rata-rata mencapai 14,26%, dan rata-rata akurasi terhadap pengukuran gelas ukur hanya 85,74%. Berdasarkan data tersebut, akan dilakukan kalibrasi untuk mengurangi kesalahan pembacaan yang dihasilkan oleh sensor *water flow* YF-S201 terhadap gelas ukur (master). Hasil kalibrasi yang diharapkan memiliki rata-rata akurasi sebesar  $\pm 10\%$  dari pembacaan gelas ukur.

Kalibrasi dilakukan dengan membuat model persamaan regresi dari data yang telah diambil dengan menggunakan persamaan (1).

$$= -0,0376 + 1,1696$$



Gambar 19. Grafik Linearitas Sensor *Water Flow* YF-S201

Setelah didapatkan persamaan regresi linear dari hasil perhitungan yang sudah dilakukan seperti pada Gambar 19, maka persamaan tersebut akan dimasukkan ke dalam program Arduino agar pembacaan dari sensor tersebut memiliki tingkat *error* yang rendah dan memiliki akurasi yang tinggi.

Setelah program untuk kalibrasi sensor *water flow* YF-S201 ditambahkan, langkah berikutnya adalah melakukan pengambilan data sampel untuk mengevaluasi keberhasilan kalibrasi tersebut. Hasil pengambilan data sampel dapat dilihat melalui Tabel 11.

Tabel 11. Data Pengukuran Debit Air Menggunakan *Water Flow* YF-S201 Setelah Kalibrasi

No	Tanggal	YF-S201 (X) (l/m)	Gelas (Y) (l/m)	X-Y	Error (%)	Akurasi (%)
1	15/05/24	0,59	0,68	0,09	13,70	86,30
2	15/05/24	0,90	0,93	0,03	3,46	96,54
3	15/05/24	1,05	1,10	0,05	4,17	95,83
4	15/05/24	1,37	1,41	0,04	2,62	97,38
5	15/05/24	1,68	1,70	0,02	0,99	99,01
6	15/05/24	2,00	2,00	0,00	0,17	99,83
7	15/05/24	2,46	2,52	0,06	2,42	97,58
8	15/05/24	2,80	2,87	0,07	2,47	97,53
9	15/05/24	3,07	3,11	0,04	1,15	98,85
10	15/05/24	3,47	3,49	0,02	0,58	99,42
11	15/05/24	4,00	4,01	0,01	0,33	99,67
12	15/05/24	4,40	4,35	0,05	1,13	98,87
13	15/05/24	5,07	4,86	0,21	4,36	95,64
14	15/05/24	5,87	5,74	0,13	2,33	97,67
15	15/05/24	6,00	5,97	0,03	0,50	99,50
Rata-rata					2,69	97,31

Berdasarkan Tabel 11, terlihat bahwa persentase *error* telah mengalami penurunan dari 14,26% menjadi 2,69%. Dengan hasil tersebut, dapat dikatakan bahwa *error* yang terjadi setelah kalibrasi kedua memiliki tingkat *error* yang sangat rendah, yaitu di bawah 3%. Selain itu, tingkat akurasi juga mengalami peningkatan dari -85,74% menjadi 97,31%. Nilai akurasi tersebut sudah memenuhi syarat karena tidak melebihi toleransi yang tertera pada *datasheet* sensor *water flow* YF-S201, yaitu  $\pm 10\%$ .

### Uji Verifikasi

Uji verifikasi dilakukan untuk memastikan sistem yang dibuat sudah berfungsi dengan baik, berikut adalah beberapa pengujian yang dilakukan:

#### 1. Verifikasi Data Sensor *Pressure Transducer*

Uji verifikasi data sensor *pressure transducer* bertujuan untuk mengetahui kesesuaian nilai yang terbaca pada Serial Monitor dengan nilai yang ada pada Firebase, Blynk dan Spreadsheets. Hasil verifikasi dapat dilihat pada Tabel 12.

Tabel 12. Uji Verifikasi Data Sensor *Pressure Transducer*

Pengujian ke-	Serial Monitor	Firebase	Blynk	Spreadsheets	Hasil
1	0,05	0,05	0,05	0,05	OK
2	0,1	0,1	0,1	0,1	OK
3	0,15	0,15	0,15	0,15	OK
4	0,2	0,2	0,2	0,2	OK
5	0,25	0,25	0,25	0,25	OK
6	0,3	0,3	0,3	0,3	OK
7	0,35	0,35	0,35	0,35	OK
8	0,4	0,4	0,4	0,4	OK
9	0,45	0,45	0,45	0,45	OK
10	0,5	0,5	0,5	0,5	OK

#### 2. Verifikasi Data Sensor *Water Flow* YF-S201

Uji verifikasi data sensor *water flow* YF-S201 bertujuan untuk mengetahui kesesuaian nilai yang terbaca pada Serial Monitor dengan nilai yang

ada pada Firebase, Blynk dan Spreadsheets. Hasil verifikasi dapat dilihat pada Tabel 13.

Tabel 13. Uji Verifikasi Data Sensor *Water Flow* YF-S201

Pengujian ke-	Serial Monitor	Firebase	Blynk	Spreadsheets	Hasil
1	1	1	1	1	OK
2	1,5	1,5	1,5	1,5	OK
3	2	2	2	2	OK
4	2,5	2,5	2,5	2,5	OK
5	3	3	3	3	OK
6	3,5	3,5	3,5	3,5	OK
7	4	4	4	4	OK
8	4,5	4,5	4,5	4,5	OK
9	5	5	5	5	OK
10	5,5	5,5	5,5	5,5	OK

#### 4. Verifikasi Mode Otomatis

Uji verifikasi dilakukan untuk memastikan bahwa mode otomatis sudah berfungsi dengan baik. Hasil uji verifikasi mode otomatis dapat dilihat pada Tabel 14.

Tabel 14. Uji Verifikasi Mode Otomatis

No.	Pengujian	Hasil yang Diharapkan	Hasil
1	Tekanan Air <= 0,45 MPa	relay 1 aktif	OK
2	Tekanan Air >= 0,45 MPa & <= 0,5 MPa	Mempertahankan kondisi yang ada	OK
3	Tekanan Air <= 0,45 MPa	relay 2 aktif	OK
4	Tekanan Air >= 0,45 MPa & <= 0,5 MPa	Mempertahankan kondisi yang ada	OK
5	Tekanan Air >= 0,5 MPa	relay 2 nonaktif	OK
6	Tekanan Air >= 0,5 MPa	relay 1 nonaktif	OK

#### 5. Verifikasi Mode Manual

Uji verifikasi dilakukan untuk memastikan bahwa mode manual sudah berfungsi dengan baik. Hasil uji verifikasi mode manual dapat dilihat pada Tabel 15.

Tabel 15. Uji Verifikasi Mode Manual

No.	Pengujian	Hasil yang Diharapkan	Hasil
1	Menekantombol atau <i>push button</i> MC 1 ON	Relay 1 aktif	OK
2	Menekantombol atau <i>push button</i> MC 1 OFF	relay 1 nonaktif	OK
3	Menekantombol atau <i>push button</i> MC 2 ON	relay 2 aktif	OK
4	Menekantombol atau <i>push button</i> MC 2 OFF	relay 2 nonaktif	OK
5	Menekantombol atau <i>push button</i> MC 1 ON dan MC 2 ON secara bersamaan	relay 1 & 2 aktif	OK
6	Menekantombol atau <i>push button</i> MC 1 OFF dan MC 2 OFF secara bersamaan	relay 1 & 2 nonaktif	OK

#### 3. Verifikasi Notifikasi

Uji verifikasi dilakukan untuk memastikan bahwa pengiriman notifikasi peringatan sudah berfungsi dengan baik. Hasil uji verifikasi mode manual dapat dilihat pada Tabel 16.

Tabel 16. Uji Verifikasi Notifikasi

No.	Pengujian	Hasil yang Diharapkan	Hasil
1	Tekanan Air <= 0,45 MPa	Notifikasi terkirim ke aplikasi dan email	OK
2	Debit Air <= 3 l/m	Notifikasi terkirim ke aplikasi dan email	OK

### IV. KESIMPULAN

*Prototype* sistem *monitoring* telah berhasil dirancang bangun menggunakan Arduino Nano dan NodeMCU ESP8266. Arduino Nano berfungsi untuk membaca data dari sensor *pressure transducer* dan sensor *water flow* YF-S201 yang digunakan untuk mengukur tekanan air dan debit air. NodeMCU ESP8266 digunakan untuk mengirimkan data sensor ke Firebase dan aplikasi Blynk melalui koneksi WIFI sehingga data dari sensor tersebut dapat diakses secara *online*. Sensor yang digunakan memiliki akurasi pembacaan sebesar 99,27% untuk pembacaan sensor *pressure transducer* dan 97,31% untuk pembacaan sensor *water flow* YF-S201. Selain itu, dalam sistem ini pencatatan data telah berhasil dilakukan secara otomatis untuk mencatat tekanan dan debit air setiap satu jam.

### DAFTAR PUSTAKA

- [1] T. Hafzara Siregar, S. Permana Sutisna, G. Eka Pramono, And M. Malik Ibrahim, "Rancang Bangun Sistem Pendeteksi Kebakaran Berbasis Iot Menggunakan Arduino," 2020. [Online]. Available: [Http://Ejournal.Uika-Bogor.Ac.Id/Index.Php/Ame/Index](http://ejournal.uika-bogor.ac.id/index.php/ame/index)
- [2] P. Prasetyawan, S. Samsugi, And R. Prabowo, "Internet Of Thing Menggunakan Firebase Dan Nodemcu Untuk Helm Pintar," *Jurnal Eltikom*, Vol. 5, No. 1, Pp. 32–39, Mar. 2021, Doi: 10.31961/Eltikom.V5i1.239.
- [3] D. A. Firmansyah, S. A. H. Ahmad, E. Priyatna, V. Firmansyah, And S. Suprijanto, "Rancang Bangun Instalasi Uji Akurasi Meter Air Untuk Studi Pengaruh Tekanan Air," *Elkomika: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, Vol. 10, No. 3, P. 529, Jul. 2022, Doi: 10.26760/Elkomika.V10i3.529.
- [4] H. Hermansyah And N. Silitonga, "Pengembangan Prototype Water Flow Meter Berbasis Iot Dengan Sistem Monitoring Menggunakan Esp 8266 Pada Pipa Di Unit Water Treatment Plant Pt. Perkebunan Nusantara Ii Pks Sawit Hulu," *Ira Jurnal Teknik Mesin Dan Aplikasinya*

- (*Irajtma*), Vol. 2, No. 3, Pp. 22–29, Dec. 2023, Doi: 10.56862/Irajtma.V2i3.75.
- [5] C. Sulaeman, G. Farhandika, A. M. Rizki, And D. Fauziah, “Prosiding Seminar Nasional Energi, Telekomunikasi Dan Otomasi Sneto 2023 Smart Metering Konsumsi Daya Listrik Berbasis Iot Dengan Pencatat Data Spreadsheet Dan Notifikasi Telegram,” 2023.
- [6] R. Y. Bait, I. Fahmi, L. F. Boesday, P. Pendidikan Teknik Elektro, And U. Nusa Cendana Jl Adisucipto, “Desain Alat Dispenser Hand Sanitizer Otomatis Menggunakan Sensor Ultrasonic Berbasis Arduino Nano,” 2023.
- [7] S. Sawidin *Et AL.*, “Prosiding The 12 Th Industrial Research Workshop And National Seminar Bandung,” 2021. [Online]. Available: [Www.Arduino.Cc](http://www.Arduino.Cc)
- [8] M. Makmun, T. Khasanah, N. Sirojudin, And \* Korespondensi, “Integrasi Programmable Logic Control Outseal Mega V.2 Dengan Nodemcu Esp826 Dengan Menerapkan Internet Of Things Outseal Mega V.2 Programmable Logic Control Integration With Nodemcu Esp826 By Implementing Internet Of Things,” *Online) Teknologi: Jurnal Ilmiah Sistem Informasi*, Vol. 13, No. 1, P. 826, 2023, Doi: 10.26594/Teknologi.V13i1.3868.
- [9] F. F. Akmal, E. Susanto, And M. R. Rosa, “Online Monitoring Dan Kontrol Besaran Pressure Dan Flow Pada Prototipe Perpipaan Minyak Dengan Menggunakan Pole Placement Pada Networked Control System Online Monitoring And Control Of Pressure And Flow In Oil Pipeline System Prototype Using Pole Placement In Networked Control System,” 2021.
- [10] R. Berlianti And Fibriyanti, “Perancangan Alat Pengontrolan Beban Listrik Satu Fasa Jarak Jauh,” *Sainetin (Jurnal Sain, Energi, Teknologi & Industri)*, Vol. 5, No. 1, Pp. 17–26, Dec. 2020.
- [11] F. Irsan Pasaribu, E. Sahnur Nasution, L. Nasution, And M. Rasidy, “Disain Alat Monitoring Real-Time Dari Kualitas Air Tambak Udang Berbasis Internet Of Things,” Vol. 6, No. 2, Pp. 29–35, 2024, Doi: 10.30596/Rele.V6i1.17200.
- [12] R. Ananda Pratama And M. Arman, “Sistem Akuisisi Data Temperatur Showcase Berbasis Iot Menggunakan Esp32 Dengan Sensor Termokopel Dan Logging Ke Google Spreadsheets,” 2023.
- [13] S. Ardhi, T. Putera Gunawan, S. Tjandra, And G. L. Dewi, “Penerapan Metode Regresi Linear Dalam Pengembangan Pengukuran Aliran Air Pada Sensor Yf-S201,” 2023. [Online]. Available: [Http://Univ45sby.Ac.Id/Ejournal/Index.Php/Industri/Index](http://Univ45sby.Ac.Id/Ejournal/Index.Php/Industri/Index)
- [14] R. Eko Cahyono *Et AL.*, “Rancang Bangun Prototipe Smart Control System Sebagai Pengendali Debit Air Berbasis Computer Control,” 2024.
- [15] R. Atma Ivory, N. Kholis, And F. Baskoro, “Review Penggunaan Sensor Suhu Terhadap Respon Pembacaan Skala Pada Inkubator Bayi,” 2021.
- [16] S. P. Sipayung, N. A. Siagian, And A. K. B. Ginting, “Analisis Prediksi Jumlah Penerimaan Mahasiswa Baru Dengan Metode Regresi Linier Sederhana,” 2024.
- [17] I. A. Rozaq, D. Tauhida, R. Primadasa, And F. Setiawan, “Uji Linearitas Sensor Konduktivitas Pada Prototipe Pengukur Kualitas Air Tambak,” 2023.