

# Penerapan IoT Pada Sistem Monitoring dan Otomatisasi Pengoperasian Pompa Air Kabel Duct Gardu Induk Tanggul

Nabil Ridho Setyawan<sup>1\*</sup>, Andrijani Sumarahinsih<sup>2</sup>, Aries Boedi Setiawan<sup>3</sup>

<sup>1,2,3</sup> Program Studi Teknik Elektro, Fakultas Teknik, Universitas Merdeka Malang, Malang

\*Penulis Korespondensi, email: [nabilridho0203@gmail.com](mailto:nabilridho0203@gmail.com)

Received: 08/12/2025

Revised: 25/01/2026

Accepted: 31/05/2026

**Abstract.** Petugas Jaringan dan Gardu Induk (JarGI) or Operator Gardu Induk (OpGI) cannot constantly check or monitor the condition of the water pump in the cable duct control tank at the substation. Therefore, it is necessary to develop a device that can be used to monitor the condition of the water pump in the cable duct control tank, allowing monitoring to be carried out anytime and anywhere without the need to be physically present at the pump location. This research involves the development of a monitoring device that utilizes an IoT concept based on ESP32 and is capable of sending notifications to users via Telegram. The study uses a literature review method for data collection, drawing from related scientific journals as well as testing the developed device. The device uses ESP32 as its core component, supported by a water flow sensor and a water float switch as input sources. The output section uses a relay to connect the device to the water pump and to send notifications through the Telegram application. The testing conducted on the device showed good results, with the water level sensor sending a signal when the float reaches a height of 8 mm, and the error factor from the water flow sensor readings showing an average value of 1.9%.

**Keywords:** Monitor, ESP32, Water pump, Cable duct, Water flow sensor, Water float switch, Telegram.

**Abstrak.** Petugas Jaringan dan Gardu Induk (JarGI) atau Operator Gardu Induk (OpGI) tidak mungkin untuk setiap saat mengecek atau memantau keadaan pompa air kabel *duct* pada bak kontrol Gardu Induk. Sehingga perlu membuat sebuah alat yang dapat digunakan untuk memonitoring keadaan pompa air pada bak kontrol kabel *duct* sehingga dapat dilakukan monitoring dimanapun dan kapanpun tanpa harus datang langsung ke lokasi pompa air. Pada penelitian ini dilakukan pembuatan alat monitoring yang menggunakan konsep IoT berbasis ESP32 dan dapat mengirimkan notifikasi menggunakan telegram kepada penggunanya. Penelitian ini menggunakan metode pengumpulan data studi literatur dimana peneliti mengambil bahan penelitian dari jurnal ilmiah terkait dan juga pengujian pada alat yang dibuat. Alat ini menggunakan ESP32 sebagai otak dari alat kemudian ditopang dengan sensor *water flow* dan juga *water float switch* sebagai sumber input dari alat ini, kemudian pada bagian *output* menggunakan *relay* untuk menghubungkan alat dengan pompa air dan juga notifikasi yang akan dikirimkan melalui aplikasi Telegram. Pengujian yang dilakukan pada alat menunjukkan hasil yang baik dengan sensor *water level* akan mengirim sinyal apabila pelampung sudah mencapai tinggi 8mm dan faktor *error* dari hasil pembacaan sensor *water flow* menunjukkan nilai rata rata sebesar 1,9%.

**Kata Kunci:** Memantau, ESP32, Pompa air, Kabel *duct*, *Water flow sensor*, *Water float switch*, Telegram.

## I. PENDAHULUAN

Dalam sistem gardu induk konvensional, peralatan utama terbagi menjadi peralatan yang berada di *switchyard* dan peralatan yang berada di dalam gedung gardu induk. Peralatan di *switchyard* berfungsi untuk menyalurkan dan menerima energi listrik tegangan tinggi, sedangkan peralatan di dalam gedung berperan dalam sistem kontrol, suplai tegangan DC, serta pengoperasian panel kubikel. Keterhubungan antarperalatan tersebut dilakukan

melalui jalur kabel *duct* yang berada di bawah permukaan tanah. Kondisi ini menyebabkan kabel *duct* memiliki potensi mengalami genangan air yang dapat merusak kabel kontrol maupun kabel daya serta mengganggu keandalan sistem gardu induk [1], [2].

Sebagai upaya pencegahan, gardu induk dilengkapi dengan pompa air untuk mengeluarkan genangan dari dalam kabel *duct*. Namun, sistem pemantauan yang masih dilakukan secara manual

mengharuskan petugas melakukan inspeksi langsung ke lokasi pompa secara berkala. Kondisi ini dinilai kurang efektif dan berpotensi menyebabkan keterlambatan penanganan apabila terjadi gangguan pada sistem pompa [3]. Beberapa penelitian menunjukkan bahwa penerapan teknologi *Internet of Things* (IoT) mampu meningkatkan efektivitas sistem monitoring air melalui pengiriman data secara *real-time* dan jarak jauh [4], [5].

Perkembangan teknologi *IoT* memungkinkan integrasi mikrokontroler ESP32 dengan berbagai sensor untuk keperluan monitoring dan kontrol sistem air. ESP32 banyak digunakan karena telah terintegrasi dengan modul *WiFi* serta memiliki kemampuan pemrosesan data yang baik [6], [7]. Selain itu, pemanfaatan aplikasi Telegram sebagai media notifikasi dan kontrol jarak jauh telah terbukti efektif karena kemudahan akses, kecepatan pengiriman data, dan fleksibilitas antarmuka pengguna [5], [7].

Berdasarkan permasalahan tersebut, penelitian ini mengusulkan perancangan dan implementasi sistem monitoring dan otomatisasi pengoperasian pompa air kabel *duct* berbasis *IoT*. Sistem ini menggunakan ESP32 sebagai pusat kendali, sensor *water level* sebagai pemicu kerja pompa, sensor *water flow* sebagai indikator kinerja pompa, serta Telegram sebagai antarmuka pengguna untuk melakukan monitoring dan pengendalian jarak jauh.

## II. METODOLOGI

Metode yang digunakan pada penelitian ini meliputi studi literatur, perancangan sistem, perancangan *hardware*, perancangan *software*, perancangan mekanik.

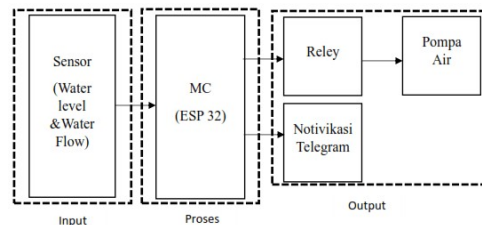
### A. Studi literatur

Studi literatur dilakukan untuk memperoleh referensi dan landasan teori yang berkaitan dengan penerapan sistem monitoring berbasis *IoT* pada sistem air. Literatur yang dikaji meliputi penggunaan mikrokontroler ESP32 sebagai perangkat utama *IoT*, penerapan sensor *water level* untuk mendeteksi ketinggian air serta penggunaan sensor *water flow* untuk memantau aliran air dan kinerja pompa [8],[9]. Selain itu, dikaji pula

penelitian yang memanfaatkan aplikasi Telegram sebagai media monitoring dan notifikasi jarak jauh. Hasil studi literatur ini digunakan sebagai dasar dalam menentukan arsitektur sistem dan metode pengujian alat.

### B. Perancangan sistem

Perancangan sistem dilakukan dengan menyusun diagram blok yang menggambarkan alur kerja sistem monitoring dan otomatisasi pompa air kabel *duct*. Sistem memiliki dua input utama, yaitu sensor *water level* dan sensor *water flow*. Data dari sensor diproses oleh ESP32, kemudian digunakan untuk mengendalikan *relay* pompa serta mengirimkan notifikasi kondisi sistem melalui aplikasi Telegram. Konsep perancangan ini mengacu pada sistem monitoring air berbasis *IoT* yang telah diterapkan pada penelitian sebelumnya [3], [4].



**Gambar 1.** Blok Diagram Sistem  
(Sumber : Dokumentasi pribadi)

Dari diagram blok yang disusun terdapat dua buah masukan yang akan diolah oleh *microcontroller* yang nantinya akan di aktualisasikan dalam bentuk gerakan mekanik pada pompa dan notifikasi yang muncul pada telegram. Berikut adalah penjelasan dari dua mode masukan yang terdapat dalam alat yang penulis buat.

#### 1. Sensor *water level*

Sensor *water level* akan mendeteksi kondisi *level* air pada bak kontrol, apabila kondisi level air sudah mencapai ambang batas yang ditentukan, maka mikrokontroler (ESP 32) akan memerintahkan *relay* untuk menyalakan pompa air dan juga mengirim notifikasi pada Telegram.

#### 2. Sensor *water flow*

Sensor *water flow* akan mendeteksi adanya aliran air pada *outlet* dari pompa air, kemudian

mikrokontroler (ESP 32) akan mengirimkan kondisi pompa air (beroperasi/tak beroperasi).

### C. Perancangan hardware

#### 1. ESP 32

ESP 32 berperan sebagai pusat *control* atau otak dari rancang bangun alat ini. Dalam ESP 32 terdapat semua program untuk menjalankan sensor sensor yang ada di dalam rancang bangun dan mengirimnya kedalam *user interface* [6], [7].

#### 2. Sensor *water level* K-0135

Sensor ini digunakan untuk mengukur atau memitigasi kenaikan muka air dalam bak *control*. Sensor ini disusun secara *vertical* sesuai dengan kebutuhan dari pembuatan rancang bangun alat ini (menyesuaikan dengan kedalaman bak kontrol). Penulis akan menggunakan beberapa sensor yang [1], [2], [4].

#### 3. Sensor *water flow*

Sensor ini digunakan untuk mengetahui apakah pompa air bekerja sesuai dengan fungsinya atau tidak. Sensor ini diletakan pada bagian *outlet* pompa air, ketika sensor *water level* memberikan perintah untuk pompa bekerja, maka pompa akan menyala dan mengeluarkan air pada bagian *outlet* pompa. Dalam waktu 1 menit sensor akan membaca aliran air dan mengirimkan notifikasi kedalam Telegram [8], [9], [10].

#### 4. Relai

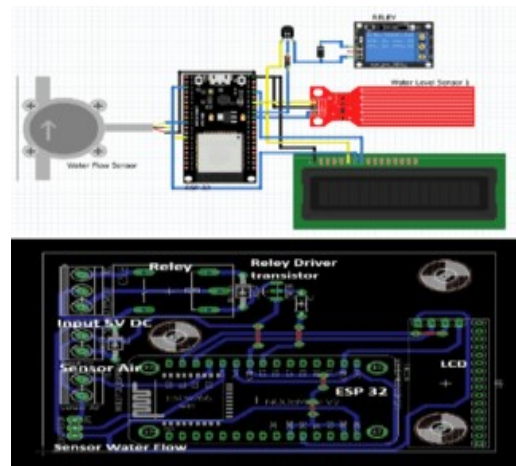
Pemasangan relai digunakan untuk menyambungkan antara kontrol dari *microcontroller* dengan sumber tegangan AC yang digunakan sebagai sumber tegangan pada motor pompa [3], [5].

#### 5. Proteksi *microcontroller*

Proteksi *microcontroller* menggunakan rangkain *relay driver transistor* dengan menggunakan dioda 1N4004 sebagai proteksi lonjakan arus saat *relai* mati, kemudian transistor C9014 bertindak sebagai saklar elektronik dan resistor 10K ohm sebagai pembatas arus dari pin *microcontroller*.

ESP32 berperan sebagai pusat kendali sistem yang mengolah seluruh data masukan dari sensor serta mengatur keluaran sistem. Sensor *water level* K-0135 berfungsi sebagai pendeteksi kenaikan muka air pada bak kontrol yang

memberikan sinyal awal kepada ESP32 untuk memicu pengoperasian pompa air. Setelah pompa aktif, sensor *water flow* yang dipasang pada outlet pompa digunakan untuk memantau aliran air sebagai indikator kinerja pompa dan memastikan proses pengosongan air berjalan dengan baik. Seluruh perintah pengoperasian pompa dikendalikan melalui *relay* yang berfungsi sebagai penghubung antara sinyal kontrol bertegangan rendah dari ESP32 dengan sumber tegangan AC pada motor pompa. Untuk menjaga keandalan dan keamanan sistem, rangkaian proteksi mikrokontroler diterapkan menggunakan *relay driver transistor* dan dioda proteksi guna melindungi ESP32 dari lonjakan arus listrik selama proses *switching*. Dengan keterpaduan antar komponen tersebut, sistem monitoring dan otomatisasi pompa air kabel *duct* dapat bekerja secara efektif dan andal. Gambar 2 merupakan desain skematik dan *PCB board* dari alat.



**Gambar 2.** Desain Skematik dan PCB Alat  
(Sumber : Dokumentasi Pribadi)

### D. Perancangan software

#### 1. Arduino IDE

Dalam pembuatan alat *monitoring online* dan otomatisasi pengoperasian pompa air kabel *duct* gardu induk berbasis arduino IoT ini, penulis menggunakan aplikasi arduino ide untuk menyusun program pada *microcontroller* (ESP 32).

Penyusunan program menggunakan Arduino IDE bertujuan agar ESP 32 dapat terhubung ke

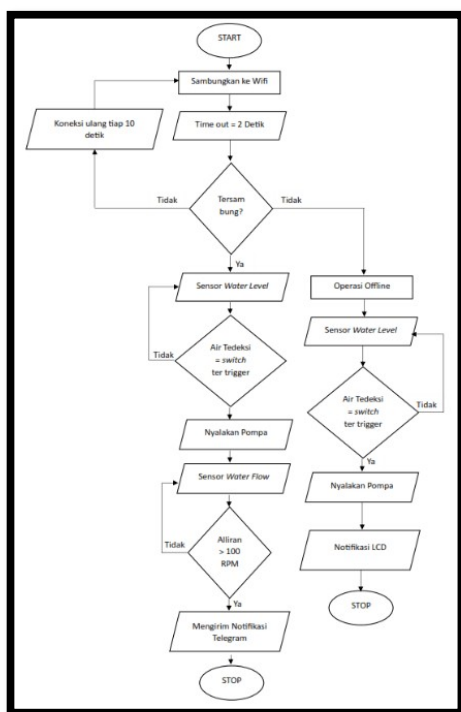
internet yang nantinya digunakan sebagai pusat kontrol dari alat ini dengan menggunakan perantara telegram.

Selain untuk menghubungkan ESP 32 dengan internet, penggunaan Arduino IDE juga bertujuan agar *hardware* yang digunakan dalam pembuatan alat ini (sensor) dapat saling terintegrasi sehingga dapat mencapai kinerja alat yang maksimal.

## 2. Aplikasi telegram

Data bacaan dari sensor yang digunakan akan ditampilkan dalam aplikasi Telegram dan akan dikirim secara *realtime* atau sesuai dengan permintaan dari pengguna. Selain sebagai penampil informasi, aplikasi Telegram ini juga bisa digunakan untuk mengatur *ON/OFF* dari pompa air secara jarak jauh tanpa harus datang ke lokasi dari pompa air.

### E. Flow chart



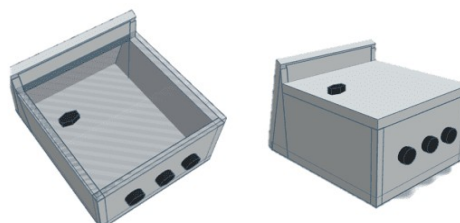
**Gambar 4. Flowchart**  
(Sumber : Dokumentasi Pribadi)

Saat alat distart, maka ESP 32 akan mencoba untuk menyambungkan ke jaringan *wifi* maupun jaringan internet yang sudah disediakan dan diprogramkan dalam ESP 32, ini akan terus

berlanjut hingga ESP 32 sudah tersambung kedalam jaringan internet tersebut. Setelah ESP 32 tersambung internet, kemudian sensor sensor dapat bekerja sesuai dengan fungsi yang sudah diprogramkan dan juga ESP 32 dapat mengirim notifikasi ke Telegram agar dapat dipantau oleh pengguna alat. Pada sensor *water level*, ketika sensor sudah mendeteksi adanya air dalam kabel *duct*, maka sensor akan mengirimkan perintah ke ESP 32 kemudian ESP 32 akan memerintahkan pompa untuk bekerja. Setelah pompa bekerja kemudian sensor *water flow* akan mendeteksi adakah aliran air pada bagian *outlet* dari pompa, jika sensor membaca adanya aliran air, maka sensor akan mengirimkan sinyal ke ESP 32 kemudian ESP 32 akan mengirimkan notifikasi jika pompa sudah bekerja. Pengiriman notifikasi kerja pompa akan ditunda selama 1 menit untuk memastikan benar benar ada air yang mengalir pada bagian *outlet* pompa. Notifikasi akan dikirim 5 menit sekali agar pengguna dapat memantau kinerja dari pompa yang digunakan untuk menyedot air dalam kabel *duct*.

### F. Perancangan mekanik

Untuk mekanik dari alat yang dibuat ini hanya menggunakan box mekanik yang digunakan sebagai tempat atau rumah untuk *hardware* kontrol (ESP 32) dan beberapa sumber tegangan agar terlindung dari percikan air ataupun udara lembab yang dapat mengakibatkan malfungsi dari alat yang dibuat.



**Gambar 3. Box Mekanik**  
(Sumber : Dokumentasi Pribadi)

Gambar 4 merupakan bentuk dari boks mekanik yang digunakan sebagai wadah untuk melindungi *hardware* dari gangguan dari luar, seperti percikan air, udara lembab dan lain sebagainya.

### III. HASIL DAN PEMBAHASAN

#### A. Pemrograman alat

Pemrograman alat dilakukan dengan menggunakan Arduino IDE 1.8.13. *Microcontroller* yang digunakan pada alat ini adalah ESP32 yang digabungkan dengan sensor *water level* dan *water flow* dan disambungkan dengan Telegram agar dapat dipantau keadaan alat darimanapun dan secara jarak jauh.

```
1 ///////////////////////////////////////////////////////////////////
2 #include "CTBot.h";
3 CTBot mybot;
4 String ssid = "Pos satpam";
5 String pass = "semeru.27";
6 ///////////////////////////////////////////////////////////////////
7 String token = "8056867904:AAFyEHEvmt159epAU3WH-bSprvD7Sc5UxI";
8 const int id = 5302198753;
9 ///////////////////////////////////////////////////////////////////
10 const int Relay_Motor =14;
11 const int Sensor_Air =13;
12 int Sens_State = 0;
13 ///////////////////////////////////////////////////////////////////
```

**Gambar 5.** Library Koneksi Internet dan Telegram

(Sumber : Dokumentasi Pribadi)

Untuk menyambungkan *microcontroller* dengan Telegram dan *wifi* atau jaringan internet diharuskan menggunakan *library* khusus yaitu *CTBot.h* yang nantinya dapat membuat *microcontroller* dapat disambungkan dengan jaringan internet dan juga *BOT* Telegram yang sudah dibuat sebagai *user interface* antara pengguna dan alat.

```
14 #include <Wire.h>
15 #include <LiquidCrystal_I2C.h>
16 // Set the LCD address to 0x27 for a 16 chars and 2 line display
17 LiquidCrystal_I2C lcd(0x27, 16, 2);
18 ///////////////////////////////////////////////////////////////////
19 unsigned long lastflash;
20 int RPM;
```

**Gambar 6.** Library LCD dan Deklarasi Sensor Water Flow

(Sumber : Dokumentasi Pribadi)

Untuk menggunakan LCD pada alat yang deprogram menggunakan Arduino harus menggunakan *library* *wire.h* dan juga *LiquidCrystal\_I2C.h* agar *microcontroller* yang diprogram dapat terhubung dengan LCD yang digunakan. Kemudian dilakukan pendeklarasian pembacaan sensor *water flow* sebagai *integer* agar dapat diolah dari hasil pembacaan analog menjadi digital sehingga dapat ditampilkan pada LCD.

```
25 void setup() {
26     // put your setup code here, to run once:
27     lcd.begin();
28     lcd.setCursor(0,0);
29     lcd.print("telegram Aktiv");
30     lcd.print(" ");
31
32     pinMode(Sensor_Air, INPUT);
33     Serial.begin(9600);
34     pinMode(Relay_Motor, OUTPUT);
35     Serial.println("Memulai telegram bot");
36
37     mybot.wifiConnect(ssid, pass);
38
39     mybot.setTelegramToken(token);
40
41     if(mybot.testConnection())
42         Serial.println("koneksi berhasil");
43     else
44         Serial.println("Koneksi Gagal ");
45 }
```

**Gambar 7.** Pendeklarasian I/O, Koneksi Internet dan Telegram  
(Sumber : Dokumentasi Pribadi)

Gambar 7 menunjukkan fungsi yang digunakan untuk pendeklarasian mana yang digunakan sebagai *input* dari alat dan *output* dari alat, mendeklarasikan *ssid* dan *pass* yang digunakan untuk menyambungkan *microcontroller* dengan jaringan internet, token dari BOT Telegram yang akan digunakan dan juga menunjukkan apakah alat sudah terkoneksi dengan jaringan internet atau belum.

```
58 TMessage msg;
59 if(mybot.getNewMessage(msg)){
60     Serial.println("Pesan masuk : "+msg.text);
61     String pesan = msg.text;
62     if(pesan=="cek"){
63         //digitalWrite(Relay_Motor, LOW); //led nodemcu menyala
64         mybot.sendMessage(id, "Telegram Aktiv");
65         mybot.sendMessage(msg.sender.id, "Telegram Aktiv");
66         lcd.setCursor(0,0);
67         lcd.print("telegram bot Aktiv");
68         lcd.print(" ");
69
70     } else if (pesan=="status"){
71         //digitalWrite(Relay_Motor, HIGH); //led nodemcu mati
72         mybot.sendMessage(id, "Status Telegram Aktiv");
73         mybot.sendMessage(msg.sender.id, "Status Telegram Aktiv");
```

**Gambar 8.** Program untuk Mengirim Pesan ke Telegram

(Sumber : Dokumentasi Pribadi)

Gambar 8 menunjukkan program masuk dalam *void loop* atau fungsi berulang, dimana dalam fungsi ini dibuat program yang akan terus menerus diolah oleh *microcontroller* pada saat alat dinyalakan. Agar pesan yang dikirim dapat dibaca oleh *microcontroller* terlebih dahulu dilakukan deklarasi *string* berupa "pesan = msg.text", hal ini dilakukan agar pesan yang masuk dibaca sebagai *string* dan

dapat dibaca oleh *microcontroller* sehingga dapat melakukan balasan dengan mengirimkan perintah pada BOT Telegram dengan jawaban yang sudah diprogram sebelumnya.

```
77 Sens_State = digitalRead(Sensor_Air); //Pembacaan Sensor Air
78 if (Sens_State == HIGH) {
79     // turn LED on:
80     digitalWrite(Relay_Motor, HIGH); //led nodemcu menyala
81     //mybot.sendMessage(id, "Pompa off");
82     //mybot.sendMessage(msg.sender.id, "Pompa On");
83     lcd.setCursor(0,0);
84     lcd.print("Pompa On");
85     lcd.print(" ");
86 } else {
87     // turn LED off:
88     digitalWrite(Relay_Motor, LOW); //led nodemcu mati
89     //mybot.sendMessage(id, "Pompa on");
90     //mybot.sendMessage(msg.sender.id, "Pompa Off");
91     lcd.setCursor(0,0);
92     lcd.print("Pompa Off");
93     lcd.print(" ");
94 }
95
```

**Gambar 9.** Pembacaan Sensor *Water Level*  
(Sumber : Dokumentasi Pribadi)

Dalam program yang dibuat, *microcontroller* akan mengubah *relay* menjadi NC apabila *Sens\_State* membaca sinyal *high* atau sensor *water level* telah mencapai *level* tertentu, kemudian jika keadaan muka air turun maka sinyal *high* dari sensor *water level* akan berubah menjadi *low* dan membuat *relay* berubah kembali menjadi NO dan pompa akan mati.

```
96 int wings= 8;
97 int RPMnew = RPM/wings;
98
99 if (RPM >= 100) {
100     // turn LED on:
101     mybot.sendMessage(msg.sender.id, "Pompa sudah On");
102 } else {
103     // turn LED off:
104     mybot.sendMessage(msg.sender.id, "Pompa Sudah Off");
105 }
106 }
```

**Gambar 11.** Program Pembacaan Sensor *Water flow*  
(Sumber : Dokumentasi Pribadi)

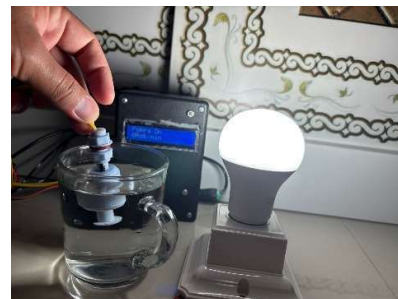
Dalam program ini dilakukan deklarasi *wings* yang merupakan jumlah bilah dalam turbin yang ada dalam sensor *water flow* kemudian deklarasi *RPMnew* yang merupakan hasil dari rumus konversi pembacaan analog dari sensor *water flow* menjadi data digital sehingga dapat dibaca oleh mikrokontroler.

```
108 void Baca_RPM()
109 {
110     noInterrupts();
111     int wings= 8;
112     int RPMnew = RPM/wings;
113     Serial.print(RPMnew);
114     Serial.print(" Rot/sec :"); //Revolutions per second
115     //lcd.clear();
116     //lcd.print(RPMnew);
117     //lcd.print("Rot/sec");
118     Serial.print((RPMnew*60));
119     Serial.println("Rot/min."); //Revolutions per minute
120     lcd.setCursor(0, 1);
121     lcd.print(RPMnew*60);
122     lcd.print("Rot/min");
123     lcd.print(" ");
124     RPM=0;
125     interrupts();
126     delay(1000); //1 second.
127 }
```

**Gambar 12.** Konversi Pembacaan Sensor *Water flow*  
(Sumber : Dokumentasi Pribadi)

Gambar 11 menunjukkan fungsi *Baca\_RPM* dimana dalam fungsi ini dibuat program khusus yang digunakan untuk mengkonversi hasil pembacaan dari sensor *water flow* menjadi data digital sehingga dapat dibaca oleh mikrokontroler dan ditampilkan dalam layar LCD dan digunakan untuk mengirim pesan keadaan pompa oleh BOT Telegram.

#### B. Pengujian sensor water level



**Gambar 10.** Pengujian Sensor *Water Level*  
(Sumber : Dokumentasi Pribadi)

Gambar 12 menunjukkan proses pengujian dari sensor *water level* yang digunakan. Pengujian dilakukan dengan cara mencelupkan sensor *water level* pada gelas berisi air sebagai pengganti dari bak kontrol pada rumah pompa air. Dari hasil pengujian yang dilakukan, sensor *water level* baru memberikan nilai *HIGH* ketika air sudah menaikan pelampung pada level 8 mm. Pada pengujian ini dilakukan penggantian aktuator dari alat yang dibuat, dari pompa air menjadi lampu *LED*. Penggantian aktuator menjadi lamput *LED*

dimaksudkan untuk mengetahui apakah *relay* yang digunakan pada alat ini dapat berfungsi dengan baik atau tidak untuk mengalirkan listrik AC yang digunakan sebagai sumber tegangan dari pompa air.

Berdasarkan hasil pengujian yang telah dilakukan, sistem monitoring dan otomatisasi pengoperasian pompa air kabel duct berbasis *IoT* dapat bekerja sesuai dengan perancangan. Pengujian sensor *water level* menunjukkan bahwa sensor mampu mendeteksi kenaikan muka air dengan baik, ditandai dengan perubahan kondisi *relay* dari *OFF* menjadi *ON* ketika ketinggian air mencapai 8 mm. Hasil ini sejalan dengan penelitian sebelumnya yang menyatakan bahwa sensor *water level* efektif digunakan sebagai pemicu kerja pompa pada sistem monitoring air berbasis *IoT* [1], [2].

**Tabel 1.** Hasil Pengujian Sensor *Water level*

| No. | Level Air | Kondisi Relai |
|-----|-----------|---------------|
| 1.  | 1 mm      | <i>Off</i>    |
| 2.  | 2 mm      | <i>Off</i>    |
| 3.  | 3 mm      | <i>Off</i>    |
| 4.  | 4 mm      | <i>Off</i>    |
| 5.  | 5 mm      | <i>Off</i>    |
| 6.  | 6 mm      | <i>Off</i>    |
| 7.  | 7 mm      | <i>Off</i>    |
| 8.  | 8 mm      | <i>On</i>     |

### C. Pengujian sensor *water flow*

Pengujian sensor *water flow* untuk mengukur kecepatan putaran turbin dari sensor *water flow* dilakukan dengan cara membandingkan hasil pembacaan dari sensor *water flow* dengan hasil pembacaan dari *digital tachometer* seperti pada Gambar 13

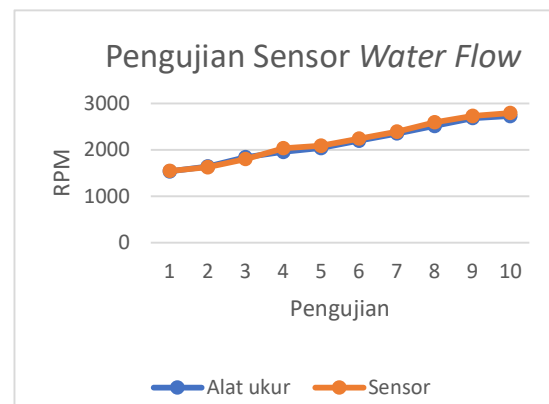
Kemudian dilakukan penghitungan *error* untuk mengerahui akurasi dari sensor *water flow* dan didapatkan data penghitungan pada Tabel 2.

**Tabel 2.** Pengujian sensor *water flow*

| No. | Alat Ukur (RPM) | Sensor <i>Water flow</i> (RPM) | Error (%) |
|-----|-----------------|--------------------------------|-----------|
| 1.  | 1538,4          | 1548,0                         | 0,63 %    |
| 2.  | 1643,2          | 1626,0                         | 1,05 %    |
| 3.  | 1844,7          | 1806,0                         | 2,1 %     |
| 4.  | 1962,4          | 2040,0                         | 3,9 %     |
| 5.  | 2045,7          | 2089,9                         | 2,1%      |

| No. | Alat Ukur (RPM) | Sensor <i>Water flow</i> (RPM) | Error (%) |
|-----|-----------------|--------------------------------|-----------|
| 6.  | 2198,5          | 2246,0                         | 2%        |
| 7.  | 2356,1          | 2395,8                         | 1,6%      |
| 8.  | 2519,7          | 2597,3                         | 2,9%      |
| 9.  | 2688,4          | 2735,9                         | 1,7%      |
| 10. | 2730,8          | 2798,8                         | 2,4%      |

Dari Tabel 2 dapat dilihat hasil pengujian dari sensor *water flow*. Pengujian sensor *water flow* dilakukan dengan membandingkan hasil pembacaan sensor dengan alat ukur digital *tachometer*. Dari sepuluh kali pengujian, diperoleh nilai *error* rata-rata sebesar 2,04% dengan standar deviasi sebesar 0,92%. Interval kepercayaan 95% menunjukkan bahwa nilai *error* berada pada rentang 1,38% hingga 2,70%. Nilai *error* tersebut masih berada di bawah batas toleransi 5%, sehingga sensor *water flow* dapat dikatakan memiliki akurasi yang baik dan layak digunakan untuk memantau kinerja pompa air. Hasil ini konsisten dengan penelitian sebelumnya yang menyatakan bahwa sensor *water flow* berbasis hall effect memiliki tingkat akurasi yang baik untuk aplikasi monitoring aliran air [9], [10], [11]. Dari sepuluh hasil pengujian, data kemudian dimasukkan kedalam grafik pembandingan antara hasil pembacaan sensor *water level* dengan pembacaan alat ukur *digital tachometer*. Untuk lebih memudahkan pembaca, hasil pengujian penulis masukan kedalam Grafik 1.



**Grafik 1.** Pengujian Sensor *Water Flow*

### C. Pengujian sistem alat



**Gambar 13.** Pengujian Sensor *Water Flow*  
(Sumber : Dokumentasi Pribadi)

Pengujian sistem secara keseluruhan menunjukkan bahwa seluruh komponen, mulai dari sensor, mikrokontroler, *relay*, hingga aplikasi Telegram, dapat terintegrasi dengan baik. Sistem mampu mengirimkan notifikasi kondisi pompa secara *real-time* serta memungkinkan pengguna melakukan monitoring dan pengendalian jarak jauh. Hasil ini mendukung penelitian terdahulu yang menyatakan bahwa pemanfaatan Telegram sebagai media monitoring IoT memberikan kemudahan dan keandalan dalam pengawasan sistem air [5], [7]. Hasil pengujian ini dapat dilihat pada Gambar 14.



**Gambar 14.** Pengujian Sistem Alat  
(Sumber : Nabil, 2025)

### IV. KESIMPULAN DAN SARAN

Adapun kesimpulan dari pembuatan alat ini adalah pompa air kabel *duct* dapat dioperasikan dan dipantau secara jarak jauh melalui aplikasi Telegram tanpa perlu melakukan pengecekan secara langsung ke rumah pompa. Keadaan kabel *duct* juga dapat diketahui secara *online* melalui pengecekan pada aplikasi Telegram. Sensor yang digunakan mampu terhubung dengan baik dengan *microcontroller* dan terhubung ke Telegram, sehingga proses monitoring dan pengoperasian dapat dilakukan dari jarak jauh. Berdasarkan hasil pengujian, sensor *water level* dan *water flow* bekerja

dengan baik, ini ditunjukkan dengan nilai rata rata *error* yang rendah, yaitu sebesar 2,04% pada sensor *water flow*, serta *switch* pada sensor *water level* yang mulai bekerja pada ketinggian 8mm. Dengan demikian, monitoring pompa air kabel *duct* dapat dilakukan kapan saja dan dimana saja hanya dengan mengecek status pompa melalui aplikasi Telegram.

Sebagai saran untuk pengembangan alat ini adalah dapat dilakukan pengujian *long-term* pada kondisi nyata sesuai dengan keadaan pengoprasian alat untuk mengetahui reliabilitas (kesalahan pengukuran atau pembacaan sensor saat operasi jangka panjang, berapa kali alat terputus dari koneksi internet dan juga keefektifan dari program penyambungan otomatis apabila alat terputus dari jaringan internet).

### UCAPAN TERIMAKASIH

Puji syukur kehadiran Allah SWT atas berkat Rahmat, hidayah dan juga Taufiq-Nya sehingga penulis dapat menyelesaikan makalah ini dengan baik dan juga tepat waktu. Makalah ini telah penulis susun semaksimal mungkin dan sebaik mungkin, tentunya juga dengan bantuan dari banyak pihak sehingga dapat membantu kelancaran dalam pembuatan makalah ini.

### REFERENSI

- [1] F. Hardiyanto and I. Imaduddin, "Perancangan *Water level* Sensor Module Menggunakan MQTT Over WiFi dan Serial Communication," *Journal of Artificial Intelligence and Digital Business (RIGGS)*, vol. 4, no. 3, pp. 3030–3034, 2025, doi: 10.31004/riggs.v4i3.2390.
- [2] H. Darmanto, L. Lamsadi, dan H. Asrul, "Monitoring Ketinggian Air Tandon Berbasis IoT Dengan ESP32 Melalui Website," *JUSTER: Jurnal Sains dan Terapan*, vol. 4, no. 2, pp. 67–73, May 2025, doi: 10.57218/juster.v4i2.1507.
- [3] R. Wicaksono, D. Riyani, R. Widayanto, and E. N. Budisusila, "Implementasi IoT dalam Pengaturan Sistem untuk Peningkatan Kualitas Distribusi Air Pamsimas di Gunungpati," *Uranus: Jurnal Ilmiah Teknik Elektro, Sains*

- dan Informatika*, vol. 3, no. 3, pp. 37–44, Sep. 2025, doi: 10.61132/uranus.v3i3.1039.
- [4] M. R. Saputra and D. Kurniawan, “Rancang Bangun Sistem Smart *Water level* Berbasis Internet of Things (IoT),” *Jurnal Teknologi dan Sistem Tertanam*, vol. 4, no. 2, pp. 98–107, 2025. doi: 10.56211/helloworld.v4i2.940
- [5] J. Novelliani dan W. Wildian, “Sistem Monitoring dan Notifikasi Penggunaan Air PDAM Berbasis Arduino dan Telegram,” *Jurnal Fisika Unand*, vol. 10, no. 2, pp. 219–224, 2021, doi: 10.25077/jfu.10.2.219-224.2021.
- [6] Muhammad Iqbal R; Setia Budi A, “Mekanisme Penyediaan Layanan Pada Perangkat *Smart Home* Berbasis ESP 32 Menggunakan Pendekatan,” *JPTIHK*, vol. 8, no. 4, 2024.
- [7] D. R. Tisna, T. Maharani, dan K. T. Nugroho, “Pemanfaatan *Chatbot* Telegram Untuk Monitorinf dan Kontrol Kualitas Air Menggunakan ESP 32,” *JIPi (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 9, no. 3, hlm. 1292–1306, Agu 2024, doi: 10.29100/jipi.v9i3.5329.
- [8] H. Arief Kusuma, M. Irfan, dan S. Nugraha, “Pengembangan Instrumen Pengukuran Aliran Air Berbasis Internet of Things (IoT),” *Jurnal Elektro dan Mesin Terapan*, vol. 7, no. 1, 2021, doi: 10.35143/elementer.v7i1.4627.
- [9] A. Adela Rashad, Ulikaryani, B. Aji Girawan, R. Ariawan, J. Sodikin, dan H. D. Hastuti, “Uji Fungsi dan Kalibrasi Sensor *Water flow* YF-S201 Berbasis Arduino Uno pada Mesin Penjernih Air Sungai,” *Jurnal Teknik Mesin, Elektro dan Ilmu Komputer*, vol. 3, no. 2, pp. 1–9, Jul. 2023, doi: 10.55606/teknik.v3i2.1850.
- [10] Juma’in Juma’in, Z. Zaenudin, M. M. Efendi, dan L. Delsi Samsumar, “Perancangan Sistem Monitoring Meteran Air Menggunakan Sensor *Water flow* pada Rumah Tangga,” *Journal of Data Analytics, Information, and Computer Science*, vol. 1, no. 4, 2022, doi: 10.70248/jdaics.v1i4.1305.
- [11] R. Husniyah dan P. Prabowo, “Development of Arduino-Based Flowmeter Sensor Props for Measuring *Water flow* in Dynamic Fluid Materials,” *Prisma Sains: Jurnal Pengkajian Ilmu dan Pembelajaran Matematika dan IPA*, vol. 10, no. 3, pp. 427–442, 2022, doi: 10.33394/j-ps.v10i3.5192.