

BAB IV

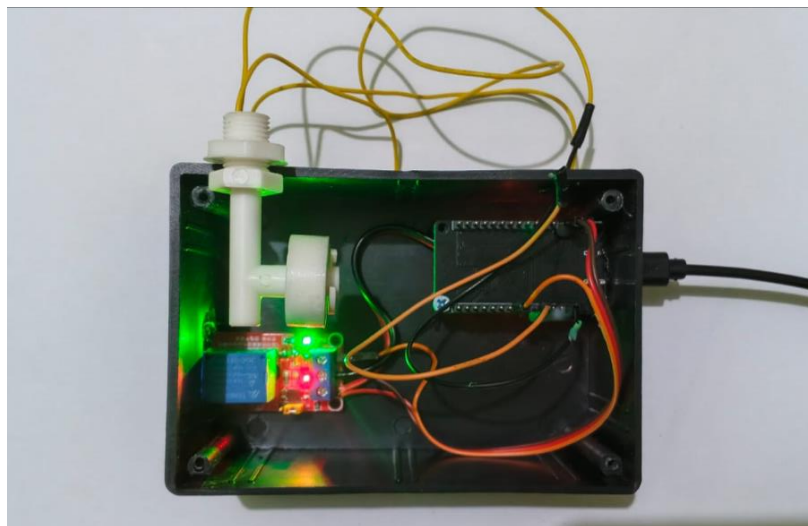
HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

Tahap implementasi bertujuan untuk memastikan bahwa sistem *Smart Water Control System* berbasis IoT dapat bekerja sesuai dengan desain yang telah dirancang, yaitu mampu mengontrol pompa air melalui *Whatsapp* serta mematikan pompa secara otomatis ketika tandon penuh menggunakan *Float Switch*.

4.1.1 Realisasi Perangkat Keras

Pada tahap ini dilakukan perakitan seluruh komponen perangkat keras sesuai dengan perancangan yang telah dijelaskan sebelumnya. Proses realisasi meliputi pemasangan ESP32, modul *relay*, pompa air, serta pengkabelan sesuai dengan diagram rangkaian yang telah dirancang sebelumnya. Perangkat keras dirakit menjadi satu sistem yang terintegrasi agar dapat bekerja secara optimal.



Gambar 4. 1 Realisasi *Smart Water Control System*

Berdasarkan Gambar 4.1, terlihat bahwa papan mikrokontroler ESP32

terhubung dengan modul *relay 1 channel* yang berfungsi sebagai saklar elektronik untuk mengendalikan pompa air. Kabel fase dari sumber listrik dihubungkan ke terminal *COM* dan *NO* pada *relay*, sehingga ketika *relay* aktif, arus listrik akan mengalir menuju pompa air dan pompa menyala.

ESP32 mendapatkan catu daya melalui adaptor USB, sedangkan modul *relay* terhubung ke pin digital ESP32 sebagai pin kendali. Rangkaian ini dirancang agar aman dan stabil saat digunakan dalam pengoperasian sehari-hari.

4.1.2 Realisasi Instalasi *Float Switch*

Pada tahap ini dilakukan pemasangan sensor *Float switch* sebagai komponen utama dalam sistem otomatisasi pengisian air. *Float switch* dipasang pada bagian atas tandon air dan berfungsi sebagai pendeteksi batas maksimum ketinggian air. Pemasangan sensor ini sangat penting karena menjadi indikator utama untuk menghentikan kerja pompa secara otomatis ketika air telah penuh.



Gambar 4. 2 Pemasangan *Float switch* pada Tandon Air

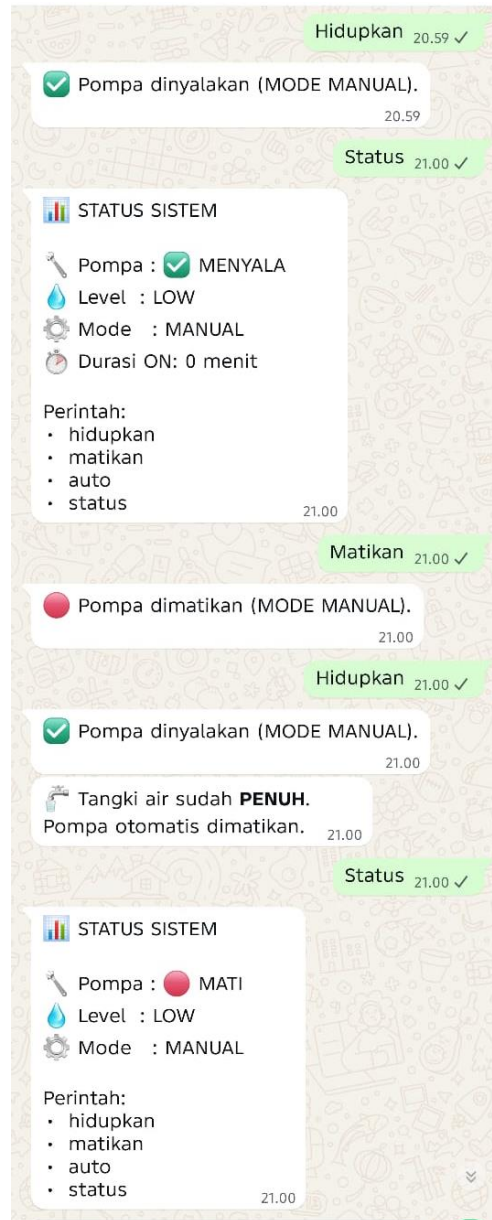
Berdasarkan Gambar 4.2, terlihat bahwa *Float switch* dipasang pada posisi horizontal di dinding bagian atas tandon air. Sensor ini diatur sedemikian rupa sehingga ketika permukaan air menyentuh pelampung, maka mekanisme saklar di dalam sensor akan berubah status menjadi *HIGH*.

Perubahan status tersebut dibaca oleh ESP32 melalui pin input digital yang telah dikonfigurasi pada program. Ketika nilai *HIGH* terdeteksi, sistem akan mengeksekusi perintah untuk mematikan *relay* sehingga pompa air berhenti bekerja. Dengan konfigurasi ini, sistem mampu mencegah terjadinya *overflow* atau luapan air pada tandon.

Penggunaan satu buah *Float switch* pada bagian atas tandon dipilih karena sistem ini dirancang hanya untuk mengontrol kondisi penuh (*full detection*). Dengan demikian, pompa air hanya akan berhenti ketika batas maksimum tercapai, sedangkan proses penyalaan tetap dikendalikan melalui perintah *Whatsapp*.

4.1.3 Implementasi Sistem Kendali dan *Monitoring* Melalui *Whatsapp*

Pada tahap implementasi ini dilakukan pengujian sistem kendali dan *monitoring* pompa air menggunakan aplikasi *Whatsapp* sebagai media komunikasi antara pengguna dan perangkat ESP32. Sistem diintegrasikan menggunakan layanan API sehingga pesan yang dikirim oleh pengguna dapat diterima, diproses, dan dieksekusi oleh mikrokontroler secara *real-time* melalui jaringan internet.



Gambar 4.3 Tampilan Pengujian Sistem Melalui *Whatsapp*

Berdasarkan Gambar 4.3, terlihat beberapa perintah yang dapat digunakan oleh pengguna, yaitu:

- hidupkan → untuk menyalakan pompa air
- matikan → untuk mematikan pompa air
- auto → untuk mengaktifkan mode otomatis
- status → untuk menampilkan kondisi sistem secara keseluruhan

Ketika pengguna mengirimkan perintah “Hidupkan”, sistem memberikan

balasan berupa notifikasi “*Pompa dinyalakan (MODE MANUAL)*”. Hal ini menunjukkan bahwa ESP32 berhasil menerima dan memproses perintah, kemudian mengaktifkan *relay* sehingga pompa air menyala.

Selanjutnya, ketika pengguna mengirimkan perintah “Status”, sistem menampilkan informasi lengkap mengenai kondisi perangkat, antara lain:

- Status pompa (MENYALA atau MATI)
- Level air (*LOW* atau kondisi lainnya)
- Mode sistem (MANUAL atau *AUTO*)
- Durasi waktu pompa menyala

Fitur ini memungkinkan pengguna melakukan *monitoring* jarak jauh tanpa harus memeriksa langsung kondisi tandon.

Apabila pengguna mengirimkan perintah “Matikan”, sistem merespons dengan pesan “*Pompa dimatikan (MODE MANUAL)*” dan *relay* akan dinonaktifkan sehingga pompa berhenti bekerja.

Selain kendali manual, sistem juga memiliki fungsi otomatis. Ketika air dalam tandon mencapai batas maksimal dan *Float switch* mendeteksi kondisi penuh, sistem akan mengirimkan notifikasi:

“*Tangki air sudah PENUH. Pompa otomatis dimatikan.*”

Pesan ini menunjukkan bahwa sistem berhasil menjalankan fungsi proteksi otomatis untuk mencegah luapan air dan pemborosan energi listrik.

Dengan demikian, implementasi kendali dan *monitoring* melalui *Whatsapp* menunjukkan bahwa sistem mampu bekerja dua arah, yaitu menerima perintah dari pengguna dan memberikan umpan balik berupa informasi status secara *real-time*.

4.1.4 Implementasi Program pada ESP32

Pada tahap ini dilakukan implementasi perangkat lunak pada mikrokontroler ESP32 menggunakan Arduino IDE. Program dirancang untuk mengintegrasikan koneksi WiFi, komunikasi MQTT, kendali *relay* sebagai penggerak pompa air, serta pembacaan sensor *Float switch* sebagai pendeteksi kondisi penuh pada tandon air.

Secara umum, alur program dimulai dari proses inialisasi pin dan koneksi jaringan, kemudian sistem menunggu perintah kendali yang dikirim melalui broker MQTT. Selain itu, sistem juga secara terus-menerus membaca status *Float switch* untuk mendeteksi perubahan level air.

1. Inialisasi Sistem dan Konfigurasi Jaringan

Berikut merupakan bagian program yang berfungsi untuk mengatur parameter koneksi WiFi, broker MQTT, serta topik komunikasi yang digunakan.

```

5  const char* ssid = "smartwater";
6  const char* password = "12345678";
7
8  // =====
9  const char* mqtt_server = "broker.hivemq.com";
10
11 const char* topicCommand = "water/tian/pump/command";
12 const char* topicStatus = "water/tian/pump/status";
13 const char* topicLevel = "water/tian/tank/level";
--

```

Gambar 4. 4 Program Inialisasi Sistem dan Konfigurasi Jaringan

Kode di atas menunjukkan bahwa sistem menggunakan jaringan WiFi dengan SSID tertentu serta terhubung ke broker MQTT publik. Terdapat tiga topik utama yang digunakan, yaitu:

- *topicCommand* untuk menerima perintah *ON/OFF*
- *topicStatus* untuk mengirim status pompa

- *topicLevel* untuk mengirim status level air

Konfigurasi pin perangkat keras ditunjukkan pada bagian berikut:

```

</> C++

#define RELAY_PIN 5
#define FLOAT_PIN 4

pinMode(RELAY_PIN, OUTPUT);
pinMode(FLOAT_PIN, INPUT_PULLUP);

digitalWrite(RELAY_PIN, HIGH);

```

Gambar 4. 5 Program Konfigurasi Pin

Program akan terus melakukan pengecekan hingga status koneksi menunjukkan berhasil terhubung. Tahapan ini sesuai dengan diagram alir sistem yang telah dijelaskan pada Bab III.

2. Implementasi Logika Kendali Pompa (*ON/OFF*)

Bagian berikut merupakan inti dari sistem kendali pompa air. Program akan membaca pesan yang masuk melalui topik perintah, kemudian mengeksekusi logika sesuai dengan isi pesan.

```

if (message == "ON") {
    digitalWrite(RELAY_PIN, LOW);
    pumpState = true;
    client.publish(topicStatus, "ON");
}

if (message == "OFF") {
    digitalWrite(RELAY_PIN, HIGH);
    pumpState = false;
    client.publish(topicStatus, "OFF");
}

```

Gambar 4. 6 Logika Kendali Poma

Ketika sistem menerima pesan “*ON*”, *relay* akan diaktifkan (aktif *LOW*) sehingga pompa air menyala. Sebaliknya, jika menerima pesan “*OFF*”,

relay dimatikan sehingga pompa berhenti bekerja. Setelah eksekusi perintah, sistem akan mengirimkan status terbaru melalui topik *topicStatus*.

3. Implementasi Pembacaan *Float Switch*

Untuk mendeteksi kondisi tandon penuh, sistem membaca status *Float switch* secara terus-menerus di dalam fungsi *loop*.

```
bool floatState = digitalRead(FLOAT_PIN);

if (floatState != lastFloatState) {
  delay(200);

  if (floatState == LOW) {
    client.publish(topicLevel, "FULL");
  } else {
    client.publish(topicLevel, "LOW");
  }

  lastFloatState = floatState;
}
```

Gambar 4. 7 Program Pembacaan *Float Switch*

Program melakukan pengecekan perubahan status sensor. Jika terjadi perubahan, sistem memberikan jeda (*debounce*) selama 200 ms untuk menghindari pembacaan yang tidak stabil.

Apabila *Float switch* bernilai *LOW*, maka sistem mengirimkan status “FULL” yang menandakan air telah mencapai batas maksimal. Sebaliknya, jika bernilai *HIGH*, sistem mengirimkan status “LOW” yang menandakan air belum penuh.

4.2 Pengujian Sistem

Pengujian sistem dilakukan untuk mengetahui kinerja dan stabilitas *Smart Water Control System* berbasis IoT dalam kondisi operasional nyata. Pengujian dilaksanakan selama 7 hari, mulai tanggal 2 Maret sampai dengan 9 Maret,

dengan tujuan untuk memastikan sistem mampu bekerja secara konsisten dalam menerima perintah, mengontrol pompa air, membaca sensor *Float Switch*, serta mengirimkan notifikasi status.



Gambar 4. 8 Implementasi Alat

4.2.1 Skenario Pengujian

Tabel Skenario Pengujian Sistem disusun untuk memastikan seluruh fungsi pada *Smart Water Control System* berbasis IoT dapat berjalan sesuai dengan perancangan yang telah dijelaskan pada Bab III. Setiap skenario dirancang untuk menguji komponen utama sistem, baik dari sisi perangkat keras maupun perangkat lunak.

Tabel 4. 1 Skenario Pengujian Sistem

No	Skenario Pengujian	Langkah Pengujian	Output yang Diharapkan
1	Uji Koneksi WiFi	Menghidupkan sistem dan memantau koneksi jaringan	ESP32 terhubung ke WiFi tanpa gagal
2	Uji Koneksi MQTT	Menghubungkan ESP32 ke broker	Status MQTT connected
3	Uji Perintah ON	Mengirim perintah ON melalui sistem	Pompa menyala dan status terkirim
4	Uji Perintah OFF	Mengirim perintah OFF	Pompa mati dan status terkirim
5	Uji <i>Float switch</i> LOW	Air belum menyentuh sensor	Status level <i>LOW</i> terkirim
6	Uji <i>Float switch</i> FULL	Air menyentuh sensor	Status FULL terkirim
7	Uji Stabilitas 24 Jam	Sistem dibiarkan aktif 1 hari penuh	Tidak terjadi <i>error</i> atau restart
8	Uji Respon Waktu	Mengukur delay perintah ON	Respon < 3 detik
9	Uji Konsistensi 7 Hari	<i>Monitoring</i> harian	Sistem bekerja stabil tanpa kegagalan

Skenario pertama dan kedua difokuskan pada pengujian konektivitas, yaitu koneksi WiFi dan koneksi MQTT. Pengujian ini bertujuan untuk memastikan bahwa ESP32 dapat terhubung ke jaringan internet dan broker komunikasi sebelum sistem menjalankan fungsi kendali. Keberhasilan pada tahap ini menjadi dasar bagi pengujian berikutnya, karena sistem IoT sangat bergantung pada stabilitas jaringan.

Skenario ketiga dan keempat menguji fungsi kendali manual pompa air melalui perintah *ON* dan *OFF*. Pengujian dilakukan dengan mengirimkan perintah melalui sistem komunikasi, kemudian diamati apakah *relay* aktif atau nonaktif sesuai perintah serta apakah status berhasil dikirim kembali sebagai umpan balik. Tahap ini bertujuan untuk memastikan logika kendali bekerja dengan benar.

Skenario kelima dan keenam berfokus pada pengujian sensor *Float Switch*.

Pada kondisi air belum penuh, sistem harus membaca status *LOW* dan mengirimkan informasi yang sesuai. Sebaliknya, ketika air mencapai batas maksimal dan menyentuh *Float Switch*, sistem harus mendeteksi kondisi *FULL* dan mengirimkan notifikasi status. Pengujian ini bertujuan untuk memastikan mekanisme proteksi otomatis berjalan dengan baik.

Skenario ketujuh dilakukan untuk menguji stabilitas sistem dalam periode waktu yang lebih panjang, yaitu selama 24 jam tanpa dimatikan. Pengujian ini bertujuan untuk mengetahui apakah sistem mengalami gangguan seperti restart otomatis, kehilangan koneksi, atau *error* komunikasi.

Skenario kedelapan menguji waktu respon sistem terhadap perintah yang dikirim. Parameter ini penting untuk mengetahui seberapa cepat sistem merespons kendali pengguna. Delay yang kecil menunjukkan bahwa sistem memiliki performa komunikasi yang baik.

Terakhir, skenario kesembilan merupakan pengujian konsistensi sistem selama 7 hari berturut-turut. Pengujian ini bertujuan untuk memastikan bahwa sistem tidak hanya bekerja sesaat, tetapi mampu beroperasi secara stabil dalam jangka waktu tertentu sesuai kebutuhan operasional di lingkungan sekolah.

Dengan adanya skenario pengujian yang terstruktur ini, maka evaluasi kinerja sistem dapat dilakukan secara sistematis dan terukur.

4.2.2 Hasil Pengujian Sistem Selama 7 Hari

Pengujian sistem dilakukan selama 7 hari berturut-turut, mulai tanggal 2 Maret sampai dengan 8 Maret. Pengujian dilakukan dalam kondisi operasional normal dengan sistem aktif selama jam penggunaan air di sekolah. Parameter yang diuji meliputi koneksi WiFi, koneksi MQTT, respon perintah *ON/OFF*,

pembacaan *Float Switch*, pengiriman status level air, serta kestabilan sistem secara keseluruhan. Berikut merupakan hasil pengujian harian yang telah dilakukan.

Tabel 4. 2 Hasil Pengujian Sistem Selama 7 Hari

No	Tanggal	Koneksi WiFi	Respon ON/OFF	Pembacaan Float	Notifikasi Status	Delay Respon	Keterangan
1	2 Maret	Berhasil	Berhasil	Berhasil	Berhasil	±1 detik	Sistem normal
2	3 Maret	Berhasil	Berhasil	Berhasil	Berhasil	±1 detik	Sistem normal
3	4 Maret	Berhasil	Berhasil	Berhasil	Berhasil	±1 detik	Sistem normal
4	5 Maret	Berhasil	Berhasil	Berhasil	Berhasil	±1 detik	Sistem normal
5	6 Maret	Berhasil	Berhasil	Berhasil	Berhasil	±1-2 detik	Sistem stabil 24 jam
6	7 Maret	Berhasil	Berhasil	Berhasil	Berhasil	±1 detik	Sistem normal
7	8 Maret	Berhasil	Berhasil	Berhasil	Berhasil	±1 detik	Evaluasi akhir berhasil

Berdasarkan hasil pengujian selama 7 hari, sistem menunjukkan performa yang stabil tanpa mengalami kegagalan fungsi. Seluruh parameter pengujian berhasil dijalankan sesuai dengan perancangan sistem.

Delay respon perintah *ON/OFF* ada diantara 1 sampai 2 detik, yang dipengaruhi oleh kualitas koneksi jaringan internet. Namun demikian, waktu respon tersebut masih dalam batas yang dapat diterima untuk sistem kendali jarak jauh berbasis IoT.

Tidak ditemukan kegagalan koneksi WiFi maupun MQTT selama periode pengujian. Pembacaan *Float switch* juga berjalan dengan baik dan mampu mendeteksi kondisi penuh secara akurat.

4.3 Analisis dan Pembahasan

4.3.1 Analisis Kinerja Sistem

Berdasarkan hasil pengujian yang telah dilakukan selama 7 hari berturut-turut, sistem *Smart Water Control System* berbasis IoT menunjukkan kinerja yang stabil dan konsisten. Selama periode 2 Maret hingga 8 Maret, tidak ditemukan kegagalan koneksi WiFi maupun gangguan komunikasi MQTT.

Sistem mampu menerima perintah *ON* dan *OFF* dengan baik serta mengeksekusinya sesuai dengan logika program. *Relay* aktif ketika menerima perintah *ON* dan nonaktif saat menerima perintah *OFF*. Selain itu, pembacaan *Float switch* juga berjalan normal tanpa kesalahan deteksi.

Tidak ditemukan kondisi *error*, *restart* otomatis, maupun kehilangan koneksi selama sistem diuji dalam kondisi operasional nyata.

4.3.2 Analisis Waktu Respon Sistem

Waktu respon sistem diukur berdasarkan selang waktu antara pengiriman perintah dan eksekusi pompa, serta waktu pengiriman notifikasi ketika tandon penuh.

Berdasarkan hasil pengamatan, rata-rata waktu respon sistem berada pada rentang kurang dari 1 detik sampai 2 detik tergantung kepada kualitas jaringan internet. Respon ini mencakup:

- Waktu penerimaan perintah *ON/OFF*
- Eksekusi *relay*
- Pengiriman status pompa
- Pengiriman notifikasi saat air penuh

Waktu respon yang cepat ini menunjukkan bahwa komunikasi MQTT yang

digunakan memiliki latensi rendah dan koneksi jaringan berada dalam kondisi stabil. Faktor utama yang mempengaruhi waktu respon adalah kualitas jaringan internet. Namun selama pengujian berlangsung, koneksi jaringan berada dalam kondisi baik sehingga delay dapat diminimalkan.

Dengan waktu respon kurang dari 1 detik, sistem dapat dikategorikan responsif dan layak digunakan dalam operasional sehari-hari.

4.3.3 Analisis Sistem Otomatisasi Tandon Air

Fungsi otomatisasi sistem diuji melalui pembacaan *Float switch* pada bagian atas tandon. Ketika air menyentuh sensor, sistem berhasil mendeteksi kondisi penuh dan langsung mengirimkan notifikasi serta menghentikan pompa.

Tidak ditemukan keterlambatan signifikan dalam proses deteksi kondisi penuh. Hal ini menunjukkan bahwa mekanisme pembacaan digital dan logika pemrograman berjalan sesuai dengan perancangan.

Dengan adanya fitur ini, sistem berhasil mencegah terjadinya *overflow* atau luapan air yang berpotensi menyebabkan pemborosan air dan energi listrik.

4.3.4 Evaluasi Sistem Terhadap Tujuan Penelitian

Berdasarkan hasil implementasi dan pengujian, sistem telah memenuhi seluruh tujuan penelitian, yaitu:

1. Berhasil merancang sistem kendali pompa air berbasis IoT menggunakan ESP32.
2. Berhasil mengimplementasikan kontrol jarak jauh melalui sistem komunikasi.
3. Berhasil mematikan pompa secara otomatis saat tandon penuh.
4. Memberikan notifikasi secara *real-time* dengan waktu respon cepat.

Dengan demikian, sistem dapat dinyatakan layak diterapkan di lingkungan sekolah sebagai solusi otomatisasi pengisian air berbasis IoT.