

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Implementasi Sistem

Pada penelitian ini telah berhasil dirancang dan di implementasikan sebuah sistem kendali lampu berbasis *Internet of Things* (IoT) menggunakan media komunikasi *WhatsApp* dan modul LoRa SX1278. Sistem ini memanfaatkan mikrokontroler ESP32 sebagai pengendali utama, serta layanan API dari Twilio dan ThingESP sebagai penghubung antara pengguna dan perangkat.

Sistem terdiri dari dua bagian utama, yaitu:

1. Node Pengirim (*Transmitter*) → ESP32 yang terhubung ke internet dan menerima perintah dari *WhatsApp*
2. Node Penerima (*Receiver*) → ESP32 yang terhubung dengan modul relay untuk mengendalikan lampu

Perintah yang dikirim oleh pengguna melalui *WhatsApp* akan diproses oleh server API, kemudian diteruskan ke ESP32 pengirim. Selanjutnya data dikirim melalui komunikasi LoRa ke ESP32 penerima untuk mengendalikan perangkat.



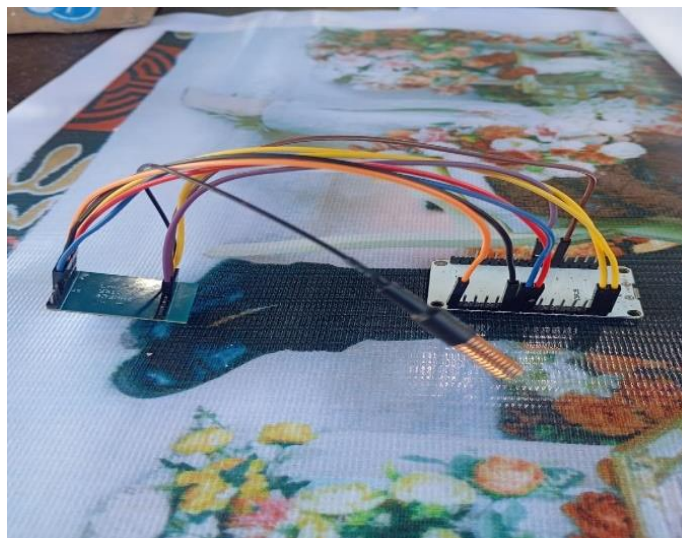
Gambar 4.1 Hasil Implementasi

4.1.1 Node Pengirim (Transmitter)

Node pengirim merupakan bagian sistem yang berfungsi menerima perintah dari pengguna melalui *WhatsApp*, kemudian mengirimkannya ke node penerima menggunakan modul LoRa. Node ini menggunakan ESP32 sebagai mikrokontroler, modul LoRa SX1278 sebagai media komunikasi, koneksi WiFi untuk akses internet, serta layanan API seperti Twilio dan ThingESP sebagai penghubung sistem.

Cara kerjanya dimulai saat pengguna mengirim perintah melalui *WhatsApp*, seperti “P” untuk informasi, “1” untuk menyalakan lampu, dan “2” untuk mematikan lampu. Perintah tersebut diproses oleh API, diteruskan ke ESP32, kemudian dikirim melalui LoRa ke node penerima.

Selain itu, node pengirim juga menerima balasan berupa status lampu (menyala atau mati) dari node penerima, yang kemudian dikirim kembali ke pengguna melalui *WhatsApp* sebagai umpan balik sistem.



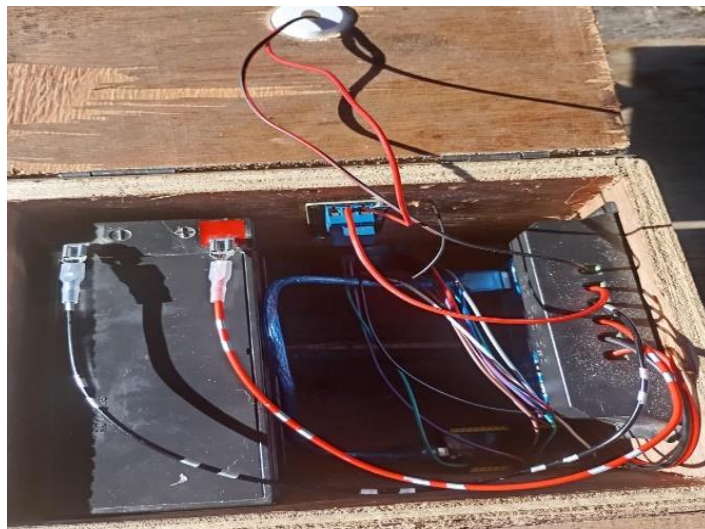
Gambar 4.2 Node Pengirim

4.1.2 Node Penerima (Receiver)

Node penerima merupakan bagian sistem yang berfungsi menerima data dari node pengirim melalui LoRa dan mengeksekusi perintah tersebut pada perangkat output, yaitu relay.

Node ini menggunakan ESP32 yang terhubung dengan modul LoRa SX1278 dan relay. Setelah data diterima, ESP32 akan memproses perintah yang masuk. Jika perintah yang diterima adalah untuk menyalakan lampu, maka relay akan aktif sehingga lampu menyala. Sebaliknya, jika perintah untuk mematikan lampu diterima, maka relay akan nonaktif dan lampu akan mati.

Node penerima kemudian mengirimkan kembali status lampu ke node pengirim sebagai umpan balik sistem.



Gambar 4.3 Node Penerima

4.2 Implementasi Program (Arduino IDE)

Implementasi sistem dilakukan menggunakan Arduino IDE untuk menulis dan mengunggah program ke mikrokontroler ESP32. Program ini memungkinkan

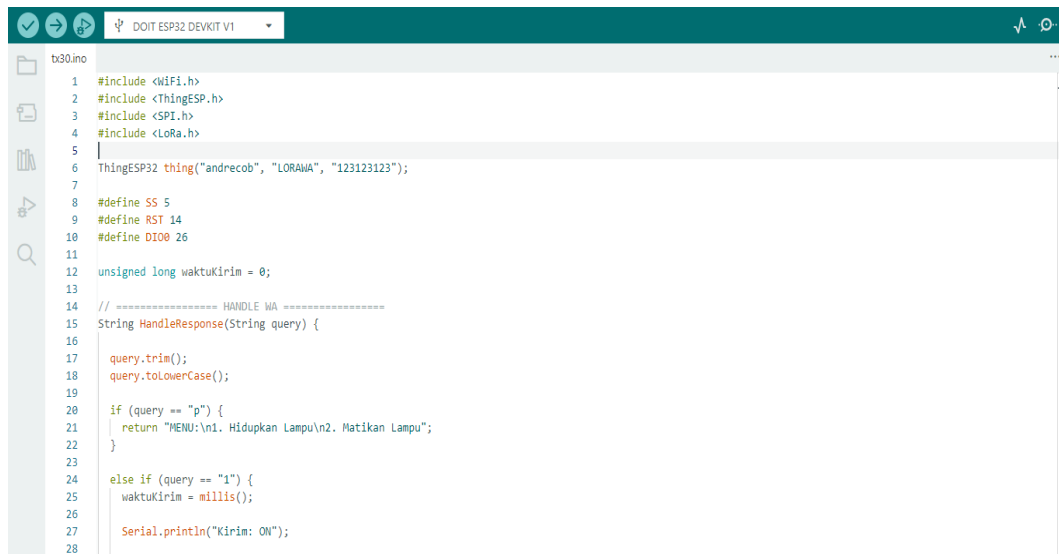
ESP32 terhubung ke jaringan WiFi, menerima perintah dari pengguna melalui *WhatsApp* menggunakan Twilio sebagai bot, serta mengirim dan menerima data ke perangkat lain menggunakan modul LoRa SX1278, dengan ThingESP sebagai server perantara.

Twilio berfungsi sebagai bot WhatsApp yang menerima perintah pengguna, seperti “hidupkan lampu” atau “matikan lampu”, dan mengirimkannya ke ThingESP, yang bertindak sebagai server. ESP32 kemudian mengambil perintah dari ThingESP, mengeksekusi perintah melalui LoRa, dan mengirimkan feedback status kembali ke server, sehingga pengguna dapat melihat kondisi perangkat secara real-time.

Agar program berjalan dengan baik, beberapa library harus di instal di Arduino IDE:

1. WiFi → (bawaan ESP32), untuk koneksi internet.
2. HTTPClient → (bawaan ESP32), untuk komunikasi HTTP dengan ThingESP.
3. SPI → (bawaan Arduino), untuk komunikasi dengan modul LoRa.
4. LoRa by Sandeep Mistry, untuk pengiriman dan penerimaan data jarak jauh.
5. ThingESP, untuk integrasi dengan server yang menampung perintah dan *feedback* dari bot.

Selain itu, tambahkan Board Manager URL berikut agar ESP32 dapat diprogram di Arduino IDE: https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

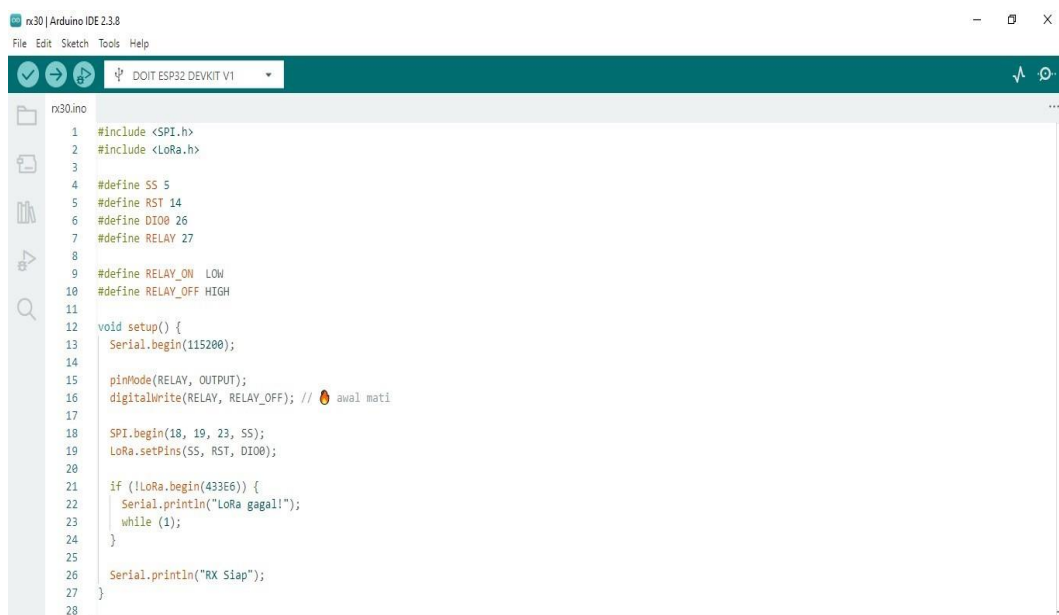


```

1 #include <WiFi.h>
2 #include <ThingESP.h>
3 #include <SPI.h>
4 #include <LoRa.h>
5
6 ThingESP32 thing("andrecob", "LORAMA", "123123123");
7
8 #define SS 5
9 #define RST 14
10 #define DIO0 26
11
12 unsigned long waktuKirim = 0;
13
14 // ===== HANDLE WA =====
15 String HandleResponse(String query) {
16
17   query.trim();
18   query.toLowerCase();
19
20   if (query == "p") {
21     return "MENU:\n1. Hidupkan Lampu\n2. Matikan Lampu";
22   }
23
24   else if (query == "1") {
25     waktuKirim = millis();
26
27     Serial.println("Kirim: ON");
28

```

Gambar 4.4 Implementasi Penerima



```

1 #include <SPI.h>
2 #include <LoRa.h>
3
4 #define SS 5
5 #define RST 14
6 #define DIO0 26
7 #define RELAY 27
8
9 #define RELAY_ON LOW
10 #define RELAY_OFF HIGH
11
12 void setup() {
13   Serial.begin(115200);
14
15   pinMode(RELAY, OUTPUT);
16   digitalWrite(RELAY, RELAY_OFF); // awal mati
17
18   SPI.begin(18, 19, 23, SS);
19   LoRa.setPins(SS, RST, DIO0);
20
21   if (!LoRa.begin(433E6)) {
22     Serial.println("LoRa gagal!");
23     while (1);
24   }
25
26   Serial.println("RX Siap");
27 }
28

```

Gambar 4.5 Implementasi Pengirim

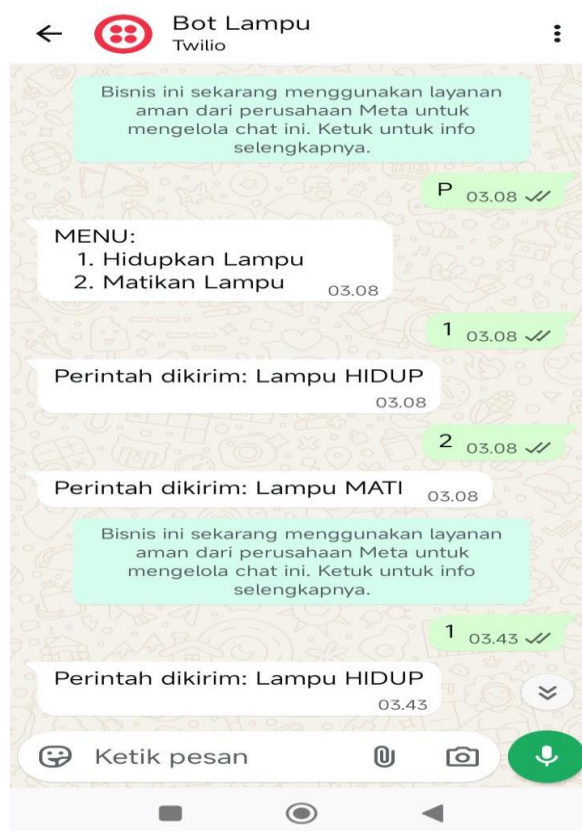
4.3 Pengujian Sistem Melalui WhatsApp

Pengujian sistem dilakukan dengan mengirimkan perintah melalui *WhatsApp* untuk memastikan bahwa sistem dapat berjalan dengan baik dan sesuai dengan yang dirancang. Pada tahap ini, pengguna berinteraksi langsung dengan sistem

menggunakan bot *WhatsApp* yang terhubung melalui Twilio dan ThingESP.

Pengguna mengirimkan perintah seperti “P” untuk menampilkan menu, “1” untuk menghidupkan lampu, dan “2” untuk mematikan lampu. Setelah perintah dikirim, sistem akan memproses pesan tersebut dan memberikan respon berupa status lampu sesuai dengan perintah yang diberikan. Selain itu, sistem juga dapat mengirimkan informasi tambahan seperti konfirmasi perintah dan kondisi perangkat secara real-time.

Pengujian ini bertujuan untuk memastikan bahwa komunikasi antara *WhatsApp*, server, dan perangkat ESP32 berjalan dengan lancar tanpa adanya kesalahan dalam penerimaan maupun eksekusi perintah.



Gambar 4.6 Perintah Bot Whatsapp

4.4 Pengujian Sistem

4.4.1 Pengujian Komunikasi LoRa

Pengujian komunikasi LoRa dilakukan untuk mengetahui kinerja pengiriman data antara node pengirim dan penerima menggunakan modul LoRa SX1278, serta memastikan data dapat diterima dengan baik.

Parameter yang digunakan meliputi frekuensi 433 MHz, bandwidth 125 kHz, dan daya pancar 18 dBm, sehingga komunikasi diharapkan berjalan stabil.

Pengujian dilakukan pada beberapa variasi jarak dengan node pengirim berada di dalam ruangan dan lingkungan padat penduduk yang memiliki banyak penghalang seperti bangunan dan pepohonan.

Pengukuran jarak dilakukan menggunakan aplikasi Google Maps dengan titik awal pengukuran ditetapkan pada posisi node *transmitter* menuju *node receiver*. Setiap jarak diuji sebanyak 5 kali percobaan, kemudian persentase keberhasilan dihitung berdasarkan perbandingan jumlah data yang berhasil diterima terhadap total percobaan. Seperti, dari 5 kali percobaan, 4 percobaan berhasil yang menunjukkan tingkat keberhasilan sebesar 80%.

Tabel 4.1 Hasil Pengujian LoRa SX1278

No	Jarak Pengujian (m)	Keberhasilan Pengiriman	Keterangan
1	25	100%	Data diterima dengan sangat baik tanpa gangguan
2	50	100%	Sinyal stabil dan tidak terjadi packet loss

No	Jarak Pengujian (m)	Keberhasilan Pengiriman	Keterangan
3	75	100%	Terdapat sedikit penurunan sinyal, namun data masih dapat diterima dengan baik
4	100	80%	Data tetap terkirim, namun mulai terjadi penurunan kualitas sinyal
5	125	40%	Kualitas komunikasi menurun, terdapat peningkatan packet loss
6	150	20%	Data masih dapat diterima, namun kondisi sinyal sangat buruk dan kurang

Berdasarkan Tabel 4.1, komunikasi LoRa bekerja sangat baik pada jarak 25 sampai dengan jarak 75 meter dengan tingkat keberhasilan tinggi. Namun, seiring bertambahnya jarak, kualitas komunikasi menurun akibat penghalang dan melemahnya sinyal, meskipun sistem masih dapat digunakan.

4.4.2 Pengujian Waktu Respon Sistem Berbasis WhatsApp

Pengujian waktu respon sistem dilakukan untuk mengetahui kecepatan sistem dalam mengeksekusi perintah yang dikirim oleh pengguna melalui *WhatsApp* hingga perangkat berhasil dikendalikan. Perintah yang digunakan berupa pesan “1” untuk menyalakan lampu (ON) dan pesan “2” untuk mematikan lampu (OFF).

Pengukuran waktu respon dilakukan menggunakan stopwatch, dimulai saat pesan berhasil dikirim dan dihentikan ketika lampu berubah kondisi sesuai perintah. Pengujian dilakukan pada beberapa variasi jarak untuk mengetahui pengaruh jarak terhadap waktu respon sistem.

Tabel 4.2 Waktu Respon Sistem

No	Jarak (m)	Perintah WhatsApp	Waktu Respon
1	25	ON/OFF	±2 detik
2	50	ON/OFF	±2 detik
3	75	ON/OFF	±2-3 detik
4	100	ON/OFF	±3 detik
5	125	ON/OFF	±3-4 detik
6	150	ON/OFF	±4 detik

Berdasarkan Tabel 4.2, waktu respon sistem berbasis *WhatsApp* berkisar antara ± 2 hingga ± 4 detik. Waktu respon cenderung meningkat seiring bertambahnya jarak antara node pengirim dan penerima. Hal ini dipengaruhi oleh proses komunikasi melalui jaringan internet dan LoRa, serta kondisi lingkungan yang memiliki banyak penghalang. Meskipun demikian, sistem masih tergolong responsif dan mampu bekerja dengan baik.

4.5 Analisis Sistem dan Evaluasi

Berdasarkan hasil pengujian yang telah dilakukan, sistem kontrol lampu berbasis LoRa dan *WhatsApp* dapat bekerja dengan baik. Komunikasi antara node pengirim dan node penerima menunjukkan tingkat keberhasilan yang cukup baik pada daerah padat penduduk. Namun, pada jarak ± 150 meter, tingkat keberhasilan pengiriman data mengalami penurunan hingga sekitar 20%.

Penggunaan teknologi LoRa terbukti mampu mendukung komunikasi jarak jauh dengan konsumsi daya yang rendah. Sementara itu, integrasi dengan *WhatsApp* memberikan kemudahan bagi pengguna dalam mengontrol perangkat tanpa memerlukan aplikasi tambahan, sehingga sistem menjadi lebih praktis dan *user-friendly*.

Namun demikian, berdasarkan hasil pengujian, terdapat beberapa kendala yang mempengaruhi kinerja sistem, seperti adanya hambatan berupa bangunan dan pepohonan yang menyebabkan penurunan kualitas sinyal LoRa. Hal ini menunjukkan bahwa kondisi lingkungan sangat berpengaruh terhadap performa komunikasi.

Sebagai evaluasi, sistem masih dapat dikembangkan lebih lanjut dengan meningkatkan posisi antena, menggunakan antena dengan gain yang lebih tinggi, atau melakukan optimasi parameter LoRa seperti *spreading factor* (parameter yang mengatur jangkauan dan kekuatan sinyal) dan *bandwidth* (parameter yang mengatur kecepatan serta kestabilan pengiriman data) untuk meningkatkan jangkauan dan kestabilan komunikasi.

Dengan demikian, sistem yang dirancang telah memenuhi tujuan penelitian, namun masih memiliki beberapa keterbatasan yang dapat diperbaiki pada pengembangan selanjutnya agar kinerja sistem menjadi lebih optimal.