

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Sistem

Tahap implementasi sistem merupakan langkah lanjutan setelah proses analisis dan perancangan selesai dilakukan. Implementasi ini mencakup penggabungan antara perangkat keras dan perangkat lunak yang telah dirancang, agar dapat bekerja secara terintegrasi sesuai dengan tujuan sistem. Sistem yang dikembangkan bertujuan untuk melakukan penyiraman tanaman cabai secara otomatis berdasarkan kelembapan tanah serta memberikan informasi kondisi lingkungan secara *real-time* melalui aplikasi Telegram.

Implementasi dimulai dengan merakit komponen perangkat keras sesuai dengan skema rangkaian. Setelah itu, perangkat lunak diunggah ke mikrokontroler, dan dilakukan integrasi antara Arduino UNO dengan NodeMCU ESP8266 untuk memastikan sistem dapat mengontrol aktuator dan mengirimkan notifikasi secara otomatis.

4.1.1 Perangkat Keras (*Hardware*)

Dalam pengembangan sistem penyiraman otomatis tanaman cabai berbasis Internet of Things, dibutuhkan sejumlah komponen perangkat keras yang dirakit sesuai kebutuhan sistem. Berikut adalah komponen-komponen utama yang digunakan:

1. Arduino UNO: Sebagai mikrokontroler utama yang mengontrol logika sistem penyiraman.
2. NodeMCU ESP8266: Modul komunikasi nirkabel untuk mengirimkan data ke Telegram Bot.

3. Sensor Soil Moisture (2 buah): Digunakan untuk membaca tingkat kelembapan tanah.
4. Sensor DHT22: Digunakan untuk memantau suhu dan kelembapan udara di sekitar tanaman.
5. Modul Relay 2 Channel: Untuk mengontrol aktif dan nonaktifnya dua pompa air secara otomatis.
6. Pompa Air 12V (2 buah): Digunakan sebagai aktuator penyiraman tanaman.
7. Power Supply 12V: Sebagai sumber daya untuk pompa dan komponen lainnya.
8. Breadboard dan Kabel Jumper: Untuk menghubungkan rangkaian secara fleksibel.
9. Box dan Dudukan Rangkaian: Untuk menjaga kerapihan dan keamanan perangkat keras.

Semua komponen dirakit sesuai skema rangkaian, dan dilakukan pengujian awal menggunakan multimeter serta serial monitor guna memastikan sensor dan aktuator bekerja dengan baik.

4.1.2 Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan dalam sistem ini terdiri dari pemrograman mikrokontroler dan integrasi layanan pesan Telegram. Pengembangan perangkat lunak dilakukan menggunakan:

1. Arduino IDE: Digunakan untuk menulis, memverifikasi, dan mengunggah kode program ke Arduino UNO dan NodeMCU ESP8266.
2. Library Pendukung: Beberapa library yang digunakan antara lain:
 - a. DHT.h untuk membaca sensor DHT22

- b. WiFiClientSecure.h dan CTBot.h untuk komunikasi Telegram
 - c. SoftwareSerial.h untuk komunikasi antar Arduino dan NodeMCU
3. Telegram Bot API: Digunakan untuk menerima dan mengirimkan pesan melalui aplikasi Telegram. Bot dibuat menggunakan BotFather dan dihubungkan ke sistem melalui token API.

Program yang dibuat mencakup:

- a. Pembacaan data dari sensor kelembapan tanah dan suhu/kelembapan udara.
- b. Pengambilan keputusan untuk mengaktifkan pompa jika tanah terdeteksi kering.
- c. Pengiriman notifikasi otomatis ke Telegram saat penyiraman berlangsung.
- d. Fitur permintaan status melalui Telegram untuk memberikan informasi suhu dan kelembapan secara real-time kepada pengguna.

Dengan integrasi perangkat keras dan perangkat lunak yang baik, sistem ini dapat berjalan otomatis dan memberikan kemudahan bagi pengguna dalam memantau dan mengontrol penyiraman tanaman secara efisien.

4.2 Rangkaian Keseluruhan Sistem

Berikut adalah rangkaian keseluruhan sistem penyiraman tanaman cabai otomatis berbasis IoT yang telah dirancang, beserta penjelasan fungsional masing-masing komponen.



Gambar 4. 1 Rangkaian Keseluruhan Sistem

Penjelasan Fungsi Masing-Masing Modul:

1. Sensor Soil Moisture

Digunakan untuk mendeteksi tingkat kelembapan tanah. Jika tanah dalam kondisi kering (di bawah ambang batas), maka sistem akan mengaktifkan pompa air untuk menyiram tanaman secara otomatis.

2. Sensor DHT22

Berfungsi untuk membaca suhu dan kelembapan udara di sekitar tanaman. Informasi ini dikirim ke pengguna melalui Telegram sebagai bagian dari laporan kondisi lingkungan.

3. Arduino UNO

Bertugas sebagai mikrokontroler utama yang membaca data dari sensor dan mengontrol aktuator (relay dan pompa air) berdasarkan logika penyiraman otomatis.

4. NodeMCU ESP8266

Berfungsi sebagai penghubung sistem ke jaringan Wi-Fi dan mengirimkan data berupa status sensor serta notifikasi penyiraman ke pengguna melalui Telegram Bot.

5. Modul Relay 2 Channel

Digunakan untuk mengendalikan dua buah pompa air. Relay diaktifkan oleh Arduino saat kelembapan tanah terdeteksi rendah.

6. Pompa Air 12V

Digunakan sebagai aktuator untuk menyiram tanaman ketika kondisi tanah kering.

7. Power Supply 12V

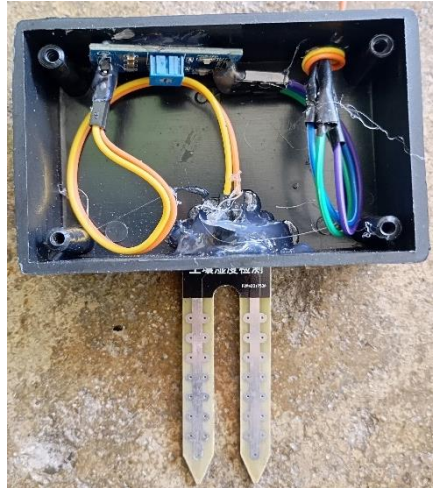
Menyuplai daya ke seluruh sistem, termasuk pompa, mikrokontroler, dan sensor.

8. Breadboard & Kabel Jumper

Digunakan untuk menyusun dan menghubungkan rangkaian antar komponen secara fleksibel dan rapi.

4.2.1 Rangkaian Soil Moisture Sensor

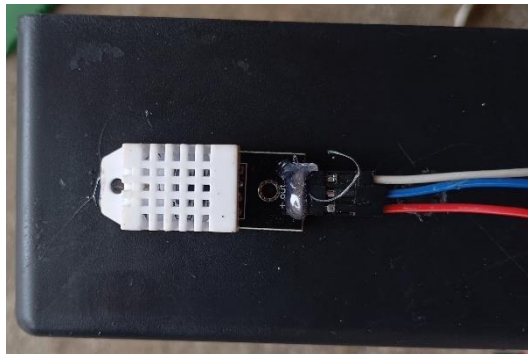
Berikut rangkaian *Soil Moisture Sensor* yang telah dirakit dan dimasukkan kedalam box project kecil agar mudah di implementasikan ke tanah.



Gambar 4. 2 Rangkaian Sensor Kelembapan Tanah

4.2.2 Rangkaian Sensor DHT22

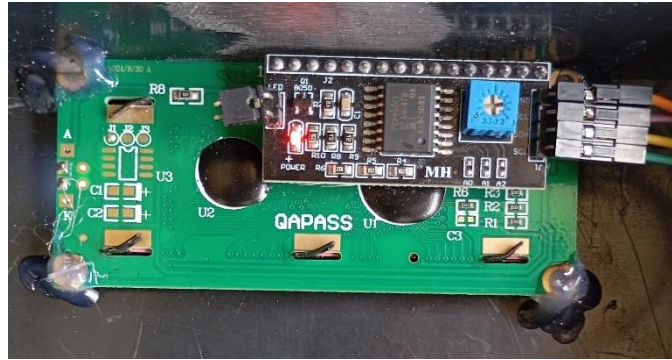
Berikut Rangkaian Sensor DHT22 untuk mengambil data suhu dan kelembapan udara secara *real time*.



Gambar 4. 3 Rangkaian Sensor DHT22

4.2.3 Rangkaian LCD (*Liquid Crystal Display*)

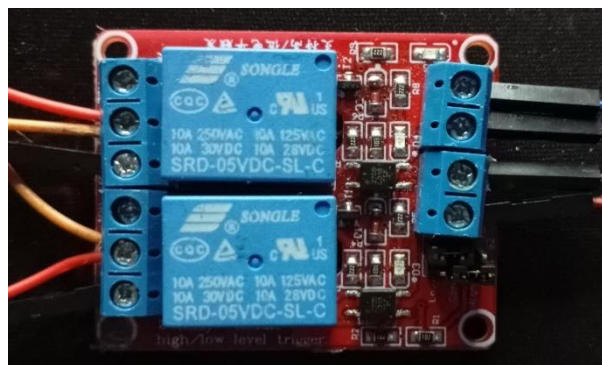
Rangkaian LCD (*Liquid Crystal Display*) yang terhubung dengan Arduino UNO didalam Box. Seperti pada gambar berikut :



Gambar 4. 4 Rangkaian LCD (*Liquid Crystal Display*)

4.2.4 Rangkaian Relay

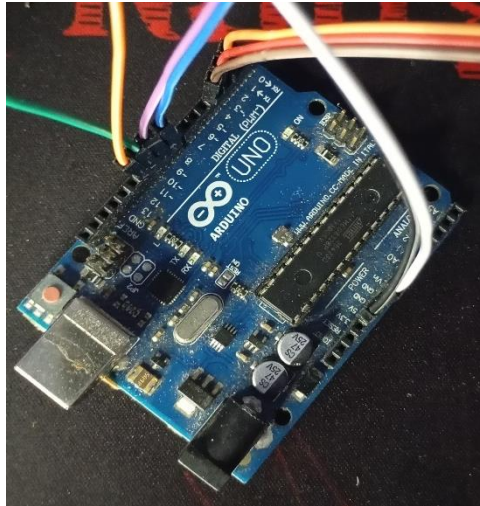
Relay dalam proyek ini digunakan sebagai saklar otomatis untuk menghidupkan dan mematikan relay ketika kondisi kelembapan tanah dibawah 40%.



Gambar 4. 5 Rangkaian Relay

4.2.5 Rangkaian Arduino UNO

Gambar berikut menunjukkan konfigurasi rangkaian Arduino UNO yang digunakan pada perancangan alat.



Gambar 4. 6 Rangkaian Arduino UNO

4.2.6 Rangkaian ESP8266

Berikut merupakan rangkaian ESP8266 yang digunakan dalam sistem ini.



Gambar 4. 7 Rangkaian ESP8266

4.2.7 Rangkaian Pompa Air

Gambar berikut memperlihatkan rangkaian dari pompa air 12 volt yang diaplikasikan dalam perancangan alat.



Gambar 4. 8 Rangkaian Pompa Air

4.2.8 Rangkaian Sensor DHT22 ke Arduino UNO

Berikut adalah rangkaian kabel Sensor DHT22 yang digunakan untuk membaca suhu dan kelembapan udara. Sensor ini terhubung langsung ke Arduino UNO.

Keterangan:

- a. VCC ke pin 5V Arduino UNO
- b. GND ke pin GND Arduino UNO
- c. Data ke pin digital D2 Arduino UNO

4.2.9 Rangkaian Soil Moisture Sensor ke Arduino UNO

Berikut adalah rangkaian kabel Soil Moisture Sensor yang digunakan untuk mendeteksi kadar kelembapan tanah. Dalam sistem ini digunakan dua sensor, masing-masing terhubung ke Arduino UNO.

Keterangan sensor kanan:

- a. VCC ke pin 5V Arduino UNO
- b. GND ke pin GND Arduino UNO
- c. AO ke pin A0 Arduino UNO

Keterangan sensor kiri:

- a. VCC ke pin 5V Arduino UNO
- b. GND ke pin GND Arduino UNO
- c. AO ke pin A1 Arduino UNO

4.2.10 Rangkaian Relay ke Arduino UNO

Berikut adalah rangkaian modul relay yang digunakan untuk mengendalikan pompa air dalam sistem. Terdapat dua relay yang masing-masing terhubung ke Arduino UNO.

Keterangan relay kanan:

- a. VCC ke pin 5V Arduino UNO
- b. GND ke pin GND Arduino UNO
- c. IN ke pin digital 7 Arduino UNO

Keterangan relay kiri:

- a. VCC ke pin 5V Arduino UNO
- b. GND ke pin GND Arduino UNO
- c. IN ke pin digital 8 Arduino UNO

4.2.11 Rangkaian ESP8266 ke Arduino UNO

Berikut adalah rangkaian koneksi modul ESP8266 yang digunakan untuk komunikasi data secara nirkabel. Modul ini mendapatkan suplai daya dari sumber eksternal yang terpisah dengan Arduino, namun tetap menggunakan satu adaptor yang dibagi dengan teknik *jumper* (cangkok) di bagian tengah kabel. Ground antara ESP8266 dan Arduino tetap disatukan untuk memastikan kestabilan komunikasi data.

Keterangan:

- a. VCC ESP8266 ke power supply eksternal 5V
- b. GND ESP8266 ke GND Arduino UNO
- c. RX D5 ESP8266 ke pin digital 11 Arduino UNO
- d. TX D6 ESP8266 ke pin digital 10 Arduino UNO

4.2.12 Rangkaian LCD ke Arduino UNO

Berikut adalah rangkaian LCD 16x2 dengan modul I2C yang digunakan untuk menampilkan data dari sistem. Penggunaan modul I2C mempermudah koneksi karena hanya membutuhkan dua pin dari Arduino UNO.

Keterangan:

- a. VCC ke pin 5V Arduino UNO
- b. GND ke pin GND Arduino UNO
- c. SDA ke pin A4 Arduino UNO
- d. SCL ke pin A5 Arduino UNO

4.3 Integrasi Telegram

Telegram berperan sebagai media notifikasi dalam sistem penyiraman otomatis berbasis Internet of Things (IoT). Pengguna dapat memantau kondisi lingkungan secara realtime dengan mengirimkan perintah /cek melalui aplikasi Telegram. Sistem akan merespons dengan mengirimkan data sensor berupa sensor kelembapan tanah kanan, kiri, suhu dan kelembapan udara. Dengan fitur ini, pengguna dapat mengetahui kondisi tanah dan lingkungan secara praktis tanpa harus mengakses perangkat secara langsung.

4.3.1 Integrasi Telegram Dengan SmartTani_Bot

Untuk menghubungkan sistem penyiraman otomatis dengan aplikasi Telegram, digunakan bot bernama SmartTani_Bot. Bot ini dibuat menggunakan BotFather di Telegram, yang menghasilkan Token API sebagai kunci untuk mengakses bot melalui program.

Bot ini diintegrasikan dengan *mikrokontroler* menggunakan koneksi *internet* melalui modul ESP8266. Ketika pengguna mengirim perintah /cek ke SmartTani_Bot, sistem akan mengambil data dari sensor dan mengirimkan balasan yang berisi informasi kelembapan tanah kanan dan kiri, suhu, serta kelembapan udara.

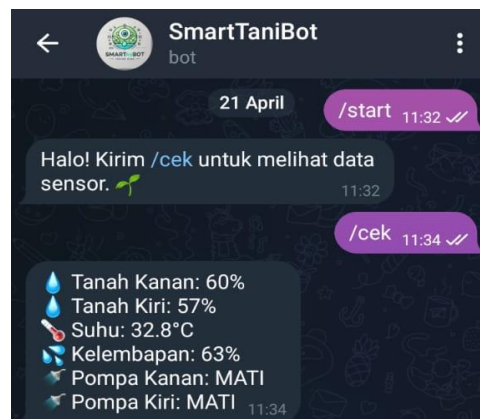
4.3.2 Pembuatan SmartTani_Bot pada Telegram

Untuk menghubungkan sistem penyiraman otomatis dengan Telegram, diperlukan sebuah bot khusus yang dibuat melalui Telegram menggunakan fitur resmi bernama BotFather. Bot yang digunakan dalam sistem ini bernama SmartTani_Bot. Proses pembuatan bot ini menghasilkan Token API, yang digunakan untuk mengintegrasikan bot ke dalam program pada Arduino IDE.

Berikut adalah langkah-langkah pembuatan bot SmartTani_Bot:

1. Buka aplikasi Telegram, lalu cari dan buka BotFather atau kunjungi link <https://t.me/BotFather>.
2. Ketik perintah /newbot untuk memulai proses pembuatan bot baru.
3. Berikan nama bot sesuai keinginan, misalnya: SmartTani.
4. Setelah itu, buat username bot dengan akhiran “bot”, misalnya: SmartTani_Bot.
5. Setelah berhasil, BotFather akan memberikan Token API. Token ini penting dan digunakan dalam kode program untuk menghubungkan sistem dengan bot tersebut.

Bot yang telah dibuat ini nantinya akan merespons perintah seperti /cek untuk mengirimkan data sensor dari sistem secara otomatis ke Telegram.



Gambar 4. 9 Tampilan Bot Telegram

4.4 Implementasi Program Monitoring Sistem

Sistem Penyiraman Otomatis Berbasis *Internet of Things (IoT)* adalah sistem yang dirancang untuk memantau kondisi tanah serta melakukan penyiraman secara otomatis berdasarkan nilai kelembapan tanah. Sistem ini menggunakan sensor *soil moisture* untuk mendeteksi kadar air di tanah, sensor DHT22 untuk membaca suhu dan kelembapan udara, serta modul ESP8266 untuk mengirim data ke Telegram.

Data hasil pembacaan sensor dapat diakses oleh pengguna melalui perintah /cek pada bot Telegram, dan hasil *monitoring* juga ditampilkan pada LCD secara langsung. Sistem ini dibangun menggunakan *mikrokontroler* Arduino UNO yang terhubung ke ESP8266, sehingga dapat terintegrasi dengan jaringan *Wi-Fi* dan mengakses layanan Telegram. Berikut adalah *script* program yang digunakan untuk membaca data dari sensor, mengontrol pompa air, dan mengirimkan data ke Telegram melalui koneksi ESP8266.

4.4.1 Script Arduino UNO

```

1  #include <Wire.h>
2  #include <LiquidCrystal_I2C.h>
3  LiquidCrystal_I2C lcd(0x27, 16, 2);
4  #include <DHT.h>
5  #include <SoftwareSerial.h>
6  SoftwareSerial espSerial(10, 11); // RX, TX
7  #define DHTPIN 2
8  #define DHTTYPE DHT22
9  DHT dht(DHTPIN, DHTTYPE);
10 #define TANAH_KANAN A0
11 #define TANAH_KIRI A1
12 #define POMPA_KANAN 7
13 #define POMPA_KIRI 8
14
15 // Kalibrasi
16 int nilaiKering = 1020;
17 int nilaiBasah = 400;
18 unsigned long waktuMulaiKanan = 0;
19 unsigned long waktuMulaiKiri = 0;
20 bool pompaKananAktif = false;
21 bool pompaKiriAktif = false;
22 const unsigned long durasiPompa = 5000; // 5 detik
23 const unsigned long jedaPompa = 10000; // 30 detik
24 unsigned long waktuSelesaiKanan = 0;
25 unsigned long waktuSelesaiKiri = 0;
26
27 void setup() {
28   Serial.begin(9600);
29   espSerial.begin(9600);
30   lcd.init();
31   lcd.backlight();

```

Gambar 4. 10 Sketch Arduino UNO Membaca Sensor

```

32 | pinMode(POMPA_KANAN, OUTPUT);
33 | pinMode(POMPA_KIRI, OUTPUT);
34 | digitalWrite(POMPA_KANAN, HIGH); // Matikan pompa awal
35 | digitalWrite(POMPA_KIRI, HIGH);
36 | dht.begin();
37 | Serial.println("Sistem Arduino siap!");
38 | }
39 | int hitungPersen(int nilai) {
40 |   return constrain(map(nilai, nilaiKering, nilaiBasah, 0, 100), 0, 100);
41 | }
42 |
43 | void loop() {
44 |   // Baca sensor
45 |   int tanahKanan = hitungPersen(analogRead(TANAH_KANAN));
46 |   int tanahKiri = hitungPersen(analogRead(TANAH_KIRI));
47 |   float suhu = dht.readTemperature();
48 |   float kelembapan = dht.readHumidity();
49 |   // Tampilkan di LCD
50 |   lcd.setCursor(0, 1);
51 |   lcd.print("R:");
52 |   lcd.print(tanahKanan);
53 |   lcd.print("% L:");
54 |   lcd.print(tanahKiri);
55 |   lcd.print("% ");
56 |   lcd.setCursor(0, 0);
57 |   lcd.print("Tem:");
58 |   lcd.print(suhu, 1);
59 |   lcd.print(" Hum:");
60 |   lcd.print(kelembapan, 0);
61 |   lcd.print("% ");
62 |   // Kontrol otomatis pompa

```

Gambar 4. 11 *Sketch* Arduino UNO Membaca Sensor (Lanjutan)

```

63 | if (tanahKanan < 40 && !pompaKananAktif && (millis() - waktuSelesaiKanan > jedaPompa)) {
64 |   digitalWrite(POMPA_KANAN, LOW);
65 |   waktuMulaiKanan = millis();
66 |   pompaKananAktif = true;
67 |   Serial.println("Pompa kanan diaktifkan");
68 | }
69 | if (tanahKiri < 40 && !pompaKiriAktif && (millis() - waktuSelesaiKiri > jedaPompa)) {
70 |   digitalWrite(POMPA_KIRI, LOW);
71 |   waktuMulaiKiri = millis();
72 |   pompaKiriAktif = true;
73 |   Serial.println("Pompa kiri diaktifkan");
74 | }
75 | // Matikan pompa setelah durasi
76 | if (pompaKananAktif && millis() - waktuMulaiKanan >= durasiPompa) {
77 |   digitalWrite(POMPA_KANAN, HIGH);
78 |   pompaKananAktif = false;
79 |   waktuSelesaiKanan = millis();
80 |   Serial.println("Pompa kanan dimatikan");
81 | }
82 | if (pompaKiriAktif && millis() - waktuMulaiKiri >= durasiPompa) {
83 |   digitalWrite(POMPA_KIRI, HIGH);
84 |   pompaKiriAktif = false;
85 |   waktuSelesaiKiri = millis();
86 |   Serial.println("Pompa kiri dimatikan");
87 | }
88 | // Handle perintah dari ESP
89 | if (espSerial.available()) {
90 |   String perintah = espSerial.readStringUntil('\n');
91 |   perintah.trim();
92 |   Serial.println("Perintah diterima: " + perintah);
93 |   if (perintah == "REQ") {

```

Gambar 4. 12 *Sketch* Arduino UNO Membaca Sensor (Lanjutan)

```

94 String balasan = "##START##\n";
95 balasan += "💧 Tanah Kanan: " + String(tanahKanan) + "%\n";
96 balasan += "💧 Tanah Kiri: " + String(tanahKiri) + "%\n";
97 balasan += "🌡 Suhu: " + String(suhu, 1) + "°C\n";
98 balasan += "💧 Kelembapan: " + String(kelembapan, 0) + "%\n";
99 balasan += "🚰 Pompa Kanan: " + String(pompaKananAktif ? "HIDUP" : "MATI") + "\n";
100 balasan += "🚰 Pompa Kiri: " + String(pompaKiriAktif ? "HIDUP" : "MATI") + "\n";
101 balasan += "##END##"; // Tanda akhir data
102 espSerial.println(balasan);
103 Serial.println("Mengirim: " + balasan);
104 delay(100);
105 }
106 else if (perintah == "NYALA") {
107   digitalWrite(POMPA_KANAN, LOW);
108   digitalWrite(POMPA_KIRI, LOW);
109   waktuMulaiKanan = millis();
110   waktuMulaiKiri = millis();
111   pompaKananAktif = true;
112   pompaKiriAktif = true;
113   espSerial.println("##START##\nSemua pompa dihidupkan selama 5 detik!\n##END##");
114 }
115 else if (perintah == "MATI") {
116   digitalWrite(POMPA_KANAN, HIGH);
117   digitalWrite(POMPA_KIRI, HIGH);
118   pompaKananAktif = false;
119   pompaKiriAktif = false;
120   espSerial.println("##START##\nSemua pompa dimatikan secara manual\n##END##");
121 }
122 }
123 delay(500);
124 }

```

Gambar 4. 13 *Sketch* Arduino UNO Membaca Sensor (Lanjutan)

4.4.2 Script ESP8266

```

1  #include <SoftwareSerial.h>
2  #include <ESP8266WiFi.h>
3  #include <WiFiClientSecure.h>
4  #include <UniversalTelegramBot.h>
5  const char* ssid = "SmartTani_Bot";
6  const char* pass = "12345678";
7  const char* botToken = "7979079631:AAGSn53-Gsbbcus96ZmeSiNkgM10zqCnSC0";
8  const int64_t chatID = 5383352245;
9  WiFiClientSecure client;
10 UniversalTelegramBot bot(botToken, client);
11 SoftwareSerial unoSerial(D1, D2); // RX, TX
12 unsigned long lastCheck = 0;
13 const unsigned long checkInterval = 3000;
14
15 void setup() {
16   Serial.begin(9600);
17   unoSerial.begin(9600);
18   client.setInsecure();
19   WiFi.begin(ssid, pass);
20   Serial.print("Menghubungkan ke WiFi");
21   while (WiFi.status() != WL_CONNECTED) {
22     delay(500);
23     Serial.print(".");
24   }
25   Serial.println("\nWiFi Terhubung. IP: " + WiFi.localIP().toString());
26   bot.sendMessage(String(chatID), "🤖 *Bot Siap Digunakan!*", "Markdown");
27 }
28
29 void loop() {
30   // Cek pesan masuk dari Telegram
31   if (millis() - lastCheck > checkInterval) {

```

Gambar 4. 14 *Sketch* ESP8266 Mengirim ke Telegram

```

32     int numNewMessages = bot.getUpdates(bot.last_message_received + 1);
33     while (numNewMessages) {
34         Serial.println("📬 Pesan baru diterima!");
35         handleMessages(numNewMessages);
36         numNewMessages = bot.getUpdates(bot.last_message_received + 1);
37     }
38     lastCheck = millis();
39 }
40 }
41
42 void handleMessages(int total) {
43     for (int i = 0; i < total; i++) {
44         String pesan = bot.messages[i].text;
45         String idPengirim = bot.messages[i].chat_id;
46
47         if (pesan == "/start") {
48             bot.sendMessage(idPengirim, "Halo! Kirim /cek untuk melihat data sensor. 📡", "");
49         }
50
51         else if (pesan == "/cek") {
52             Serial.println("📡 Mengirim REQ ke Arduino...");
53             unoSerial.println("REQ"); // Kirim perintah ke Arduino
54
55             String dataBalik = bacaDariUno(); // Baca respon dari Arduino
56             bot.sendMessage(idPengirim, dataBalik, "Markdown");
57         }
58     }
59 }
60
61 // Fungsi baca data dari Arduino
62 String bacaDariUno() {

```

Gambar 4. 16 Sketch ESP8266 Mengirim ke Telegram (Lanjutan)

```

61 // Fungsi baca data dari Arduino
62 String bacaDariUno() {
63     String buffer = "";
64     bool mulai = false;
65     unsigned long timeout = millis();
66     while (millis() - timeout < 3000) {
67         if (unoSerial.available()) {
68             String baris = unoSerial.readStringUntil('\n');
69             baris.trim();
70             Serial.println("[ESP] Terima: " + baris);
71
72             if (baris == "##START##") {
73                 mulai = true;
74                 buffer = "";
75             } else if (baris == "##END##") {
76                 break;
77             } else if (mulai) {
78                 buffer += baris + "\n";
79             }
80         }
81         yield(); // Cegah reset watchdog
82     }
83     if (buffer == "") buffer = "❗ Tidak ada respon dari Arduino.";
84     return buffer;

```

Gambar 4. 15 Sketch ESP8266 Mengirim ke Telegram (Lanjutan)

4.4.3 Pengujian Sistem

Setelah seluruh proses perancangan dan integrasi selesai, sistem diuji untuk memastikan bahwa semua komponen berfungsi sesuai dengan perencanaan.

1. LCD (Liquid Crystal Display)

Pengujian ini bertujuan untuk memverifikasi bahwa data sensor kelembapan tanah dan suhu udara yang ditampilkan di LCD benar-benar sesuai dengan hasil pembacaan sensor yang sebenarnya tanpa adanya perbedaan data. Hasil pengujian menunjukkan bahwa LCD berhasil menampilkan informasi mengenai kelembapan tanah kanan, kelembapan tanah kiri, suhu udara, serta kelembapan udara secara akurat dan konsisten. Data yang ditampilkan juga selaras dengan data yang dikirimkan melalui aplikasi Telegram, sehingga dapat disimpulkan bahwa LCD dapat digunakan sebagai indikator visual yang handal, akurat, dan responsif untuk memantau kondisi lingkungan tanaman secara real-time selama sistem dioperasikan.



Gambar 4. 17 LCD Menampilkan Data

Hasil pengujian menunjukkan bahwa LCD berhasil menampilkan informasi kelembapan tanah kanan, kelembapan tanah kiri, suhu, dan kelembapan udara secara akurat. Data yang ditampilkan juga konsisten dengan data yang dikirimkan melalui Telegram, sehingga LCD dapat digunakan sebagai indikator visual yang handal untuk memantau kondisi lingkungan secara real-time.

2. Relay dan Pompa Air

Pengujian ini bertujuan untuk memastikan bahwa relay yang mengontrol pompa air berfungsi sesuai dengan nilai kelembapan tanah yang telah ditentukan.

Tabel 4. 1 Tabel Relay

NO	Keterangan Baterai	Aksi Relay
1	Kelembapan Tanah 0%	ON
2	Kelembapan Tanah 10%	ON
3	Kelembapan Tanah 20%	ON
4	Kelembapan Tanah 30%	ON
5	Kelembapan Tanah 40%	ON
6	Kelembapan Tanah 50%	OFF
7	Kelembapan Tanah 60%	OFF
8	Kelembapan Tanah 70%	OFF
9	Kelembapan Tanah 80%	OFF
10	Kelembapan Tanah 90%	OFF
11	Kelembapan Tanah 100%	OFF

Berdasarkan Tabel 4.1, relay akan aktif dan menghidupkan pompa air apabila nilai kelembapan tanah kurang atau sama dengan 40%. Sebaliknya, apabila nilai kelembapan tanah di atas 40%, relay akan nonaktif dan pompa air akan berhenti. Pengujian ini membuktikan bahwa sistem mampu melakukan penyiraman otomatis berdasarkan kondisi kelembapan tanah yang telah ditetapkan.

3. Telegram

Pengujian Telegram bertujuan untuk memastikan bahwa bot SmartTani_Bot dapat berfungsi sebagai media monitoring secara realtime.



Tabel 4. 2 Telegram Mengirimkan Data Sensor

Saat pengguna mengetik perintah /start, bot SmartTani_Bot mengirimkan pesan sambutan berisi instruksi penggunaan. Selanjutnya, saat perintah /cek dikirim, bot akan membalas dengan data kondisi lingkungan yang terdiri dari:

- a. Tanah Kanan: 92%
- b. Tanah Kiri: 62%
- c. Suhu: 36.6°C
- d. Kelembapan: 55%

Pengujian ini menunjukkan bahwa SmartTani_Bot dapat memberikan laporan sensor secara realtime dengan baik, sehingga pengguna dapat memantau kondisi lahan kapan saja melalui aplikasi Telegram.