

LAMPIRAN-LAMPIRAN

Lampiran 1 Kode Node MCU

```
#include <ESP8266WiFi.h> // Library untuk koneksi WiFi pada ESP8266
#include <FirebaseESP8266.h> // Library untuk berinteraksi dengan Firebase dari
ESP8266
#include <LiquidCrystal_I2C.h> // Library untuk LCD dengan I2C
#include <NewPing.h> // Library untuk sensor ultrasonik
#include <UniversalTelegramBot.h> // Library untuk mengontrol bot Telegram
#include <WiFiClientSecure.h> // Library untuk membuat koneksi aman melalui HTTPS

// Koneksi Parameter
#define FIREBASE_HOST "https://iot-kebun-durian-default-rtdb.asia-
southeast1.firebaseio.com/" // URL Firebase Realtime Database
#define FIREBASE_AUTH "AlzaSyANytUBHAqdFBAFkl2hclF8lQrj5M7L9YI" // Token API
KEY otentikasi Firebase
#define WIFI_SSID "iPhone"
#define WIFI_PASSWORD "arohhhhh"

// Koneksi Pin
#define PIN_TRIG_ULTRASONIK D6
#define PIN_ECHO_ULTRASONIK D5
#define PIN_RELAY_POMPA D7
#define PIN_SCL_LCD D1
#define PIN_SDA_LCD D2

// Membuat objek sensor ultrasonik
NewPing jarakTandonAir(PIN_TRIG_ULTRASONIK, PIN_ECHO_ULTRASONIK, 200);

// Inisialisasi LCD dengan alamat I2C 0x27 dan ukuran 16x2
LiquidCrystal_I2C lcd(0x27, 16, 2);

// Konfigurasi bot Telegram
#define BOT_TOKEN "7216154318:AAFvpAS0bG76vud1TBD0EplhuuX6c7Tvgek" //
Token bot Telegram
String chatIDs = "7469765745"; // ID pengguna yang sudah ditetapkan untuk
menerima notifikasi
WiFiClientSecure client; // Objek klien untuk koneksi HTTPS aman
UniversalTelegramBot bot(BOT_TOKEN, client); // Objek bot Telegram dengan token
yang disediakan
unsigned long lastTimeBotRan; // Variabel untuk melacak waktu terakhir bot berjalan
unsigned long lastNotifSentTime = 0; // Waktu terakhir pesan notif dikirim
const int requestBotSendInterval = 250; // Waktu jeda antara setiap permintaan global
pada bot (250 ms atau 0,25 detik)
const unsigned long notifSendInterval = 5000; // Waktu jeda khusus notifikasi pada
bot (5000 ms atau 5 detik)

// Definisi jalur Firebase
```

```

FirebaseData firebaseData; // Objek untuk mengelola data Firebase
String pathRelayPompa = "/nodemcu/relay_pompa"; // Jalur relay pompa di Firebase
String pathKapasitasAir = "/nodemcu/kapasitas_air"; // Jalur kapasitas air di Firebase
String pathMode = "/nodemcu/mode"; // Jalur mode sistem di Firebase
String pathNotif = "/nodemcu/notif"; // Jalur untuk notifikasi di Firebase
String pathSoil1T1 = "/wemos1/soil1_t1"; // Jalur sensor tanah 1 (area pangkal) pohon 1 di Firebase
String pathSoil2T1 = "/wemos1/soil2_t1"; // Jalur sensor tanah 2 (area luar) pohon 1 di Firebase
String pathRelayS1 = "/wemos1/relay_s1"; // Jalur relay selenoid pohon 1 di Firebase
String pathSoil1T2 = "/wemos2/soil1_t2"; // Jalur sensor tanah 1 (area pangkal) pohon 2 di Firebase
String pathSoil2T2 = "/wemos2/soil2_t2"; // Jalur sensor tanah 2 (area luar) pohon 2 di Firebase
String pathRelayS2 = "/wemos2/relay_s2"; // Jalur relay selenoid pohon 2 di Firebase

// Spesifikasi tandon air
const float tinggiEmber = 30; // Tinggi ember dalam cm
const float kapasitasEmber = 20; // Kapasitas ember dalam liter

// Spesifikasi sensor soil
const int rangeValueSoil = 500; // Range nilai dari sensor soil yang ditetapkan untuk nilai < adalah basah atau >= adalah kering

// Fungsi untuk menghitung kapasitas air berdasarkan sensor ultrasonik
float getKapasitasAir() {
    int jarak = jarakTandonAir.ping_cm(); // Mengukur jarak air dengan sensor ultrasonik
    float tinggiAir = tinggiEmber - jarak; // Menghitung tinggi air di ember
    float kapasitasAir = (tinggiAir / tinggiEmber) * kapasitasEmber; // Mengkonversi tinggi air ke kapasitas liter
    kapasitasAir = constrain(kapasitasAir, 0, kapasitasEmber); // Membatasi nilai kapasitas air agar berada dalam batas ember
    return kapasitasAir;
}

// Fungsi untuk membaca nilai float dari Firebase
float getFirebaseFloatValue(String path, float defaultValue) {
    if (Firebase.getFloat(firebaseData, path)) {
        return firebaseData.floatData();
    } else {
        if (!Firebase.getFloat(firebaseData, path)) {
            Firebase.setFloat(firebaseData, path, defaultValue);
        }
        return defaultValue;
    }
}

// Fungsi untuk membaca nilai integer dari Firebase
int getFirebaseIntValue(String path, int defaultValue) {

```

```

if (Firebase.getInt(firebaseData, path)) {
return firebaseData.intData();
} else {
if (!Firebase.getInt(firebaseData, path)) {
Firebase.setInt(firebaseData, path, defaultValue);
}
return defaultValue;
}
}

// Fungsi untuk mengupdate nilai float ke Firebase
void updateFirebase(String path, float value) {
Firebase.setFloat(firebaseData, path, value); // Update nilai ke Firebase
}

// Fungsi untuk me-reset semua relay (mematikan relay)
void resetRelays() {
updateFirebase(pathRelayPompa, 0); // Mematikan relay pompa
updateFirebase(pathRelayS1, 0); // Mematikan relay selenoid tanaman 1
updateFirebase(pathRelayS2, 0); // Mematikan relay selenoid tanaman 2
}

// Fungsi untuk menampilkan data pada LCD
void updateLCD(float sisaAir, int soil1T1, int soil2T1, int soil1T2, int soil2T2, int relayS1,
int relayS2) {
lcd.clear();

// T = Tanaman, S = Selenoid, A = Air, L = Liter, K = Kering, B = Basah, | = Tanda
Pemisah
// Simbol untuk kelembaban tanah K untuk kering B untuk basah
char simbolT1 = (soil1T1 >= rangeValueSoil || soil2T1 == 1) ? 'K' : ((soil1T1 <
rangeValueSoil && soil2T1 == 0) ? 'B' : 'B');
char simbolT2 = (soil1T2 >= rangeValueSoil || soil2T2 == 1) ? 'K' : ((soil1T2 <
rangeValueSoil && soil2T2 == 0) ? 'B' : 'B');

// Baris pertama
lcd.setCursor(0, 0);
lcd.print("T1:");
lcd.print(simbolT1);
lcd.print(" S1:");
lcd.print(relayS1);
lcd.print(" | A(L)");

// Baris kedua
lcd.setCursor(0, 1);
lcd.print("T2:");
lcd.print(simbolT2);

```

```

lcd.print(" S2:");
lcd.print(relayS2);
lcd.print(" | ");
lcd.print(sisaAir);
}

// Fungsi untuk membaca status notifikasi dari Firebase
int getNotifStatus() {
    return getFirebaseIntValue(pathNotif, 1); // Default adalah 1 ketika pada database
    belum ada (notifikasi hidup)
}

// Fungsi untuk mengirim informasi kepada semua pengguna
void sendNotifBot() {
    // Dapatkan data dari sensor
    float sisaAir = Firebase.getFloat(firebaseData, pathKapasitasAir) ?
    firebaseData.floatData() : 0;
    int soil1T1 = Firebase.getInt(firebaseData, pathSoil1T1) ? firebaseData.intData() : 0;
    int soil2T1 = Firebase.getInt(firebaseData, pathSoil2T1) ? firebaseData.intData() : 0;
    int soil1T2 = Firebase.getInt(firebaseData, pathSoil1T2) ? firebaseData.intData() : 0;
    int soil2T2 = Firebase.getInt(firebaseData, pathSoil2T2) ? firebaseData.intData() : 0;

    // Menentukan kondisi tanah
    String kondisiTanah1Soil1 = (soil1T1 >= rangeValueSoil) ? "Kering" : ((soil1T1 <
    rangeValueSoil) ? "Basah" : "Basah");
    String kondisiTanah1Soil2 = (soil2T1 == 1) ? "Kering" : ((soil2T1 == 0) ? "Basah" :
    "Basah");
    String kondisiTanah2Soil1 = (soil1T2 >= rangeValueSoil) ? "Kering" : ((soil1T2 <
    rangeValueSoil) ? "Basah" : "Basah");
    String kondisiTanah2Soil2 = (soil2T2 == 1) ? "Kering" : ((soil2T2 == 0) ? "Basah" :
    "Basah");

    // Menyusun pesan yang akan dikirim
    String notif = "Notifikasi Informasi Kondisi Tanaman\n\n";
    notif += "*Kondisi Tanah Tanaman 1*\n";
    notif += "- Sensor Soil 1: " + kondisiTanah1Soil1 + " (" + String(soil1T1) + ")\n";
    notif += "- Sensor Soil 2: " + kondisiTanah1Soil2 + " (" + String(soil2T1) + ")\n\n";
    notif += "*Kondisi Tanah Tanaman 2*\n";
    notif += "- Sensor Soil 1: " + kondisiTanah2Soil1 + " (" + String(soil1T2) + ")\n";
    notif += "- Sensor Soil 2: " + kondisiTanah2Soil2 + " (" + String(soil2T2) + ")\n";

    // Memecah string chatIDs menjadi ID individual, dipisahkan dengan koma
    int startIdx = 0;
    int commIdx = chatIDs.indexOf(',', startIdx);

    // Kirim pesan ke semua chat_id yang ada dalam chatIDs
    while (commIdx != -1) {
        String chat_id = chatIDs.substring(startIdx, commIdx);
        bot.sendChatAction(chat_id, "typing");
    }
}

```

```

bot.sendMessage(chat_id, notif, "Markdown");
startIdx = commIdx + 1;
commIdx = chatIDs.indexOf(',', startIdx);
}

// Kirim pesan ke ID terakhir (atau satu-satunya jika tidak ada koma)
if (startIdx < chatIDs.length()) {
String chat_id = chatIDs.substring(startIdx);
bot.sendChatAction(chat_id, "typing");
bot.sendMessage(chat_id, notif, "Markdown");
}
}

// Fungsi untuk menangani perintah bot Telegram
void handleBotCommands() {
int numNewMessages = bot.getUpdates(bot.last_message_received + 1); //
Mendapatkan update pesan baru
int mode = getFirebaseIntValue(pathMode, 1); // Memeriksa mode sistem
(otomatis/manual)

for (int i = 0; i < numNewMessages; i++) {
String chat_id = bot.messages[i].chat_id; // Mendapatkan ID chat Telegram (siapa
yang chat)
String text = bot.messages[i].text; // Mendapatkan teks pesan pengguna
bot.sendChatAction(chat_id, "typing"); // Menampilkan status "typing" sebelum
merespon

// Menangani perintah "/start"
if (text == "/start") {
String welcome = "Selamat datang di Sistem Alat Penyiraman Otomatis Pada
Perkebunan Pohon Durian Berbasis IoT.\n\n";
welcome += "*Gunakan perintah:*\n";
welcome += "/start - Mulai Sistem\n";
welcome += "/mode - Ganti Mode Otomatis/Manual\n";
welcome += "/notif - Hidupkan/Matikan Notifikasi\n";
welcome += "/info - Lihat Informasi Sistem\n";
welcome += "/on1 - ON Penyiraman Tanaman 1\n";
welcome += "/off1 - OFF Penyiraman Tanaman 1\n";
welcome += "/on2 - ON Penyiraman Tanaman 2\n";
welcome += "/off2 - OFF Penyiraman Tanaman 2\n\n";
bot.sendMessage(chat_id, welcome, "Markdown");
}

// Menangani perintah "/info"
else if (text == "/info") {
float sisaAir = getFirebaseFloatValue(pathKapasitasAir, 0);
int relayPump = getFirebaseIntValue(pathRelayPompa, 0);
int soil1T1 = getFirebaseIntValue(pathSoil1T1, 0);
int soil2T1 = getFirebaseIntValue(pathSoil2T1, 0);

```

```

int soil1T2 = getFirestoreIntValue(pathSoil1T2, 0);
int soil2T2 = getFirestoreIntValue(pathSoil2T2, 0);
int relayS1 = getFirestoreIntValue(pathRelayS1, 0);
int relayS2 = getFirestoreIntValue(pathRelayS2, 0);
int notif = getFirestoreIntValue(pathNotif, 0);
// Menentukan kondisi sistem berdasarkan data Firebase
String kondisiMode = (mode == 1) ? "Otomatis" : "Manual";
String kondisiNotif = (notif == 1) ? "Hidup" : "Mati";
String kondisiTanah1Soil1 = (soil1T1 >= rangeValueSoil) ? "Kering" : ((soil1T1 <
rangeValueSoil) ? "Basah" : "Basah");
String kondisiTanah1Soil2 = (soil2T1 == 1) ? "Kering" : ((soil2T1 == 0) ? "Basah" :
"Basah");
String kondisiTanah2Soil1 = (soil1T2 >= rangeValueSoil) ? "Kering" : ((soil1T2 <
rangeValueSoil) ? "Basah" : "Basah");
String kondisiTanah2Soil2 = (soil2T2 == 1) ? "Kering" : ((soil2T2 == 0) ? "Basah" :
"Basah");
String kondisiPompa = (relayPump == 1) ? "Hidup" : "Mati";
String kondisiSolenoid1 = (relayS1 == 1) ? "Hidup" : "Mati";
String kondisiSolenoid2 = (relayS2 == 1) ? "Hidup" : "Mati";
// Menyusun informasi sistem untuk dikirim ke pengguna
String info = "Informasi Sistem Alat Penyiraman Otomatis Pada Perkebunan Pohon
Durian Berbasis IoT.\n";
info += "\n*NodeMCU (Parent)*\n";
info += "- Mode Sistem: " + kondisiMode + "\n";
info += "- Status Notifikasi: " + kondisiNotif + "\n";
info += "- Sisa Air Tandon: " + String(sisaAir) + "L\n";
info += "- Kondisi Pompa: " + kondisiPompa + "\n";
info += "\n*Wemos (Child - Tanaman 1)*\n";
info += "- Kondisi Sensor Soil 1: " + kondisiTanah1Soil1 + " (" + String(soil1T1) + ")\n";
info += "- Kondisi Sensor Soil 2: " + kondisiTanah1Soil2 + " (" + String(soil2T1) + ")\n";
info += "- Kondisi Solenoid Valve: " + kondisiSolenoid1 + "\n";
info += "\n*Wemos (Child - Tanaman 2)*\n";
info += "- Kondisi Sensor Soil 1: " + kondisiTanah2Soil1 + " (" + String(soil1T2) + ")\n";
info += "- Kondisi Sensor Soil 2: " + kondisiTanah2Soil2 + " (" + String(soil2T2) + ")\n";
info += "- Kondisi Solenoid Valve: " + kondisiSolenoid2;
// Mengirim pesan ke pengguna
bot.sendMessage(chat_id, info, "Markdown");
}

// Menangani perintah "/notif"
else if (text == "/notif") {
int currentNotif = getNotifStatus();
int newNotifStatus = (currentNotif == 1) ? 0 : 1; // Beralih antara 1 dan 0
updateFirestore(pathNotif, newNotifStatus); // Perbarui status notifikasi di Firebase
if (newNotifStatus == 1) {
bot.sendMessage(chat_id, "Notifikasi telah dihidupkan. Silahkan gunakan /notif untuk
mengubahnya kembali.", "");
} else {

```

```

bot.sendMessage(chat_id, "Notifikasi telah dimatikan. Silahkan gunakan /notif untuk mengubahnya kembali.", "");
}
}

// Menangani perintah "/mode"
else if (text == "/mode") {
// Mengganti mode sistem (otomatis/manual)
int newMode = mode == 1 ? 0 : 1;
updateFirebase(pathMode, newMode);
resetRelays(); // Reset semua relay (default 0/mati)
if (newMode == 1) {
bot.sendMessage(chat_id, "Mode diubah ke Otomatis. Silahkan gunakan /mode untuk mengubahnya kembali.", "");
} else {
bot.sendMessage(chat_id, "Mode diubah ke Manual. Silahkan gunakan /mode untuk mengubahnya kembali.", "");
}
}

// Perintah manual diizinkan hanya dalam mode manual
else if (mode == 0) {
if (text == "/on1") {
updateFirebase(pathRelayS1, 1); // Menghidupkan selenoid tanaman 1
updateFirebase(pathRelayPompa, 1); // Menghidupkan pompa
bot.sendMessage(chat_id, "Pompa Selenoid Tanaman 1 berhasil dihidupkan.", "");
}
else if (text == "/off1") {
updateFirebase(pathRelayS1, 0); // Mematikan selenoid tanaman 1
bot.sendMessage(chat_id, "Pompa Selenoid Tanaman 1 berhasil dimatikan.", "");
}
else if (text == "/on2") {
updateFirebase(pathRelayS2, 1); // Menghidupkan selenoid tanaman 2
updateFirebase(pathRelayPompa, 1); // Menghidupkan pompa
bot.sendMessage(chat_id, "Pompa Selenoid Tanaman 2 berhasil dihidupkan.", "");
}
else if (text == "/off2") {
updateFirebase(pathRelayS2, 0); // Mematikan selenoid tanaman 2
bot.sendMessage(chat_id, "Pompa Selenoid Tanaman 2 berhasil dimatikan.", "");
}
}
// Menangani perintah yang tidak tersedia di mode manual
else {
bot.sendMessage(chat_id, "Maaf, perintah tidak tersedia. Silakan gunakan perintah yang sesuai!", "");
}
}

// Menangani perintah yang tidak tersedia di mode otomatis
else {

```

```

if (text == "/on1" || text == "/off1" || text == "/on2" || text == "/off2") {
  bot.sendMessage(chat_id, "Maaf, tidak dapat mengatur sistem dalam mode
otomatis. Gunakan /mode untuk beralih ke manual sebelum mengatur sistem!", "");
}
else {
  bot.sendMessage(chat_id, "Maaf, perintah tidak tersedia. Silakan gunakan perintah
yang sesuai!", "");
}
}
}
}

// Fungsi setup untuk inisialisasi sistem
void setup() {
  Serial.begin(9600); // Memulai komunikasi serial dengan kecepatan 9600 baud
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD); // Menghubungkan ke jaringan WiFi
  Serial.print("Sistem IoT Kebun Durian\n");
  Serial.print("Booting...\n");
  Serial.print("Koneksi ke WiFi");

  // Menunggu sampai terhubung ke WiFi
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("\nTerkoneksi WiFi");

  // Menginisialisasi koneksi ke Firebase
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);

  // Inisialisasi LCD
  lcd.init();
  lcd.backlight();

  // Inisialisasi pin relay pompa
  pinMode(PIN_RELAY_POMPA, OUTPUT);
  digitalWrite(PIN_RELAY_POMPA, LOW);

  // Menginisialisasi koneksi aman untuk bot Telegram
  client.setInsecure();
}

// Fungsi utama yang berjalan berulang kali
void loop() {
  if (WiFi.status() != WL_CONNECTED) {
    WiFi.disconnect();
    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
    while (WiFi.status() != WL_CONNECTED) {
      delay(500);

```

```

Serial.print(".");
}
Serial.println("Menghubungkan ulang ke WiFi");
}

// Membaca kapasitas air dari sensor dan memperbarui di Firebase
float sisaAir = getKapasitasAir();
updateFirebase(pathKapasitasAir, sisaAir);

// Membaca mode sistem, status relay, dan sensor tanah dari Firebase
int mode = getFirebaseIntValue(pathMode, 1);
int relayPump = getFirebaseIntValue(pathRelayPompa, 0);
int soil1T1 = getFirebaseIntValue(pathSoil1T1, 0);
int soil2T1 = getFirebaseIntValue(pathSoil2T1, 0);
int soil1T2 = getFirebaseIntValue(pathSoil1T2, 0);
int soil2T2 = getFirebaseIntValue(pathSoil2T2, 0);
int relayS1 = getFirebaseIntValue(pathRelayS1, 0);
int relayS2 = getFirebaseIntValue(pathRelayS2, 0);

// Memperbarui tampilan LCD dengan informasi terbaru
updateLCD(sisaAir, soil1T1, soil2T1, soil1T2, soil2T2, relayS1, relayS2);

// Logika untuk menghidupkan/mematikan relay pompa berdasarkan status relay
solenoid
if (relayS1 == 1 || relayS2 == 1) {
    updateFirebase(pathRelayPompa, 1); // Jika salah satu relay solenoid hidup, hidupkan
    relay pompa
} else {
    updateFirebase(pathRelayPompa, 0); // Jika kedua relay solenoid mati, matikan relay
    pompa
}

// Mengontrol status relay pompa secara fisik
digitalWrite(PIN_RELAY_POMPA, relayPump == 1 ? HIGH : LOW);

// Mode otomatis untuk mengontrol relay berdasarkan kelembaban tanah
if (mode == 1) {
    // Jika tanah pohon 1 kering, hidupkan relay
    if (soil1T1 >= rangeValueSoil || soil2T1 == 1) {
        updateFirebase(pathRelayS1, 1); // Hidupkan relay solenoid pohon 1
    } else if (soil1T1 < rangeValueSoil && soil2T1 == 0) {
        updateFirebase(pathRelayS1, 0); // Matikan relay solenoid pohon 1 jika tanah basah
    }
    // Jika tanah pohon 2 kering, hidupkan relay
    if (soil1T2 >= rangeValueSoil || soil2T2 == 1) {
        updateFirebase(pathRelayS2, 1); // Hidupkan relay solenoid pohon 2
    } else if (soil1T2 < rangeValueSoil && soil2T2 == 0) {
        updateFirebase(pathRelayS2, 0); // Matikan relay solenoid pohon 2 jika tanah basah
    }
}

```

```

}

// Membaca status notifikasi dari Firebase
int notifStatus = getNotifStatus();
// Jika notifikasi dihidupkan (notifStatus == 1), kirim notifikasi setiap 5 detik
if (notifStatus == 1) {
  if (millis() - lastNotifSentTime > notifSendInterval) {
    sendNotifBot(); // Kirim notifikasi
    lastNotifSentTime = millis();
  }
}

// Menjalankan perintah bot Telegram
if (millis() > lastTimeBotRan + requestBotSendInterval) {
  handleBotCommands();
  lastTimeBotRan = millis();
}
}

```

Lampiran 2 Kode Wemos 1

```

#include <ESP8266WiFi.h> // Library untuk koneksi WiFi pada ESP8266
#include <FirebaseESP8266.h> // Library untuk berinteraksi dengan Firebase dari
ESP8266

// Koneksi Parameter
#define FIREBASE_HOST "https://iot-kebun-durian-default-rtdb.asia-
southeast1.firebaseio.com/" // URL Firebase Realtime Database
#define FIREBASE_AUTH "AlzaSyANytUBHAqdFBAFkl2hclF8lQrj5M7L9YI" // Token API
KEY otentikasi Firebase
#define WIFI_SSID "iPhone"
#define WIFI_PASSWORD "arohhhhh"

// Koneksi Pin
#define SOIL1_PIN A0
#define SOIL2_PIN D4
#define RELAY_PIN D5

// Objek untuk mengelola data Firebase
FirebaseData firebaseData;

// Definisi jalur Firebase
String pathSoil1T1 = "/wemos1/soil1_t1"; // Jalur sensor tanah 1 (area pangkal) pohon
1 di Firebase
String pathSoil2T1 = "/wemos1/soil2_t1"; // Jalur sensor tanah 2 (area luar) pohon 1
di Firebase
String pathRelayS1 = "/wemos1/relay_s1"; // Jalur relay selenoid pohon 1 di Firebase

```

```

// Fungsi setup untuk inialisasi sistem
void setup() {
  Serial.begin(9600); // Memulai komunikasi serial dengan kecepatan 9600 baud
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD); // Menghubungkan ke jaringan WiFi
  Serial.print("Sistem IoT Kebun Durian (Tanaman 1)\n");
  Serial.print("Booting...\n");
  Serial.print("Koneksi ke WiFi");

  // Menunggu sampai terhubung ke WiFi
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("\nTerkoneksi WiFi");

  // Menginisialisasi koneksi ke Firebase
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);

  // Inisialisasi pin relay selenoid
  pinMode(RELAY_PIN, OUTPUT);
  digitalWrite(RELAY_PIN, LOW); // Pastikan relay mati pada awalnya
}

// Fungsi untuk mengupdate nilai sensor tanah ke Firebase
void updateSoilSensor() {
  int soil1Value = analogRead(SOIL1_PIN);
  int soil2Value = digitalRead(SOIL2_PIN);
  if (Firebase.setInt(firebaseData, pathSoil1T1, soil1Value)) {
    Serial.print("Berhasil update ");
    Serial.print(pathSoil1T1);
    Serial.print(" Nilai: ");
    Serial.println(soil1Value);
  } else {
    Serial.print("Gagal update ");
    Serial.println(firebaseData.errorReason());
  }
  if (Firebase.setInt(firebaseData, pathSoil2T1, soil2Value)) {
    Serial.print("Berhasil update ");
    Serial.print(pathSoil2T1);
    Serial.print(" Nilai: ");
    Serial.println(soil2Value);
  } else {
    Serial.print("Gagal update ");
    Serial.println(firebaseData.errorReason());
  }
}

// Fungsi untuk mengontrol relay selenoid berdasarkan data Firebase

```

```

void updateRelayStatus() {
  if (Firebase.getInt(firebaseData, pathRelayS1)) {
    int relayStatus = firebaseData.intData(); // Ambil nilai relay dari Firebase
    digitalWrite(RELAY_PIN, relayStatus == 1 ? LOW : HIGH); // Atur relay sesuai nilai
    Serial.print("Berhasil ambil ");
    Serial.print(pathRelayS1);
    Serial.print(" Nilai: ");
    Serial.println(relayStatus);
  } else {
    Serial.print("Gagal ambil ");
    Serial.println(firebaseData.errorReason());
  }
}

// Fungsi utama yang berjalan berulang kali
void loop() {
  updateSoilSensor(); // Update nilai sensor tanah ke Firebase
  updateRelayStatus(); // Update status relay selenoid dari Firebase

  delay(1000); // Delay sistem 1 detik
}

```

Lampiran 3 Kode Wemos 2

```

#include <ESP8266WiFi.h> // Library untuk koneksi WiFi pada ESP8266
#include <FirebaseESP8266.h> // Library untuk berinteraksi dengan Firebase dari
ESP8266

// Koneksi Parameter
#define FIREBASE_HOST "https://iot-kebun-durian-default-rtdb.asia-southeast1.firebaseio.com/" // URL Firebase Realtime Database
#define FIREBASE_AUTH "AlzaSyANytUBHAqdFBAFkl2hclF8lQrj5M7L9YI" // Token API
KEY otentikasi Firebase
#define WIFI_SSID "iPhone"
#define WIFI_PASSWORD "arohhhhh"

// Koneksi Pin
#define SOIL1_PIN A0
#define SOIL2_PIN D4
#define RELAY_PIN D5

// Objek untuk mengelola data Firebase
FirebaseData firebaseData;

// Definisi jalur Firebase
String pathSoil1T2 = "/wemos2/soil1_t2"; // Jalur sensor tanah 1 (area pangkal) pohon
2 di Firebase

```

```

String pathSoil2T2 = "/wemos2/soil2_t2"; // Jalur sensor tanah 2 (area luar) pohon 2
di Firebase
String pathRelayS2 = "/wemos2/relay_s2"; // Jalur relay selenoid pohon 2 di Firebase

// Fungsi setup untuk inisialisasi sistem
void setup() {
  Serial.begin(9600); // Memulai komunikasi serial dengan kecepatan 9600 baud
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD); // Menghubungkan ke jaringan WiFi
  Serial.print("Sistem IoT Kebun Durian (Tanaman 2)\n");
  Serial.print("Booting...\n");
  Serial.print("Koneksi ke WiFi");

  // Menunggu sampai terhubung ke WiFi
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("\nTerkoneksi WiFi");

  // Menginisialisasi koneksi ke Firebase
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);

  // Inisialisasi pin relay selenoid
  pinMode(RELAY_PIN, OUTPUT);
  digitalWrite(RELAY_PIN, LOW); // Pastikan relay mati pada awalnya
}

// Fungsi untuk mengupdate nilai sensor tanah ke Firebase
void updateSoilSensor() {
  int soil1Value = analogRead(SOIL1_PIN);
  int soil2Value = digitalRead(SOIL2_PIN);
  if (Firebase.setInt(firebaseData, pathSoil1T2, soil1Value)) {
    Serial.print("Berhasil update ");
    Serial.print(pathSoil1T2);
    Serial.print(" Nilai: ");
    Serial.println(soil1Value);
  } else {
    Serial.print("Gagal update ");
    Serial.println(firebaseData.errorReason());
  }
  if (Firebase.setInt(firebaseData, pathSoil2T2, soil2Value)) {
    Serial.print("Berhasil update ");
    Serial.print(pathSoil2T2);
    Serial.print(" Nilai: ");
    Serial.println(soil2Value);
  } else {
    Serial.print("Gagal update ");
    Serial.println(firebaseData.errorReason());
  }
}

```

```
}

// Fungsi untuk mengontrol relay selenoid berdasarkan data Firebase
void updateRelayStatus() {
  if (Firebase.getInt(firebaseData, pathRelayS2)) {
    int relayStatus = firebaseData.getIntData(); // Ambil nilai relay dari Firebase
    digitalWrite(RELAY_PIN, relayStatus == 1 ? LOW : HIGH); // Atur relay sesuai nilai
    Serial.print("Berhasil ambil ");
    Serial.print(pathRelayS2);
    Serial.print(" Nilai: ");
    Serial.println(relayStatus);
  } else {
    Serial.print("Gagal ambil ");
    Serial.println(firebaseData.errorReason());
  }
}

// Fungsi utama yang berjalan berulang kali
void loop() {
  updateSoilSensor(); // Update nilai sensor tanah ke Firebase
  updateRelayStatus(); // Update status relay selenoid dari Firebase

  delay(1000); // Delay sistem 1 detik
}
```