





Lampiran

Hasil Pengujian

Kode	Pengujian	Case	Hasil yang diharapkan
TS-01	Pengujian Soil Moisture Sensor	Soil Moisture membaca nilai basah dan kering dan mengeluarkan nilai resistensinya	
TS-02	Pengujian LCD 16x2	LCD menerima data pembacaan dari soil moisture	
TS-03	Pengujian LED	LED menerima data pembacaan dari soil moisture	
TS-04	Pengujian ThingSpeak	Data dikirimkan melalui wifi	

Kode Program

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <ESP8266WiFi.h>
#include <ThingSpeak.h>

// Pin Definitions
#define RED_LED D5
#define GREEN_LED D6
#define YELLOW_LED D7
#define BUTTON_PIN D3
#define SOIL_SENSOR_PIN A0

// LCD Setup
LiquidCrystal_I2C lcd(0x27, 16, 2); // Alamat LCD

// WiFi Credentials
const char* ssid = "iPhone"; // SSID WiFi
const char* password = "1234567890"; // Password WiFi

// ThingSpeak Credentials
unsigned long channelID = 2731498; // ThingSpeak Channel ID
const char* apiKey = "3X3IXY8NUSXSYB6N"; // API Write Key

// Variables
bool isMeasuring = false;
int soilValue = 0;
float moisturePercent = 0;
```

```

WiFiClient client;

void setup() {
  // Initialize serial communication
  Serial.begin(115200);

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

  // Initialize pins
  pinMode(RED_LED, OUTPUT);
  pinMode(GREEN_LED, OUTPUT);
  pinMode(YELLOW_LED, OUTPUT);
  pinMode(BUTTON_PIN, INPUT_PULLUP);

  // Default LED state
  digitalWrite(RED_LED, LOW);
  digitalWrite(GREEN_LED, HIGH);
  digitalWrite(YELLOW_LED, LOW);

  // Display welcome message
  lcd.setCursor(0, 0);
  lcd.print("Pengukur Kadar Air");
  lcd.setCursor(0, 1);
  lcd.print("1461900003");

  // Initialize WiFi connection
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
}

```

```

void loop() {
  if (digitalRead(BUTTON_PIN) == LOW && !isMeasuring) {
    isMeasuring = true;
    startMeasurement();
    isMeasuring = false;
  }
}

void startMeasurement() {
  // Indicate measurement start
  digitalWrite(GREEN_LED, LOW);
  digitalWrite(YELLOW_LED, HIGH);

  // Show loading bar on LCD
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Proses Mengukur");
  lcd.setCursor(0, 1);

  for (int i = 0; i < 16; i++) {
    lcd.print("=");
    delay(200);
  }

  // Read soil moisture value (langsung ambil nilai mentah dari sensor)
  soilValue = analogRead(SOIL_SENSOR_PIN);

  // Convert soil value to percentage (based on the sensor range)
  moisturePercent = map(soilValue, 653, 508, 0, 100);
  moisturePercent = constrain(moisturePercent, 0, 100); // Limit the value
  between 0 and 100
}

```

```

// Print raw sensor value and percentage to Serial Monitor
Serial.print("Soil Sensor Value: ");
Serial.println(soilValue);
Serial.print("Moisture Percentage: ");
Serial.println(moisturePercent);

// Display result on LCD
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Kadar Air: ");
lcd.setCursor(0, 1);
lcd.print(moisturePercent);
lcd.print("%");

// Control LEDs based on moisture percentage
if (moisturePercent >= 0 && moisturePercent <= 13) {
  // Hijau (Kering)
  digitalWrite(GREEN_LED, HIGH);
  digitalWrite(YELLOW_LED, LOW);
  digitalWrite(RED_LED, LOW);
} else if (moisturePercent >= 14 && moisturePercent <= 18) {
  // Kuning (Sedang)
  digitalWrite(GREEN_LED, LOW);
  digitalWrite(YELLOW_LED, HIGH);
  digitalWrite(RED_LED, LOW);
} else if (moisturePercent >= 19 && moisturePercent <= 100) {
  // Merah (Basah)
  digitalWrite(GREEN_LED, LOW);
  digitalWrite(YELLOW_LED, LOW);
  digitalWrite(RED_LED, HIGH);
}

```

```
// Send moisture percentage to ThingSpeak
ThingSpeak.setField(1, moisturePercent); // Field 1 is for the moisture
percentage
ThingSpeak.writeFields(channelID, apiKey);

// Indicate measurement end
delay(1000);
}
```