Journal of Engineering, Technology & Applied Science
p-ISSN: 2721-7949, e-ISSN: 2721-8090

Original Research Paper

Integration of Weather API and IoT in an Automatic Irrigation System Based on Soil Moisture for Sugarcane Farming

Khasanul Khakim[1], **Rini Puji Astutik**[1*]

[1] *Department of Electrical Engineering, Faculty of Engineering, Universitas Muhammadiyah Gresik. Gresik, Indonesia.*

Article History
Received:
02.03.2025

Revised:
28.03.2026

Accepted:
06.04.2026

***Corresponding Author:**
Khasanul Khakim
Email:
khasanulkhakim@gmail.com

Abstract: National sugarcane productivity in Indonesia faces significant challenges due to climate change and a reliance on rain-fed lands, as is the case in Banjarejo Village, Grobogan Regency. Prolonged drought conditions frequently lead to unstable crop growth and diminished harvest yields. This study aims to design and implement an adaptive, Internet of Things (IoT)-based automated irrigation system by integrating soil moisture sensors with weather forecast data obtained via the OpenWeatherMap Application Programming Interface (API). The system utilizes an ESP32 microcontroller as its central control unit and solar panels as its primary energy source to support sustainable agriculture. The research methodology encompasses hardware design, employing Capacitive v1.2 soil moisture sensors, a 200 Wp solar module, and a 100 Ah battery; as well as software development utilizing JSON deserialization methods on the Blynk platform. Testing results demonstrate that the sensors exhibit a remarkably high level of accuracy, with an average error rate of merely 0.528%. The integration of weather data enables the system to make more intelligent irrigation decisions; the pump activates only when soil moisture levels fall below 40% and no rainfall is forecast. Over a one-month testing period, the system achieved a 100% success rate in executing its control logic and maintaining real-time data synchronization. The use of solar panels also proved effective, generating a peak power output of 87.4 Watts under bright, sunny conditions. In conclusion, this integration of IoT technology, real-time weather data, and renewable energy successfully enhances water-use efficiency and remote monitoring effectiveness, offering an innovative solution for sugarcane farmers as they navigate dynamic environmental conditions.

Keywords: ESP32, IoT, Smart Agriculture, Smart Irrigation, Weather API.

1. Introduction

The sugar industry is a highly strategic sector of the Indonesian economy, providing basic necessities and serving as a source of livelihood for sugarcane farmers. However, national sugar production still faces various challenges, including low sugarcane productivity, which leads to dependence on sugar imports [1]. One of the main factors affecting sugarcane productivity is climate change, particularly in rain-fed agricultural areas [2].

In areas such as Desa Banjarejo, Kecamatan Gabus, Kabupaten Grobogan, the dependence on rainfall makes the agricultural sector highly vulnerable to climate change and prolonged drought. This condition causes unstable sugarcane growth and may even significantly reduce crop yields [3]. Therefore, solutions are needed to optimize water use and adapt to environmental conditions efficiently.

Along with technological developments, the application of the Internet of Things (IoT) in agriculture has grown rapidly, particularly in automatic irrigation systems that use soil moisture sensors [4]. However, previous research shows that IoT-based irrigation systems generally focus on integrating sensors and real-time data, thus requiring further development to enable more adaptive decision-making in response to dynamic environmental conditions [5].

The developed system uses an ESP32 microcontroller as the control center [6], solar panels as the primary energy source [7], and the Blynk IoT platform for remote monitoring and control [8]. This integration enables the system to activate or pause irrigation based on soil moisture conditions and weather forecasts, thereby optimizing water use [9].

Based on this, there is a research gap in integrating IoT-based automatic irrigation systems with Application Programming Interface (API)-based weather data to improve the accuracy of watering decisions. Therefore, this study proposes the development of an IoT-based automatic irrigation system that integrates soil moisture sensors with weather data from the OpenWeatherMap API [10]. and uses solar panels as the primary power source [11].

This research aims to design and implement an automatic irrigation system that optimizes water use, increases farmer efficiency, and supports sustainable sugarcane productivity. The main contribution of this research is the integration of IoT technology, real-time weather data, and renewable energy systems into a single, adaptive, and efficient system, which is expected to be an innovative solution for modern agriculture, especially for sugarcane crops on rain-fed land.

2. Literature Review

2.1. Internet of Things

The Internet of Things (IoT) is a paradigm that enables physical devices to communicate and exchange data over the internet without requiring continuous human intervention. IoT systems typically consist of sensing devices, communication modules, data processing units, and cloud-based platforms that collectively support real-time monitoring and control. This concept has become fundamental across sectors, including agriculture, due to its ability to provide timely, accurate data for decision-making [12].

In agricultural applications, IoT plays a crucial role in enabling precision farming by integrating environmental sensors with automated control systems. Soil moisture, temperature, and humidity sensors continuously collect field data, which is then transmitted to microcontrollers for processing. The processed data can trigger automated responses, such as irrigation, based on predefined thresholds, improving efficiency and reducing resource wastage [13].

In this study, the IoT architecture consists of a capacitive soil moisture sensor v1.2 as the input device, an ESP32 microcontroller as the processing unit, and the Blynk platform for monitoring and control. This layered architecture reflects a typical IoT system design, integrating sensing, processing, and visualization into a unified system. Previous studies have demonstrated that such IoT-based irrigation systems significantly enhance automation, responsiveness, and operational efficiency [14].

2.2. Smart Agriculture and Water Use Efficiency

Smart agriculture refers to the application of advanced technologies such as IoT, artificial intelligence, and data analytics to improve agricultural productivity and sustainability. One of the primary objectives of smart agriculture is optimizing resource use, particularly water, a critical factor in crop production. Efficient water management is essential to ensure sustainable agricultural practices, especially in water-scarce regions [15].

Water use efficiency (WUE) is a key performance indicator in agricultural systems, representing the relationship between water consumption and crop yield. In crops such as sugarcane, water availability significantly affects growth, biomass accumulation, and sugar production. Over-irrigation can lead to nutrient leaching and water waste, while under-irrigation can reduce crop productivity. Therefore, maintaining optimal soil moisture conditions is essential [16].

The implementation of sensor-based automatic irrigation systems enables precise control of water application based on real-time soil conditions. Using soil moisture sensors allows irrigation to be applied only when necessary, reducing unnecessary water use. Previous research indicates that IoT-based irrigation systems improve water-use efficiency and support environmentally sustainable farming practices [17].

2.3. Sensor Data Integration and Weather APIs

Application Programming Interfaces (APIs) are standardized interfaces that allow different software systems to communicate and exchange data efficiently. In IoT-based agricultural systems, APIs play a crucial role in integrating external data sources, such as weather information, into the decision-making process. This integration enhances system intelligence by incorporating environmental predictions alongside real-time sensor data [18].

The combination of soil moisture sensor data with weather forecasts allows irrigation systems to become more adaptive and predictive. For example, if rainfall is predicted, the system can delay irrigation even if soil moisture levels are low, thereby preventing overwatering. This approach represents an advancement over traditional systems that rely solely on local sensor data without considering external environmental factors [19].

Weather service providers such as OpenWeatherMap offer real-time and forecast data through open APIs, enabling seamless integration into IoT platforms [20]. By leveraging these APIs, systems can retrieve parameters such as rainfall probability, temperature, and humidity, which are then processed to generate more accurate control decisions. Studies have shown that integrating weather data improves system accuracy, efficiency, and adaptability in smart farming applications [21].

2.4. Environmental Condition-Based Automatic Control System

An automatic control system is designed to regulate system behavior based on input signals and predefined logic. In irrigation systems, automatic control ensures that water is supplied to plants according to their requirements without manual intervention. These systems can be classified as open-loop or closed-loop, depending on whether feedback is used [22].

In this research, a closed-loop control system is implemented, where soil moisture data serves as feedback to determine irrigation actions. The system uses rule-based logic to control the water pump, ensuring that irrigation occurs only when necessary. This feedback mechanism allows the system to continuously adjust its operation in response to changing environmental conditions [23].

Additionally, integrating weather data enhances the control strategy by introducing predictive capabilities. The system not only reacts to current soil conditions but also anticipates future environmental changes. This combination of reactive and predictive control improves system efficiency and reduces unnecessary water usage, making it suitable for sustainable agricultural applications [24].

2.5. The Role of Sensors and Microcontrollers in IoT Systems

Sensors are essential components in IoT systems as they provide real-time data from the physical environment. In agricultural applications, soil moisture sensors are widely used to measure soil volumetric water content. Accurate sensing is critical because irrigation decisions depend heavily on the reliability of the collected data [25].

The capacitive soil moisture sensor v1.2 is commonly used due to its advantages over resistive sensors, including higher durability and resistance to corrosion. It operates by measuring changes in capacitance caused by variations in soil moisture levels. This makes it more stable and suitable for long-term deployment in agricultural environments [26].

Microcontrollers, such as the ESP32, act as the central processing unit in IoT systems. The ESP32 is widely adopted due to its built-in Wi-Fi capability, low power consumption, and high processing performance. It enables seamless communication between sensors and cloud platforms, allowing real-time data transmission, processing, and system control [6].

2.6. System Monitoring Using the IoT Platform (Blynk)

IoT platforms provide an interface for remotely monitoring and controlling connected devices. Blynk is a widely used IoT platform that supports various microcontrollers, including ESP32, and offers an intuitive user interface for system interaction. It allows users to visualize sensor data and control devices through mobile or web applications [27].

One of the key advantages of Blynk is its flexibility and ease of use. The platform provides drag-and-drop widgets that enable users to design custom dashboards without extensive programming knowledge. This simplifies the development process and makes IoT systems more accessible to users with varying technical backgrounds [28].

Blynk utilizes cloud-based infrastructure, enabling real-time data storage and remote access. This enhances system reliability and scalability, as users can monitor and control the system from anywhere with an internet connection. Research shows that using IoT platforms like Blynk improves monitoring efficiency, control flexibility, and overall system performance [29].

2.7. Utilization of Solar Energy in Irrigation Systems

Renewable energy plays an important role in supporting sustainable agricultural systems. Solar energy, in particular, is widely used due to its availability, environmental friendliness, and long-term cost-effectiveness. Solar panels convert sunlight into electrical energy, which can power irrigation systems in remote areas [30].

In off-grid agricultural settings, solar-powered irrigation systems provide an independent energy source, reducing reliance on conventional electricity. This is especially beneficial in rural or isolated regions where access to the electrical grid is limited or unavailable. The integration of solar energy ensures continuous system operation without significantly increasing operational costs [31].

By combining solar energy with IoT-based irrigation systems, the overall sustainability of the system is increased. This not only reduces carbon emissions but also supports efficient resource management. Previous studies have shown that solar-powered irrigation systems significantly contribute to sustainable agriculture by increasing energy efficiency and reducing environmental impacts [32].

3. Methodology

3.1. System Design

System design is a stage that generally describes the components used as inputs and outputs, as well as the system's workflow, which proceeds sequentially from start to finish. The entire process is depicted in the system block diagram shown in Figure 1.

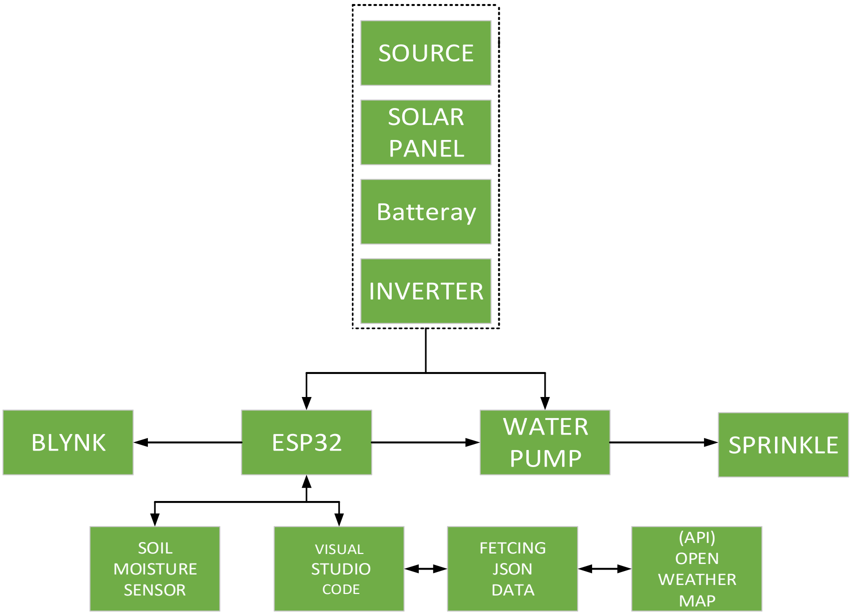


Figure 1. System Block Diagram

Based on Figure 1, it can be explained that there are components used as the main source in this system, namely solar power generators consisting of solar panels used to absorb solar energy and convert it into electrical energy, batteries used to store electrical energy that has been generated from solar panels, and inverters used to convert DC current sources to AC, input consisting of soil moisture sensors and data results from Visual Studio Code through a data parsing process using JSON taken from the Open Weather Map application programming interface (API) which is used as input, then all data is sent to the ESP32 microcontroller for processing, the soil moisture sensor and the Open Weather Map application programming interface (API) are used as determinants of whether the process will continue to the output or will continue to be processed. The results of the soil moisture sensor and data obtained from the OpenWeatherMap application programming interface (API) will be displayed on an IoT platform via a Blynk application for remote monitoring, with output to a water pump and sprinkler to drive them.

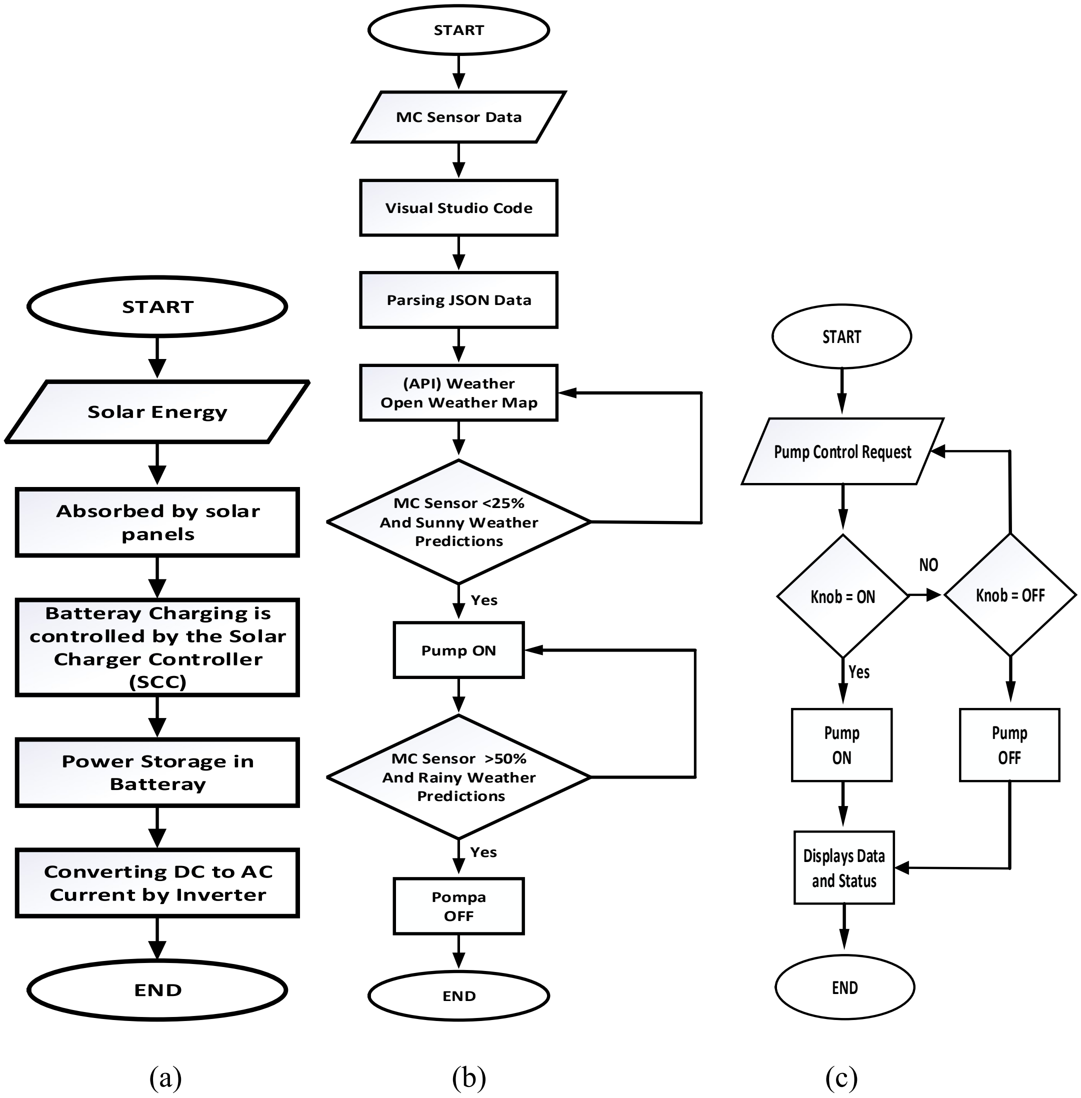


Figure 2. Flowchart for:
(a) Solar Panel Working System
(b) Main Work System
(c) Working System in the BLYNK Application

Based on Figure 2 (a), the working system process begins when the solar panel absorbs sunlight energy and converts it into electrical energy. The electrical energy produced by the solar panel is regulated by the Solar Charger Controller (SCC) to maintain a stable power flow before being stored in the battery. Then the inverter converts it into direct current (DC), and then the direct current (DC) is converted into alternating current (AC) using a (DC) to (AC) inverter before the power source is distributed to the system.

Based on Figure 2 (b), the process begins with data reading in Visual Studio Code using the JSON deserialization method (Arduino JSON) from the OpenWeatherMap weather application programming interface (API) and a soil moisture sensor, then sending the data to the ESP32 for further processing. Data taken from the Open Weather Map weather application programming interface (API) is in the form of weather forecasts (sunny, cloudy, rainy), weather hours, humidity (air humidity), Celsius temperature, and Fahrenheit temperature. Soil moisture sensor data is recorded as humidity values or soil water content. If the soil moisture value is more than 25% and the weather prediction is (sunny, cloudy, rainy), the sensor reading will continue to repeat until the soil moisture value is less than 25%, after which the ESP32 microcontroller will give a command to turn on the pump until the soil moisture value is more than 50%, and the pump will turn off when the soil moisture is more than 50%. In the event of flooding or standing water, the automatic system will not activate because the sensor will detect high humidity under the system's control logic. The pump will only activate if the soil moisture is at the dry end and there is no forecast for rain. Therefore, in flooding or standing water conditions, the system is safe, and no additional watering is required.

Based on Figure 2 (c), the system starts from the data obtained from the application programming interface (API) of the OpenWeatherMap weather and the soil moisture sensor, which will be sent to the BLYNK application software by ESP32, which will later be used for a remote monitoring system. Then the BLYNK application displays the data value parameters from the application programming interface (API) of the OpenWeatherMap weather and the soil moisture sensor. The values displayed by the application programming interface (API) of the OpenWeatherMap are in the form of weather prediction data (sunny, cloudy, rainy), and the values displayed by the soil moisture sensor are in the form of humidity values or water content in the soil. Displays the water pump status: ON (sunny) or OFF (cloudy or rainy). When ON, the pump flows water to the sprinkler and sprays water onto the fields; when OFF, the pump is off.

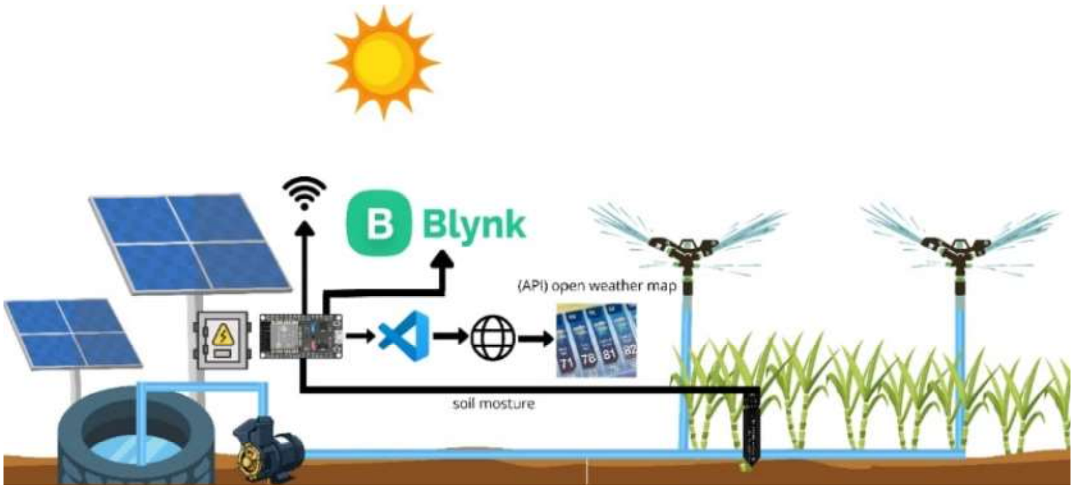


Figure 3. Design in the field

Figure 3 illustrates the system design in the field. This IoT-based automatic watering system utilizes weather APIs and soil moisture sensors for sugarcane plants. This system is designed to monitor and water crops according to a program generated by parsing weather data (APIs) from OpenWeatherMap and soil moisture sensors. This system is expected to achieve greater efficiency in monitoring and watering sugarcane plants.

3.1. Hardware Design Planning

Hardware design is a description of the physical components to be used as a reference for implementing a real-world system. In this study, a monitoring system was designed to assess the performance of the electronic circuit before it was implemented in the system being developed. The electronic circuit used consists of input, process, and output components, an explanation of which can be seen in Figure 4.

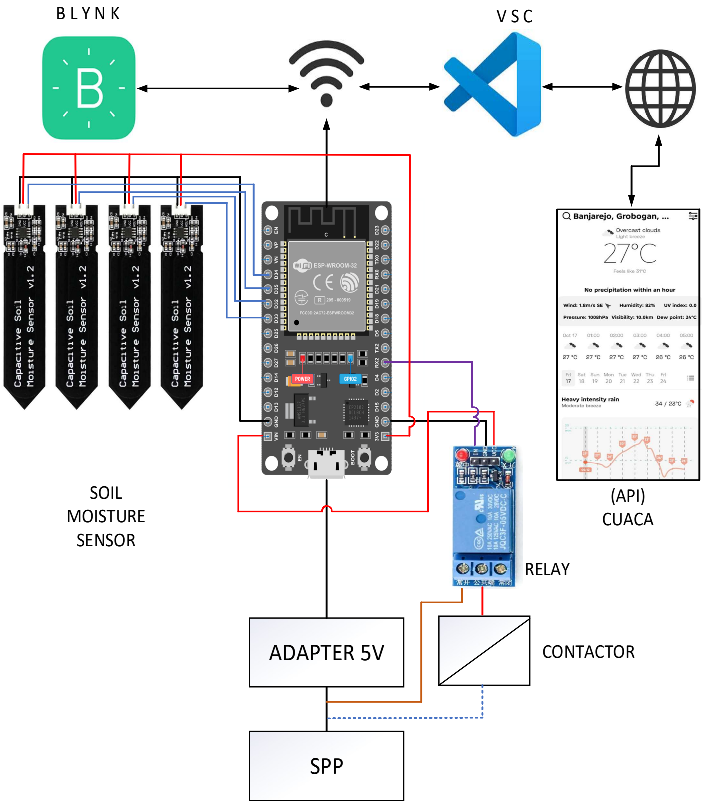


Figure 4. Monitoring System Design

The monitoring system design in this study uses one sensor and the results of weather API parsing using Visual Studio Code as inputs, and relays as outputs connected to pins on the ESP32 microcontroller. The four soil moisture sensors are connected to GPIO pin 33 for sensor one, GPIO pin 32 for sensor two, GPIO pin 35 for sensor three, and GPIO pin 34 for sensor four, and the relay is connected to GPIO pin 17.

The ESP32 microcontroller was chosen because it includes a Wi-Fi module, which is useful for sending sensor data and weather API parsing results to the BLYNK application, which will later be used for remote monitoring. This aims to make it easier for farmers, even if they don't have the free time to visit the fields. The ESP32 microcontroller uses a 5V adapter taken from the output of a solar power plant (spp).

3.2. Software Design Planning

The software design in this study has two components: retrieving weather data from the API using Visual Studio Code and JSON, and enabling remote monitoring via the BLYNK application.

1) Parsing JSON Data

As shown in Figure 5, parsing JSON means converting it from text to a structured format. The ESP32 nodeMCU sends an HTTP GET request to the API provider at https://ibnux.github.io/open-importer/cuaca/501233.json for OpenWeatherMap data. The API replies with a JSON response. Ensure the response code is 200 to confirm success. When data is received, use ARDUINO JSON to handle it. After the ESP32 nodeMCU receives JSON data, it parses and deserializes it using the

Arduino JSON library.

Figure 5. JSON Data Parsing Schematic

The data is then rendered more easily accessible. The JSON parsing program code can be seen in Figure 6.

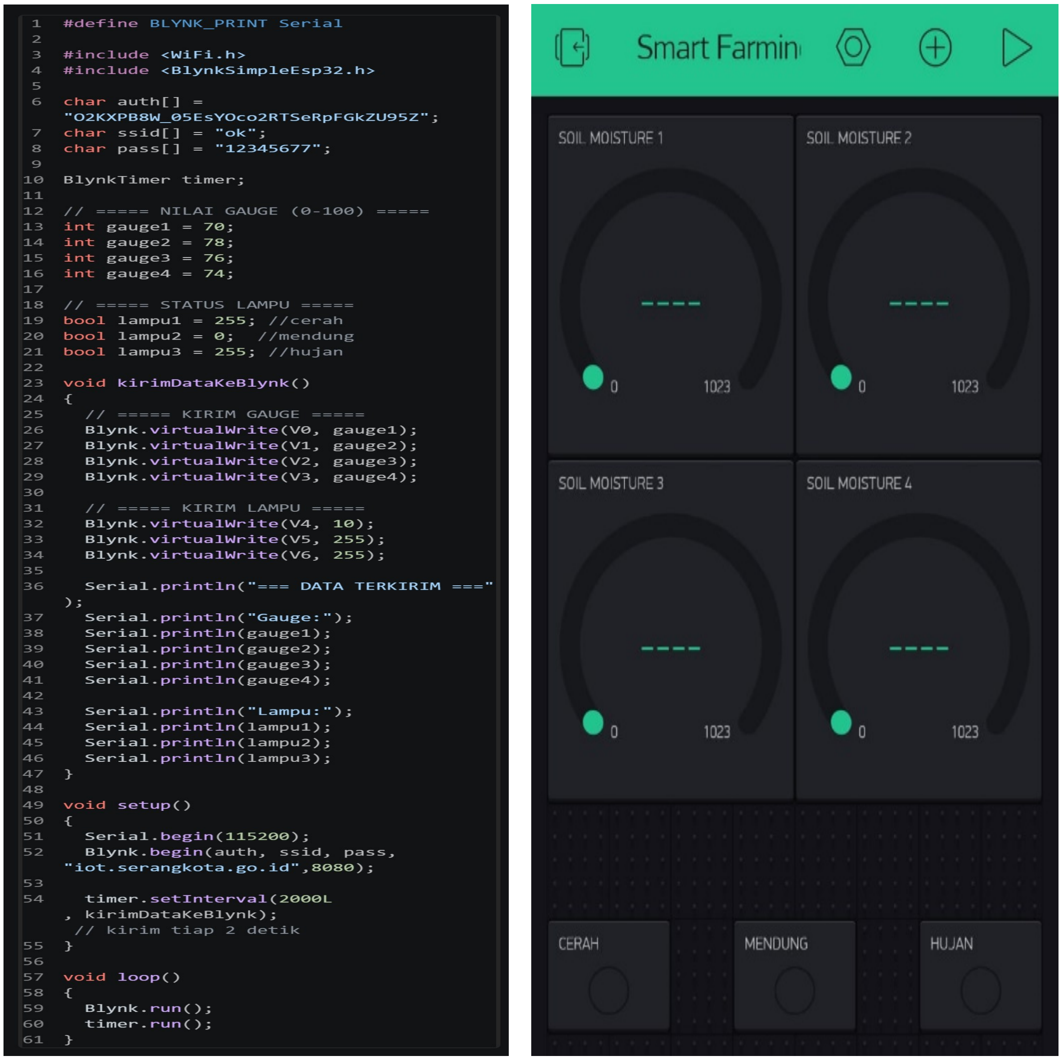
```
#include <Arduino.h>
#include <WiFi.h>
#include <HTTPClient.h>
#include <ArduinoJson.h>

const char* ssid = "khakim";
const char* password = "12345678";

// API resmi BMKG
const char* url =
"https://api.bmkg.go.id/publik/prakiraan-
cuaca?adm4=33.15.08.2014";

unsigned long lastUpdate = 0;
const unsigned long interval = 60000;

void ambilCuaca() {
  HTTPClient http;
  http.begin(url);
  int httpCode = http.GET();

  if (httpCode == 200) {
    String payload = http.getString();
    DynamicJsonDocument doc(32768);
    deserializeJson(doc, payload);

    JsonObject lokasi = doc["lokasi"];
    JsonObject cuaca =
doc["data"][0]["cuaca"][0][0];
```

```
Serial.println("====== UPDATE CUACA
BMKG ======");
    Serial.printf("Lokasi: %s, %s\n", (const
char*)lokasi["desa"], (const
char*)lokasi["kecamatan"]);
    Serial.printf("Kondisi: %s\n", (const
char*)cuaca["weather_desc"]);
    Serial.printf("Suhu: %d C, Kelembapan:
%d%%\n\n", cuaca["t"].as<int>(),
cuaca["hu"].as<int>());
  } else {
    Serial.println("HTTP request gagal");
  }

  http.end();
}

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("\nWiFi Terhubung");
  ambilCuaca();
}

void loop() {
  if (millis() - lastUpdate >= interval) {
    lastUpdate = millis();
    ambilCuaca();
  }
}
```

Figure 6. Program code for parsing JSON data

2) Monitoring System Design

The software design for remote monitoring on the BLYNK application in this research includes several input components. These consist of data taken from the weather application programming interface (API) using Visual Studio Code and soil moisture sensors. Both data sources can be monitored in real time via the IoT platform, specifically the BLYNK application. Before monitoring, data processing is managed by the ESP32, which is programmed with Visual Studio Code. The steps for connecting the ESP32 with the BLYNK application can be seen in Figure 7.



```
#define BLYNK_PRINT Serial

#include <WiFi.h>
#include <BlynkSimpleEsp32.h>

char auth[] =
"O2KXPB8W_05EsYOco2RTSeRpFGkZU95Z";
char ssid[] = "ok";
char pass[] = "12345677";

BlynkTimer timer;

// ===== NILAI GAUGE (0-100) =====
int gauge1 = 70;
int gauge2 = 78;
int gauge3 = 76;
int gauge4 = 74;

// ===== STATUS LAMPU =====
bool lampu1 = 255; //cerah
bool lampu2 = 0;   //mendung
bool lampu3 = 255; //hujan

void kirimDataKeBlynk()
{
  // ===== KIRIM GAUGE =====
  Blynk.virtualWrite(V0, gauge1);
  Blynk.virtualWrite(V1, gauge2);
  Blynk.virtualWrite(V2, gauge3);
  Blynk.virtualWrite(V3, gauge4);

  // ===== KIRIM LAMPU =====
  Blynk.virtualWrite(V4, 10);
  Blynk.virtualWrite(V5, 255);
  Blynk.virtualWrite(V6, 255);

  Serial.println("=== DATA TERKIRIM ==="
);
  Serial.println("Gauge:");
  Serial.println(gauge1);
  Serial.println(gauge2);
  Serial.println(gauge3);
  Serial.println(gauge4);

  Serial.println("Lampu:");
  Serial.println(lampu1);
  Serial.println(lampu2);
  Serial.println(lampu3);
}

void setup()
{
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass,
"iot.serangkota.go.id",8080);

  timer.setInterval(2000L
, kirimDataKeBlynk);
 // kirim tiap 2 detik
}

void loop()
{
  Blynk.run();
  timer.run();
}
```

(a) (b)

Figure 7. Blynk Application:
(a) Code
(b) Display

3.3. Solar Panel Planning

Solar panels serve as a key component that will provide the primary source of electrical energy. Therefore, careful calculations of solar panel and battery capacity are essential. Inverter selection is also crucial, as it significantly affects the efficiency and effectiveness of electrical energy use in this device's design. A detailed explanation includes:

1) Solar Panel Calculation

This first point explains the calculation process for solar panels used as a power source for a full-day agricultural irrigation system. The calculation begins by determining the total power requirement using the Equation (1).

$$Wh = P \, x \, h \tag{1}$$

Where:
Wh = Power consumption/day (Watt hours)
P = Load power used (Watts)
h = Hourly usage (Hours)

The power capacity used by each component in an assumed 12-hour time can be seen in Table 1.

Tabel 1. Power Required by the System

No	Components	Power (Watts)	Usage (Hours)	Total	Total Power (Wh)
1	water pump PS-128 BIT	290	2	1	580
2	Contactor	10	2	1	20
3	ESP32 Microcontroller	1.25	12	1	15
4	1-Channel Relay Module	0.4	12	1	4.8
5	Capacitive Soil Moisture Sensor v1.2	0.05	12	1	0.6
				Total	620.4

As shown in Table 1, the power used by each component can be calculated to determine the required capacity of the solar panel. In Indonesia, the optimal sunlight intensity occurs from 09.00 WIB to 14.00 WIB, when sunlight falls perpendicular to the solar panel and the environmental temperature is high. Thus, the effective time for absorbing solar energy can be assumed to be 5 hours per day. Based on this, the solar panel's capacity can be calculated using Equation (2).

$$Wp = \frac{Wh}{t} \tag{2}$$

Where:
Wp = Solar panel capacity (Watt peak)
Wh = Power consumption/day (Watt hours)
t = 5 hours

This study requires a solar panel capacity of 620.4/5 hours = 124.08 Wp or 200 Wp to meet the load power requirements, ensuring optimal system operation.

2) Battery Capacity Calculation

The second point discusses calculating the battery capacity used in the system. Batteries store electrical energy when sunlight intensity is high, and this energy is then reused when light intensity decreases or when the system requires additional power. To obtain the appropriate battery capacity, precise and accurate calculations are required. To determine battery capacity, Equation (3) can be used:

$$Ah = \frac{Wh}{v} \tag{3}$$

Where:
Ah = Battery Capacity (Ampere-hours)
Wh = Daily Usage (Watt-hours)
V = Voltage (Volt)

In this study, the required battery capacity is 620.4 Wh / 12 = 51.7 Ah. To ensure optimal operation, a 12 V 100 Ah battery is used.

3) Inverter Capacity Calculation

This third point explains the design of an inverter, which converts direct current (DC) to alternating current (AC) so the resulting electrical energy can operate the pump. In the design process, calculating inverter capacity is required to meet load requirements. This inverter capacity can be determined using Equation (4):

$$P_{total} = W_{max} + (25\% \; x \; W_{max}) \tag{4}$$

Where:
P_{total} = Total required power capacity (Watts)
W_{max} = Peak load power (Watts)
25% = Safety faktor/reserver power to accommodate initial starting currents (inrush current)

In this study, the required inverter capacity is 620.4 Wh. Inverter design, the lost power calculation, or the amount of power lost during system operation, is performed using Equation (5):

$$P_{loss} = P_{in} - P_{out} \tag{5}$$

Where:
P_{loss} = Power lost (Watts)
P_{in} = Input power (Watts)
P_{out} = Power out (Watts)

The P_{in} comes from the total power produced by the battery, which is 1,200 Watts, while to determine the P_{out} value, Equation (6) is used:

$$P_{out} = P_{in} \; x \; \eta \tag{6}$$

Where:
P_{out} = Power out (Watts)
P_{in} = Input power (Watts)
η = Efficiency

The inverter efficiency is 90%, so the Pout is 1,200 Watts × 90% = 1,080 Watts.

4. Finding and Discussion

After the prototype process was completed, a series of tests was conducted, including sensor and solar panel tests, as well as a comprehensive evaluation of system performance.

Figure 8. System Design Prototype Results

This testing aimed to assess the system to ensure that the results obtained were in accordance with the initial design. All testing activities were conducted in the sugarcane plantation area of Desa Banjarejo, Kecamatan Gabus, Kabupaten Grobogan, Central Java. The data obtained was then analyzed to assess the performance and reliability of the system, as seen in Figure 7.

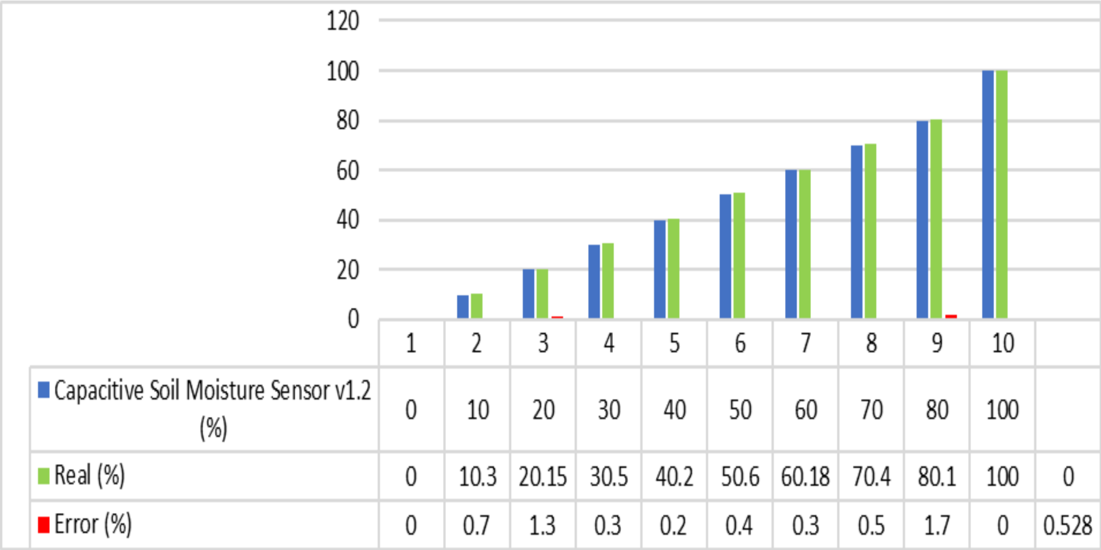
4.1. Sensor Testing

Sensor testing is divided into two parts: testing the Capacitive Soil Moisture Sensor v1.2 and parsing JSON data from OpenWeatherMap. Testing is carried out in stages using sensor data and JSON-parsed data to determine whether its performance meets the system's functional requirements. This testing compares the detection results with the actual values. Sensor testing is carried out with Equation (7), which can be seen in Table 2.

$$Error(\%) = \frac{(Real\ Value - Sensor\ Value)}{Real\ Value} \times 100 \tag{7}$$

Table 2. Capacitive Soil Moisture Sensor v1.2 Test Data

No.	Capacitive Soil Moisture Sensor v1.2	Real (%)	Error (%)
1	0	0	0
2	10	10.3	0.7
3	20	20.15	1.3
4	30	30.5	0.3
5	40	40.2	0.2
6	50	50.6	0.4
7	60	60.18	0.3
8	70	70.4	0.5
9	80	80.10	1.7
10	100	100	0
		Averge Error	0.528



	1	2	3	4	5	6	7	8	9	10	
🟦 Capacitive Soil Moisture Sensor v1.2 (%)	0	10	20	30	40	50	60	70	80	100	
🟩 Real (%)	0	10.3	20.15	30.5	40.2	50.6	60.18	70.4	80.1	100	0
🟥 Error (%)	0	0.7	1.3	0.3	0.2	0.4	0.3	0.5	1.7	0	0.528

Figure 9. Graph of Capacitive Soil Moisture Sensor v1.2 test results

As seen from Table 2, the results of the test show that the value of the Capacitive Soil Moisture sensor v1.2 with the actual value has an average error of 0.528%, which means that the Capacitive Soil Moisture sensor v1.2 is hereby declared accurate, as seen in Figure 8.

Figure 10 shows the placement of the v1.2 capacitive soil moisture sensor, positioned close to the sprinkler to maximize soil data collection accuracy. This placement ensures the sensor detects immediate changes in soil moisture after irrigation, thus more accurately reflecting field conditions. The v1.2 sensor measures changes in capacitance caused by soil water content; higher water content results in higher capacitance readings.

Figure 10. Testing of the Capacitive Soil Moisture Sensor v1.2

As shown in Figure 11, parsing JSON data for 1 month produces accurate real-time weather prediction values. In this system, the Capacitive Soil Moisture Sensor v1.2 and the API application programming interface (OpenWeatherMap) with JSON data parsing are used to improve performance and accuracy. As the main program of the system, equation (7) is used to calculate the reliability of the system's successful operation. The Capacitive Soil Moisture Sensor v1.2 measures soil water content and moisture by detecting changes in capacitance. The selection of the Capacitive Soil Moisture Sensor v1.2 offers greater corrosion resistance and higher accuracy. The correlation between the Capacitive Soil Moisture Sensor v1.2 and the JSON data parsing results is crucial for decision-making about the device's performance.

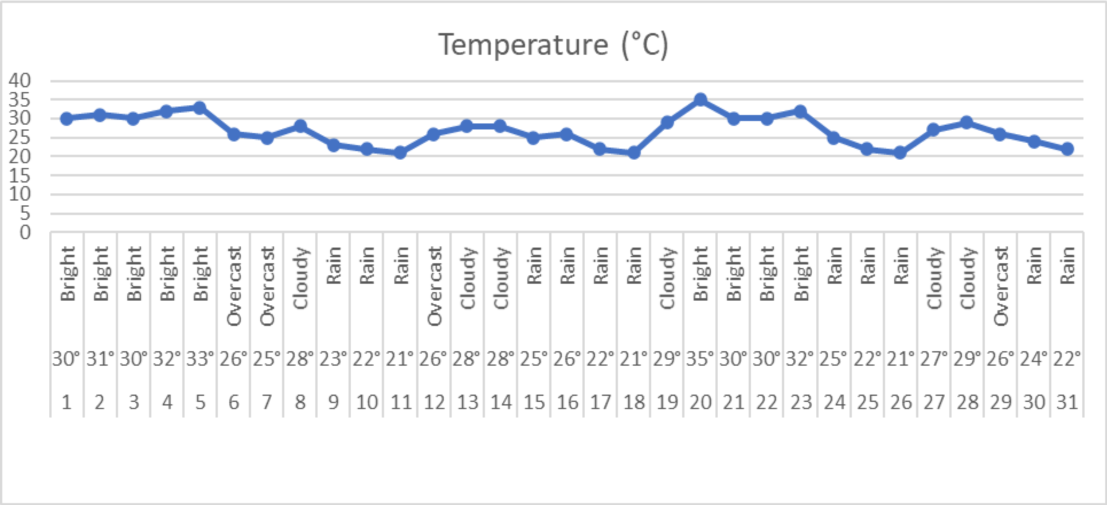


Figure 11. JSON Parsing Data Test Results

$$Reliability - \frac{(Number\ of\ Correct\ Conditions)}{Total\ Number\ of\ Tests} \times 100\% \quad (8)$$

4.2. Solar Panel Testing

This solar panel testing measures the voltage, current, and power generated by the solar power generation system (SPP) as a function of the incident sunlight intensity. The stages of solar panel testing can be seen in Figure 12.

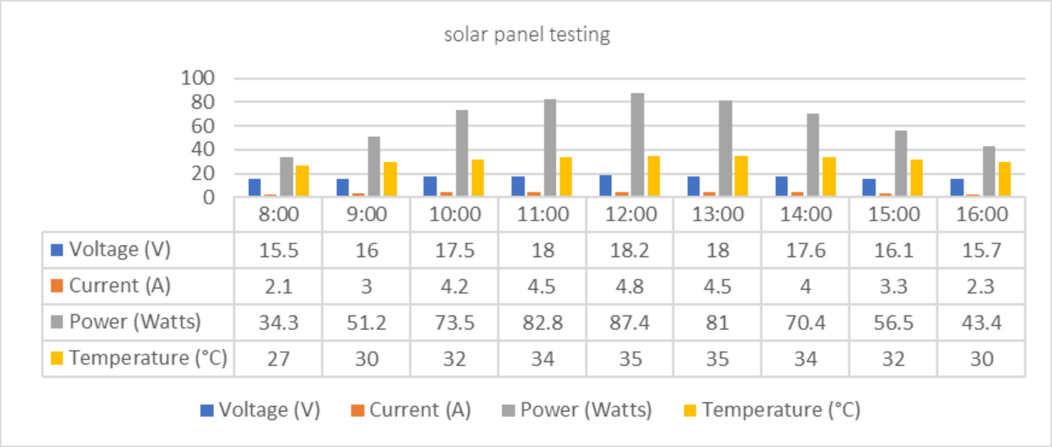
Figure 12. Solar Panel Testing

Figure 12 shows solar panels installed on the roof of a building near the plantation area to capture optimal sunlight and minimize the risk of loss or unwanted incidents if placed below. The results of the solar panel test can be seen in Table 3.

Table 3. Solar Panel Testing

Time (WIB)	Voltage (V)	Current (A)	Power (Watts)	Temperature (°C)
08:00	15.5	2.1	34.3	27
09:00	16.0	3.0	51.2	30
10:00	17.5	4.2	73.5	32
11:00	18.0	4.5	82.8	34
12:00	18.2	4.8	87.4	35
13:00	18.0	4.5	81.0	35
14:00	17.6	4.0	70.4	34
15:00	16.1	3.3	56.5	32
16:00	15.7	2.3	43.4	30

Based on Table 3, it can be seen that the solar panel test was conducted on 1/03/2026 in sunny weather. The solar panel test lasted for 8 hours starting from 08:00 to 16:00 WIB, with the highest voltage value at 12:00 WIB and tending to be high when the weather is sunny accompanied by a high temperature increase, with this it can be concluded that there is a correlation between weather conditions and voltage absorption by solar panels, while the voltage tends to decrease towards the afternoon, this also occurs when the weather is cloudy and the temperature decreases as can be seen in Table 5.



	8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00
🟦 Voltage (V)	15.5	16	17.5	18	18.2	18	17.6	16.1	15.7
🟧 Current (A)	2.1	3	4.2	4.5	4.8	4.5	4	3.3	2.3
⬜ Power (Watts)	34.3	51.2	73.5	82.8	87.4	81	70.4	56.5	43.4
🟨 Temperature (°C)	27	30	32	34	35	35	34	32	30

Figure 13. Solar Panel Testing Graph

4.3. Overall Tool Testing

The final testing phase was the overall system testing. This testing was conducted on a sugarcane plantation in Desa Banjarejo, Kecamatan Gabus, Kabupaten Grobogan. The purpose of this testing was to ensure that the system's performance met expectations and that the hardware and software were synchronized. This testing included the solar power plant (SPP), the sensors used, and the automated monitoring and control system on the sugarcane plantation.

Figure 14. Testing of Tools on Sugarcane Plantations

Based on Table 4, observations over a one-month period from March 1–31, 2026, show that the Internet of Things (IoT)-based automatic watering system demonstrated optimal performance in controlling soil moisture. The main parameters observed included soil moisture, ambient temperature, weather conditions, and electrical parameters such as voltage, current, and power.

The soil moisture values obtained from the sensors showed significant variation, ranging from

25% to 85%. Under low-humidity conditions (below 40%), the system automatically activated the water pump. This was observed on March 1, 4, and 20, when soil moisture values were in the 25–30% range, and the pump was on. Conversely, when soil moisture exceeded 50%, the pump tended to be off. This indicates that the system was operating according to its designed control logic, which is to maintain optimal soil moisture. Furthermore, weather conditions obtained from the parsed JSON data also affected soil moisture. During rainy conditions, soil moisture was high, reaching 70% to 85%, so the pump remained inactive. Meanwhile, in sunny conditions, soil moisture tends to decrease, so the pump is activated to maintain soil moisture.

Table 4. Overall Tool Testing

No	Date	Average Soil moisture Sensor (%)	Data Parsing Json temperature	Weather Json Data Parsing	Highest Voltage (V)	Highest Current (A)	Highest Power (Watts)	Pump
1	1/03/2026	30	30°	Bright	18.2	4.8	87.4	ON
2	2/03/2026	60	31°	Bright	18	4.2	85.2	OFF
3	3/03/2026	60	30°	Bright	19.2	4.9	88.2	OFF
4	4/03/2026	30	32°	Bright	20	4.9	88.5	ON
5	5/03/2026	50	33°	Bright	20	5.1	88.9	OFF
6	6/03/2026	60	26°	Overcast	15.1	3.1	70,1	OFF
7	7/03/2026	62	25°	Overcast	14.2	2.9	69.9	OFF
8	8/03/2026	60	28°	Cloudy	17.5	4.1	80.2	OFF
9	9/03/2026	70	23°	Rain	12.2	3.2	60.2	OFF
10	10/03/2026	78	22°	Rain	11.4	3.5	59.5	OFF
11	11/03/2026	70	21°	Rain	11.1	3.1	55.5	OFF
12	12/03/2026	60	26°	Overcast	15.3	3.1	69.1	OFF
13	13/03/2026	55	28°	Cloudy	17.3	4.4	75.5	OFF
14	14/03/2026	53	28°	Cloudy	17.5	4.4	76.1	OFF
15	15/03/2026	70	25°	Rain	11.7	3.1	60.1	OFF
16	16/03/2026	75	26°	Rain	11.5	2.9	58.3	OFF
17	17/03/2026	82	22°	Rain	12.1	2.8	57.7	OFF
18	18/03/2026	85	21°	Rain	11.9	2.7	56.9	OFF
19	19/03/2026	60	29°	Cloudy	17.8	4.2	74.5	OFF
20	20/03/2026	25	35°	Bright	19.8	5.1	90.1	ON
21	21/03/2026	60	30°	Bright	18.2	4.4	88.5	OFF
22	22/03/2026	70	30°	Bright	18.9	4.6	87.5	OFF
23	23/03/2026	73	32°	Bright	18.2	4.2	87.2	ON
24	24/03/2026	70	25°	Rain	11.1	3.1	55.5	OFF
25	25/03/2026	76	22°	Rain	12.3	2.3	52.2	OFF
26	26/03/2026	80	21°	Rain	11.5	2.5	51.1	OFF
27	27/03/2026	65	27°	Cloudy	17.9	4.1	74.5	OFF
28	28/03/2026	66	29°	Cloudy	16.3	4.4	71.9	OFF
29	29/03/2026	69	26°	Overcast	15.3	3.1	69.1	OFF
30	30/03/2026	80	24°	Rain	11.1	2.9	56.1	OFF
31	31/03/2026	80	22°	Rain	11.3	2.5	55.5	OFF

In terms of electrical parameters, the highest power is generally observed in sunny conditions, indicating that the energy source is being utilized. Conversely, in rainy conditions, power tends to decrease as sunlight intensity decreases. Based on 31 days of testing, the system shows a 100%

success rate in executing the control logic. This is indicated by the match between soil moisture conditions and the system's decision to activate or deactivate the pump. Thus, the system has a very high level of reliability. Although the 100% result, this value is based on a limited test scenario (31 days), so long-term testing is needed for further validation.

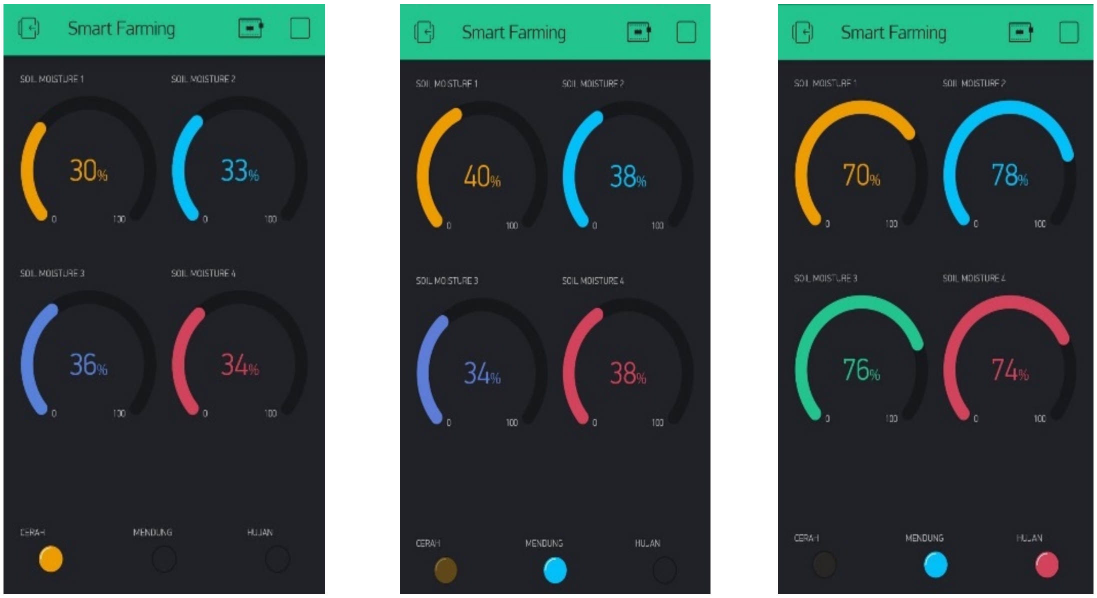


Figure 15. BLYNK Display When the System Working

Figure 15 shows three weather conditions, sunny, cloudy, and rainy, displayed in the Blynk application from Weather API data. The display provides real-time weather information, which is used as an additional parameter for irrigation decisions. The Weather API and IoT integration in the automatic soil moisture-based irrigation system for sugarcane fields operate optimally. Steps include sensor readings, JSON parsing, microcontroller data processing, and solar energy use. The system synchronizes all components to ensure data is displayed accurately in the application. By combining soil moisture and weather data, the system adapts and more efficiently determines watering needs. The system relies on multiple parameters to increase water-use efficiency, improve reliability, and support sustainable IoT-based smart agriculture.

5. Conclusion

Testing of the IoT-based automated irrigation system at the sugarcane plantation in Banjarejo Village yielded highly accurate and reliable results. The Capacitive v1.2 soil moisture sensor operated with a high degree of precision, exhibiting an average error rate of just 0.528%. The system's performance was further optimized through the integration of weather data from the OpenWeatherMap API, presented via the Blynk application; ensuring that irrigation decisions were based not only on soil conditions but also on real-time weather forecasts.

In terms of energy efficiency, the use of solar panels proved effective as a self-sustaining power source, demonstrating a strong correlation between solar intensity and generated power; peak power output reached 87.4 Watts during sunny daytime conditions. Throughout the 31-day testing period in March 2026, the automated control system achieved a 100% success rate in executing its operational logic. The pump consistently activated when soil moisture levels dropped below 40% and remained inactive during rainfall or when moisture levels were sufficient; consequently, the system proved highly effective in enhancing water usage efficiency and supporting the implementation of sustainable smart agriculture practices.

For future development, the use of more adaptive control methods, such as fuzzy logic or machine learning, as well as longer-term testing and larger-scale field testing, is recommended. This system has the potential to be developed into a more reliable, efficient, and sustainable smart irrigation solution to support technology-based agriculture.

References

[1] E. Johnson, H. Jusril, L. Pratiwi, S. Trisnasari, A. M. Thow, and C. N. Rachmi, "Looking towards the sweet, sweet future: a political economy analysis of sugar and nutrition policy in Indonesia," *Public Health Nutr.*, vol. 28, no. 4, pp. 1–9, 2025, doi: 10.1017/S1368980025100566.

[2] A. Muis, S. Wulandari, I. K. Ardana, and A. Wahyudi, "Understanding climate adaptation practices among small-scale sugarcane farmers in Indonesia: The role ofclimate risk behaviors, farmers' support systems, and crop-cattle integration," *Resour. Environ. Sustain.*, vol. 13, no. 10, pp. 1–10, 2023, doi: 10.1016/j.resenv.2023.100129.

[3] N. Qin, Q. Lu, G. Fu, J. Wang, K. Fei, and L. Gao, "Assessing the drought impact on sugarcane yield based on crop water requirements and standardized precipitation evapotranspiration index," *Agric. Water Manag.*, vol. 275, no. 7, pp. 1–15, 2023, doi: 10.1016/j.agwat.2022.108037.

[4] S. Gupta *et al.*, "Smart agriculture using IoT for automated irrigation, water and energy efficiency," *Smart Agric. Technol.*, vol. 12, no. 6, pp. 1–11, 2025, doi: 10.1016/j.atech.2025.101081.

[5] A. Morchid, Z. Said, A. Y. Abdelaziz, P. Siano, and H. Qjidaa, "Fuzzy logic-based IoT system for optimizing irrigation with cloud computing: Enhancing water sustainability in smart agriculture Abdennabi," *Smart Agric. Technol.*, vol. 11, no. 3, pp. 1–14, 2025, doi: 10.3390/iot4030012.

[6] G. P. Pereira, M. Z. Chaari, and F. Daroge, "IoT IoT-Enabled Smart Drip Irrigation System Using ESP32," *Mol. Divers. Preserv. Int.*, vol. 4, no. 7, pp. 221–243, 2023, doi: 10.3390/iot4030012.

[7] D. Balamurali *et al.*, "A solar-powered, internet of things (IoT)-controlled water irrigation system supported by rainfall forecasts utilizing aerosols: a review," *Environ. Dev. Sustain.*, vol. 10, no. 12, pp. 1–40, 2025, doi: 10.1007/s10668-024-05953-z.

[8] A. Abu and A. A. A. Shabaneh, "Design of a smart hydroponics monitoring system using an ESP32 microcontroller and the Internet of Things," *MethodsX*, vol. 11, no. 6, pp. 1–9, 2023, doi: 10.1016/j.mex.2023.102401.

[9] M. Afzal, I. A. Saeed, M. N. Sohail, M. H. Saad, and M. R. Sarker, "IoT-Enabled Adaptive Watering System With ARIMA-Based Soil Moisture Prediction for Smart Agriculture," *IEEE Access*, vol. 13, no. 12, pp. 27714–27728, 2025, doi: 10.1109/ACCESS.2025.3532447.

[10] Y. Dong, B. Werling, Z. Cao, and G. Li, "Implementation of an in-field IoT system for precision irrigation management," *fortiers*, vol. 10, no. 2, pp. 1–11, 2024, doi: 10.3389/frwa.2024.1353597.

[11] N. Rodrigues *et al.*, "Sugarcane Yield Estimation Using Satellite Remote Sensing Data in Empirical or Mechanistic Modeling: A Systematic Review," *Mol. Divers. Preserv. Int.*, vol. 10, no. 2, pp. 1–26, 2024, doi: 10.3390/ rs16050863.

[12] M. Španer, M. Trunti˘, and D. Hercog, "IoT-Based Off-Grid Solar Power Supply: Design, Implementation, and Case Study of Energy Consumption Control Using Forecasted Solar Irradiation," *Appl. Sci.*, vol. 15, no. 12, pp. 1–28, 2025, doi: 10.3390/ app152212018.

[13] Y. Boukri *et al.*, "Analysis and experimental implementation of affordable smart irrigation system using IoT to reduce agricultural costs and minimize water usage," *Appl. Water Sci.*, vol. 10, no. 8, pp. 1–13, 2026, doi: 10.1007/s13201-025-02727-4.

[14] I. A. Ameen and M. A. Al-sheikh, "Wireless Sensor Networks for Smart Gardening : ESP-NOW and Blynk IoT Integration for Water and Energy Optimization," *Int. J. Comput. Electron. Asp. Eng.*, vol. 5, no. 3, pp. 98–112, 2024, doi: 10.26706/ijceae.5.3.20240804.

[15] X. Ye, N. Shirakawa, K. Sano, and S. Zhang, "An IoT-based predictive irrigation scheduling framework for precision soil moisture control in greenhouses," *Agric. Water Manag.*, vol. 329, no. 4, pp. 1–20, 2026, doi: 10.1016/j.agwat.2026.110334.

[16] F. G. Delgado, J. M. T. Arbeláez, and D. F. C. Castañeda, "Assessment of photosynthetic and photothermal responses to water limitation in three sugarcane (Saccharum spp .) genotypes

using the photoacoustic technique," *J. Saudi Soc. Agric. Sci.*, vol. 7, no. 6, pp. 1–14, 2025, doi: 10.1007/s44447-025-00050-7.

[17] M. A. Alsultan *et al.*, "A Battery-Less UHF RFID Sensor for Soil Moisture Monitoring," *IEEE J. RADIO Freq. Identif.*, vol. 9, no. 5, pp. 286–294, 2025, doi: 10.1109/JRFID.2025.3572843.

[18] R. Pujadas, E. Valderrama, and W. Venters, "The value and structuring role of web APIs in digital innovation ecosystems : The case of the online travel ecosystem," *Res. Policy*, vol. 53, no. 2, pp. 1–26, 2024, doi: 10.1016/j.respol.2023.104931.

[19] G. S. P. Lakshmi, P. N. Asha, G. Sandhya, S. V. Sharma, S. Shilpashree, and S. G. Subramanya, "An intelligent IOT sensor coupled precision irrigation model for agriculture," *Meas. Sensors*, vol. 25, no. 9, pp. 1–10, 2023, doi: 10.1016/j.measen.2022.100608.

[20] N. Yanti, F. Z. Rachman, D. Lesmideyarti, and M. E. Prasetyo, "Optimizing Real-Time Weather Data and Information Services with Frameworks and Application Programming Interfaces," *JITE (Journal Informatics Telecommun. Eng.*, vol. 9, no. 1, pp. 430–441, 2026, doi: 10.31289/jite.v9i2.16399.

[21] I. Bagus Sunandari and A. Royhan, "AKCU : A Web-Based Application for Real-Time Weather and Activity Recommendations," *Int. J. Comput. Res.*, vol. 1, no. 1, pp. 30–39, 2026, doi: 10.11591/jjrip.v99.i1.pp1-1x.

[22] K. Dixit, A. Ranjan, K. A. Faizan, and P. Kiran, "Automated Irrigation System Based on Soil Moisture Sensor," *Int. J. Res. Appl. Sci.*, vol. 10, no. 7, pp. 475–479, 2022, doi: 10.22214/ijraset.2022.45285.

[23] A. U. Rahayu, Linawati, N. P. Sastra, and I. B. Gede Maunaba, "The Evolution of Weather-Based Deep Learning in Smart Irrigation: A Systematic Review of Sustainable Approaches and Perspectives," *Indones. J. Electron.*, vol. 8, no. 2, pp. 141–160, 2026, doi: 10.35882/ijeeemi.v8i1.285.

[24] L. Huang *et al.*, "Advances and prospects of closed-loop precision irrigation for synergistic water-salt-phosphorus regulation in saline-alkali soils," *Agric. Water Manag.*, vol. 325, no. 8, pp. 1–16, 2026, doi: 10.1016/j.agwat.2026.110187.

[25] S. Dey, M. Rahman, and N. Karmakar, "Design and Experimental Analysis of Time Domain Reflectometry-Based Chipless RFID Soil Moisture Sensor," *IEEE Access*, vol. 13, no. 9, pp. 185284–185295, 2025, doi: 10.1109/ACCESS.2025.3626485.

[26] R. Nandi and D. Shrestha, "Assessment of Low-Cost and Higher-End Soil Moisture Sensors across Various Moisture Ranges and Soil Textures," *Mol. Divers. Preserv. Int.*, vol. 24, no. 9, pp. 1–13, 2024, doi: 10.3390/s24185886.

[27] B. Gunawan and R. P. Astutik, "Implementation of Solar Energy for an IoT-Based Agricultural Irrigation and Monitoring," *J. Eng. Technol. Appl. Sci.*, vol. 7, no. 3, pp. 189–203, 2025, doi: 10.36079/lamintang.jetas-0703.958.

[28] K. V. Vardhan, Y. K. Sai, and S. Musala, "IoT-Based Home Automation Using Blynk Application and Google Assistance," *Algorithms Comput. Theory Eng. Appl. Adv. Sci. Technol. Innov.*, vol. 18, no. 10, pp. 1–8, 2025, doi: 10.1007/978-3-031-72747-4_18.

[29] P. S, "Blynk 2.0 based Smart Electricity Monitoring Meter," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 11, no. 1, pp. 1312–1323, 2023, doi: 10.22214/ijraset.2023.48792.

[30] S. Ahmed *et al.*, "A Comprehensive Review of Solar Photovoltaic Systems : Scope , Technologies , Applications , Progress , Challenges , and Recommendations," *IEEE Access*, vol. 13, no. 3, pp. 69723–69750, 2025, doi: 10.1109/ACCESS.2025.3558539.

[31] F. Shaik, S. S. Lingala, and P. Veeraboina, "Effect of various parameters on the performance of solar PV power plant : a review and the experimental study," *Sustain. Energy Res.*, vol. 10, no. 10, pp. 1–23, 2023, doi: 10.1186/s40807-023-00076-x.

[32] Z. Aini, Kunaifi, A. Wanda, E. Ismaredah, and W. Anjarjati, "Solar Irrigation System in Indonesia: Practical Assessment and Evaluation for Converting Fossil Fuels with Solar Energy," *IOP Conf. Ser. Earth Environ. Sci. Pap.*, vol. 927, no. 1, pp. 1–9, 2022, doi: 10.1088/1755-1315/927/1/012022.