

INTERNET OF THINGS: AUTOMATIC PLANT WATERING SYSTEM USING ANDROID

Ridwan Siskandar^{1✉}, Muhammad A. Fadhil¹, Billi Rifa Kusumah², Irmansyah³, Irzaman³

¹Computer Engineering Study Program, College of Vocational Studies, IPB University

²Faculty of Marine and Fisheries Technology, Nahdlatul Ulama Cirebon University

³Physics Department, IPB University

✉Komunikasi Penulis, email : ridwansiskandar@apps.ipb.ac.id

DOI:<http://dx.doi.org/10.23960/jtep-lv9i4.297-310>

Naskah ini diterima pada 19 Juli 2020; revisi pada 9 September 2020;
disetujui untuk dipublikasikan pada 26 November 2020

ABSTRACT

Internet of Things (IoT) is a system that connects devices directly or indirectly to the internet. The device can work with remote control. One application of the IoT system on the watering system is able to provide an approach to the ease in the process of growth and development of plants. The research carried out was applied to an IoT-based smart plant watering device. The tool is supported with a soil moisture measuring sensor that acts as a benchmark to determine the condition of soil moisture and automatic control of the process of watering plants. The process of watering plants is scheduled in the morning and evening. Information, as long as the device is run will be displayed on the LCD and screen of an Android-based smartphone application. The data can be a comparison value to determine the suitability of soil moisture data in plants. The volume of water supplied in the watering process also affects the level of soil moisture. IoT-based smart plant sprinklers can automate watering by measuring the percentage of soil moisture as a benchmark for providing water during the watering process. Remote watering control can also be done on this tool by using a WiFi signal to the same access point that is connected to the smartphone client and microcontroller.

Keyword: agriculture, automatic sprinklers, engineering, IoT

I. INTRODUCTION

Progress in the field of information technology and embedded systems in the era of digitalization is increasingly leading to the study of control and automation systems. The topic of microcontroller-based watering plant automation integrated with internet of things (IoT) is very interesting to review, as did previous researchers namely: (Islami, 2018; Pambudi *et al.*, 2020; Sasmoko and Horman, 2020). Previous researchers have succeeded in developing the application of sensor automation on plant watering devices in daily life in all disciplines including smart farming: PGPR technology for the sustainability of dry land agriculture. (Ekawati, 2013), the design of an automatic plant sprinkler uses an arduino uno microcontroller (Dean Hansen, 2015), automatic watering of plant media based on smart phone and wireless fuzzy sensor network (Amir *et al.*, 2017), irrigation automation based

on internet of things (Prasetyo *et al.*, 2018), watering and lighting systems in smart gardens use context-aware based technology (Giovannie dan Sumaryo, 2018), automated watering and irrigation system using arduino uno (Patil and Shah, 2019), control device engineering for aquaponic monitoring system (Siskandar and Kusumah, 2019) design and build tools for watering and fertilizing plants automatically with an internet of things-based monitoring system (Windyasari, 2019), automatic tomatoes plant watering system using internet of things internet of things: automatic sprinklers in prototyping greenhouse using smartphone based android (Nasution *et al.*, 2020), smart plant watering design using a smartphone and an Arduino microcontroller based on the internet of thing (Pambudi *et al.*, 2020), water flow monitoring system and automatic sprinkling in iot-based greenhouses with esp8266 and blynk (Sasmoko and Horman, 2020).

In general, treatment of plant maintenance, such as the watering process that is often done by most plant farmers is still done manually by farmer workers (Giovannie dan Sumaryo, 2018; Rahman, 2018; Windyasari, 2019). The use of manual systems has weaknesses, one of them is a human error because the assessment of crop farmers is still subjective (Cobantoro *et al.*, 2019; Kurniawan *et al.*, 2015; Ratnawati dan Silma, 2017).

IoT is a system that connects devices directly or indirectly to the internet. The device can work with remote control. One application of the IoT system on the watering system can provide an approach to the ease in the process of growth and development of plants (Prasetyo *et al.*, 2018; Prasetyo dan Yusuf, 2020; Islami, 2018; Lu *et al.*, 2013; Mora *et al.*, 2018). One of the main factors of plant growth and development is precisely the size of soil humidity.

In accordance with the statement above, this researcher developed a research study entitled "Internet of things: Automatic plant watering system using android". The tool is supported with a soil moisture sensor that acts as a benchmark to determine soil moisture conditions and automatic control of the process of watering plants. The process of watering plants is scheduled in the morning (07.00 - 10.00 WIB) and in the afternoon (16.00 - 19.00 WIB). Data will be displayed on the LCD and Android-based smartphone applications while the tool is running. The data can be a comparison value to determine the suitability of soil moisture data in

plants. The volume of water that is regulated in the watering process will affect the level of soil moisture. IoT-based smart plant sprinklers can automate watering by measuring the percentage of soil moisture as a benchmark for providing water during the watering process. Remote watering control can be done on this tool by using a wifi signal to the same access point that is connected to the smartphone client and microcontroller.

II. MATERIALS AND METHODS

The method used in this research is the System Development Life Cycle (SDLC) with the waterfall model. In this method, there are four stages, namely analysis, design, implementation and testing (Bhavsar *et al.*, 2020; Stefanus and Andry, 2020). Stages of analysis are ways to find out the problems and needs that will be used in the process of making tools. The analysis phase is divided into two, namely problem analysis and needs analysis. Analysis of the problem is done by looking for information or references which relate to the problem at hand. Next is to analyze the needing of materials and tools in making tools. The needing for tools and materials, which are used in research, are shown in Table 1.

The design stage is the planning process to get a solution of existing problems. The design includes the design of block diagrams, flow diagrams, electronic circuits (hardware), software (software), Android-based smartphone applications, and mechanical tools.

Table 1. Equipment and Material Needs

No.	Materials and Tools	Function	Volume
1	NodeMcu V3	Data processing.	1 piece
2	Adaptor/ battery	Electric power source	1 piece
3	Water pump	Water output / booster	1 piece
4	Relay	Electric current control	1 piece
5	RTC DS3231	Timer	1 piece
6	RGB LED		1 piece
7	LCD 20 x 4	Displays sensor reading data	1 piece
8	I2C	LCD supporting components	1 piece
9	Soil moisture sensor YL-69	Reading soil moisture	1 piece
10	Access point	Spreading of the wifi signal	1 piece
11	Smartphone	Control based andorid	1 piece

2.1. Illustration of Component Connection Design

Figure 1 explains the operating functions of the prototype tool. The power supply serves as an electrical conductor. The power supply is connected directly to the NodeMcu V3 ESP8266 module with the aim of turning on the module function. The voltage received by the module will generate a voltage value of 5 volts on each analog pin and digital pin on the NodeMcu V3 ESP8266 module. Voltage generation will perform the function of an electronic module connected to the NodeMcu V3 ESP8266 in accordance with the pins that have been initialized at the software design stage. Access point functions as a modem that spreads wifi signals. At the same time, the client attribute is a smartphone that has been installed.

2.2. Design of Flow Chart

Figure 3 shows the tool flow diagram. The condition of soil moisture is divided into three conditions, namely: dry, damp and wet. The three soil moisture conditions are measured from the large percentage of soil moisture values read by the YL-69 soil moisture sensor. Calibration of the categorization of soil moisture conditions is based on the value of the percentage of humidity obtained from observations of water supply in the scope of the planting media (pot) with a diameter of 12 cm. The average plant growth is ideal at 25% - 49% humidity (Amir *et al.*, 2017; Cobantoro *et al.*, 2019; Ratnawati and Silma, 2017). Growing media on sprinklers weighs 0.28 kg, so it requires 20.94 liters of water every day. The water pump on sprinklers has a water discharge of 0.06 liters / second. Seeing this, the

total time (dry conditions) used for watering is 5.83 minutes. The total water demand and the length of watering time needed are adjusted to the scheduling of morning watering (07.00 - 10.00 WIB) and afternoon (16.00 - 19.00 WIB).

The principle works, divided into two, namely: (1) the sprinklers work in automatic mode and (2) the sprinklers work in the control mode of the smartphone. When the sprinklers work automatically if the sensor detects a large percentage of soil moisture value of less than 30%, the LCD will display information that reads "DRY", "RED LED" lights up and the water pump automatically water the plants. If the sensor detects a large percentage of soil moisture value more than equal to 30% and less than 45%, the LCD will display information that reads "SOFT", "GREEN LED" lights up, and the water pump automatically turns off. If the sensor detects a large percentage of soil moisture value more than equal to 45%, the LCD will display information that reads "WET", "GREEN LED" lights up and the water pump automatically turns off. Furthermore, the RTC module that functions as a time data storage and time setting will provide real-time time data to be displayed on the LCD. Data is displayed on the LCD with the format "date.month.year" and "hour: minute: second". The data will continue to be updated according to the time reading process looping in the program code. With the time of watering the plants, so when the condition of the plant is detected dry even outside the watering hours, the water pump will not turn on. This process will continue with the program looping period every second, and system functions will stop if

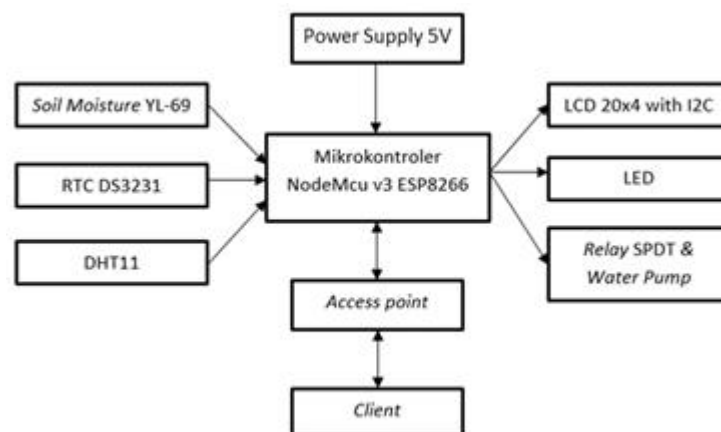


Figure 1. Illustration of Component Connection Design

the power supply on the microcontroller is disconnected. When the sprinklers work in the control mode of the smartphone, the user can control the water pump at any time while connected to the access point.

2.3. Electronic Circuit Design (Hardware)

The design of electronic circuits (hardware) is shown in Figure 3. Figure 3 explains that in designing an electronic circuit (hardware) three parts are needed, namely: input components (RTC DS3231, soil moisture YL-69 and batteries), process components (NodeMcu V3) and output components (RGB LED, water pump, relay and LCD 20 x 4).

1. Software Design

Software design consists of library initialization, pin and component type initialization, void setup () function, sensor read function, coding for checking conditions of soil moisture, coding for setting watering time, coding for checking

connectivity and coding for functions of the client request.

The "Wire.h" library functions as a serial structure call to display information that takes place on the microcontroller on the monitor screen. Library "RTCLib.h" functions as a declaration in running the RTC DS3231 function, the library "LiquidCrystal_I2C.h" functions as a declaration in carrying out LCD functions with the I2C module to shorten the pins that are on the LCD, and the library "ESP8266Wifi.h" functions as a declaration in carrying out the ESP8266Wifi function. The more detailed libraries initialization is shown in Figure 4a.

Figure 4b shows the initialization of pin and component type. The #define command functions as the initialization of the pin used for each component "int" on the soil moisture variable is a type of data used in storing sensor readings. "Structure" in the RTC and LCD is an

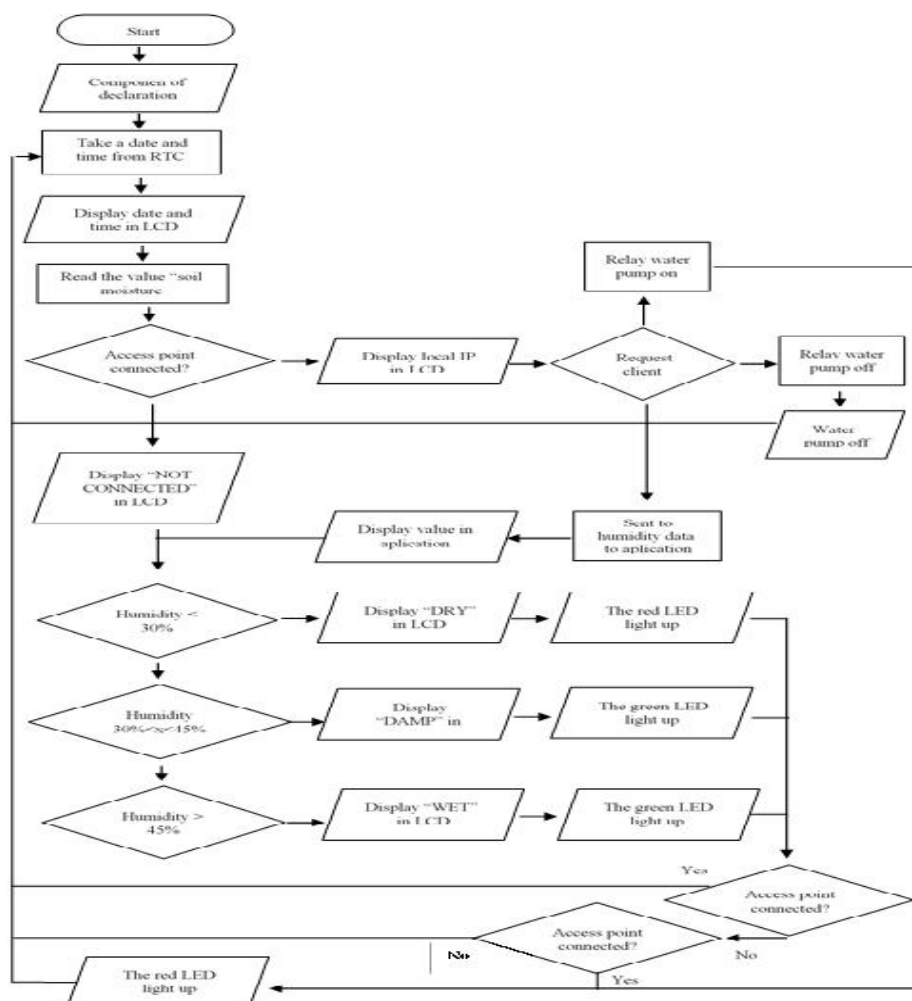


Figure 2. Flow Diagram of The Sprinkler Method

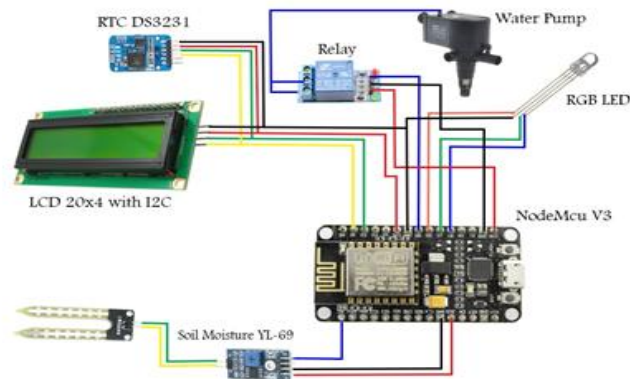


Figure 3. Electronic Network Design (Hardware)

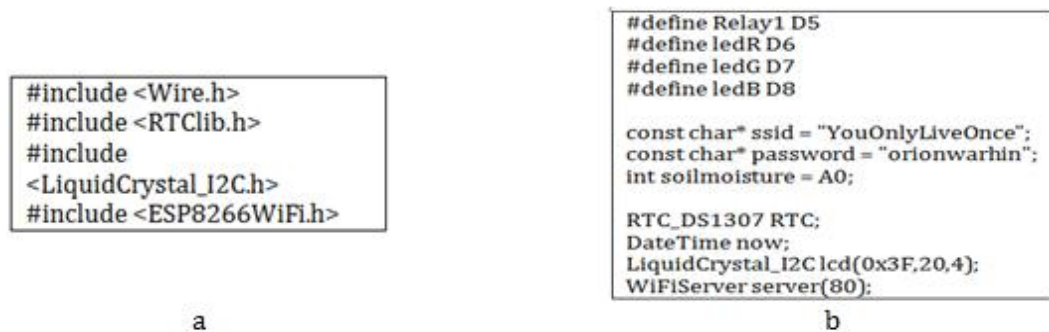


Figure 4. (a) Initializing The Library and (b) Initialization of Pin and Component Types

initialization of the type of component used. At the initialization of the LCD variable is named LCD with memory address 0x3F and the size of the LCD is 20x4.

The void setup function is used to declare a function on an initialized component. The void setup function is shown in Figure 5. Some variables in writing this program are declared using the same name as the original name of the component. This is used to make it easier to recognize variables that have already been declared. "Relay1" and "led" are declared as pins that function as outputs. The digital pins of "Relay1" and "led" are declared with *active high* and *low*. *Active high* functions to connect the voltage-current to the pin used. Whereas active low is the opposite.

The function of reading sensor data is shown in Figure 6. The "smRead ()" function is a function to get the ADC value from the soil moisture sensor readings, with the "smValue" variable as the ADC value storage variable. The "return" command functions to reverse the obtained integer value, then put it in the "smRead" function. The purpose of reducing the value of

1023 with the ADC value is to reverse the results of the ADC value which should read the humidity sensor is getting wetter, the smaller the ADC value, the wetter the soil moisture, the ADC value will be even greater. While the "smReadPercent ()" function is a function used to get a percentage value from the ADC value. The "smValuePercent" variable stores the compression results of the ADC value to percent with the map command, the ADC value with the vulnerable value 0-1023 is changed to the vulnerable value 0-100. The purpose of compressing is to make it easier to analyze the calculation of soil moisture values based on the water content in the soil.

The next process is to declare the void loop function. This function aims to loop while the program is running. The contents of this function in the form of commands display information both serially on the monitor and the LCD screen. Information displayed on the LCD is information on soil moisture conditions measured in real time and execution settings for plant watering times. The program code for checking soil moisture conditions is shown in Figure 7a. The program code for setting watering time is shown in Figure 7b.

```

void setup() {
  Serial.begin(115200);
  RTC.begin();

  pinMode(Relay1,OUTPUT);
  pinMode(ledR,OUTPUT);
  pinMode(ledG,OUTPUT);
  pinMode(ledB,OUTPUT);
  digitalWrite(Relay1,HIGH);
  digitalWrite(ledR,LOW);
  digitalWrite(ledG,LOW);
  digitalWrite(ledB,LOW);

  introShow();
}

void loop() {
  timeShow();

  if (WiFi.status() != WL_CONNECTED){
    lcd.setCursor(3,1);
    lcd.print("NOT CONNECTED");
    chkShow();
    Serial.println();
    Serial.println("WiFi NOT connected");
    smShow();
    ledShow();
    irrigationShow();
    delay(3000);
  }

  else {
    Serial.println();
    Serial.println("Connected My IP is : ");
    Serial.println(WiFi.localIP());
    lcd.setCursor(3,1);
    lcd.print(WiFi.localIP());
    server.begin();

    Serial.println();
    Serial.println("WiFi connected");
    appShow();
    smShow();
    ledShow();
    delay(3000);
  }
}

```

Figure 5. Function Void Setup

```

int smRead(){
  int smValue = analogRead(soilmoisture);
  return 1023 - smValue;
}

int smReadPercent(){
  int smValue = 1023 - analogRead(soilmoisture);
  float smValuePercent = map(smValue,0,1023,0,100);
  return smValuePercent;
}

```

Figure 6. Function Read Sensor Data

<pre> if (smReadPercent() < 30){ lcd.print("DRY"); digitalWrite(ledR,HIGH); digitalWrite(ledG,LOW); digitalWrite(ledB,LOW); delay(100); } else if (smReadPercent() >= 30 andand smReadPercent() < 45){ lcd.print("DUMP"); digitalWrite(ledR,LOW); digitalWrite(ledG,HIGH); digitalWrite(ledB,LOW); delay(100); } else { lcd.print("WET "); digitalWrite(ledR,LOW); digitalWrite(ledG,LOW); digitalWrite(ledB,HIGH); delay(100); } </pre> <p style="text-align: center;">a</p>	<pre> if (now.hour() == 7 andand smReadPercent() < 30 now.hour() == 8 andand smReadPercent() < 30 now.hour() == 9 andand smReadPercent() < 30 now.hour() == 16 andand smReadPercent() < 30 now.hour() == 17 andand smReadPercent() < 30 now.hour() == 18 andand smReadPercent() < 30){ digitalWrite(Relay1,LOW); delay(200); } else { digitalWrite(Relay1,HIGH); } </pre> <p style="text-align: center;">b</p>
--	--

Figure 7. (a) Examination of Soil Moisture Conditions and (b) Setting Watering Time for Plants

Figure 8a explains the program code for the connectivity check function. In the program code, the function “WiFi.begin(ssid, password);” explain how the microcontroller process tries to connect to the access point. The ssid variable contains a string data type named SSID. The password variable contains a password which functions as a code to connect to the initial initialized SSID. The function “Server.begin();” is the command code to run NodeMcu as a server (data processor). This command will run if the connection to the access point is already connected. Writing the program code for sending and receiving client requests is shown in Figure 8b.

2. Designing Android-Based Smartphone Applications

Application created with the help of the MIT App Inventor platform. The platform can be accessed online, making it easier for users to create applications without having to download the application maker program. At MIT App Inventor the program code initialization is done by arranging command blocks that have been provided on the platform. The arrangement of blocks forms puzzles that are interconnected. The order of the blocks determines the process of reading the command code. Making an application for this tool is composed of four parts

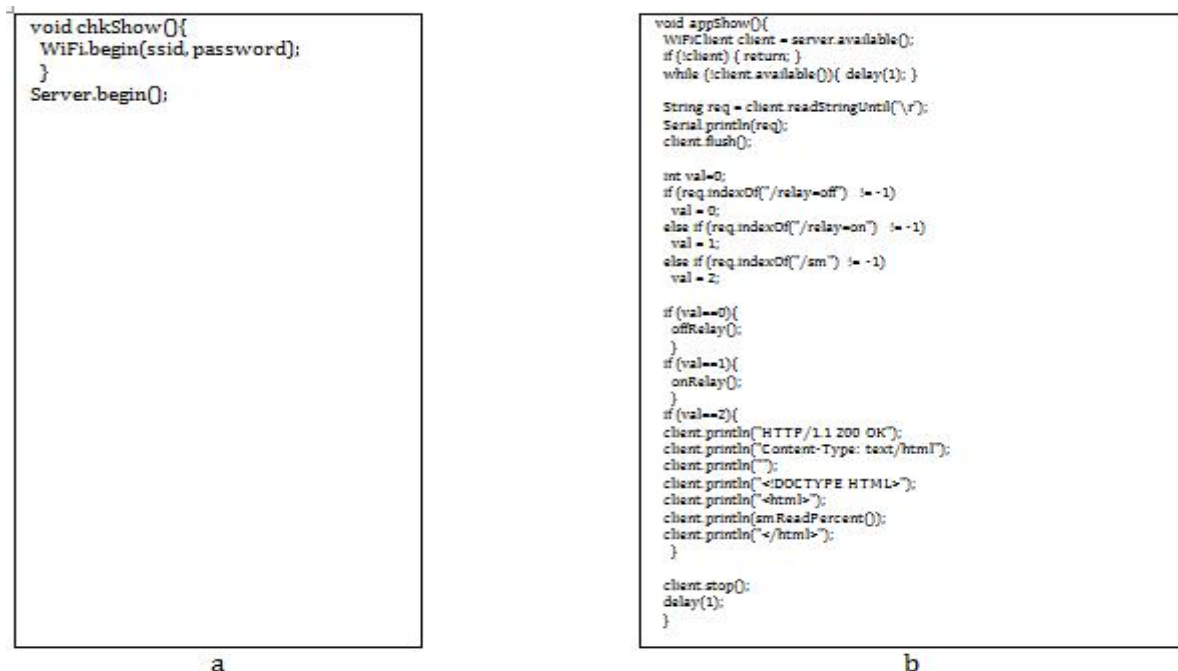


Figure 8. (a) Program Code for The Connectivity Check function and (b) Program Code for Client Request Functions



Figure 9. (a) Components of a Splash Screen and (b) Components from The Main Menu

of the display, namely: splash screen, main menu, soil moisture monitoring and water pump control.

Splash screen is the initial display when opening an application. The splash screen display on the application tool contains the application title, opening logo on the application and the text "Watering Plants". The components of the splash screen are shown in Figure 9a.

In making *splash screens*, the "Horizontal Arrangement" component functions as a layout. Component "label" functions as a label to embed text then the component "clock" as a timer in processing commands in the form of a period the splash screen appears.

The second stage is making the main menu. The main menu function is a display that functions as a container selection of functions provided in the application. In the main menu contains two choices of functions which later serve as a user display too, namely monitoring soil moisture and water pump control. The components of the main menu display are shown in Figure 9b.

In the main menu components, the "Image" component functions like a component to display images on the main menu view. The "button" component functions as a button that determines display options or further functions available on the main menu. The "clock" component functions as a timer to set the time in the process of rotating images available on the main menu.

The third stage is the making of soil moisture monitoring display where this display is one of the functions available on the main menu that has been explained previously. Soil moisture monitoring is a user interface that functions to display data on the percentage of soil moisture read / sent by the NodeMcu module, and to update sensor reading data per three seconds. The components of the soil moisture monitoring display are shown in Figure 10a.

In the components of monitoring soil moisture monitoring, the component "label" functions to display the percentage value of soil moisture with a calculation of time per three seconds (setting the time using the function of the clock component). The "button" component functions as close screen command or can be interpreted as a function to return to the display before this screen is opened. The "webSM" component functions as a connecting component with the HTTP protocol which contains the IP address obtained from the access point along with the data string as a command code to get the percentage value of soil moisture from the measurement results of the sensor.

The fourth stage is to make the water pump control display. This display is one of the functions available in the main menu options previously described. Water pump control is a user interface that functions to display the control button in the form of turning on and off the water pump that is connected to the NodeMcu module. Unlike the previous soil moisture monitoring display, in this water pump



Figure 10. (a) Components of Monitoring Soil Moisture and (b) Components Making Up The Water Pump Control Display

control display, there is no process of updating data at a particularly vulnerable time, so there is no need for components in the form of a clock or which normally functions as a timer. The components making up the water pump control display are shown in Figure 10b.

In the constituent components of the water pump control, there are two button components with "Button1" functioning as a control button to turn on and turn off the water pump while "Button2" functions as a close screen on display. There are two web components, namely "webRelayOn" containing an IP address as a client request process for the NodeMcu module to connect current to the relay or turning on the water pump and "webRelayOff" containing an IP address as the client request process for the NodeMcu module to break the current in the relay or turn off the water pump.

3. Mechanical Mechanics Planning

The watering prototype made with the size of the box (media storage pots) is 39x27x20 cm³. Pot stored in the box has a diameter of 12 cm. Watering pipes are stored just below the plant. Watering pipes are given a hole with a diameter of 2 mm every 1 cm distance. The electronic component is stored in the front of the pot storage box. More clearly, the mechanical design of the sprinklers is shown in Figure 11.

At the implementation stage, a merger is made between electronic devices (hardware), software (software) and application tools. After everything is integrated, then the testing stage is carried out. Stages of testing are carried out to determine the suitability of the tools made to make tools. Tests on the equipment include:

testing the function of the sprinklers, testing the application of watering plants, testing the integration of electronic circuits and software, testing the reading of soil moisture sensors in the smartphone control mode and testing the sprinklers when in automatic mode.

III. RESULTS AND DISCUSSION

In sprinklers, the implemented plant is the Lily paris plant with a planting medium of 12 cm in diameter. The sprinkler display is shown in Figure 12a. The electronic circuit (hardware) contained in the component box consists of NodeMcu V3, LCD 20x4, LED, soil moisture YL-69, RTC DS3231 and SPDT relay arranged together. The results of the electronic circuit (hardware) are shown in Figure 12b.

3.1. Testing The Function of The Watering Tool

This test is carried out with the aim of the tool functioning properly according to or not. The results of testing the function of the sprinklers are shown in Table 2

3.2. Application of Plant Watering

Based on the design, the smartphone client has been installed using the App Inventor application. There are four display interfaces in the application of the plant watering tool, namely the splash screen interface display, main menu, soil moisture monitoring and water pump control. The interface of the splash screen, main menu, soil moisture monitoring and water pump control is shown in Figure 13.

Taking the results of the design of the application is done by using a screenshot technique on the

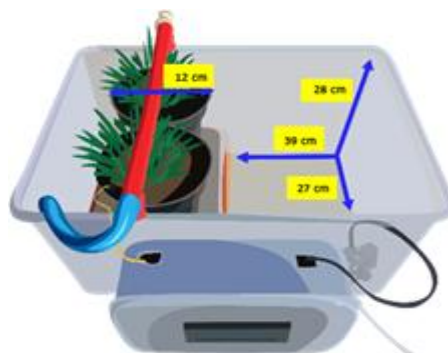


Figure 11. Mechanical Design of Sprinklers

Table 2. Testing The Function of Sprinklers

No	Tool Components	Status	Information
1	NodeMcu V3	Good	Runs well, can take commands from sensors and run water pumps
2	Adaptor/ baterai	Good	Going well, can provide electricity well
3	Water pump	Good	Going well, can function according to the program created
4	Relay	Good	Going well, can function according to the program created
5	RTC DS3231	Good	Functioning according to the program created
6	RGB LED	Good	Functioning according to the program created
7	LCD 20 x 4	Good	Running well can bring up the display interface
8	I2C	Good	Functioning according to the program created
9	Sensor soil moisture YL-69	Good	Going well, the sensor can read the moisture in the growing media
10	Access point	Good	Runs well, there is a microcontroller connectivity process function access point

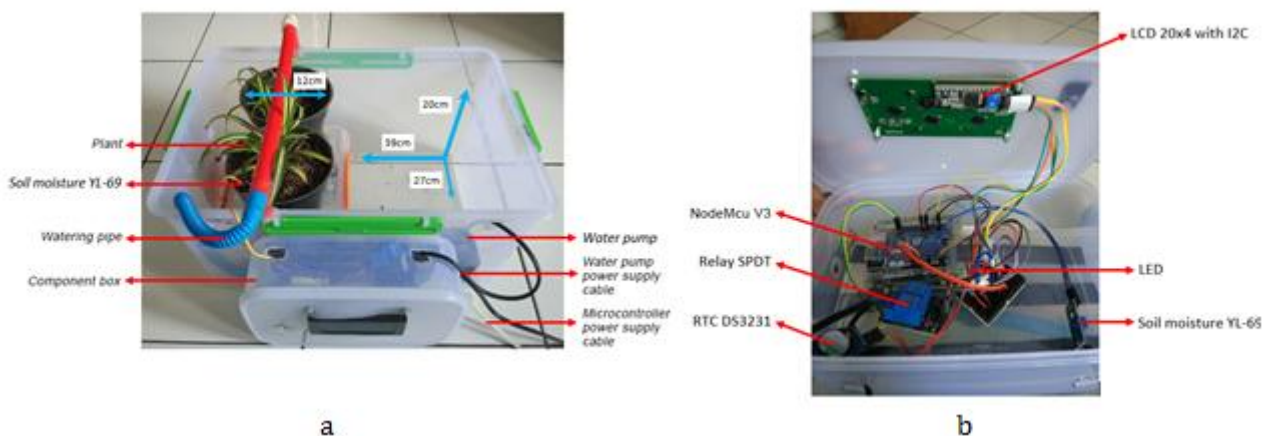


Figure 12. (a) Display of plant sprinklers and (b) The Results of The Electronic Circuit (Hardware)

smartphone screen that has the application installed from this tool. To run the application function, the smartphone must be connected to wifi which also connects to the same access point used by the NodeMcu module. But in some conditions, there is a connection failure between the client and access point. These conditions result in the failure of data transactions in the form of not updating the monitoring and control functions of the application. If the connectivity fails, then the application will display information failed to connect. The display failed to connect on each screen is shown in Figure 14.

3.3. Testing of Electronic Circuit Integration, Sprinkler Software and Software Applications

Testing is done by looking at the response of the microcontroller to the status of connectivity with the access point. If the microcontroller is not connected to the access point, the LCD will display information in the form of the words "NOT CONNECTED". If the microcontroller is successfully connected to the access point, it will

display information in the form of a local IP on the LCD screen. Information display Connected and not connected access points on the LCD is shown in Figure 15.

Checking the microcontroller connectivity information with the access point shown in Figure 24 is done every three seconds. The next test is displaying information on the monitoring function on the application of the plant watering tool, namely the percentage value of soil moisture obtained from the results of the soil moisture sensor reading. This process is also done every three seconds for updating the data. If the microcontroller fails to connect then the error message display is shown in Figure 17. If the microcontroller is connected, the information displaying the percentage of soil moisture in the application is shown in Figure 16.

3.4. Testing the Soil Moisture Sensor Readout on Smartphone Control Mode

The testing phase is carried out to find out that the entire system designed especially the soil

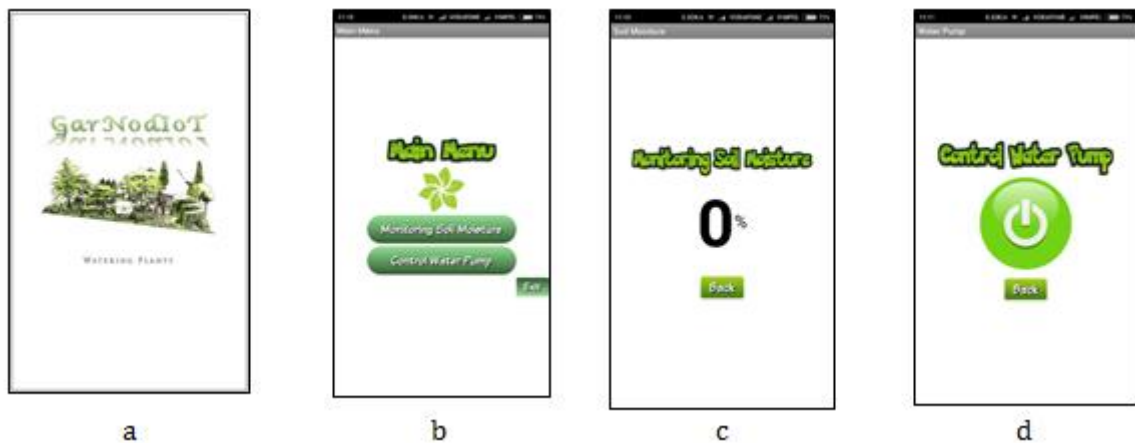


Figure 13. Display Interface: (a) *Splash Screen*, (b) *Main Menu*, (c) *Monitoring Soil Moisture*, and (d) *Control Water Pump*



Figure 14. Display Fails to Connect to Screen (a) *Sensor Reading*, (b) *on Water Pump Button*, and (c) *Off Water Pump Button*



Figure 15. (a) *The Information Display is not Connected to The Access Point on The LCD* and (b) *The Information Display is Connected to The Access Point on The LCD*

moisture sensor is running well. Testing is done by installing sensors into the soil in the planting media (pots) with different soil conditions (dry, damp and wet). The results of testing the soil moisture conditions are shown in Figure 17.

3.5. Testing the Sprinklers When Automatic Mode

The results of the overall sprinkler test when the automatic mode condition is shown in Table 3. The test is carried out by taking into account the



Figure 16. Display Information on Soil Moisture Percentage in The Application

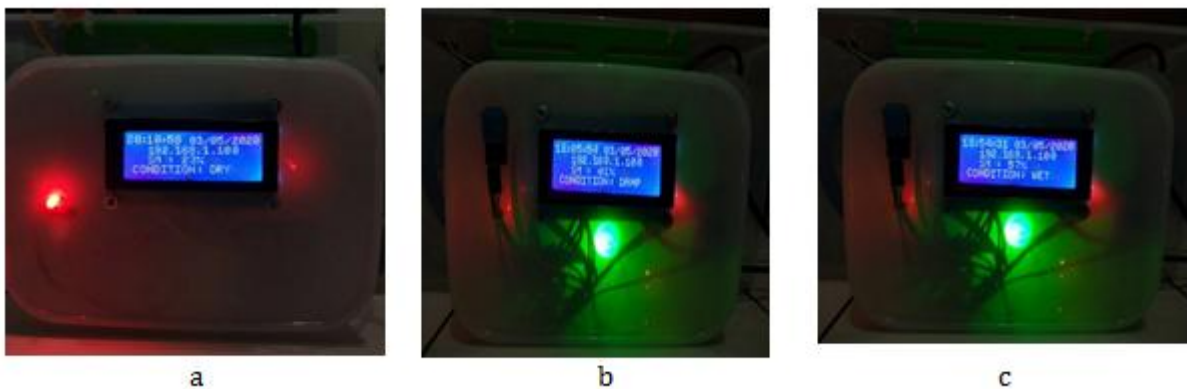


Figure 17. Soil Moisture Test Results (a) Dry Conditions, (b) Damp Conditions, and (c) Wet Conditions

Table 3. Sprinkler Test Results When in Automatic Mode

Time (o'clock)	Percentage of Soil Moisture (%)	Water Pump	LED	Soil Moisture Conditions
06.00	23	Off	On	Dry
09.15	18	On	On	Dry
11.10	57	Off	Off	Wet
14.20	41	Off	Off	Moist
16.05	33	Off	Off	Moist
16.50	25	On	On	Dry

condition of whether the water pump and LED are on during the watering process in the morning and evening.

Based on Table 3, the test results can be concluded that the tool is running well according to the watering schedule. Water pump will only turn on if the condition of soil moisture is dry or the percentage of soil moisture is below 30% in the morning watering schedule (07.00-10.00) and the afternoon watering schedule (16.00-19.00), while the LED will still light as a notification if

the condition of dry soil moisture or the percentage of soil moisture is below 30% regardless of the schedule for watering the plants.

IV. CONCLUSION

Making smart plant sprinklers based on IoT (Internet of Things) can be a solution approach to plant growth and development activities. In this tool describe the scheduling of morning and evening watering like watering that is done by

crop farmers manually. Indicators of soil moisture conditions are displayed on the LCD and smartphone applications for flexible and practical monitoring. The working principle of this tool is that if the soil moisture conditions are wet or meet the threshold value criteria, then the watering process will stop, so that water use will be more efficient. The ability to control the water channel using a WiFi network with an Android-based smartphone application can provide a prototype approach to the developing IoT (Internet of Things) system. All functions that have been mentioned in this prototype have been tested and succeeded as intended.

REFERENCES

- Prasetyo, A., Yusuf, A.R., Litanianda, Y. 2018. Otomasi Irigasi Janggolan Berbasis Internet Of Things. *Multitek Indonesia* 13(2): 104-109.
- Amir, F., Rahmawati, D., Ulum, M. 2017. Penyiraman Tanaman Media Otomatis Berbasis Telepon Seluler Plintar dan Jaringan Sensor Fuzzy Tanpa Kabel. *Seminar Nasional Matematika dan Aplikasinya*, 21 Oktober 2017, Universitas Airlangga, Surabaya: 355-361.
- Prasetyo, A. dan Yusuf, A.R. 2020. Integrated device electronic untuk sistem irigasi tetes dengan kendali Internet of Things. *Jurnal Ilmiah Teknologi Informasi Asia* 14(1): 1-7.
- Bhavsar, K., Shah, V., Gopalan, S. 2020. Scrumbanfall/ : an agile integration of scrum and kanban with waterfall in software engineering. *International Journal of Innovative Technology and Exploring Engineering* 9(4): 2075-2084.
- Giovannie, C.P., Sumaryo, S., Ekaputri, C. 2018. Sistem penyiraman dan pencahayaan pada kebun pintar menggunakan teknologi berbasis Context Awar. *e-Proceeding of Engineering* 5(3): 4280-4287.
- Cobantoro A F, Setyawan M B, Budi W M A. 2019. Otomasi greenhouse berbasis mikrokomputer RASPBERRY PI. *Jurnal Ilmiah Teknologi Informasi Asia* 13(2): 115-124.
- Hansen, D., Hoendarto, G., Lina, L. 2015. Perancangan Perangkat Penyiram Tanaman Otomatis Menggunakan Mikrokontroler Arduino Uno. *Jurnal InTekSis* 4(2): 64-75.
- Ekawati, I. 2013. Smart Farming/ : Teknologi Pggpr Untuk Keberlanjutan Pertanian Lahan Kering. *Prosiding Seminar Nasional Optimalisasi Sumberdaya Lokal di Era Revolusi Industri 4.0*: 615-622.
- Islami F A. 2018. Algoritma decision tree pada sistem penyiraman tanaman otomatis berbasis Internet of Things. *Jurnal Ilmiah Teknologi Dan Rekayasa* 23(1): 66-77.
- Kurniawan M A, Sunarya U, Nurmantris D A. 2015. Alat penyiraman otomatis berbasis mikrokontroler dengan Android sebagai media monitoring. *E-Proceeding of Applied* 1(2): 1543-1551.
- Lu, X., Liu, W., Wang, H., Sun, Q. 2013. Robot control design based on smartphone. *The 25th Chinese Control and Decision Conference (CCDC)*: 2820-2823.
- Mora, H., Signes-Pont, M.T., Gil, D., Johnsson, M. 2018. Collaborative working architecture for IoT-based applications. *Sensors*, 18(6): 1676. <https://doi.org/10.3390/s18061676>.
- Nasution, I.S., Iskandar, M.R., Jayanti, D.S. 2020. Internet of things: Automatic sprinklers in prototyping greenhouse using smartphone based android. *IOP Conference Series: Earth and Environmental Science*, 425: 012069. <https://doi.org/10.1088/1755-1315/425/1/012069>
- Pambudi, A.S., Andryana, S., Gunaryati, A. 2020. Rancang bangun penyiraman tanaman pintar menggunakan smartphone dan mikrokontroler arduino berbasis Internet of Thing. *Jurnal Media Informatika*

- Budidarma*, 4(2): 250–256. <https://doi.org/10.30865/mib.v4i2.1913>
- Patil, V.B., Shah, A.B. 2019. Automated watering and irrigation system using Arduino UNO. *International Journal of Innovative Science and Research Technology*, 4(12): 928–932.
- Rahman, A. 2018. Penyiraman tanaman secara otomatis menggunakan propeler berbasis IoT. *ITEJ (Information Technology Engineering Journals)* 3(1): 1-8.
- Ratnawati, R., Silma, S. 2017. Sistem kendali penyiram tanaman menggunakan propeller berbasis Internet of Things. *Jurnal Inspiration* 7(2): 147-158.
- Sasmoko, D., Horman, R. 2020. Sistem Monitoring Aliran Air Dan Penyiraman Otomatis Pada Rumah Kaca Berbasis IOT Dengan Esp8266 Dan Blynk. *CIRCUIT: Jurnal Ilmiah Pendidikan Teknik Elektro* 4(1): 1–10.
- Siskandar, R., and Kusumah, B.R. 2019. Control device engineering for aquaponic monitoring system. *Aquacultura Indonesiana* 20(2): 72-73. <http://dx.doi.org/10.21534/ai.v20i2.151>
- Stefanus, M., dan Andry, J.F. 2020. Pengembangan aplikasi e-learning berbasis web menggunakan model waterfall pada SMK Strada 2 Jakarta. *Jurnal Fasilkom* 10(1): 1-10.
- Vina S.W.P.A.B. 2019. Rancang bangun alat penyiraman dan pemupukan tanaman secara otomatis dengan sistem monitoring berbasis Internet of Things. *Prosiding Seminar Nasional Sains, Teknologi, dan Sosial Humaniora, UIT 2019*: 21 hlm.