SITEKIN: Jurnal Sains, Teknologi dan Industri, Vol. 22, No. 2.280 - 294
ISSN 2407-0939 print/ISSN 2721-2041 online

Smarthome Design Using Raspberry Pi 3 Based on Internet Of Things (IoT)

Urief Arsani Atma Putra[1.], L. Ahmad Syamsul Irfan Akbar[2.], Cipta Ramadhani[3.]
[1.2.3] Department of Electrical Engineering, Faculty of Engineering, University Of Mataram, Indonesia
Jln. Majapahit No. 62 , Kec. Selaperang, Kota Mataram, Prov. Nusa Tenggara Barat. 83115
Email: urifarsani69@gmail.com, laluirfan@gmail.com, ramdhani.cipta@gmail.com

ABSTRACT

Advances in Internet of Things (IoT) technology have driven the development of smart, efficient, and widely accessible home automation systems. This study aims to determine how to design and integrate Raspberry Pi technology with Bylnk to control relays remotely manually using widgets and automatically using LDR and DHT11 sensors. The research method employs an engineering research approach focused on the design, development, and testing of a prototype smart home system based on the Internet of Things (IoT) using Raspberry Pi 3. The process begins with a literature review on IoT concepts and Raspberry Pi characteristics, followed by the design of hardware and software that integrates the Raspberry Pi 3, LDR and DHT11 sensors, a four-channel relay module, and the Blynk platform for the user interface. After assembly, the system is tested through experimental scenarios observing LED responses to four input conditions: light intensity, ambient temperature, automatic timer, and manual commands via the Blynk app. The collected data is quantitative observational, including LED status (on/off), response time, and analyzed descriptively and comparatively by comparing actual results against the success criteria established in the design. The test results show that the system operates reliably in executing each control function, both automatic and manual, with fast and accurate responses. This research contributes to the development of a modular, cost-effective, and easily implementable smart home system in real residential environments.

Keywords: Smart home, Internet of Things (IoT), Raspberry Pi 3, Blynk

Introduction

Science and technology (IPTEK) is advancing rapidly, especially in the field of engineering technology. One example of a technology that has been widely developed is the Internet of Things (IoT), which can connect all devices to the internet and enable them to communicate with each other via the internet [1]. To support this, remote control is needed. Currently, Raspberry Pi 3 is the technology that can meet this need [2].

Outdoor activities often make it difficult to control and monitor electrical appliances in the home [3]. Humans are highly dependent on electricity, making it one of the basic necessities of life. It is common to find situations where homeowners forget to turn off the lights when they are outside the house, forcing them to return and conduct an inefficient check—both in terms of time and finances, such as the cost of gasoline to return home again [4].

Research on the design of smart homes using Raspberry Pi 3 based on the Internet of Things (IoT) to improve security and efficiency has been widely conducted. Research by Kusuma, dkk. developed an internet-connected home door security system with RFID and sensor features for real-time monitoring [5]. Surya Angga Pranata, dkk. designed a tool for detecting illegal logging using sound sensors and cameras with notifications sent to Telegram [6]. Andrian, dkk. built a mobile-based remote lighting control system using Raspberry Pi and Python. Meanwhile [7], Ayyina Dewi Parwati, dkk. designed a smart camera that can monitor an area and send automatic alerts to a smartphone [8]. Based on previous studies, such as home door security systems, illegal logging detectors, simple light controllers, and Raspberry Pi-based smart cameras, most focus on a specific function or limited use without integrating various sensors comprehensively into a single system. Additionally, there have been few studies that comprehensively combine automatic and manual control for household devices, particularly in time-, light-, and temperature-based lighting control integrated through the Blynk platform and Raspberry Pi 3. This highlights the need for the development of a more functional, flexible, and responsive smart home system that can adapt to various environmental conditions and be easily controlled remotely.

The development of the Internet of Things (IoT) concept aims to create efficient and automatic connections between various objects. The Internet of Things (IoT) works by connecting various electrical devices over any distance without user intervention. The Internet of Things (IoT) has developed innovations such as smart homes, where this technology is used as an automated control system to manage electronic devices in homes and offices via the internet [9]. Blynk is a platform that supports the control of Arduino, Raspberry Pi, ESP8266, WEMOS D1, and similar modules via the internet. Blynk is a platform for creating graphical interfaces for projects that can be implemented using a drag-and-drop widget method, making it very easy to use [10]. This study designed and built a smart home prototype that integrates Raspberry Pi 3 with the Blynk platform and light, temperature, and time sensors to create an automated and remote control system for electrical appliances, particularly lights. This system not only relies on manual control via an application but also operates automatically to adapt to environmental conditions, making it more efficient and adaptive to user needs. By integrating various technologies into a unified system, this research offers an innovative solution to the challenges users often face when monitoring and controlling electrical devices while away from home.

Based on the above explanation, a study was conducted to answer how to design and build a simple smart home prototype, but focused on the needs of performing many tasks that hinder movement to turn lights on or off remotely, as well as adding automatic sensors such as light, temperature, and time sensors. To create this concept, a device is needed that can control like a remote control but with unlimited range. In this study, the researcher utilized Bylnk technology combined with a Raspberry Pi 3 as the main controller to ensure the desired concept functions effectively and connects it to the internet to enable control without limitations on distance or location.

Research Methods

This research method applies an engineering research approach that focuses on the design, construction, and testing of a smart home system prototype based on the Internet of Things (IoT) using Raspberry Pi 3. The process begins with a literature review on IoT concepts and Raspberry Pi characteristics, followed by the design of hardware and software that integrates the Raspberry Pi 3, LDR and DHT11 sensors, a four-channel relay module, and the Blynk platform for the user interface. After assembly, the system is tested through experimental scenarios observing LED responses to four input conditions: light intensity, ambient temperature, automatic timer, and manual commands via the Blynk app. The collected data is quantitative observational, including LED status (on/off), response time, and analyzed descriptively and comparatively by comparing actual results against the success criteria established in the design.

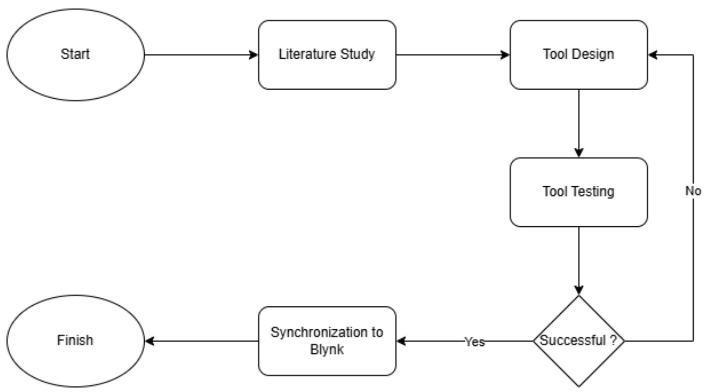


Figure 1 Research Diagram

Figure 1 below is a flowchart of the research or development process for an Internet of Things (IoT)-based device. The process begins with an initial stage of literature review to gather relevant information and references as a basis for device design. Following this, device design is carried out, which includes component assembly and system programming. The designed device then enters the testing phase to verify its functionality. If the testing results are unsuccessful, improvements are made by returning to the design phase. However, if the testing is successful, the process continues with synchronization to the Blynk application so that the device can be controlled and monitored in real-time via a smartphone. Once synchronization is complete and the system functions properly, the research process is deemed complete.

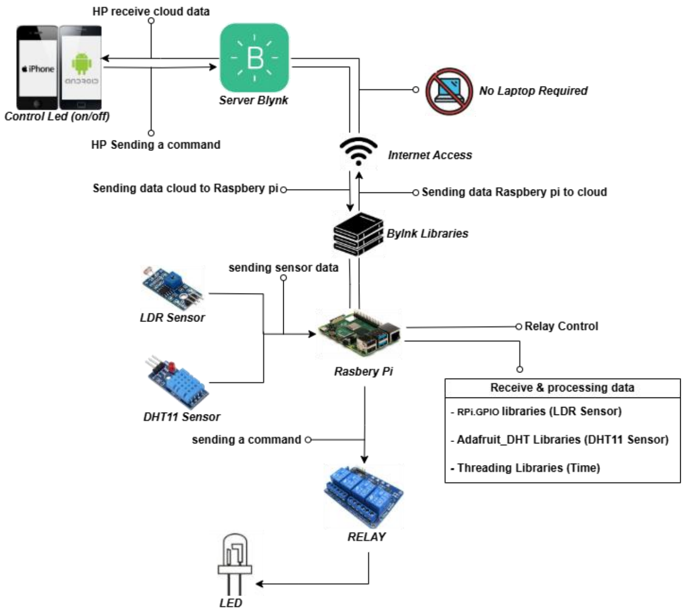


Figure 2 Smarthome System and Architecture

Figure 2 shows the architecture of an Internet of Things (IoT)-based monitoring and control system that utilizes Raspberry Pi as the control center. This system consists of several main components, namely sensors, Raspberry Pi, relay modules, LEDs, and the Blynk application connected via a smartphone. LDR and DHT11 sensors are used to detect light intensity, temperature, and humidity. Data from the sensors is sent to the Raspberry Pi for analysis, which then controls the relay module based on the data. The relay subsequently controls the LED to turn on or off as an indicator. Additionally, the Raspberry Pi is connected to a Wi-Fi network and can communicate with the Blynk app, enabling users to monitor environmental conditions and control the LED directly via a smartphone.

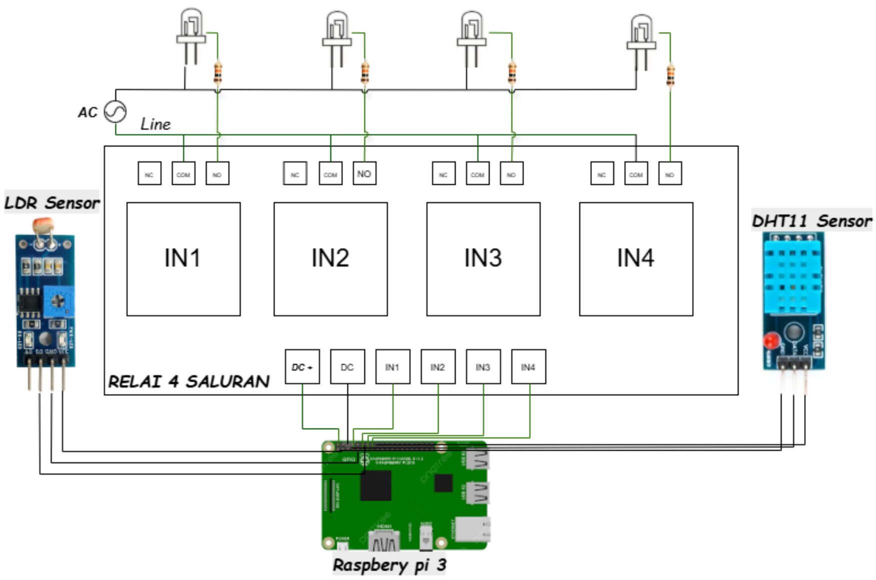


Figure 3 Circuit Diagram

Table 1 Result analysis of planting media

No	Pin Raspberry	Output
1	5V	VCC (Relay)
2	GND	GND (Relay)
3	GPIO14	IN1 (Relay)

SITEKIN: Jurnal Sains, Teknologi dan Industri, Vol. 22, No. 2, pp.280 - 294
ISSN 2407-0939 print/ISSN 2721-2041 online

4	GPIO15	IN2 (Relay)
5	GPIO23	IN3 (Relay)
6	GPIO24	IN4 (Relay)
7	3V3	VCC (DHT11)
8	GND	GND (DHT11)
9	GPIO4	DAT (DHT11)
10	5V	VCC (LDR)
11	GND	GND (LDR)
12	GPIO17	DO(LDR)

Figure 3 and Table 1 illustrate the wiring diagram for the entire home automation circuit. For the 4-channel relay module, the relay's VCC pin is connected to the Raspberry Pi's 5V pin to supply power, and the relay's GND pin is connected to the Raspberry Pi's GND pin. The four relay control pins—IN1, IN2, IN3, and IN4—are each connected to the Raspberry Pi's GPIO pins, namely GPIO14, GPIO15, GPIO23, and GPIO24. These pins are used to control each relay channel. Next, for the DHT11 temperature and humidity sensor, the DHT11 VCC pin is connected to the Raspberry Pi's 3.3V pin, while the DHT11 GND is connected to the Raspberry Pi's GND pin. The DHT11 sensor's data pin, DAT, is connected to the Raspberry Pi's GPIO4 pin to read temperature and humidity data. Finally, for the digital LDR light sensor, the LDR's VCC pin is connected to the Raspberry Pi's 5V pin, and the LDR's GND pin is connected to the Raspberry Pi's GND pin. The digital output from the LDR sensor (DO pin) is connected to the Raspberry Pi's GPIO17 pin to read the lighting conditions (bright or dark).

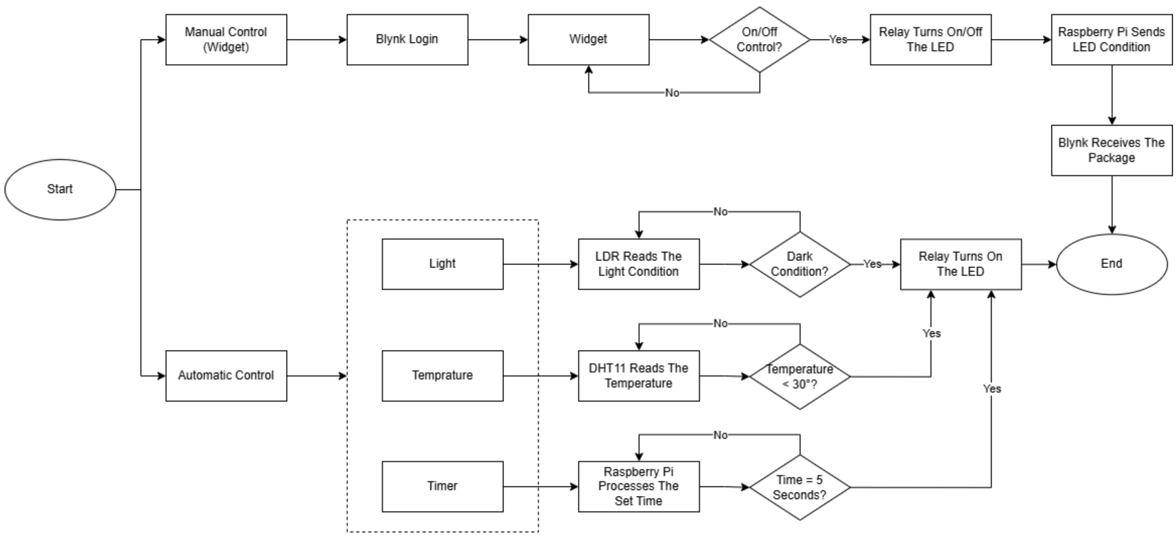


Figure 4 System Flowchart

Figure 4 shows the workflow of the Raspberry Pi-based automatic and manual LED light control system with sensor integration and the Blynk application. The system starts with two control options: manual control and automatic control. In manual control mode, the user first logs into the Blynk application, then uses the control widget to turn the LED on or off. If the user activates the control, the relay will turn the LED on or off, and the Raspberry Pi will send the LED status to the Blynk server, so that the user can view the status through the application. This process ends after Blynk receives the data packet.

Meanwhile, in automatic control mode, the system utilizes three main parameters, namely light, temperature, and time [11]. An LDR sensor is used to read light conditions. If the detected light condition is dark, the relay will turn on the LED. If not, the LED remains off and the LDR returns to reading the light condition. For temperature readings, the system uses a DHT11 sensor. If the detected temperature is below 30°C, the LED will be turned on. If not, the LED remains off, and the DHT11 will resume temperature readings. Finally, for time control, the system processes the predefined time interval of 5 seconds. If this time is reached, the LED will be turned on. If not, the LED remains off until the predefined time is reached again.

Results and Discussion

Results

Hardware Implementation

The hardware assembly for designing the smarthome in this study was carried out based on the schematic circuit configuration shown in Figure 3.

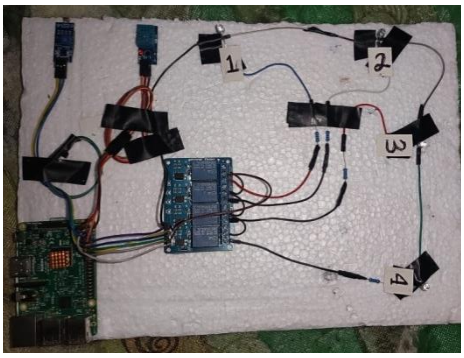


Figure 5 System Prototype

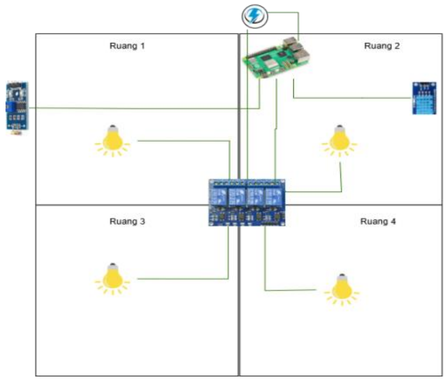


Figure 6 Prototype Implementation Diagram

Figure 5 shows the physical appearance of the designed system, while Figure 6 shows the implementation diagram of the designed system. This system uses a Raspberry Pi 3 microcontroller as the main control unit, equipped with a 4-channel relay module that functions to regulate the flow of electric current to the LED load. Additionally, an LDR sensor is used to detect ambient lighting conditions (bright or dark), and a DHT11 sensor is employed to monitor the air temperature around the device.

In this prototype, there are four LEDs, each controlled by a different mechanism:

1. LED 1 is automatically controlled based on the readings from the LDR sensor. The LED will turn on when the sensor detects low light conditions and will turn off again when bright conditions are detected.
2. LED 2 is automatically controlled based on the temperature read by the DHT11 sensor. The LED will turn on when the temperature exceeds 30°C and will turn off when the temperature drops below that threshold.
3. LED 3 is controlled using an automatic timer system, which turns it on for 5 seconds and then off for 5 seconds in a repeating cycle.
4. LED 4 is manually controlled via the Blynk app connected to the Raspberry Pi 3, allowing users to control the LED remotely using a smartphone.

Software Implementation

The software uses a Raspberry Pi 3 controlled via the Internet of Things (IoT) with the Blynk application. The program is written in Python with supporting modules such as RPi.GPIO for GPIO pin control, Adafruit_DHT for the DHT11 temperature sensor, and BlynkLib for communication with the Blynk application.

GPIO Initialization and Configuration

The program initializes the Raspberry Pi GPIO pins that will be used to control four LEDs via relays, as well as read inputs from a light sensor (LDR) and a DHT11 temperature sensor.

```
Start the program

Disable GPIO warnings
Set GPIO mode to BCM numbering

Define GPIO pins:
    LED_CAHAYA    ← GPIO 14  (Output for light sensor)
    LED_SUHU      ← GPIO 15  (Output for temperature sensor)
    LED_TIMER     ← GPIO 23  (Output for automatic timer)
    LED_MANUAL    ← GPIO 24  (Output for manual control via Blynk)
```

```
    SENSOR_CAHAYA ← GPIO 17  (Input from digital light sensor/LDR)
    SENSOR_SUHU   ← GPIO 4   (Input from temperature sensor DHT11)

Set GPIO pin modes:
    Set LED_CAHAYA as OUTPUT
    Set LED_SUHU as OUTPUT
    Set LED_TIMER as OUTPUT
    Set LED_MANUAL as OUTPUT
    Set SENSOR_CAHAYA as INPUT

Set all LEDs to initial OFF state:
    LED_CAHAYA  ← LOW
    LED_SUHU    ← LOW
    LED_TIMER   ← LOW
    LED_MANUAL  ← LOW
```

Initializing Blynk

The program uses authentication tokens from Blynk to connect the Raspberry Pi to the cloud-based Blynk application.

```
Start the program

Import the Blynk library

Store the Blynk authentication token in a variable:
    BLYNK_AUTH ← 'T6YesasbjImjfknc3WR6He406pb0EU2d'

Connect to the Blynk server using the following information:
    - Authentication token: BLYNK_AUTH
    - Server: blynk.cloud
    - Port: 80
```

LED Control with Blynk Virtual Pins

Manually control LEDs through the Blynk app using virtual pins 0 to 3. Any change in the value of a virtual pin will change the status of the associated LED.

```
If the Blynk application sends data to Virtual Pin V0:
    If the value ≠ 0:
        Turn ON the light sensor LED
        Display "Cahaya ON"
    If the value = 0:
        Turn OFF the light sensor LED
        Display "Cahaya OFF"

If the Blynk application sends data to Virtual Pin V1:
    If the value ≠ 0:
        Turn ON the temperature LED
        Display "Suhu ON"
    If the value = 0:
        Turn OFF the temperature LED
```

SITEKIN: Jurnal Sains, Teknologi dan Industri, Vol. 22, No. 2.280 - 294
ISSN 2407-0939 print/ISSN 2721-2041 online

```
        Display "Suhu OFF"

If the Blynk application sends data to Virtual Pin V2:
    If the value ≠ 0:
        Turn ON the timer LED
        Display "Timer ON"
    If the value = 0:
        Turn OFF the timer LED
        Display "Timer OFF"

If the Blynk application sends data to Virtual Pin V3:
    If the value ≠ 0:
        Turn ON the manual LED
        Display "Manual ON"
    If the value = 0:
        Turn OFF the manual LED
        Display "Manual OFF"
```

Light Sensor Monitoring (LDR)
The LDR sensor is connected to the digital GPIO pin. The data received is HIGH or LOW, indicating light or dark conditions. LED 1 is automatically controlled based on this light sensor.

```
Function: monitor_light_sensor

    Repeat continuously:
        If the light sensor detects a bright condition:
            Display "Bright! LED OFF"
            Turn OFF the light LED (set GPIO pin to HIGH)

        If the sensor detects a dark condition:
            Display "Dark! LED ON"
            Turn ON the light LED (set GPIO pin to LOW)

        Wait for 1 second before repeating
```

DHT11 Temperature Sensor Control
The DHT11 temperature sensor is read every 2 seconds. If the temperature exceeds a certain limit (30°C), LED 2 will light up as an indicator.

```
Import the DHT11 sensor library

Set the temperature sensor to use:
    DHT_SENSOR ← DHT11

Set the temperature threshold:
    TEMPERATURE_LIMIT ← 30°C

Function: kontrol_suhu

    Repeat continuously:
```

```
        Read temperature and humidity data from the DHT11 sensor

        If temperature reading is successful:
            Display the temperature and humidity values
            If temperature ≥ TEMPERATURE_LIMIT:
                Turn ON the temperature LED (set GPIO to HIGH)
                Display "Hot! LED ON"
            If temperature < TEMPERATURE_LIMIT:
                Turn OFF the temperature LED (set GPIO to LOW)
                Display "Cool! LED OFF"

        If temperature reading fails:
            Display "Failed to read temperature sensor!"

        Wait for 2 seconds before repeating
```

Automatic LED Timer (LED 3)
LED 3 turns on and off alternately every 5 seconds using time.sleep() in an infinite loop.

```
Function: toggle_timer

    Repeat continuously:
        Turn ON the timer LED (set GPIO to HIGH)
        Display "LED ON for 5 seconds"
        Wait for 5 seconds

        Turn OFF the timer LED (set GPIO to LOW)
        Display "LED OFF for 5 seconds"
        Wait for 5 seconds
```

Starting Programs and Running Threads
The sensor and timer functions are run in separate threads so that they run in parallel, while the main loop is responsible for running the Blynk process.

```
Import the threading and time libraries

Call  the  check_blynk_connection()  function  to  ensure  Blynk  is
connected

Run the following functions concurrently (in separate threads):
    - monitor_light_sensor (automatic light sensor monitoring)
    - kontrol_suhu (automatic temperature monitoring)
    - toggle_timer (automatic LED timer control)

Start the main loop:
    While the program is running:
        Execute  blynk.run()  to  handle  commands  from  the  Blynk
application
        Wait for 0.1 seconds
```

```
If the user interrupts the program (KeyboardInterrupt):
    Display "Program stopped."
    Clean up all GPIO configurations
```

Blynk Connection Check Function

This function runs periodically to ensure that the Raspberry Pi remains connected to the Blynk server.

```
Function: check_blynk_connection

    Display message:
        "Raspberry Pi successfully connected to Blynk 2.0!"

    Run the check_blynk_connection function again after 10 seconds
```

Blynk Application Settings

The Blynk app is used to manually control LED 4. The Blynk app is set up by downloading the Blynk IoT app from the Play Store or App Store, then creating a new template on the blynk.cloud website by selecting the Raspberry Pi device and WiFi connection.

```
#define BLYNK_TEMPLATE_ID "TMPL6jYZMeLGv"
#define BLYNK_TEMPLATE_NAME "Home Automaton with Raspberry Pi3"
#define BLYNK_AUTH_TOKEN "T6YesasbjImjfknc3WR6He406pb0EU2d"
```

Figure 7. Bylnk token

After that, create a device from the template that has been created and copy the Auth Token as shown in Figure 7 to be entered into the Python program. In the Blynk application, add a Button widget associated with Virtual Pin V3 and set it in switch mode (ON/OFF) as shown in Figure 8.

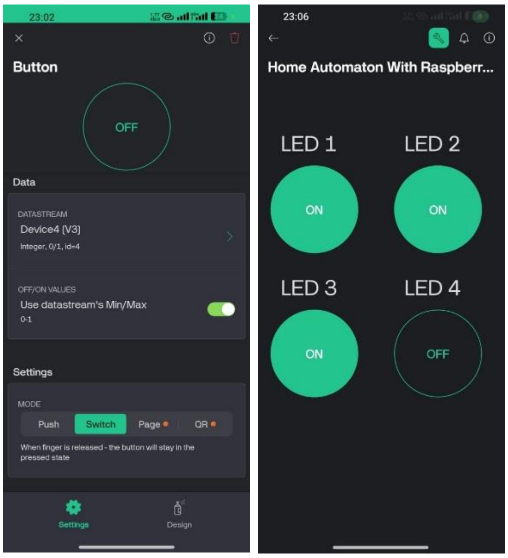


Figure 8 Virtual pin & Blynk application display

The obtained Auth Token is entered into the Python code to connect the Raspberry Pi with Blynk. After the program is run on the Raspberry Pi, LED 4 can be controlled manually via the buttons available in the Blynk application. After all stages of the system development are completed, testing of the light control and monitoring functions is performed using the Blynk application, as shown in Figure 8.

SITEKIN: Jurnal Sains, Teknologi dan Industri, Vol. 22, No. 2, pp.280 - 294
ISSN 2407-0939 print/ISSN 2721-2041 online

Testing

LED Testing 1 Based on LDR Sensor

This test was conducted to ensure that LED 1 can be controlled automatically based on ambient lighting conditions, i.e., only distinguishing between light and dark conditions. The testing process was carried out by adjusting the lighting around the LDR sensor and observing the response of LED 1. The test results are presented in Table 3 below:

Table 2 LED 1 Test Results

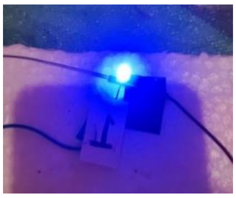
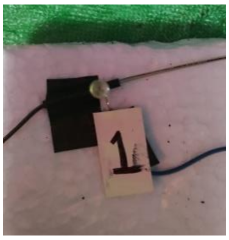
Environmental Conditions	LED Status 1
dark! LED On	
Bright! LED Off	

Table 2 shows the results of testing LED 1 controlled by a light sensor (LDR). Based on the test, when the LDR sensor detects dark environmental conditions, LED 1 will turn on automatically. Conversely, when the environmental conditions are bright, LED 1 will turn off. This shows that the system works well according to its designed function.

LED Testing 2 Based on Temperature Sensor

This test aims to evaluate the response of LED 2 to changes in the ambient temperature. The temperature sensor used controls LED 2 with a specific temperature threshold. From the test results, LED 2 turns on when the ambient temperature exceeds 30°C and turns off when the temperature is below 30°C. Table 3 presents the test results obtained.

Table 3 LED Test Results 2

Temperature (°C)	LED Status 2
Temperature: 27 0c, Humidity: 86.6%, Cold LED On	
temperature: 30 0c, humidity: 93%, Hot! LED Off	

temperature: 32 0c, humidity: 95%, Hot! LED Off	

Table 3 shows the results of testing LED 2 controlled by the DHT11 temperature sensor. Based on the test, LED 2 only lights up when the sensor reads a temperature of 27°C. Meanwhile, at temperatures of 30°C and 32°C, LED 2 is off. This shows that the system is designed to turn on LED 2 only at temperatures below 30°C. This experiment indicates that the temperature control system has successfully functioned according to the designed logic, which is to activate the LED as an indicator when the temperature is within a certain range.

LED Testing 3 Based on Time Scheduling

This test aims to ensure that LED 3 can turn on and off alternately at a predetermined time interval, namely 5 seconds on and 5 seconds off. The observation results using a stopwatch show that LED 3 successfully follows the specified time pattern with good accuracy. The test data can be seen in Table 4.

Table 4 LED 3 Test Results

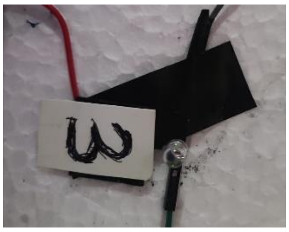
Status on Raspberry	LED Status 3
LED On for 5 seconds	
LED Off for 5 seconds	

Table 4 shows the results of testing LED 1 controlled by a light sensor (LDR). Based on the test, when the LDR sensor detects dark environmental conditions, LED 1 will turn on automatically. Conversely, when the environmental conditions are bright, LED 1 will turn off. This shows that the system works well according to its designed function.

Testing 4 LEDs via the Blynk Application

This test was conducted to test the ability to control LED 4 remotely through the Blynk application connected to Raspberry Pi 3. The test results showed that the ON and OFF commands from the application were successfully received and responded to by the system. Table 5 summarizes the results of this test.

SITEKIN: Jurnal Sains, Teknologi dan Industri, Vol. 22, No. 2, pp.280 - 294
ISSN 2407-0939 print/ISSN 2721-2041 online

Table 5 LED 4 Test Results

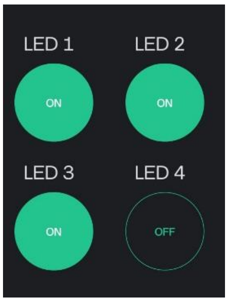
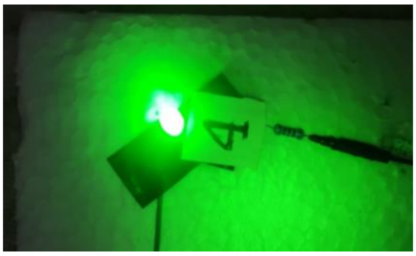
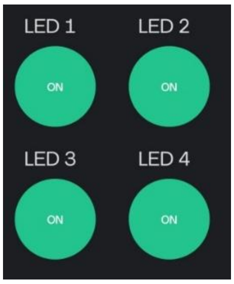
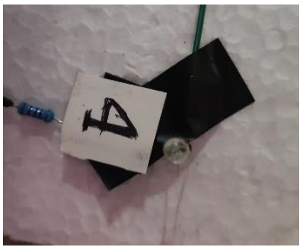
Actions on the Application	LED Status 4
Pressing the on button 	
Pressing the off button 	

Table 5 shows the test results of LED 4 controlled manually through the Blynk app. In this test, LED 4 can be turned on and off directly by the user using the virtual buttons available in the Blynk application on the smartphone. When the ON button is pressed the LED 4 will turn on, and when the OFF button is pressed the LED 4 will turn off. The response from the system takes place in real-time according to the commands given through the application. The results of this test show that the communication between the microcontroller system and the Blynk platform runs well and the manual control feature works as designed.

Overall System Testing

After each LED is tested individually, an overall system test is conducted to ensure that all components can work in unison and in accordance with the conditions that have been designed. This test involves all control elements, namely the light sensor (LDR), temperature sensor (DHT11), automatic timer, and manual control through the Blynk application. The following are the overall system test results:

Table 6 Overall Smarthome System Testing Results

No	Light Condition	Temperature (°C)	Timer Status	Blynk Button	LED 1 (LDR)	LED 2 (Temp)	LED 3 (Timer)	LED 4 (Manual)
1	Dark	32	Active (On 5s)	OFF	ON	OFF	ON	OFF
2	Bright	30	Active (Off 5s)	ON	OFF	OFF	OFF	ON
3	-	27	-	-	-	ON	-	-

SITEKIN: Jurnal Sains, Teknologi dan Industri, Vol. 22, No. 2.280 - 294
ISSN 2407-0939 print/ISSN 2721-2041 online

Based on the test results presented in Table 4.5, it can be concluded that the designed smarthome system has functioned properly and in accordance with the design objectives. LED 1 lights up when the environment is dark and will turn off automatically when detected as bright by the light sensor (LDR). LED 2 lights up only when the temperature is below 30°C and will turn off when the temperature reaches or exceeds that value, according to the data read by the DHT11 temperature sensor. LED 3 is controlled by an automatic timing system that sets the on and off cycle every 5 seconds alternately and repetitively without user intervention. Meanwhile, LED 4 functions well in receiving manual commands from the user through the Blynk application, both to activate and deactivate the LED in real-time. The results of this test prove that all system components can work simultaneously (parallel) without interruption, and the Raspberry Pi 3 is able to handle various input and output processes in multitasking with stability. Thus, this Internet of Things (IoT)-based smarthome system is feasible to be applied in everyday life, especially for controlling home electrical devices automatically or remotely.

The results of this study not only show that the system successfully controls LEDs according to light conditions, temperature, time, and manual commands, but also prove that the system has the flexibility of controlling other electronic equipment. In line with research [12] smarthomes function to control various kinds of electronic devices in the home. In this study, the LED acts as a load simulation, so technically this system can be applied to control household electrical devices such as room lights, fans, water pumps, or other equipment connected through relay modules. With the Raspberry Pi 3's ability to handle various inputs and outputs simultaneously and the support of remote control through the Blynk application, this system proves to have potential as an initial solution in the development of a smarthome that is more extensive, efficient, and responsive to user needs.

Discussion

This research successfully designed and implemented an Internet of Things (IoT)-based smart home system by utilizing Raspberry Pi 3 as the control center. The system supports two operating modes, namely automatic and manual, through the Blynk application. In the test, the LEDs successfully turn on and off based on three automation scenarios namely light sensor (LDR), temperature sensor (DHT11), and time scheduling, and can be controlled manually remotely using a smartphone. The test results show that the system is able to perform control functions in four different scenarios with a fast and accurate response. The use of LDR sensors proved effective in detecting lighting conditions to automatically control LEDs, in line with the working principle of IoT-based light control systems as described by [13], which states that LDR can be used for automatic lighting optimization in smart home systems. Meanwhile, the DHT11 sensor is able to detect the ambient temperature in real-time and provide a control response to temperatures above a specified threshold, proving that this system can function adaptively to the environment.

This finding supports the IoT theory which states that devices can be interconnected through the internet network to perform functions automatically and efficiently without direct intervention from the user [1]. The implementation of Raspberry Pi as the main controller proved effective in managing data from various sensors and controlling output devices, as stated by [2] that Raspberry Pi has high integration capabilities for IoT-based automation systems. However, the advantage of this research lies in the integration of multi-sensors in one comprehensive automation system, which has not been widely raised in previous research. For example, [7] only designed a mobile-based light control system without sensor integration, while [5]focused more on home door security with RFID features.

The utilization of the Blynk application in this system makes it easy for users to control and monitor devices in real-time through an interactive graphical interface, in accordance with the findings [10] regarding the ease of developing IoT projects using the platform. The authentication feature through tokens in Blynk also guarantees the security of user access to the system. In addition, according to [14], combining IoT-based control with mobile applications allows the system to be more flexible in supporting scalable and secure automation. [15] also emphasized that a secure and cloud-based smart home system must prioritize access authentication and control scalability, which has been accommodated in this system through the use of Blynk tokens and wireless networks. Furthermore, research by [12] showed that the Raspberry Pi-based home automation system has high reliability and is suitable for low-cost residential environments, supporting the contribution of this research to the development of smart homes that are modular, cost-effective, and easy to implement. The system is also a solution to the user's problem of efficiently controlling home appliances when outside the home, for example when forgetting to turn off electrical appliances.

Conclusion

This research successfully designed and built a prototype of Internet of Things (IoT)-based smarthome system integrated with Blynk application, using Raspberry Pi 3 as the main control center. The system consists of a 4-channel relay module, LED, LDR light sensor, and DHT11 temperature sensor, each of which is controlled automatically or manually according to its function. The ability of the system to respond to predetermined test parameters, namely light intensity (light/dark), ambient temperature (with a threshold of 30°C), automatic timing (5 seconds on and 5 seconds off), and manual control through the Blynk application. The test results show that all functions are successfully executed: LED 1 responds to light conditions, LED 2 responds to temperature, LED 3 follows the automatic time interval, and LED 4 responds to commands from the application in real-time. This research also opens up opportunities for the application of cost-effective, flexible, and easy-to-implement home control systems widely, especially to improve energy efficiency and user comfort in daily life. The developed system also provides a basis for further development towards a more complex and integrated smart home system, both in terms of functionality and scalability. This is an important first step towards transforming conventional homes into smart homes that are adaptive to environmental conditions and user needs.

Acknowledge

.

This research is expected to be developed further so that in the future it can be implemented on a larger scale, so that the smart home system designed can be applied in real life in residential and other building environments.

References

[1] A. A. Laghari, K. Wu, R. A. Laghari, M. Ali, and A. A. Khan, “A Review and State of Art of Internet of Things (IoT),” 2022. doi: 10.1007/s11831-021-09622-6.

[2] A. Zein and E. S. Eriana, “PERANCANGAN INTERNET OF THINGS (IOT) SMART HOME,” *SAINSTECH J. Penelit. DAN Pengkaj. SAINS DAN Teknol.*, 2022, doi: 10.37277/stch.v31i2.1156.

[3] I. T. Yuniahastuti, S. Kartikawati, and ..., “Implementasi Peralatan Elektronik Berbasis Mikrokontroller dengan Aplikasi Blynk pada Mata Kuliah Instalasi Listrik,” *Snpte Semin. …*, pp. 55–60, 2023.

[4] A. P. Launuru, G. Manu, H. K. Tupan, and R. Hutagalung, “Rancang Bangun Sistem Kontrol Nirkabel On-Off Peralatan Listrik Dengan Perintah Suara Menggunakan Smartphone Android,” *J. SIMETRIK*, 2021, doi: 10.31959/js.v11i1.570.

[5] W. Kusuma *et al.*, “Pi Berbasis Internet of Things,” vol. 2, pp. 74–85, 2023.

[6] I. P. G. Surya Angga Pranata, I. M. Suartika, and I. W. Arta Wijaya, “ALAT PENDETEKSI PENEBANG LIAR MENGGUNAKAN SENSOR SUARA BERBASIS IoT-RASPBERRY PI,” *J. SPEKTRUM*, vol. 8, no. 3, p. 141, 2021, doi: 10.24843/spektrum.2021.v08.i03.p17.

[7] M. Adrian, “Sistem Kontrol Pencahayaan Internet Of Things (IOT) Menggunakan Raspberry Pi Berbasis Ponsel,” *J. Portal Data*, vol. 2, no. 10, pp. 1–11, 2022, [Online]. Available: http://portaldata.org/index.php/portaldata/article/view/260%0Ahttp://portaldata.org/index.php/portaldata/article/download/260/230

[8] Ayyina Dewi Parwati, “Implementasi Dan Perancangan Kamera Nirkabel Menggunakan Raspberry Pi,” *Tek. J. Ilmu Tek. dan Inform.*, vol. 2, no. 3, pp. 77–81, 2022, doi: 10.51903/teknik.v2i3.161.

[9] A. Restu Mukti, C. Mukmin, E. Randa Kasih, D. Palembang Jalan Jenderal Ahmad Yani No, S. I. Ulu, and S. Selatan, “Perancangan Smart Home Menggunakan Konsep Internet of Things (IOT) Berbasis Microcontroller,” *J. JUPITER*, vol. 14, no. 2, pp. 516–522, 2022.

[10] R. Berlianti and Fibriyanti, “Perancangan Alat Pengontrolan Beban Listrik Satu Phasa Jarak JauhMenggunakan Aplikasi Blynk Berbasis Arduino Mega,” *Jurnal, Sain, Energi, Teknol. Ind.*, vol. 5 (1), no. 1, pp. 17–26, 2020, [Online]. Available: https://doi.org/10.31849/sainetin.v5i1.6398

[11] F. Al Ghifari, A. Anjalni, D. Lestari, and U. Al Faruq, “Perancangan Dan Pengujian Sensor Ldr Untuk Kendali Lampu Rumah,” *J. Kumparan Fis.*, vol. 5, no. 2, pp. 85–90, 2022, doi: 10.33369/jkf.5.2.85-90.

[12] V. C. H. Chan, S. M. Beaudette, K. B. Smale, K. H. E. Beange, and R. B. Graham, “A subject-specific approach to detect fatigue-related changes in spine motion using wearable sensors,” *Sensors (Switzerland)*, vol. 20, no. 9, 2020, doi: 10.3390/s20092646.

[13] A. Husna, H. T. Hidayat, and Mursyidah, “Penerapan IoT Pada Sistem Otomatisasi Lampu Penerangan Ruangan Dengan Sensor Gerak Dan Sensor Cahaya Menggunakan Android,” *J. Teknol. Rekayasa Inf. dan Komput.*, vol. 3, no. 1, pp. 2581–2882, 2019.

[14] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications,” *IEEE Commun. Surv. Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015, doi: 10.1109/COMST.2015.2444095.

[15] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, “Fog Computing for the Internet of Things: Security and Privacy Issues,” *IEEE Internet Comput.*, vol. 21, no. 2, pp. 34–42, 2017, doi: 10.1109/MIC.2017.37.