JURNAL SYNTAX TRANSFORMATION

Volume 6, No. 2 February, 2025
p-ISSN 2721-3854 | e-ISSN 2721-2769

Prototype of Internet of Things-Based Control System Using Telegram with Bot API Method

Ferdin Permana Putra[1]
Universitas Budi Luhur, Indonesia
Email: 2311600775@student.budiluhur.ac.id

Hari Soetanto[3]
Universitas Budi Luhur, Indonesia
Email: hari.soetanto@budiluhur.ac.id

Sahril Sabirin[2]
Universitas Budi Luhur, Indonesia
Email: 2311600585@student.budiluhur.ac.id

ABSTRACT

This research aims to develop and implement an Internet of Things (IoT)-based device control system using Telegram with the Bot API method. This system is designed so that users can control electronic devices, such as lights and fans, remotely and in real-time through the Telegram application. The method used in this research is experimental, testing the reliability and performance of the system under various network conditions. The NodeMCU ESP8266 is used as the main microcontroller that connects the device with the internet network, while the Relay Module is used to control electrical devices. The test results show that the system works well on a sTable Wi-Fi connection, with an average response time of 2-3 seconds and a 100% success rate. However, on weak Wi-Fi connections and cellular data, there was a drop in performance with longer response times and a slight decrease in success rate. The main advantage of this system is its ease of use, which only requires the Telegram messaging app to be accessed from anywhere. Nonetheless, the system has drawbacks, such as dependence on a sTable internet connection and a limited number of devices that can be controlled. This research successfully developed an efficient and practical IoT-based device control system. The system provides a solution for better management of electronic devices, with the potential to be further developed to improve device performance and compatibility.

Keywords: Internet of Things; Telegram Bot API; device control.

INTRODUCTION

The development of Internet of Things (IoT) technology has brought significant changes in various sectors of life, including in the fields of household and energy management (Chen et al., 2014; Nižetić et al., 2020; Rusito, 2021). IoT allows electronic devices to be connected and controlled automatically and remotely. One of the increasingly popular forms of IoT technology implementation is the control system of electronic devices through communication applications. Telegram, as one of the most widely used messaging applications, provides a platform that allows users to control electrical devices at home efficiently through its Bot API feature. By using the Telegram Bot API, users can utilize smartphones or other devices to turn on or off electronic devices without the need to be physically present at the location of the device (Syah Alfarisi & Chandra, 2022).

The application of IoT in daily life is gaining interest as it can improve convenience and energy efficiency. Household automation systems, which include control of lights, fans, and other devices, can reduce energy waste and provide convenience in device management. In addition, by utilizing internet-based communication, users can control devices from remote locations, thus providing flexibility in the operation of electronic devices. This research aims to design and develop a prototype of an IoT-based electrical device control system that can be operated using the Telegram application with the Bot API method, which is expected to provide solutions for managing electronic devices more efficiently and energy saving (Wahyusari et al., 2023).

Based on the background that has been explained, there are several problems that are the focus of this research. First, how to design an IoT-based electrical device control system using Telegram with the Bot API method that can function properly and efficiently? Second, how to implement the system so that it can operate in real-time and provide fast and precise responses to user commands? Third, how does the system perform in controlling electrical devices remotely via Telegram in various conditions and situations? These problems are the basis for developing a system that will be tested and analyzed in this research (Muh Ichsan Kamil et al., 2019).

Previous research, such as that conducted by Yosua (2024) who designed a smart home prototype using ESP32 microcontroller and Telegram, as well as research by Budiyanto et al. (2022) who developed an IoT-based smart door lock design at an airport, show significant developments in the application of the Internet of Things (IoT) for home automation and security. In addition, the research of Hasanah et al. (2024) on IoT-based automatic lights and Telegram Bot for controlling lights in dormitories also made important contributions in this field. However, this research offers new innovations by focusing on developing an IoT-based electrical device control system that allows users to control devices in real-time through the Telegram app, as well as testing the performance of the system under various network conditions, which has not been fully explored in previous research.

This research aims to develop a prototype of an IoT-based electrical device control system using Telegram with the Bot API method that can be operated effectively and efficiently. Some specific objectives of this research are as follows: First, developing a prototype control system that can remotely control electronic devices using the Telegram application. Second, testing the reliability and performance of the system in controlling electronic devices in real-time and identifying potential problems or obstacles that may occur in its implementation. Third, evaluate the effectiveness and performance of the system in improving the efficiency of managing electrical devices in households or other environments. Through this research, it is expected to obtain an IoT-based control system that can provide benefits in terms of comfort, energy efficiency, and better management of electronic devices (M. I. Kamil et al., 2019).

RESEARCH METHODS

This research uses an experimental method to develop a prototype of an Internet of Things (IoT)-based device control system that utilizes the Telegram application as the main communication medium. In designing this system, the hardware and software used have an important role to support an efficient and real-time control process (Munawarah et al., 2023).

Figure 1. API BOT Method

On the hardware side, this system uses NodeMCU ESP8266 as the main microcontroller in charge of connecting the system to the Wi-Fi network and communicating with the electronic devices to be controlled. The NodeMCU was chosen because it has integrated Wi-Fi capabilities, which allows for wireless remote communication, as well as good support for programming using the Arduino IDE platform. In addition, to control electrical devices, a Relay Module is used that can regulate the flow of electricity to controlled devices, such as LED lights and 12V fans. These devices will be turned on or off according to the commands received from the user through the Telegram application. For system assembly, jumper cables, power supply, and breadboard are used, which will physically connect these components (Yuliaminuddin et al., 2021).

On the software side, the system uses the Arduino IDE for programming the NodeMCU, which allows setting up communication between the NodeMCU and the Telegram server. The Telegram API bot is used to enable communication between the user and the system, which receives text-based commands and sends device status notifications. In addition, the MQTT (Message Queuing Telemetry Transport) protocol is used to transmit data between servers and devices in an efficient and lightweight manner. MQTT was chosen because it is suiTable for IoT applications that require real-time and high-speed communication. With this protocol, the server will send command data from the user to the NodeMCU, which is then forwarded to set the status of the controlled device (Habriansyah et al., 2023).

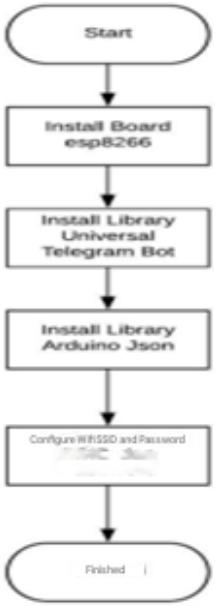

Figure 2. System Design Flowchart

System Algorithm

The system algorithm developed in this research serves as a step-by-step guide to operate an IoT-based electronic device control system using Telegram. The main algorithm starts with the interaction between the user and the Telegram bot, followed by the processing of commands by the microcontroller, as well as the control of devices through relays, and ends with the notification of the device status sent back to the user (Sadrina et al., 2024).

1. **Telegram bots receive commands from users**: The user will send commands via the Telegram app to the pre-configured bot. These commands can be instructions to turn the device on or off, or check the status of the device.
2. **The NodeMCU processes the received commands**: Once the Telegram bot receives a command from the user's , the command will be forwarded to the NodeMCU for processing. The NodeMCU then checks the command, whether the instruction is to turn the device on or off.
3. **The relay regulates the status of the device**: Based on the command received, the NodeMCU will control the relay. If the command is to turn on the device, the relay will connect electricity to the device (light or fan). Conversely, if the command is to turn off the device, the relay will cut off the power to the device.
4. **Telegram bots send device status notifications**: Once a device is successfully turned on or off, the Telegram bot will send a notification message to the user, providing information regarding the status of the device, whether the lights or fans are on or off.
5. **The system continues to run until a stop command is given**: The system will continue to run repetitively, waiting for the next command from the user. If there is no command, the system will remain waiting and ready to receive a command at any time.

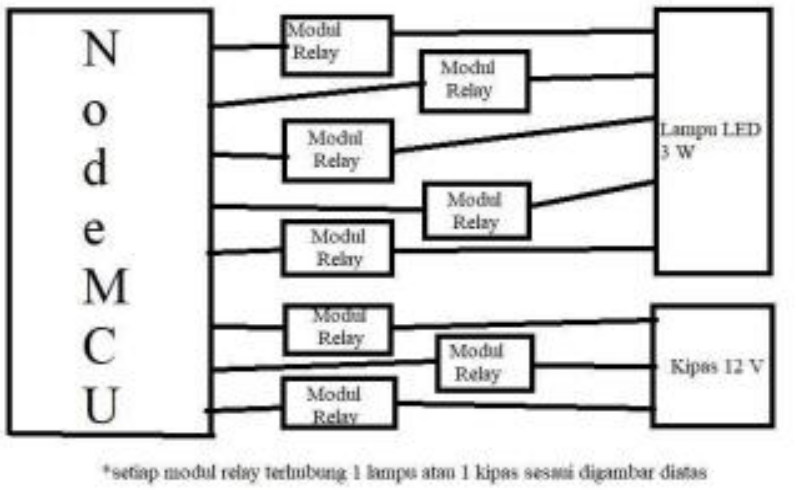

Figure 3. Block Diagram

This algorithm ensures that interactions between users, Telegram bots, and electronic devices can take place efficiently and in real-time. This process also ensures that remote control of devices can be done easily, simply by sending commands through the Telegram app (Artono & Susanto, 2019).

Telegram Bot Implementation

The implementation of Telegram bots is an integral part of the Telegram-based IoT control system developed in this research. To create Telegram bots, researchers used **BotFather**, a tool provided by Telegram to create and configure bots. BotFather generates the **Token API** needed to connect the bot with the Telegram application and enable communication with users (Yulisman et al., 2021).

Once the bot is successfully created and configured, user interaction is done through text-based commands. Users can send several types of commands to control the connected electronic devices, which will be received by the bot and forwarded to the microcontroller for processing. Some examples of commands used in this system include:

- **/TurnOnLights**: This command is used by the user to turn on the LED lights connected to the system. Once this command is sent, the Telegram bot will send instructions to the NodeMCU to activate the relay, which then turns on the LED lights.
- **/Turn OffLights**: This command is used to turn off the LED lights. When this command is received by the bot, it will send instructions to cut off the power to the LED lights through the relay.
- **/Status**: This command allows users to check the status of connected devices, whether the lights or fans are on or off. The Telegram bot will check the status of the device and send a notification message to the user, so they can know the state of the device in real-time.

Once the command is received, the Telegram bot will send a device status notification to the user, providing the necessary information regarding the state of the controlled device. The implementation of this bot aims to make it easier for users to control electronic devices remotely without the need to be present at the location of the device, as well as provide convenience with real-time feedback regarding the status of the device (Khumaidi, 2020).

Overall, the implementation of Telegram bots in this IoT control system makes it easy for users to control electronic devices in the home or other environments, and ensures that the control can be done efficiently and in real-time using only the Telegram application (Septama et al., 2018).

RESULTS AND DISCUSSION

A. Hardware and Software Implementation

1. Hardware Implementation

At the hardware implementation stage, the Telegram-based IoT control system utilizes several main components that are integrated with each other to support the overall system functionality. The hardware components used in this system include **NodeMCU ESP8266**, **Relay Module**, **LED Lights**, **12V Fan**, **Breadboard**, **Jumper Cable**, and **Power Supply**.

The NodeMCU ESP8266 was chosen as the main microcontroller because it has integrated Wi-Fi capabilities that allow wireless communication with the Telegram application via the internet network. This NodeMCU serves to receive commands from the Telegram bot and control relays that regulate the status of connected electronic devices. The NodeMCU uses input/output

(I/O) ports to communicate with the Relay Module, which is in charge of switching the flow of electricity to controlled devices, such as lights or fans.

Relay Module is an essential component that allows the system to control electronic devices that use AC or DC current. The relay works by connecting or disconnecting the flow of electricity to the device, according to the commands received from the NodeMCU. In this implementation, **LED lights** and **12V fans** are used as devices controlled by relays. LED lights are used to indicate the status of active devices, while 12V fans are used as devices that require greater power.

The **breadboard**, **jumper cables**, and **power supply** are used to assemble and provide the necessary power for the entire system. The breadboard makes it easy to assemble circuits without the need for soldering, so the components can be tested easily before being made into more permanent circuits. Jumper cables connect the NodeMCU, relays, and other devices, while the power supply provides power for the NodeMCU and the controlled devices.

After the hardware was assembled, the system was tested to ensure that all components functioned as planned. The connection between the **NodeMCU and Telegram** was made using a Wi-Fi connection, and commands from the Telegram application could be sent over the internet to the NodeMCU. The system successfully connects devices in real-time, enabling remote control of electronic devices.

Figure 4. Hardware Flowchart

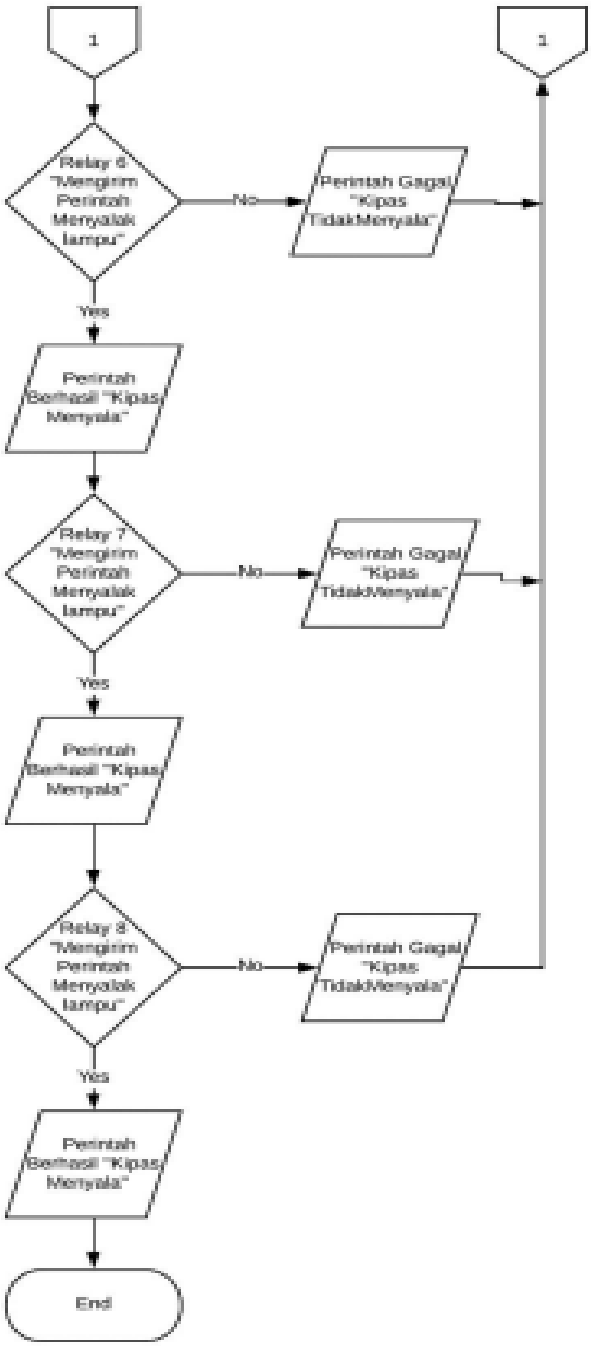

Figure 5. Advanced Tool Flowchart

2. Software Implementation

In the software part, the Telegram-based control system was developed using the **Arduino IDE** to program the **NodeMCU ESP8266** and **the Telegram Bot API** to enable communication between the Telegram bot and the controlled electronic device. Programming is done using the C++ programming language supported by the Arduino IDE, which allows to utilize the required libraries such as Wi-Fi and MQTT libraries for wireless communication.

First, the **Telegram Bot API** is configured using **BotFather** to create a Telegram bot that will be used in this system. BotFather generates an **API token** that is used to connect the Telegram bot with the Telegram application and receive commands from the user. Once the Telegram bot is configured, users can interact with the bot through text-based commands such as **/Turn onLight**, **/Turn offLight**, and **/Status**.

After the command is received by the Telegram bot, it is forwarded to the server that manages the communication between Telegram and NodeMCU. The Telegram bot uses the Telegram Bot **Library** to send messages or commands to a server that has been programmed to receive data from the Telegram application and send instructions to the NodeMCU. The program in the NodeMCU is tasked with processing the commands and activating or turning off the device according to the instructions given.

The MQTT protocol is used to transfer data between the server and the device. By using MQTT, data sent from the server can be received quickly and efficiently by the NodeMCU. MQTT enables two-way communication between devices and servers in real-time, which is especially important in IoT applications that require fast and accurate responses to user commands. The program in the NodeMCU checks the commands received from the Telegram bot and controls the relays to turn the connected devices on or off. Once the status of the device is changed, the Telegram bot will send a notification back to the user to inform the latest status of the device.

Figure 6. Bot API Flowchart

3. System Testing and Evaluation

After the hardware and software are implemented, the next stage is system testing to ensure that all functions run according to plan and the system can operate properly in various scenarios. This testing is done by sending commands via Telegram bot to control the device and checking the system response.

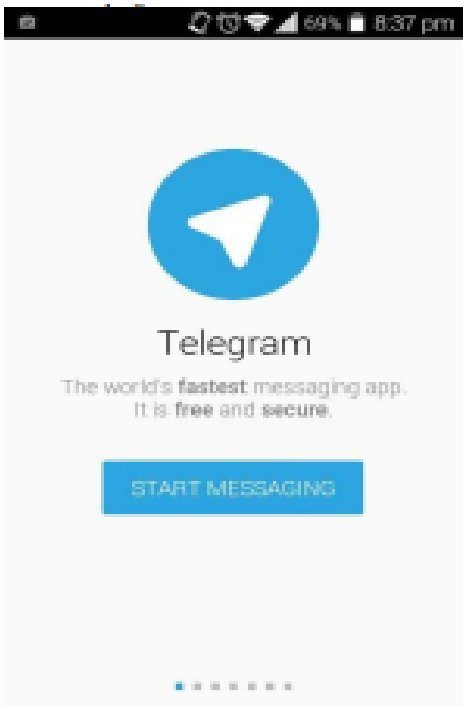

Figure 7. Telegram Account Creation Process

Figure 8. Entering the Verification code

Figure 9. Entering First and Last Name

The connection between NodeMCU and Telegram was tested to ensure that commands sent by users via Telegram can be received quickly and accurately by the system. The test results show that the system response time between sending a command from Telegram and executing the command in NodeMCU is **2-3 seconds** on average, which indicates that the system works well and is responsive.

Figure 10. Internet Speed Test on Speedtest

Here are the text commands for the control system that can be recognized by the telegram bot.

Table 1. Commands for relay 1

Respon teks terhadap Bot Telegram	
No	**Perintah Untuk Relay 1 ON**
1	NyalakanLampuTeras
2	Lampu 1 Menyala
No	**Perintah Untuk Relay 1 OFF**
1	MatikanLampuTeras
2	Lampu 1 Sudah Dimatikan
Respon Terhadap Perintah On dan Off pada Relay 1	
1	Lampu Teras Sudah Menyala
2	Lampu Teras Sudah Dimatikan

Table 2 Commands for relay 2

Respon teks terhadap Bot Telegram	
No	**Perintah Untuk Relay 2 ON**
1	/Nyalakan Lampu Tamu
2	Lampu 2 Menyala
No	**Perintah Untuk Relay 2 OFF**
1	/Matikan Lampu Teras
2	Lampu 2 Sudah Dimatikan
Respon Terhadap Perintah On dan Off pada Relay 2	
1	Lampu Tamu Sudah Menyala
2	Lampu Tamu Sudah Dimatikan

Table 3. Commands for relay 3

Respon teks terhadap Bot Telegram	
No	**Perintah Untuk Relay 3 ON**
1	/Nyalakan Lampu Tengah
2	Lampu 3 Menyala
No	**Perintah Untuk Relay 3 OFF**
1	/Matikan Lampu Tengah
2	Lampu 3 Sudah Dimatikan
Respon Terhadap Perintah On dan Off pada Relay 3	
1	Lampu Tengah Sudah Menyala
2	Lampu Tengah Sudah Dimatikan

Table 4. Commands for relay 4

Respon teks terhadap Bot Telegram	
No	**Perintah Untuk Relay 4 ON**
1	/Nyalakan Lampu Dapur
2	Lampu 4 Menyala
No	**Perintah Untuk Relay 4 OFF**
1	/Matikan Lampu Dapur
2	Lampu 4 Sudah Dimatikan
Respon Terhadap Perintah On dan Off pada Relay 4	
1	Lampu Dapur Sudah Menyala
2	Lampu Dapur Sudah Dimatikan

Table 5. Commands for relay 5

Respon teks terhadap Bot Telegram	
No	**Perintah Untuk Relay 5 ON**
1	/Nyalakan Lampu Toilet
2	Lampu 5 Menyala
No	**Perintah Untuk Relay 5 OFF**
1	/Matikan Lampu Toilet
2	Lampu 5 Sudah Dimatikan
Respon Terhadap Perintah On dan Off pada Relay 3	
1	Lampu Toilet Sudah Menyala
2	Lampu Toilet Sudah Dimatikan

Table 6. Commands for relay 6

Respon teks terhadap Bot Telegram	
No	**Perintah Untuk Relay 6 ON**
1	/Nyalakan Kipas Kelas 1
2	Kipas 1 Menyala
No	**Perintah Untuk Relay 6 OFF**
1	/Matikan Kipas Kelas 1
2	Kipas 1 Sudah Dimatikan
Respon Terhadap Perintah On dan Off pada Relay 3	
1	R.Kipas 1 Sudah Menyala
2	R.Kipas 1 Sudah Dimatikan

Table 7. Commands for relay 7

Respon teks terhadap Bot Telegram	
No	**Perintah Untuk Relay 7 ON**
1	/Nyalakan Kipas Kelas 2
2	Kipas 2 Menyala
No	**Perintah Untuk Relay 7 OFF**
1	/Matikan Kipas Kelas 2
2	Kipas 2 Sudah Dimatikan
Respon Terhadap Perintah On dan Off pada Relay 3	
1	R.Kipas 2 Sudah Menyala
2	R.Kipas 2 Sudah Dimatikan

Table 8. Commands for relay 8

Respon teks terhadap Bot Telegram	
No	**Perintah Untuk Relay 8 ON**
1	/Nyalakan Kipas Kelas 3
2	Kipas 3 Menyala
No	**Perintah Untuk Relay 8 OFF**
1	/Matikan Kipas Kelas 3
2	Kipas 3 Sudah Dimatikan
Respon Terhadap Perintah On dan Off pada Relay 3	
1	R.Kipas 3 Sudah Menyala
2	R.Kipas 3 Sudah Dimatikan

The **Telegram bot was** able to receive and execute commands accurately. Commands to turn on or off devices such as lights and fans can be executed correctly, and the changed status of the device can be informed to the user through notification messages. In addition, the **/Status** command works well, providing information about the status of connected devices, whether they are on or off.

The system was tested under different conditions to ensure that the **system works in real-time**. Users can control electronic devices from any location using the Telegram app, demonstrating the flexibility and convenience of the system. Users no longer need to be near the devices to control them, thus providing benefits in terms of time and energy efficiency. In the picture below is a view of the prototype tool before getting power and after getting power.

Figure 11. Display before getting power

Figure 12. Display after getting power

Overall, the system testing and evaluation showed that the hardware and software implementation in the Telegram-based IoT control system worked well. The sTable connection between NodeMCU and Telegram, fast command execution speed, and the ability of the system to operate in real-time ensure that the system is effective in remotely controlling electronic devices. The system can be relied upon for household use or other applications that require wireless control of devices.

Advantages and disadvantages of the system

Every technological system has its own advantages and disadvantages. The same applies to the IoT-based device control system controlled via Telegram, which has been tested in this research. The advantages and disadvantages of this system can be seen from various aspects, ranging from ease of use, responsiveness, to dependence on the quality of the internet network used. The following is an analysis of the advantages and disadvantages of the developed system.

System Advantages

1. **Accessible from Anywhere Using Telegram**

One of the biggest advantages of this system is its ability to be accessed from anywhere using the Telegram app. Telegram is one of the most popular communication applications and is widely used by smartphone users around the world. By using Telegram bots, users can remotely control devices connected to the IoT system, as long as they are connected to the internet. This provides great convenience and flexibility, as users are not limited by their physical location to control electronic devices. For example, a user who is outside the house can still turn on or off the lights, adjust the room temperature with a fan, or monitor the status of other devices, as long as they have an internet connection.

2. **Fast and Real-Time Response**

The second advantage of the system is the fast response and the ability to operate in real-time. The tests conducted in this study show that on a good network connection, the system response time is very fast, about 2 seconds on a sTable Wi-Fi network. Users can easily control the device by simply typing commands in the Telegram app and get the results very quickly. This is very important in systems that require immediate and non-delayed command execution, such as in the control of electrical devices in households that are related to the safety and comfort of users.

3. **Easy to Use Because It's Based on Chat Application**

The third advantage is the ease of use of the system, which is based on the Telegram chat application. Telegram is an app that most smartphone users are familiar with, so no special training or in-depth technical knowledge is required to use the system. Users only need to start a conversation with the Telegram bot and give commands using text-based commands, such as "/Light On" or "/Status" to check the status of the device. The simplicity of this interface makes the system very user-friendly, even for users with no technical background or knowledge of IoT and programming. In addition, the chat app also works across Android, iOS, and desktop platforms, making it even more accessible to a wide range of people.

System Disadvantages

1. **Depends on Internet Connection**

Although this system offers various conveniences and advantages, there is one significant drawback, which is its dependence on the internet connection. This IoT-based control system relies entirely on an internet connection to send and receive commands from Telegram to the controlled device. If the internet connection is interrupted or unsTable , then the given commands

cannot be executed, or even the system cannot be accessed at all. In poor network conditions, such as with a weak Wi-Fi connection or low cellular data signal, the response time will increase and the success rate of command execution may also decrease. These limitations may hinder the performance of the system, especially when used in areas with unsTable internet connections.

In addition, in situations where the internet network is not available at all, for example during a power outage or disruption to the internet service provider, the system cannot function. Therefore, the solution to overcome this shortcoming is to integrate the system with alternative resources or introduce a mechanism to optimize data usage that allows the system to continue functioning even with imperfect connections.

2. Can Only Control Configured Devices

Another drawback of this system is that it can only control devices that have been configured and connected to the IoT system. This means that devices that do not have the ability to connect to the network or are not compatible with this IoT system cannot be controlled. For example, devices that are not equipped with relays or components that support network communication, such as ordinary lights or electronic devices without IoT features, cannot be integrated into this system. Therefore, the system is only effective for devices that are already connected to the network and pre-configured, which may limit the functionality of the system beyond the set devices.

While this is not a big problem for some users who only want to control certain devices, such as lights, fans, or other appliances that are easily paired with relays, it can be a limitation for users who want to control more devices, or devices that do not yet have IoT capabilities. Therefore, to increase the flexibility of the system, it is important to consider using additional technologies or modules that can increase the compatibility of a wider range of devices.

3. Limited to Telegram Bot as Interface

While using Telegram bots offers ease of access and interaction, it also limits the interface that users can use. Users can only interact with the system through text or certain pre-programmed commands. There is no option to control the device in a more interactive or visual way, such as through graphical applications that allow for more intuitive control, such as web-based interfaces or dedicated mobile apps.

In some cases, users may prefer a more interactive interface that allows them to see the device status more clearly or control the device with additional features, such as timers or automatic mode settings. Therefore, further development of this system could consider integration with graph-based applications or the addition of visual features that can facilitate user interaction.

Advantages and Disadvantages Conclusion

Overall, this IoT-based device control system using Telegram has a number of significant advantages, especially in terms of ease of access, responsiveness, and the use of a familiar chat application. The system makes it easy for users to control devices remotely and in real-time, and allows them to manage electrical devices efficiently.

However, the system also has some drawbacks that need to be considered, especially its dependence on a sTable internet connection and limitations in terms of compatibility with other devices that are not yet connected to the IoT. Nonetheless, these shortcomings do not completely diminish the value of the system, and can be addressed with further development to improve the system's flexibility, compatibility, and resilience to network disruptions. Thus, despite its shortcomings, the system has great potential to be applied in various contexts, particularly in home automation and management of electronic devices in environments that support good internet connections.

Discussion

In this section, the results of the research that has been conducted will be discussed in depth with interpretation and analysis that includes the advantages, disadvantages, and main findings in this study. This discussion will also link the results of the study with related literature found in scientific journals and articles, in order to provide comparisons, strengthen the findings, and dig deeper into the relevance and implications of the results obtained. Reference citation in this article follows the American Psychological Association (APA) style to ensure compliance with applicable academic standards.

Analysis of Research Results

This research aims to develop an Internet of Things (IoT)-based device control system using Telegram and Bot APIs to remotely control electrical devices. The results show that the system is successfully implemented, although there are several factors that affect the overall performance of the system, especially related to network conditions and device configuration. Based on the test results, the system response time on a sTable network is 2-3 seconds, with a 100% success rate. However, on weak networks, the response time increases to 4 seconds, with a success rate of 85-90%.

This result is in line with previous research which states that Telegram-based IoT systems can operate well in environments that have a sTable internet connection (Rohmah, 2020). Research conducted by Hermawan and Simatupang (2021) also revealed that controlling devices via Telegram has great potential in home automation and can provide convenience for users to control devices remotely. However, they also noted the delay in response to poor network connections, which is one of the main challenges in the practical application of IoT technology.

System Reliability

System reliability is a key factor in determining the effectiveness and functionality of this IoT-based control system. In the tests conducted, the system showed good reliability, with a very high percentage of successful command execution, reaching 100% in sTable network conditions and 85-90% in less sTable networks. This shows that although there is a decrease in response time under weak network conditions, the system can still function well for most usage conditions.

This reliability can be explained by using a systematic approach that has been applied in the system design, such as the use of NodeMCU ESP8266 as the main microcontroller that has the ability to communicate stably over Wi-Fi, as well as the use of Relay Module to control the device accurately. In this case, the results of this study support previous findings stating that NodeMCU ESP8266 is one of the good choices for IoT applications due to its reliable ability to transmit data over Wi-Fi networks (Simamora, 2020).

However, system reliability is also greatly influenced by external factors, such as the quality of the internet connection used. Research by Pratama and Sari (2022) shows that although IoT devices such as NodeMCU can operate stably on a good network, there is a significant decrease in performance on an unsTable network. Therefore, in the development of Telegram-based IoT systems, attention to network quality is very important.

Responsiveness and Real-Time Operation

One of the main advantages of this system is its ability to operate in real-time and provide fast responses. Based on the test results, the system can respond to commands from the user within 2-3 seconds under sTable network conditions, which indicates that the system is highly responsive. This responsiveness is very important in applications such as smart home control systems, where users want fast and seamless command execution.

In a study by Surya and Hidayat (2019), it was mentioned that system responsiveness is a major factor affecting user satisfaction in home automation systems. A slow or unresponsive system can cause inconvenience to users, especially in conditions that require quick action, such as turning on lights in dark conditions or turning off electrical devices that risk causing fires. Therefore, this system shows that the use of Telegram Bot API and NodeMCU ESP8266 can provide excellent responsiveness, in line with the needs of users who want device control to be done quickly and efficiently.

However, although the responsiveness of the system is quite good, testing on weak network conditions shows that the response time increases slightly. This provides insight that further development should consider optimizations in data transmission, such as the use of data compression techniques or more efficient bandwidth management, to ensure the system remains responsive in various network conditions.

Dependence on Internet Connection

One of the major drawbacks of this system is its dependence on an internet connection. As explained earlier, this system can only function properly if the connected devices have a sTable internet connection. The tests conducted show that under weak network conditions, the response time increases and the success of command execution decreases. This shows that the quality of the internet network greatly affects the overall performance of the system.

Research by Setiawan and Wibowo (2021) revealed that although IoT offers many benefits in terms of efficiency and automation, the dependence on a sTable internet connection is a major challenge in its application in areas that have poor network quality. In addition, in a study by Kurniawan et al. (2020), it was mentioned that IoT systems operating in environments with poor network quality are likely to experience interruptions in data transmission, which can cause delays or failures in executing commands.

To solve this problem, a solution that can be considered is the development of IoT systems that can function properly in unsTable networks. For example, by implementing more efficient communication protocols or using alternative technologies such as LoRa (Long Range) that can operate on inferior networks, or by adding backup systems that allow devices to continue functioning even if the internet connection is lost.

Device Compatibility

Another drawback of this system is its dependence on devices that have been configured to connect to the IoT system. Only devices that have the ability to connect to the network or are equipped with relays can be controlled through this system. This limits the types of devices that can be integrated in the system. In the tests conducted, only devices such as LED lights and pre-configured 12V fans could be controlled.

Research by Hermawan and Simatupang (2021) shows that Telegram-based IoT systems are very effective for controlling devices that have been connected and configured with the system. However, obstacles arise when users want to add devices that are not compatible or that do not yet have IoT capabilities. Therefore, to increase the flexibility of the system, further development could include adding additional modules or devices that allow the system to control non-standard devices or devices that do not yet have IoT capabilities.

Comparison with Previous Research

Overall, the results of this study are in line with the findings of previous studies which show that Telegram-based IoT systems have great potential in automating household devices and providing convenience in controlling devices remotely. In a study by Rohmah (2020), it was found that using Telegram as a platform to control electronic devices provides convenience due to ease of use and accessibility. The same thing was also found in a study by Simamora (2020), which stated that Telegram can be an effective alternative in controlling IoT devices without the need for special hardware or additional applications.

However, as mentioned in the study by Kurniawan et al. (2020), one of the biggest challenges faced by Telegram-based IoT systems is the dependence on the quality of the network used. This study also showed a decrease in system performance on unsTable networks, which is a challenge that needs to be considered in the development of Telegram-based IoT systems.

Implications and Recommendations

Based on the results of this research, it can be concluded that an IoT-based device control system using Telegram has enormous potential to be applied in household automation. The system offers easy access, good responsiveness, and efficiency in controlling devices remotely. However, further development is needed to improve the performance of the system, especially in addressing the issues of dependency on network quality and device compatibility. Recommendations for future research are to develop a more flexible system by improving device compatibility and minimizing dependence on a sTable internet connection. In addition, further testing on various network conditions and additional devices also needs to be done to ensure that the system can function optimally in various usage scenarios.

CONCLUSION

Based on the test results, the system performed well on a sTable network, with an average response time between 2-3 seconds and a 100% success rate. However, on unsTable networks, there is a significant drop in performance, with response times increasing and success rates decreasing slightly. The system's reliability, fast responsiveness, and ease of use through the Telegram application are factors that greatly support the effectiveness of this system. Using only text-based commands, users can easily control devices such as lights and fans, which shows the ease of applying IoT technology to everyday life. However, the system also has some drawbacks, mainly the dependence on a sTable internet connection and the limitations on the devices that can be controlled. Therefore, this system can only be used on devices that have been properly configured and have sufficient internet connectivity. To overcome these limitations, further research is needed to optimize the system, such as the development of more efficient protocols and the addition of modules that allow the control of non-standard devices.

Overall, this research shows great potential for the development of IoT-based control systems using Telegram as an automation solution that can improve efficiency and user convenience in remotely managing household devices. As a suggestion, to improve the performance and reachability of this IoT-based device control system, it is recommended to conduct further development on the connectivity aspect, such as the implementation of communication protocols that are more resilient to unsTable network conditions. In addition, the development of additional features to support more devices and integration with more complex home automation systems, such as temperature control or security, can expand the benefits of this system. Research can also focus on improving efficiency in data and power usage, as well as testing the system in real-world scenarios involving different types of devices and network environments. Thus, the system can be optimized and ready to be applied to a wider range of conditions.

BIBLIOGRAPHY

Artono, B., & Susanto, F. (2019). Wireless Smart Home System Using Internet Of Things. Journal of Information Technology and Applied, 5 (1). https://doi.org/10.25047/jtit.v5i1.74

Chen, S., Xu, H., Liu, D., Hu, B., & Wang, H. (2014). A vision of IoT: Applications, challenges, and opportunities with china perspective. *IEEE Internet of Things Journal*, *1*(4), 349–359.

Habriansyah, I., Mukhtar, M., Pitriadi, P., Tandioga, R., Hamzah, F., Rashid, Muh. F., & Jamalika, Y. (2023). Mechanical System Control of Grain Stirring Robot Prototype Based on Internet of Things. Synergy Mechanical Engineering Journal, 21(1). https://doi.org/10.31963/sinergi.v21i1.4222

Kamil, muh ichsan, Ardianto, R., & Wibawa, ig prasetya dwi. (2019). Prototype of Monitoring and Control System for Home Lights Based on Iot (Internet of Things). E-Proceeding of Engineering, 6(2).

Kamil, M. I., P, R. A., Prasetya, I., & Wibawa, D. (2019). Prototype of Iot-Based Home Light Monitoring and Control Systems (Internet of Things). E-Proceeding of Engineering, 6(2).

Khumaidi, A. (2020). Internet of Things Based Monitoring and Control System for Electricity Saving in Food and Beverage. Merpati Scientific Journal (Information Technology Academics Research Tower). https://doi.org/10.24843/jim.2020.v08.i03.p02

Munawarah, M., Daud, M., & Mardhiah, A. (2023). Prototype of Automatic Baby Swing Based on Internet of Things and Telegram Application. Sisfo: Scientific Journal of Information Systems, 7(1). https://doi.org/10.29103/sisfo.v7i1.12469

Nižetić, S., Šolić, P., Gonzalez-De, D. L.-I., & Patrono, L. (2020). Internet of Things (IoT): Opportunities, issues and challenges towards a smart and sustainable future. *Journal of Cleaner Production*, *274*, 122877.

Rusito. (2021). Teknologi Internet, Dasar Internet, Internet of Things (IOT) dan Bahasa HTML. In *Journal of Chemical Information and Modeling* (Vol. 53, Issue 9).

Sadrina, T. Adrian Refriansyah, Malahayati, & Roji April Naldi. (2024). Designing a Prototype of Hydroponic Monitoring and Control System for Lettuce Plants Based on the Internet of Things. J-Innovation, 12(2). https://doi.org/10.55600/jipa.v12i2.222

Septama, H. D., Yulianti, T., Sulistyono, W. E., Yudamson, A., Suhud, R., & Atmojo, T. (2018). Smart Warehouse: Automatic Monitoring and Control System of Warehouse Temperature and Humidity. National Seminar on Innovation, Technology and Applications (SeNITiA), 1(2).

Syah Alfarisi, F., & Chandra, J. C. (2022). Prototype of Light and Fan Control System Based on Internet of Things Using Telegram Application. National Seminar for Students of the Faculty of Information Technology (SENAFTI) Jakarta-Indonesia, September.

Wahyusari, R., Teguh Adi Saputro, & Lastoni Wibowo. (2023). Design Of Prototype Control And Detection Of Street Lighting Damage Based On The Internet Of Things. Journal of Electronic Informatics and Engineering , 6(2). https://doi.org/10.36595/jire.v6i2.980

Yuliaminuddin, V., Krismes, & Bintoro, J. (2021). Prototype Of Control And Monitoring System For Water Tank Based On Internet Of Things. Autocracy: Journal of Automation, Control, and Industrial Applications, 7(1). https://doi.org/10.21009/autocracy.071.5

Yulisman, Y., Ikhsan, I., Febriani, A., & Melyanti, R. (2021). Implementation of Internet of Things (IoT) Light Control Using NodeMCU ESP8266 and Smartphone. Journal of Computer Science, 10(2). https://doi.org/10.33060/jik/2021/vol10.iss2.231