Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th October 2025. Vol. 5. No. 1; e-ISSN: 2808-4519

Developing an Android Application for Internet of Things (IoT) Based Light Control using Android Studio

Chandra Aprillian[1*], Arnes Sembiring[2], Milli Alfhi Syari[3]

[1,3] Teknik Informatika, STMIK Kaputama, Binjai, Indonesia
[2] Teknik Informatika, Universitas Medan Area, Medan, Indonesia
chandraaprilian98@gmail.com[1*], arnessembiring@staff.uma.ac.id[2], milli.fhisya@gmail.com[3]

Abstract

Technological advancements are driving the development of home automation systems, one of which is Internet of Things (IoT)-based light control. Popular applications such as Blynk are often used in these systems, but they have limitations in automation flexibility, interface design, and backend control that make further development less than optimal. This research aims to develop an Android application to control IoT-based lights by utilizing the Firebase Realtime Database as a more flexible and independent backend. The system is designed using Android Studio and integrated with the NodeMCU ESP32 and the FC-04 sound sensor to enable both online and offline light control. The research method uses prototyping, which includes creating an initial version of the application, testing, and iterative improvements based on feedback. Testing was conducted using the Arduino IDE for IoT devices, Android Studio for mobile applications, and Black Box Testing to verify system functionality. The test results show that the application is able to control lights in real-time over the internet and can still be controlled using sound sensors when offline. Available features include ON/OFF control, automatic schedule settings, and light status monitoring through Firebase. This system offers a cost-effective, flexible solution that does not rely on third-party platforms, so it has the potential to be further developed to support other smart home systems.

Keywords: *Android Studio; Firebase; Internet of Things; Light Control; NodeMCU ESP32*

1. Introduction

Current technological advances simplify various aspects of life, including the implementation of smart homes or home automation [1]. One widely used technology is the Internet of Things (IoT), which allows users to control home devices remotely via smartphones [2]. An example of this application is an IoT-based light control system that allows users to turn lights on or off without touching a switch [3].

One popular platform for connecting IoT devices to smartphones is Blynk. Its ease of use, intuitive visual interface, and integration with microcontrollers like the ESP32 make Blynk widely used in small-scale IoT development [4]. However, this platform has limitations, such as a lack of flexible automation features and UI design, limited backend control, and difficulties with advanced integration due to its closed ecosystem. This presents challenges for developers seeking to build more complex IoT systems.

Alternatively, the Firebase Realtime Database can be used as a cloud backend to store and manage IoT device data [5]. Firebase enables real-time data exchange between Android apps and IoT devices without the need for conventional servers. Furthermore, Firebase's integration with Android Studio simplifies the development of Kotlin- or Java-based applications for IoT systems [3].

Based on this background, this study aims to develop an Android application to control IoT-based lights using the Firebase Realtime Database. This application is expected to be a more flexible solution, easy to develop, and meet the functional requirements of IoT systems. Furthermore, this research can serve as a reference for developers who want to build IoT systems without relying on closed platforms like Blynk.

2. Literature Review

2.1 Android Application

Android is the most popular mobile operating system and is widely used on various devices worldwide [6]. In the context of the Internet of Things (IoT), Android applications serve as an interface for monitoring and controlling devices in real time via an internet connection. One such application is a home lighting control system that allows users to turn lights on or off wirelessly via Wi-Fi with an intuitive interface [7]. This system can be developed using the Flutter framework, the Dart programming language, and the Firebase Realtime Database to manage data and light control commands [8]. Thus, Android applications play a crucial role in IoT-based lighting control systems, providing users with the convenience and flexibility to efficiently manage home lighting.

2.2 Automatic Lights

Automatic lighting based on sound sensors is an innovative lighting system that utilizes sound waves to automatically turn lights on or off, making them more practical and energy-efficient [9]. The sound sensor can detect clapping or voice commands, then sends a signal to the microcontroller to turn the lights on or off without pressing a switch. Furthermore, the sensor's sensitivity can be adjusted to prevent the lights from turning on due to unwanted background noise. This sound threshold setting improves energy efficiency and user comfort because the lights only respond to the appropriate voice commands [10].

2.3 Internet of Things (IoT)

The Internet of Things (IoT) is a technology that connects physical devices through an internet network to collect, process, and transmit data in real time, thereby increasing efficiency in various fields such as smart homes, healthcare, and the manufacturing industry [11]. IoT aims to expand the benefits of internet connectivity for information exchange, remote control, and the integration of various devices [12]. Furthermore, IoT extends beyond the interconnection of physical devices, encompassing a broad ecosystem supported by various technologies [13].

As it develops, IoT plays a crucial role in the automation and control of household devices, including lighting, which can be controlled through an Android app. This technology allows users to control lighting remotely via smartphone, in line with the trend of integrating IoT and artificial intelligence (AI) to improve efficiency and personalize services.

2.4 Android Studio

Android Studio is an Integrated Development Environment (IDE) developed by Google in 2013 as an extension of Eclipse, with features such as a Gradle-based build system, layout editor, and Google Cloud Platform integration [14]. This IDE simplifies the development of Internet of Things (IoT)-based light control applications using the Java or Kotlin programming languages, with an XML-based user interface. Libraries like Retrofit and Volley enable IoT application data management to control lighting devices in real time over an internet connection [15].

Another advantage of Android Studio is its multi-platform debugging and testing capabilities, allowing developers to test applications without physical devices. This feature makes Android Studio a top choice for developing innovative and efficient IoT applications [16].

2.5 Firebase

Firebase is Google's integrated platform that brings together a variety of advanced features and services for apps, including a mobile backend, analytics, and growth features [17]. One key feature is the Firebase Realtime Database, a NoSQL database that enables real-time data storage and synchronization. This feature is ideal for Internet of Things (IoT)-based applications, where data from IoT devices can be sent, stored, and updated instantly.

The Firebase Realtime Database makes it easy for developers to manage IoT data by providing a serverless infrastructure, eliminating the need for developers to worry about managing servers. Furthermore, Firebase supports integration with various platforms, such as Android Studio, simplifying app development.

3. Analysis and Design

3.1 Research Methodology

This research uses the prototyping method, which develops the system iteratively based on testing feedback. The process begins with requirement analysis, interface design for the Android application, and programming the NodeMCU ESP32 with the FC-04 sound sensor integrated into Firebase Realtime Database as a real-time communication backend. After initial implementation, black box testing is conducted to validate functionalities, including online light control and offline control via the sound sensor. This approach ensures that the system meets real-world usage scenarios and is ready for further smart home development.

3.2 Overall Device Circuit

The entire tool assembly consists of many components designed to ensure that the tool functions according to its intended purpose.

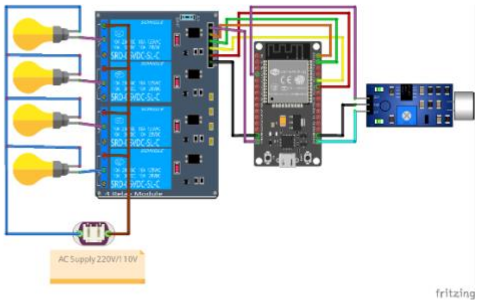

Fig. 1: Overall Device Circuit

Information :

1. AC Lamp: Functions as the main load that will be controlled to turn on or off using a relay, either automatically via a sound sensor or manually via an Android app.
2. 4-Channel Relay Module (SRD-05VDC-SL-C): Acts as an electronic switch that disconnects and connects AC current to the lamp based on commands from the NodeMCU ESP32.
3. NodeMCU ESP32: The main microcontroller that processes signals from the sound sensor, sends and receives data from Firebase, and controls the relay to turn the lamp on or off.
4. FC-04 Sound Sensor: Detects clapping or other sounds that trigger the lamp to turn on or off automatically.
5. Smartphone with IoT App: Serves as a user interface to control the lamp remotely via the Firebase Realtime Database.
6. 110/220V AC Power Supply: Power source to power the lamp, which is connected to the relay for automatic control.

3.3 Use case Diagram

This application is designed to allow users to control household devices such as lights using a smartphone easily and efficiently. The simple interface allows users to turn devices on and off directly, as well as set device operating times as needed. The use case illustrates user interaction with the system, where from the main menu users can check device status, change display modes, and exit the application. The ON/OFF control feature provides ALL ON and ALL OFF functions to control all devices simultaneously, while the schedule setting feature allows users to select times, save, and delete automatic schedules. All functions in this system are interconnected through include relationships, which indicate that these features are an essential part of the overall application system.

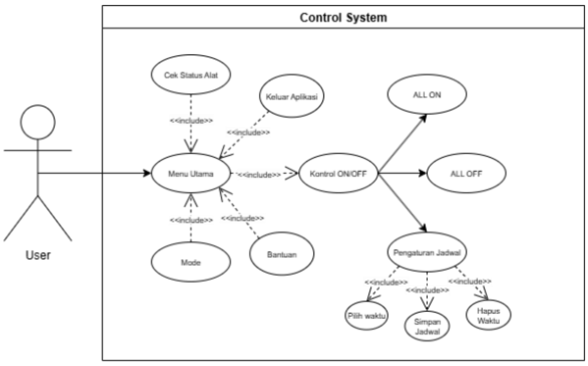

Fig. 2: Use case diagram

4. Results and Discussion

This section discusses the results of testing and implementing the developed IoT-based device control system, which allows users to control lights and other devices through an Android application. The system uses Firebase as a backend for real-time data communication between the application and the IoT device, in this case the NodeMCU ESP32. Through the Android application, users can turn the lights on or off, set the ON/OFF time, and monitor the device status. The application is designed with a user-friendly interface, providing manual control buttons and automatic timing features according to a specified schedule. The test results show that the system functions well, where the device control process runs smoothly, accurately, and responsive to user commands.

4.1 Alpha Testing

Alpha testing was conducted using the Black Box Testing method to ensure that the developed system meets user needs and expectations. This method focuses on testing software functionality from the user's perspective without examining the internal structure or program code. Thus, testing assesses whether each feature performs as expected based on specifications. The results of testing using the Black Box method on this IoT-based device control system are presented in the table below.

Table 1. Black box testing

No	Test Scenario	Test Case	Expected results	Result	Status
1	Splash Screen Display	Open Application	When the application is run, the user will see a splash screen first before entering the home screen.	Appropriate	Valid
2.	Verify the functionality of the "Dark Mode" button	Toggle switch "Dark Mode"	The app goes to dark mode (background goes dark, text goes white).	Appropriate	Valid
3.	Check the "Check Device Status" button	Check the "Check Device Status" button	The device status appears as "ONLINE" or "OFFLINE".	Appropriate	Valid
4.	Verify the "ALL ON" button	Tap the "ALL ON" button	All Relays should be on.	Appropriate	Valid
5.	Verify the "ALL OFF" button	Tap the "ALL OFF" button	All Relays should be off.	Appropriate	Valid
6.	Verify the “SET” button on all devices.	Tap the “SET” button	Displays the menu to set the device ON/OFF schedule.	Appropriate	Valid
7.	Verify individual device controls	“Toggle Switch” for any device (Headlights, etc.)	Users should be able to set ON/OFF for each individual device.	Appropriate	Valid
8.	Test the "ON Time" function	Set the time for "ON Time" to 6:00 AM	The time setting must be saved and reflected in the “ON Time” label.	Appropriate	Valid
9.	Test the "OFF Time" function	Set the time for "OFF Time" to 7:00 AM	The time setting must be saved and reflected in the “OFF Time” label.	Appropriate	Valid
10.	Test the “Dark Mode” toggle during device interaction	Toggle "Dark Mode" and device control (Lights)	Apps must maintain “Dark Mode” even when interacting with the device.	Appropriate	Valid
11.	Verify the “HELP” button	Tap the “HELP” button	Displays the Help menu for using the application.	Appropriate	Valid
12.	Verify the “BACK” button	Tap the “BACK” button	Display/Return to previous menu	Appropriate	Valid
13.	Verify the "EXIT" button	Tap the “EXIT” button	Exit the application	Appropriate	Valid

4.2 Home Interface Display Results

The app's interface is designed to be simple with high-contrast colors, such as white text on a black background and bold purple buttons, to improve readability. Basic features like dark mode and the "Headlight" device controls work well. However, it is recommended to add tooltips or status labels to clarify device information, as well as reorder the "SET" buttons so they are not too close together. Additional testing is needed to ensure optimal display on standard-sized screens.

Fig. 3: Home Interface Display Result

4.3 Schedule setting interface display results

The schedule settings interface uses a dark mode consistent with the homepage, with a dark background and bright buttons for clear contrast. Users can select the ON/OFF time using the AM/PM selector, save the settings with the green "Save" button, or delete them with the red "Clear Time" button. This simple and intuitive design promotes ease of use, especially in low-light conditions.

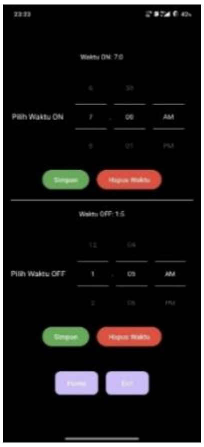

Fig. 4: Schedule setting interface display results

4.4 Sound Sensor Test Results

The sound sensor can be used when the device is not connected to the internet. The sound sensor works by detecting the sound of clapping, as described in the table below.

Table 2. The logic of applause

Number of applause	Results
1x Clap	Relay 1 ON/OFF
2x Clap	Relay 2 ON/OFF
3x Clap	Relay 3 ON/OFF
4x Clap	Relay 4 ON/OFF
5x Clap	All Relay OFF

Fig. 5: Sound Sensor Test Results

4.5 Overall Device Test Results

After the hardware device is programmed into the microcontroller and has been executed using the downloader, the program automatically enters the microcontroller.

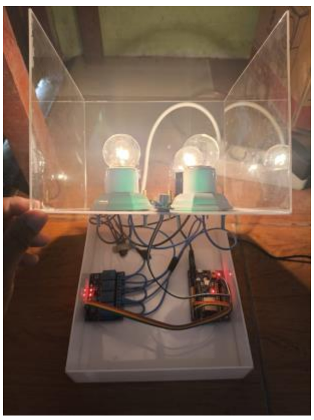

Fig. 6: Overall Device Test Results

5. Conclusion

Based on the research results, it can be concluded that the developed Android application is able to control Internet of Things (IoT)-based lights effectively by utilizing the Firebase Realtime Database as a real-time data communication medium. The development process includes designing a simple and easy-to-operate user interface, coding the Android application in Android Studio, and integration with a NodeMCU ESP32-based IoT device connected to a relay module. The test results show that the application is able to turn the lights on and off in real-time, respond to user commands quickly, and send control data to the device without significant obstacles, both via the internet network and locally using a voice sensor. This capability allows users to control the lights remotely anytime and anywhere as long as an internet connection is available, thus providing convenience, flexibility, and energy efficiency. Overall, this research proves that the integration of Android applications and Firebase-based IoT devices can be a practical smart home solution, and has the potential to be further developed by adding various other IoT devices to support more comprehensive home automation.

References

[1] M. Dg Patabo, S. Sawidin, S. Tuwongkesong, T. M. Kereh, A. P. Waroh, and J. J. Lapon, "Rancang Bangun Sistem Kontrol Lampu Ruangan Berbasis Iot Arduino Cloud Design and building of a room light control System based on iot arduino cloud," *Jambura Journal of Electrical and Electronics Engineering*, vol. 227, 2024.

[2] T. Setiawan and A. N. Anwar, "IMPLEMENTASI PROTOTYPE INTERNET OF THINGS PADA KENDALI LAMPU RUMAH MENGGUNAKAN BOT TELEGRAM DAN NODEMCU ESP8266," *JORAPI : Journal of Research and Publication Innovation*, vol. 1, no. 3, 2023.

[3] T. Sulistyorini, N. Sofi, and E. Sova, "PEMANFAATAN NODEMCU ESP8266 BERBASIS ANDROID (BLYNK) SEBAGAI ALAT ALAT MEMATIKAN DAN MENGHIDUPKAN LAMPU," *JUIT*, vol. 1, no. 3, 2022.

[4] A. Herlina, M. Irfan Syahbana, M. Adi Gunawan, and M. Miftahul Rizqi, "Sistem Kendali Lampu Berbasis Iot Menggunakan Aplikasi Blynk 2.0 Dengan Modul Nodemcu Esp8266," 2022. [Online]. Available: http://jurnal.bsi.ac.id/index.php/insantek

[5] Raken Putra Athallah, Astri Novianty, and Faisal Candrasyah Hasibuan, "Pengembangan Backend Untuk Aplikasi Monitoring dan Kontrol Air Purifier Berbasis IoT dan Sistem Cerdas," Dec. 2024.

[6] N. Rismawati, *DASAR-DASAR ANDROID STUDIO DAN MEMBUAT APLIKASI MOBILE SEDERHANA Penulis: Ika Parma Dewi, Lativa Mursyida, Agariadne Dwinggo Samala*. 2021. [Online]. Available: www.penerbitwidina.com

[7] L. Lestari, S. Syahwin, and T. Haramaini, "Pemanfaatan Teknologi Internet of Things untuk Kendali Lampu menggunakan Android," *Blend Sains Jurnal Teknik*, vol. 2, no. 2, pp. 112–124, Aug. 2023, doi: 10.56211/blendsains.v2i2.312.

[8] Z. T. A. Salasa and R. Achmad, "RANCANG BANGUN SISTEM KONTROL LAMPU RUMAH PADA TIAP RUANGAN BERBASIS ANDROID," vol. 12, no. 1, pp. 32–37, 2024.

[9] C. P. Subrata, "PERANCANGAN SISTEM LAMPU OTOMATIS MENGGUNAKAN SENSOR SUARA," *Jurnal Deli Sains Informatika*, vol. 1, no. 2, 2022.

[10] A. O. Rahmawati, U. Nurjanah, C. M. Sabilla, and F. I. Komputer, "Perancangan Sistem Kendali Otomatis Lampu Menggunakan Sensor Suara Berbasis Arduino," Apr. 2024. [Online]. Available: https://journal.mediapublikasi.id/index.php/logic

[11] H. Taherdoost, "Security and Internet of Things: Benefits, Challenges, and Future Perspectives," *Electronics (Basel)*, vol. 12, no. 8, p. 1901, Apr. 2023, doi: 10.3390/electronics12081901.

[12] S. Sos. ,M. Si. Ade Putra Ode Amane *et al.*, *PEMANFAATAN DAN PENERAPAN INTERNET OF THINGS (IOT) DI BERBAGAI BIDANG*. PT. Sonpedia Publishing Indonesia, 2023. [Online]. Available: www.sonpedia.com

[13] M. Husni, S. Figo, A. F. Fadhilah, and A. P. Y. Tampubolon, *Teknologi IoT*. Penerbit Andi, 2025. [Online]. Available: https://books.google.co.id/books?id=Q_xlEQAAQBAJ

[14] S. Sibuea, M. Ikhsan Saputro, A. Annan, and Y. Bowo Widodo, "JURNAL INFORMATIKA DAN TEKNOLOGI KOMPUTER APLIKASI MOBILE COLLECTION BERBASIS ANDROID PADA PT. SUZUKI FINANCE INDONESIA," *Maret*, vol. 2, no. 1, 2022.

[15] J. J. Sidauruk, "Rancang Bangun Sistem Lampu Otomatis Berbasis Internet of Things dengan Aplikasi Raspberry Pi," *Jurnal Pendidikan Sains dan Komputer*, vol. 2, no. 02, pp. 345–249, Oct. 2022, doi: 10.47709/jpsk.v2i02.1744.

[16] Yohanis Ndapa Deda, Hermina Disnawati, Otaget Daniel, and Rooselyna Ekawati, "Development of Android-Based Learning Media for High School Students in Indonesia: A Systematic Review of Literature," *Jurnal Iqra' : Kajian Ilmu Pendidikan*, vol. 8, no. 1, pp. 402–417, Sep. 2023, doi: 10.25217/ji.v8i1.3483.

[17] R. F. Purnomo, O. W. Purbo, and R. Z. A. Aziz, *Firebase: Membangun Aplikasi Berbasis Android*. Andi Offset, 2021. [Online]. Available: https://books.google.co.id/books?id=iosaEAAAQBAJ