# Comparative Analysis of AES (Advanced Encryption Standard) and Blowfish Algorithms in Camera Module IoT Applications

**Nurul Hidayati[1], Ikrimatuz Zulaykhah[2], Waluyo Waluyo[3], Adzikirani Adzikirani[4]**

[2,3,4] Digital Telecommunication Network Study Program, Department of Electrical Engineering, State Polytechnic of Malang, 65141, Indonesia.

[1] Telecommunication Engineering Study Program, Department of Electrical Engineering, State Polytechnic of Malang, 65141, Indonesia.

[1]ikrimatuz07@gmail.com, [2]nurulhid8@polinema.ac.id, [3]waluyo@polinema.ac.id, [4]adzikirani@polinema.ac.id

*Abstract*— **The development of technology is currently becoming very advanced and sophisticated which results in many data leaks by irresponsible individuals, so data security is mandatory. Data that is easily hacked is usually images sent through social media. This research aims to secure image transmission data using Telegram BOT. The selection of social media is because Telegram does not yet have encryption for data transmission security. The methods used are AES (Advanced Encryption Standard) and Blowfish because AES is the strongest algorithm today while Blowfish was once named the best and patent-free algorithm. So it is appropriate to conduct a comparative analysis on AES and Blowfish. Duration testing carried out using unequal sizes has an encryption process value of 100-200 ms consistently. For the duration description time of the smallest and largest image has the same value of 300 ms. For the same image size, the better the AES algorithm, the longer the duration of time while the Blowfish algorithm has a faster time than AES. As for the description process Blowfish Algorithm has the longest duration and AES 192 has the fastest duration.**

***Keywords*— AES Algorithm, Blowfish, Camera Module, Encryption, IoT.**

## I. INTRODUCTION

The development of increasingly advanced and sophisticated technology is an advantage for industry and the general public to be utilized in everyday life [1]. Where industry or society can monitor conditions in real time without having to go to the place they want to monitor using a cloud server, owners can view data anytime and anywhere using a computer or even a smartphone. This is certainly inseparable from data security and protection because of the many cases of leaked personal data transmission that are vulnerable to being infiltrated by hackers, so that data transmission security is currently mandatory [2]. All of these conditions make cybersecurity a major challenge in IoT devices with demands for confidentiality, data integrity, authentication & authorization, availability, privacy standards & regulations, and periodic system updates [3]. In this scenario, cryptography can be an effective step to ensure the confidentiality, integrity and authentication & authorization of data that passes through IoT devices [4]. Cryptography can also be a solution to secure data stored or passing through a network [5].

Internet of Things(IoT) is a network that connects various objects that have an identity and IP address, so that they can communicate with each other and exchange information about themselves and the environment they sense [6]. In IoT there are layers that have a role in maintaining data confidentiality. This study also applies one of the IoT layers, namely the application layer, where the information received can later be used as intelligent technology. The application layer provides an information interpretation process [7] one of which uses a camera module that will later take pictures to encrypt the data so that its security is maintained. Currently, not all encryption can guarantee good data security, so it requires a good cryptographic method to maintain data confidentiality [8].

In this study, sending images/data securely requires encryption methods, namely AES (Advanced Encryption Standard) and Blowfish to conduct a comparative analysis. Both methods can encrypt and decrypt information using symmetric ciphertext blocks [9]. The reason for using the AES method is because it is the latest cryptographic algorithm standard published by NIST (National Institute Standard Technology) and was chosen because the algorithm uses 128, 192, and 256-bit cryptographic keys to encrypt and decrypt data in 128-bit blocks which makes it more secure from external attacks [10]. Then the reason for using the Blowfish Algorithm is because the cryptographic method is fast, compact, simple, has varying levels of security and is not patented so it is free for various uses [11]. These methods will later be compared and the best results will be used in this study. This study uses Telegram. The advantages of telegram are that it is easy to use, cloud-based, maintains the quality of the files sent, has strong encryption, is free, can reach 200,000 users in a group, is synchronized, can send documents up to 2GB in size, and can build telegram API tools [12]. There is a feature in the form of an API (Application Programming Interface) that allows developers to create bots and integrate various services into the platform, expand application functionality according to user needs [13].

In the previous study entitled "Comparison of Advanced Encryption Standard (AES) and Blowfish Cryptographic Algorithm Performance on Text on the Android Platform" it can be seen that the AES and Blowfish methods have almost the same algorithm speed performance on Android devices with ARM processors [14]. This is one of the reasons why this study uses this method and whether the results of this study are almost the same or different.

The purpose of this study is to conduct a Comparative Analysis of the Use of AES (Advanced Encryption Standard) and Blowfish Algorithms in IoT Camera Module Applications. The way the study works is when entering a command to take data/images from the telegram application, the camera module will take the image. Here, later using a telegram bot to give the command to take images for the camera module. Before the image is sent, it will be encrypted first by the AES and Blowfish Algorithms on PHP. Telegram has a fairly complete and developed Bot API, making it possible to be used as a smart bot that can respond to messages from the public [15]. To display the image that has been sent by the camera module, the recipient must type the telegram bot that has been sent to describe the encrypted image.

The results of the analysis will later be compared to determine which is the best algorithm to use in the IoT Camera Module Application

## II. METHOD

In the research that will be conducted, a method or stage is needed that is useful as a guideline in the implementation to complete this research in order to achieve the desired goals. The following are the stages of research that will be conducted; as shown Fig. 1.

This research follows a series of systematic stages, beginning with a comprehensive literature study to gather basic knowledge and relevant information from various sources such as journals, e-books, and materials related to the system to be built. Following that, application design creation is carried out, which includes making UML, flowcharts, and user interface (UI) design to detail the features to be implemented. The next stage is system development or coding, where the designed features are transformed into computer program code using the PHP language. Then, testing of the coded program is conducted, using supporting tools such as API (Webhook), to ensure the system functions according to the established goals and needs. Results analysis is carried out by collecting data through research, interviews, or literature study to produce a user requirements document that defines user desires for the system. Finally, after all stages are completed, a report is created that summarizes the entire research process and results, documenting the application design and implementation comprehensively.

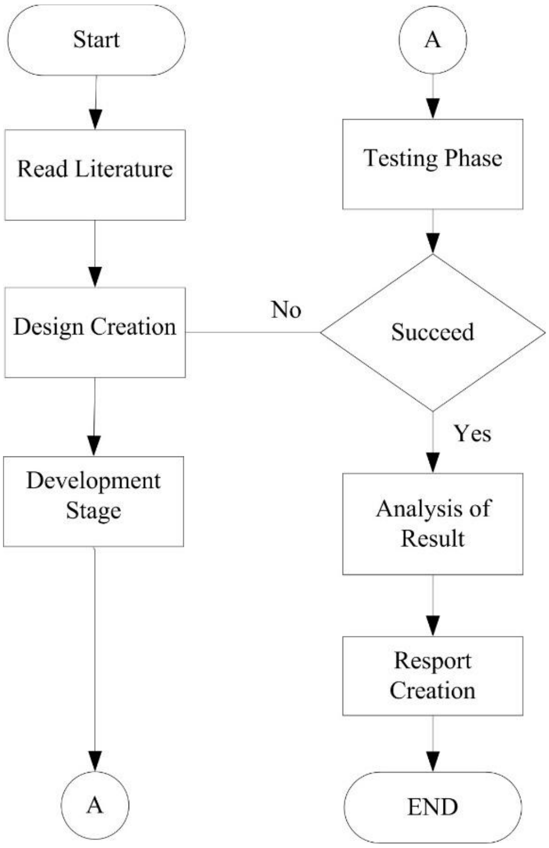


Figure 1. The stages of research that will be conducted

The process begins when the user opens the Telegram application on their smartphone and enters a command to retrieve data or images. The data or images are then captured through the camera module (ESP32 CAM). Before being sent, the data or images are encrypted via a webhook using the AES and Blowfish algorithms. These algorithms transform the original data (plaintext) into unrecognizable ciphertext through UDP and TCP/IP protocols, maintaining the confidentiality of the data or images. Finally, the image or data recipient must enter a key or BOT code in the Telegram chat column to display the encrypted data or images, as shown in Fig. 2.

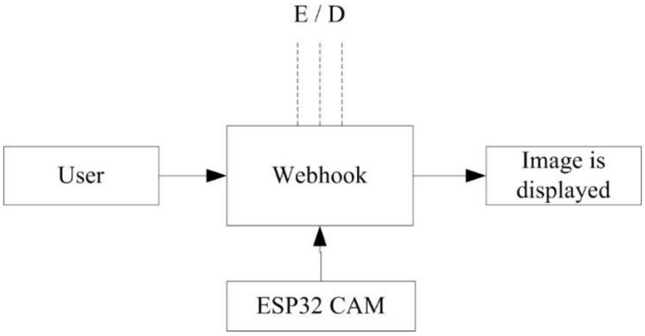


Figure 2. System Block Diagram

Fig. 3 illustrates the workflow of a system designed to capture and secure images through the Telegram application.

The process begins when a Telegram user sends a command to capture an image. This command is received by the ESP32 CAM, a camera module that then captures the image. The resulting image is then entered into a database via a webhook. This webhook enables real-time communication between Telegram and the database, ensuring the image is processed promptly. The next step is to encrypt the image using the AES and Blowfish algorithms. This encryption process transforms the image into a format that cannot be read without the correct decryption key, maintaining the confidentiality of the image during transmission. The encrypted image is then sent back to Telegram.

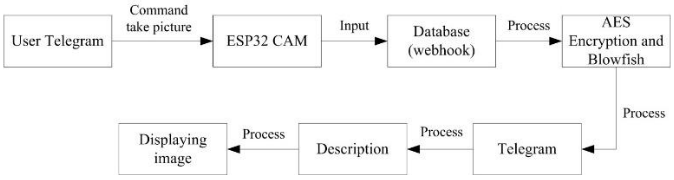
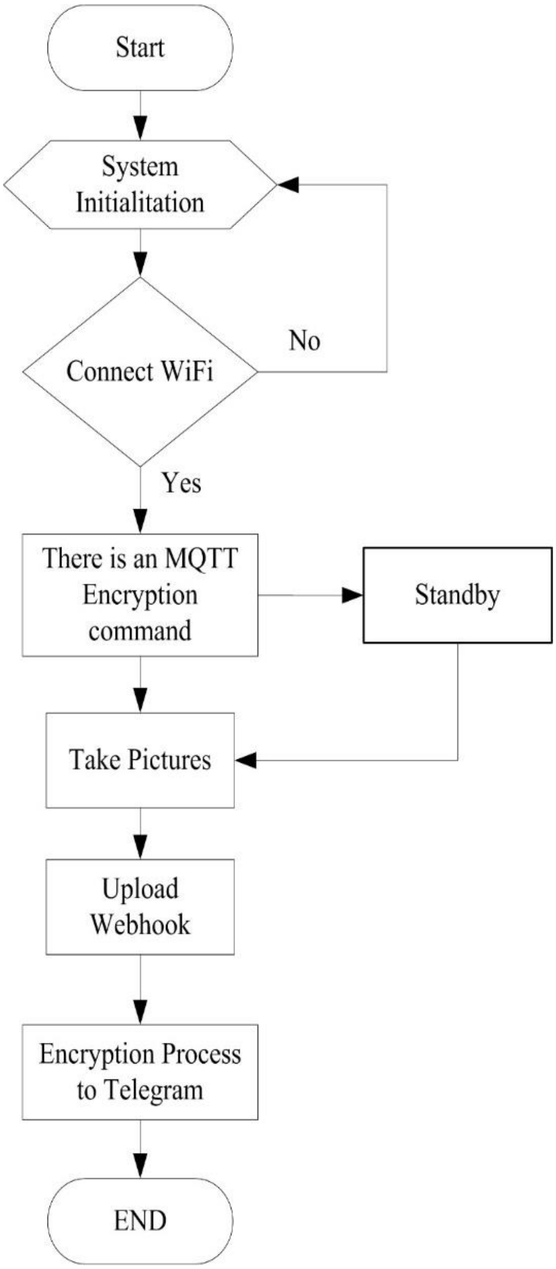


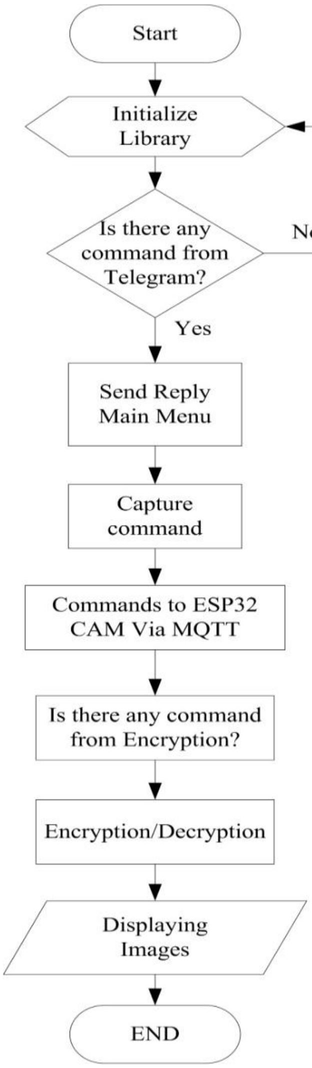
Figure 3. Flowchart

To view the image, the user must perform a description or decryption. This process requires the appropriate key to restore the image to its original format. Once decryption is complete, the image is processed and finally displayed to the user in the Telegram application. Overall, this system is designed to capture images remotely using Telegram, secure those images through encryption, and allow authorized users to view them after decryption. The use of a webhook enables seamless integration between Telegram, the database, and the encryption/decryption processes, creating an efficient and secure system.

In Fig. 4 Flowchart ESP32 Cam (a) will be initialized the system which is then connected to WiFi. If not connected then it will re-initialize if connected then in the Telegram APK there will be an encryption command on MQTT, the process takes place in the standby box. After that there will be a picture taken. The image will be uploaded to Webhook then the encryption process will be carried out which will later be sent to Telegram. In Fig. 3 Flowchart Webhook (b) there is a flow diagram that starts from initializing the library then getting a command from Telegram. If there is no command then it will re-initialize in the library. If there is a command then there will be a reply from the main menu. Then there is a command to take a picture, after that there will be a command to ESP32 CAM via MQTT. Next there will be a command from encryption/description. After the process is complete the image will be displayed



(a)



(b)

Fig. 4 ESP 32 CAM Flowchart (a) Webhook Flowchart (b)

## III. RESULTS AND DISCUSSION

### *A. Speed Test Using AES 128*

TABLE I
TESTING AES 128 IN DIFFERENT FILES SIZES

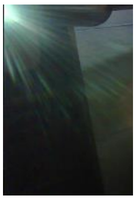
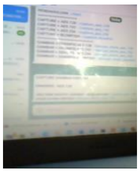
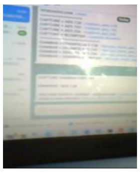
| No | Original Image | Image Description | Encryption Time (ms) | Description Time (ms) | Size(KB) |
|---|---|---|---|---|---|
| 1. | | | 200 ms | 300 ms | 17.9 KB |
| 2. | |  | 200 ms | 400 ms | 21.2 KB |
| 3. |  |  | 200 ms | 300 ms | 25.8 KB |
| 4. | | | 200 ms | 400 ms | 28.2 KB |
| 5. | | | 200 ms | 300 ms | 32.5 KB |

Table I shows that the average encryption time process has the same speed, which is 200 ms. While for the description process, the duration of the time is uncertain, such as the size of 17.9 KB showing the same time as the size of 32.5 KB, which is 300 ms. In fact, the table can be seen that the size of 17.9 KB is the smallest size and 32.5 KB is the largest size of the trial of this method.

### *B. Speed Test Using AES 192*

TABLE II
TESTING AES 192 IN DIFFERENT FILES SIZES

| No | Original Image | Image Description | Encryption Time (ms) | Description Length (ms) | Size(KB) |
|---|---|---|---|---|---|
| 1. | | | 100 ms | 400 ms | 24.7 KB |

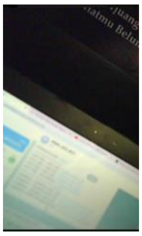
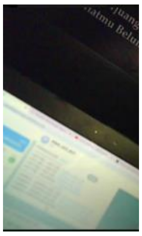
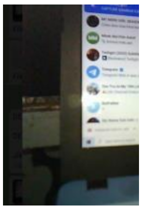
| No | Original Image | Image Description | Encryption Time (ms) | Description Length (ms) | Size(KB) |
|---|---|---|---|---|---|
| 3. |  |  | 200 ms | 400 ms | 28.8 KB |
| 4. |  | | 200 ms | 300 ms | 35.7 KB |

Table II shows the average time of the thesis process is 180 ms. While the description process shows different and inconsistent times. Like the size of 19.5 KB which is the smallest size, the value is the same as 38.5 KB which is the largest size with a duration value of 300 ms. Different from the size of 24.7 KB and 28.8 KB which have a duration of 400 ms.

### C. Speed Test Using AES 256

TABLE III
TESTING AES 256 IN DIFFERENT FILES SIZES

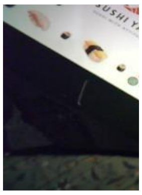
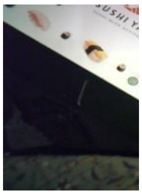
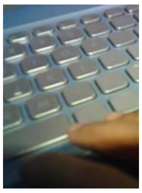
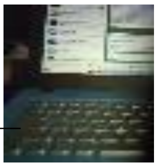
| No. | Original Image | Image Description | Encryption Time (ms) | Description Length (ms) | Size(KB) |
|---|---|---|---|---|---|
| 1. | | | 100 ms | 300 ms | 19.5 KB |
| 2. |  |  | 200 ms | 300 ms | 30.8 KB |
| 3. | |  | 100 ms | 300 ms | 34.5 KB |

Table III shows that the encryption process has an average value of 160 ms. While the description process shows a consistent and stable time of 300 ms.

### D. Speed Test Using Blowfish

TABLE IV
TESTING BLOWFISH IN DIFFERENT FILES SIZES

| No. | Original Image | Image Description | Encryption Time (ms) | Description Length (ms) | Size(KB) |
|---|---|---|---|---|---|
| 1. | | | 100 ms | 300 ms | 25.9 KB |
| 2. | |  | 200 ms | 300 ms | 30.4 KB |
| 3. | | | 100 ms | 300 ms | 32.1 KB |
| 4. | | | 200 ms | 300 ms | 37.8 KB |

| No. | Original Image | Image Description | Encryption Time (ms) | Description Length (ms) | Size(KB) |
|---|---|---|---|---|---|
| 5. | | | 200 ms | 300 ms | 41.1 KB |

Table IV shows that the encryption process has an average value of 160 ms. While the description process has a consistent and stable duration of 300 ms. Which value is the same as the AES 256 process.

### *E. Speed Test Using AES 128, 192, 256 and Blowfish with Same Files Size*

TABLE V
TESTING AES 128, 192, 256, AND BLOWFISH IN SAME FILES SIZES

| No. | Method | Original Image | Image Description | Encryption Time (ms) | Description Length (ms) | Size (KB) |
|---|---|---|---|---|---|---|
| 1. | AES 128 | | | 412 | 627.55 | 75.7 |
| 2. | AES 192 | | | 425 | 599.1 | 75.7 |
| 3. | AES 256 | | | 447 | 612.19 | 75.7 |
| 4. | Blowfish | | | 333 | 2090.1 | 75.7 |

Table V shows the results of the duration analysis of the AES 128, 192, 256 and Blowfish algorithms with the same image size of 75.7 KB. In AES 128, it shows that the encryption process in seconds is faster than AES 192 and 256, while in Blowfish the encryption time is faster than AES, which is 3330 ms. While in the long duration, the description of the Blowfish algorithm is longer than AES, which is 2090.1 ms.

TABLE VI
TESTING AES 128, 192, 256, AND BLOWFISH IN SAME FILES SIZES

| No. | Method | Original Image | Image Description | Encryption Time (ms) | Description Length (ms) | Size (KB) |
|---|---|---|---|---|---|---|
| 1. | AES 128 | [-] Searching IP Location Finder: certifiedhack<br>[+] Url: certifiedhacker.com<br>[+] IP: 162.241.216.11<br>[+] Status: success<br>[+] Region: Utah<br>[+] Country: United States<br>[+] City: Salt Lake City<br>[+] ISP: Unified Layer<br>[+] Lat & Lon: 40.7608 & -111.891<br>[+] Zipcode: 84138<br>[+] TimeZone: America/Denver<br>[+] AS: AS46606 Unified Layer | [-] Searching IP Location Finder: certifi<br>[+] Url: certifiedhacker.com<br>[+] IP: 162.241.216.11<br>[+] Status: success<br>[+] Region: Utah<br>[+] Country: United States<br>[+] City: Salt Lake City<br>[+] ISP: Unified Layer<br>[+] Lat & Lon: 40.7608 & -111.891<br>[+] Zipcode: 84138<br>[+] TimeZone: America/Denver<br>[+] AS: AS46606 Unified Layer | 446 | 620.66 | 112 |
| 2. | AES 192 | | | 472 | 615.51 | 112 |
| 3. | AES 256 | | | 489 | 658.29 | 112 |
| 4. | Blowfish | | | 331 | 3080 | 112 |

Table VI shows the results of the analysis with a size of 112.6 KB which is larger than the previous image table. In this table, the duration of the AES Algorithm from 128, 192 to 256 will take longer to perform the encryption process, while in the Blowfish Algorithm the encryption duration is faster than the three AES Algorithms, namely 3310 ms. While the old description process to display the Blowfish Algorithm image is 3080 ms longer than the three AES Algorithms.

## IV. CONCLUSION

The results obtained from this study are the duration of the encryption process carried out on the four methods with different sizes using manual calculations have consistent values. While in the description process the duration of time has inconsistent values. AES 128, 192, 256, and Blowfish Algorithm have the fastest duration to carry out the encryption process. Unlike the old description process, the fastest duration is AES 128 while Blowfish has the longest duration. In this study, image sending can be kept confidential because the larger the bit used, the better the security. In addition, this study is only a single user where if other people want to get the image/data, they must have a Telegram BOT owned by the user.

## REFERENCES

[1] Smith, J., & Johnson, A. (2023). The Impact of Technology on Industrial Growth. Journal of Industrial Technology, 15(2).

[2] Kamaljit Singh, Et Al. (2023). Data Security and Privacy in The Era Of Big Data: Challenges And Solutions. Journal of Big Data, 10(1), 1-25.

[3] A. Banafa, "Three Key Challenges Facing Iot," Ieee Internet Ofthings, March 2017. [On Line]. Available: Https://Iot.Ieee.Org/Newsletter/March-2017/Three-Major-Challenges-Facing-Iot.Html/. [Accessed Wednesday May 2023].

[4] B.J. Mohd And T. Hayajneh, "Lightweight Block Ciphers for Iot: Energy Optimization and Survival Techniques," Ieee Access, Vol. 6, P. 35 966–35 978, 2018.

[5] Va Thakor, Ma Razzaque and Mr Khandaker, "Lightweight Cryptography for Iot: A State-Of-The-Art," Researchgate, No. 19, P. 2, 2020.

[6] F. Adani And S. Salsabil, "Internet of Things: History Of Technology And Its Application," Technology Issues, Vol. 14 No 2, No. 8, P. 1, 2019.

[7] Najib, W., Sulistyo, S., & Widyawan. (2020). Review Of Threats And Security Solutions In Internet Of Things Technology. National Journal Of Electrical Engineering And Information Technology, 9(4), 1-2.

[8] M. Conti, A. Gangwal, And S. Ruj, "The Evolution of Cryptographic Attacks: A Survey," Acm Computing Surveys, Vol. 54, No. 6, Article 128, 2021. Doi: 10.1145/3460418.

[9] Ravi Kumar, Et Al. (2023). Performance Analysis of Aes And Blowfish Algorithms for Image Encryption. Multimedia Tools and Applications, 82(10), 15678-15690.

[10] Hidayatulloh, Nw, Tahir, M., Amalia, H., & Basyar, Na (2023). Getting To Know Advance Encryption Standard (Aes) As A Cryptographic Algorithm In Securing Data. Digital Transformation Technology, 1-3.

[11] S. Sitinjak, Y. Fauziah And J., "File Cryptography Application Using Blowfish Algorithm," National Informatics Seminar 2010, No. 9, Pp. C-79, 2010.

[12] Erlina. (2022). Analysis Of The Use Of Telegram As A Media For Learning Arabic In The Industrial Era 4.0. Shaut Al-'Arabiyah.

[13] Widya, Ma, & Airlangga, P. (2020). Development Of Telegram Bot Engine Using Webhook Method In Order To Improve E-Government Service Time. Journal Of Science And Technology, 14-18.

[14] Mt Rahman, A. Pinandito And Es Pramukantoro, "Comparison of Advanced Encryption Standard (Aes) And Blowfish Cryptographic Algorithm Performance On Text On Android Platform," Journal Of Information Technology And Computer Science Development, Vol. 1, No. 9, P. 1559, 2017.

[15] Wali, A., Sulistyanto, A., & Defisa, T. (2022). Development of A Database Server Filesystem Monitoring System With Alerts Using Telegram Bots. Jayakarta Informatics Management Journal, 292-296.