

Video Steganography Encryption on Cloud Storage for Securing Digital Image

Chrisna Joshua Sergio Prasetyo^{a1}, I Putu Gede Hendra Suputra^{a2}, Luh Arida Ayu Rahning Putri^{a3}, I Made Widiartha^{a4}, I Ketut Gede Suhartana^{a5}, Anak Agung Istri Ngurah Eka Karyawati^{a6}

^aInformatics Department, Faculty of Mathematics and Natural Sciences, University of Udayana
South Kuta, Badung, Bali, Indonesia

¹sergio.febraro@gmail.com

²hendra.suputra@unud.ac.id

³rahningputri@unud.ac.id

⁴madewidiartha@unud.ac.id

⁵ikg.suhartana@unud.ac.id

⁶eka.karyawati@unud.ac.id

Abstract

Cloud storage is a data storage service in cloud computing that allows stored data to be shared and accessed via the internet. Cloud storage is usually used to store personal data such as files, photos, or videos with so that these data can be accessed anywhere via the internet without the need to use physical storage media. However, cases of data leaks in cloud storage still occur which causes personal data stored in cloud storage to be accessed by other people who do not have access.

The Client-Side Steganography Encryption on Cloud Storage Application was developed using the Modified Least Significant Bit (LSB) method and Advanced Encryption Standard (AES) algorithm. This desktop-based application was developed to protect personal data of digital images embedded in a video so that unauthorized parties cannot view the data. This application is developed to be a data security solution on cloud storage to prevent theft of personal data by non-existent parties.

From the test results, the developed application can receive input, process the input, and produce the desired output. The image from the extraction process from video also does not change at all in terms of visual or visible. The results obtained from this test is the PSNR value with an average of 36.395 dB. Good PSNR value is above 30 dB and indicates that the quality of the extracted image is good and also indicates that the developed application can protect digital images embedded in videos.

Keywords: Steganography, Cryptography, Digital Image, Video, Cloud Storage

1. Introduction

Cloud computing is one of the technologies that is growing rapidly. One of the *cloud computing* services is *cloud storage*. Cloud storage is a data storage service that commonly used to store personal files such as images, videos, documents, and any other files.

Most of the cloud storage services is already use their own security protocol to maintain their data in it. But there are several cases of data breaching in *cloud storage* which made the data stored in *cloud storage* can be accessed by unauthorized person. Gmail, Google's email service, tops the list for phishing (27.8%) and keylogger (29.8%) cases [1].

This Google's email service is synchronized with their cloud storage, Google Drive. With many cases of data breaching [1], so an additional security is needed to secure data that stored in Google Drive. This data breaching risk can be prevented by using steganography and cryptography techniques.

Least Significant Bit (LSB) Modification method provides security protection by hiding an encrypted message into a container media. Least Significant Bit Modification method provide good protection if combined with Advanced Encryption Standard (AES) algorithm [2].

Advanced Encryption Standard (AES) algorithm turns the message into unreadable files and can be decrypted using the same key that used in the encryption process. This encryption-decryption processing time depends on the size of the data [3].

Based on the problem, the authors intend to develop Video Steganography Encryption on Cloud Storage application using Least Significant Bit Modification Method combined with Advanced Encryption Standard Algorithm encryption. This desktop-based application developed with the aim of protecting personal data that stored in cloud storage. This application is developed to be a data security solution on cloud storage to prevent data breaching by unauthorized parties.

2. Research Methods

2.1 Advanced Encryption Standard (AES) Algorithm

In general, cryptography can be interpreted as the science and art of encryption which aims to maintain the security and confidentiality of data. Cryptography supports the needs of two aspects of information security, secrecy (protection of the confidentiality of information data) and authenticity (protection against counterfeiting and unwanted alteration of information). The process of scrambling the message is called encryption and when tidying the scrambled message, it is called decryption. The initial message that has not been scrambled or that has been tidied up is called plaintext, while messages that have been scrambled are called ciphertext [4].

Advanced Encryption Standard (AES) is a symmetric ciphertext block that can encrypt and decrypt information. This algorithm works using cryptography key 128, 192, or 256 bits. This different type of keys will affect total number of rounds used in this algorithm [5].

There are four transformations in encryption using AES algorithm [3]. These transformations are SubBytes (change every byte state with the byte in the S-Box table), ShiftRows (shifting bit process, where the leftmost bit will be moved to the rightmost bit (bit rotation)), MixColumns (operate every element that is in a column in state), and AddRoundKey ((XOR between states with RoundKey). The four transformations will be repeated in 10, 12, or 14 rounds, depending on the type of key used (128, 192, and 256 bits). The encryption flow for this application can be seen on Figure 1.

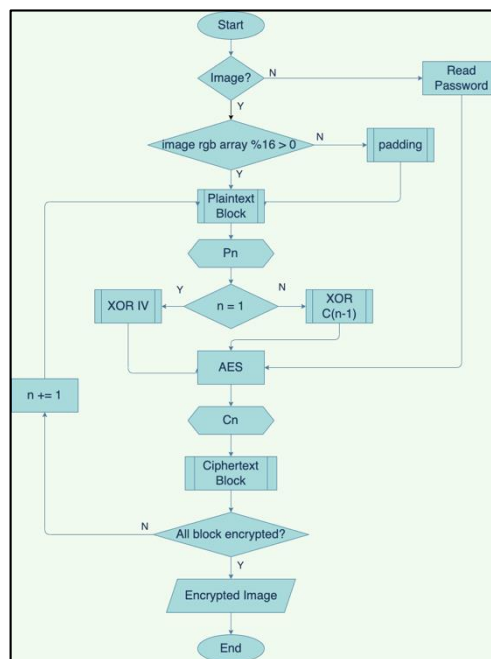


Figure 1. AES Encryption Flowchart

Decryption using AES algorithm used the inverse transformation of the AES Encryption. These transformations are InvSubBytes (change every byte state with the byte in the Inverse S-Box table), InvShiftRows (shifting bit process, where the rightmost bit will be moved to the leftmost bit (bit rotation)), InvMixColumns (each column in the state is multiplied by the multiplication matrix in AES). AddRoundKey is a self-inverse transformation on condition that uses the same key. The decryption flow for this application can be seen on Figure 2.

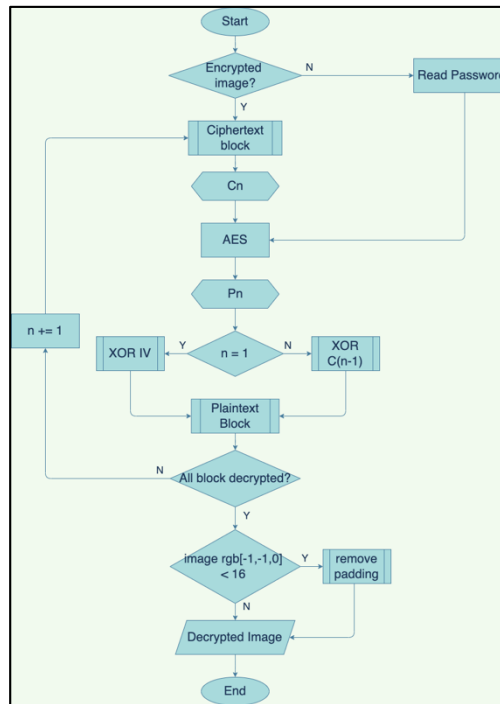


Figure 2. AES Decryption Flowchart

2.2 Least Significant Bit (LSB) Modification Method

Steganography is the art of hiding messages in digital media in such a way that other people do not realize that there is a message in the media. In the field of computer security, steganography is used to hide confidential data, when encryption cannot be done or at the same time as encryption [6].

The Least Significant Bit (LSB) is the least significant bit in a binary data. The LSB bit is located to the right of the binary number sequence. The opposite of LSB is MSB (Most Significant Bit) which is the most important part in a binary sequence that lies to the left of the binary number sequence.

The arrangement of bits in a byte that describes the suitable bit to replace is bit LSB, because this replacement only changes the byte value one higher or one lower than the previous value. Suppose the bytes in the image represent the color certain, then the change in the LSB bit does not change the color significantly and the changes are undetectable by the human eye [7].

For this application, the embedding process flow starts with user input in the form of images, videos, and passwords. Then the image resolution will be resized according to the video resolution. The video input then extracted for each frame. The first to third frames will be used for the image insertion process that has been encrypted with AES encryption. The four Most Significant Bit (MSB) images will be inserted into the first four Least Significant Bit (LSB) frames, four Least Significant Bit (LSB) images will be inserted into the second four Least Significant Bit (LSB) frames, and the third frame will be used for inserts the image resolution in string format after adjusting the video resolution.

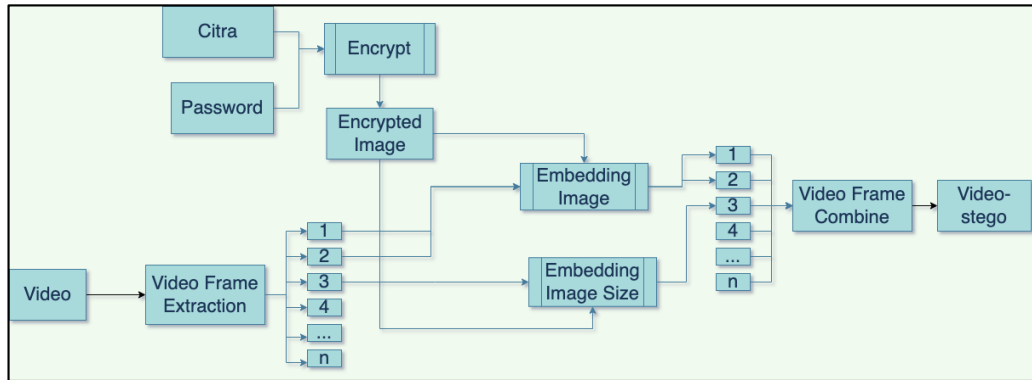


Figure 3. LSB Embedding Flowchart

The extraction process flow starts from entering the user in the form of stego-video and password. The video input is then extracted for each frame. The first to the third frames will be used for the image extraction process from stego-video. The first four Most Significant Bit (MSB) images will be taken from the first four Least Significant Bit (LSB) frames, four Least Significant Bit (LSB) images will be taken from the second four Least Significant Bit (LSB) frames. The results of the extraction of these two frames are then recorded and combined into a single unit, then the data is converted into a digital image.

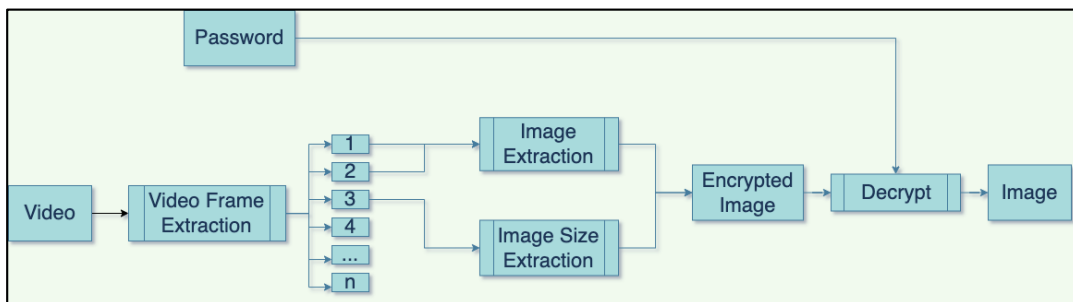


Figure 4. LSB Extraction Flowchart

2.3 Waterfall Method

Waterfall method provides sequential software life-flow approach starting from analysis, design, coding, and testing [8]. The development of this application will be using the waterfall method. Analysis stage will determine the capabilities that must be possessed by the system to meet what user needs.

1. The programming language used in the development of this application is Python.
2. This application is a desktop-based application with a minimum version of Python is 3.8 or newer.
3. This application is required to install several Python libraries used in this application before used.

In the design stage, the requirements needed in the development of this application are as follows.

- a. Front Page
 On the front page of this application, two options will be displayed, embedding and extraction, where the user can choose according to what the user wants. If the user selects embedding it will go to the embedding process page, whereas if the user selects extraction, it will go to the extraction process page.
- b. Embedding Page
 On the embedding process page, users are required to input a digital image file with .JPEG extension and a video file with .MP4 extension. Users are also required to enter a password that will be used in the digital image encryption process. After that, the user can click submit button and it will enter into the encryption and embedding process.
- c. Embedding Result Page

On the embedding results page, it will be displayed that the embedding process has been successful. After that, the user can upload the stego-video to Google Drive using the Google Drive API or save it in local storage.

d. Extraction Page

On the extraction process page, the user is required to input a stego-video file in that has a digital image in it. Users are also required to enter the same password during the insertion process which will be used in the digital image decryption process. After that, the user can click submit button and will enter into the decryption and extraction process.

e. Extraction Result Page

On the extraction results page, the extracted digital image will be displayed. Users can download the extracted digital image to be stored on their device.

3. Result and Discussion

The implementation of the development of the application will be made based on the design stage that translates the design into a program.

3.1 Coding

The implementation of this system is that the application can encrypt digital images, embed digital images into video, extract digital images from video, and decrypt encrypted images.

a. Application Interface

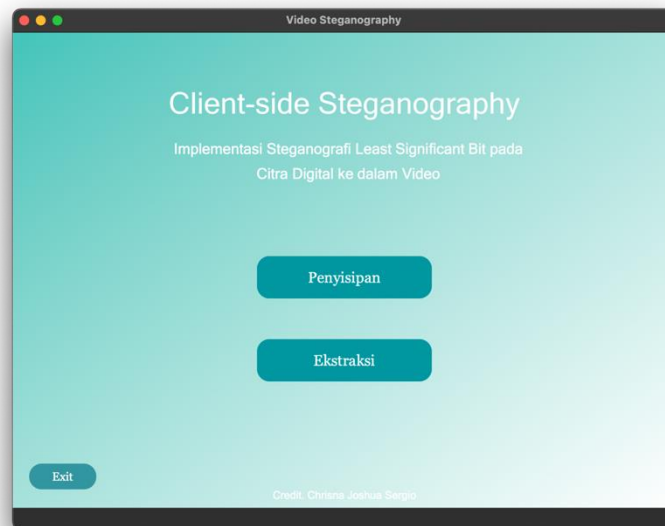


Figure 5. Main Menu Page

The main page display contains the main menus in the application. There are two menus in this page, Embedding Menu, which leads to the embedding page and Extraction Menu, which leads to the extraction page. On this page, there is also an exit button that can be used to exit the application.

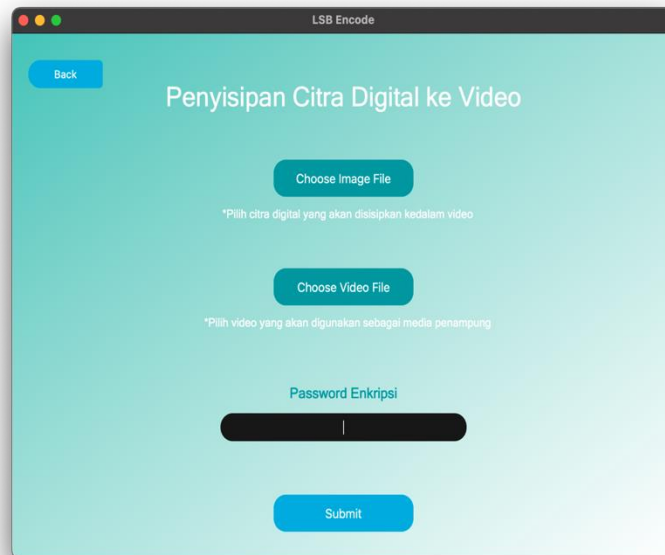


Figure 6. Embedding Page

The embedding page related to the process of embedding a digital image into a video. There are three inputs that must be entered by the user to carry out the embedding process. There are four buttons and one text field on this page. The choose image file button is used to handle image input, the choose video file button is used to handle video input, the back button is used to return to the main page, the text field is used to handle password input, and submit button is used to process all the inputs.

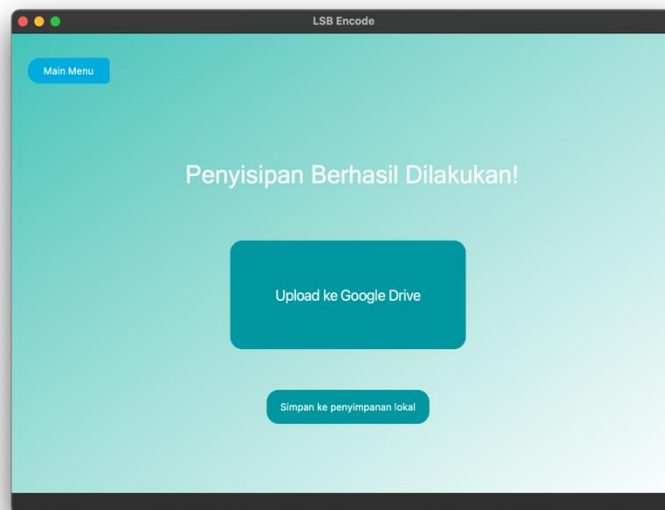


Figure 7. Embedding Result Page

The embedding result page indicates that the process of embedding a digital image into a video has been successfully carried out. On this page, there are three buttons that can be used by the user. The save to local storage button is used to save the stego-video to the user's local storage, the upload to Google Drive button is used to upload the stego-video to the user's Google Drive via Google Drive API, and the main menu button is used to return to the main page.

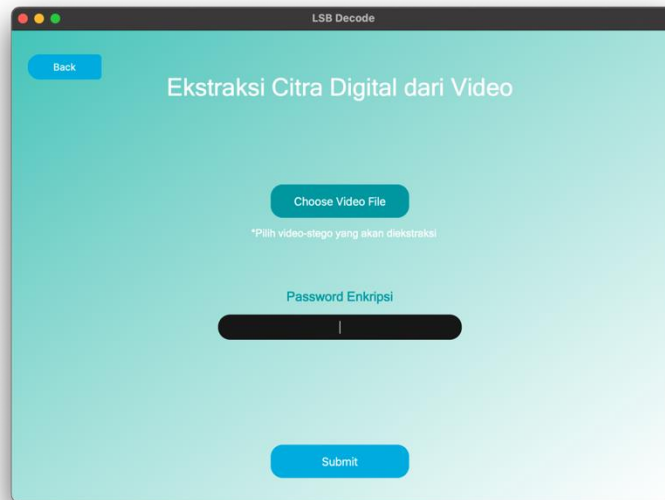


Figure 8. Extraction Page

Extraction page related to the process of extracting digital images from stego-video. There are two inputs that must be entered by the user to carry out the extraction process. There are three buttons and one text field on this page. The choose video file button functions to handle video input that has been embedded with an image, the back button is used to return to the main page, the text field is used to handle password input, and submit button is used to perform the extraction process from both inputs.

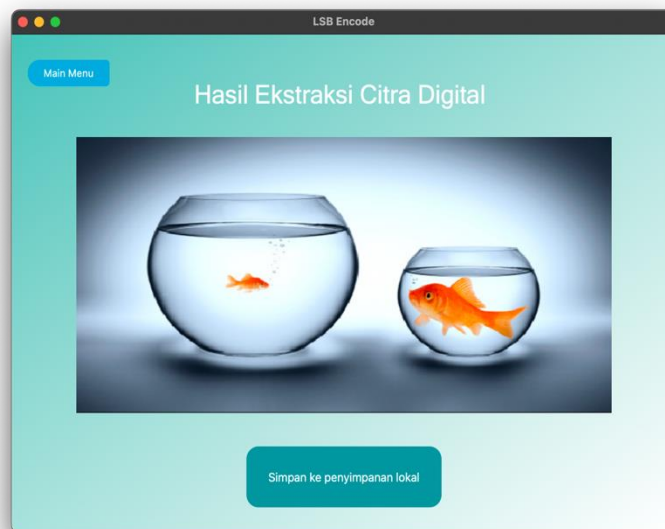



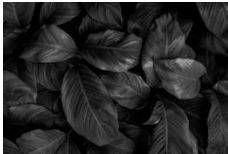


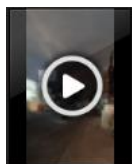
Figure 9. Extraction Result Page

The extraction result page indicates that the digital image extraction process from stego-video has been successfully carried out. On this page, users can see a preview of the successfully extracted image. This image preview display uses a fixed size and is not the original size of the extracted image. There are two buttons on this page. The save to local storage button is used to save the extracted image to the user's local storage and the main menu button is used to return to the main page.

3.2 Testing

The research data that will be used for testing this system is 3 digital image files and 2 video files. These data can be seen in Table 1.

Table 1. Research Data

No.	Filename	Resolution	File Size	Preview
1	Citra1.jpeg	528x324	49 KB	
2	Citra2.jpeg	626x418	61 KB	
3	Citra3.jpeg	4608×2963	2,2 MB	
4	Video1.mp4	640×360	784 KB	
5	Video2.mp4	720x1280	4,2 MB	

Testing the quality of the extraction results is carried out to find out how the quality of the image after it is extracted from stego-video. The test is carried out using the Peak Signal Noise Ratio (PSNR) method by comparing the original image and the extracted image. Peak Signal to Noise Ratio (PSNR) is the ratio between the maximum value of the measured signal and the amount of noise that affects the signal [9]. PSNR is measured in decibels. The results of the extraction quality test can be seen in Table 2.

Table 2. Extraction Quality Test Results

No.	Video File (.mp4)	Image File (.jpeg)	MSE	PSNR (dB)
1	Video1	Citra1	5.774328466691948	40.516
		Citra2	20.01752914951989	35.117
		Citra3	34.879654431216935	32.705
2	Video2	Citra1	5.367029898379105	40.833
		Citra2	15.276360013860014	36.291
		Citra3	33.308293336533076	32.905
Average				36.395

A good image is an image that has a small MSE value. The smaller the MSE value, the more similar an image is to the original image where the value of each pixel location is the same. The smaller the MSE value, the higher the PSNR value. In other words, the image is said to be good if the PSNR value is above 30 dB [10].

Table 4.3 shows the results of the extraction quality using the PSNR method. The table displays information in the form of MSE and PSNR values from the comparison of the original image and the extracted image. The highest PSNR value from this test is 40.516 dB in Video1 and Citra1 data, while the lowest PSNR value from this test is 32.705 dB in Video1 and Citra3 data. From this test, the average PSNR value is 36.395 dB and shows that the quality of the extracted image is said to be good. From this test, it can be concluded that the reduction in the extracted image is not significant because the original image and the extracted image still look the same visually. The quality of the extraction results is influenced by the resolution of the image along with the resolution of the video used as a media container. The MSE value is quite large because the image frame size is adjusted to the video frame size.

4. Conclusion

The quality of extraction with steganography technique using Modified Least Significant Bit and Advanced Encryption Standard methods can be said to be good. This is evidenced by the PSNR value in testing the quality of the extraction results. The average PSNR value obtained is 36.395 dB. Good PSNR value is above 30 dB and indicates that the quality of the insertion and extraction results is good. Based on the previous statement, the application has succeeded reaching the goal by securing digital image in video and extracting it from the stego-video without any data lost.

This application can be developed in terms of maintaining the quality of the extracted image against treatments such as compression, to reduce the stego-video file size and the extracted image file size. It can also be developed to run on various operating systems, so users with a base other than desktop can use this application.

References

[1] K. Thomas et al., "Data Breaches, Phishing, or Malware? Understanding the Risks of Stolen Credentials," in Proceedings of the ACM Conference on Computer and Communications Security, Oct. 2017, pp. 1421–1434. doi: 10.1145/3133956.3134067.

[2] E. Nirmala, "Penerapan Steganografi File Gambar Menggunakan Metode Least Significant Bit (LSB) dan Algoritma Kriptografi Advanced Encryption Standard (AES) Berbasis Android," Jurnal Informatika Universitas Pamulang, vol. 5, no. 1, 2020, [Online]. Available: <http://openjournal.unpam.ac.id/index.php/informatika36>

- [3] F. Muharram, H. Azis, and A. R. Manga, "Analisis Algoritma pada Proses Enkripsi dan Dekripsi File Menggunakan Advanced Encryption Standard," *Prosiding Seminar Nasional Ilmu Komputer dan Teknologi Informasi*, vol. 3, no. 2, 2018.
- [4] H. Mukhtar, *Kriptografi untuk Keamanan Data*. Yogyakarta: Deepublish, 2018.
- [5] D. Darwis, R. Prabowo, and N. Hotimah, "Kombinasi Gifshuffle, Enkripsi AES dan Kompresi Data Huffman untuk Meningkatkan Keamanan Data," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 4, p. 389, Oct. 2018, doi: 10.25126/jtiik.201854727.
- [6] R. D. Kusuma, M. Syafaat, and A. Setiawan, "Aplikasi Steganografi Video dengan Metode Least Significant Bit (LSB) untuk Alat Bantu pada Foto Film Militer," *Jurnal Elkasista*, vol. 1, 2020.
- [7] U. A. Anti, A. H. Kridalaksana, and D. M. Khairina, "Steganografi pada Video Menggunakan Metode Least Significant Bit (LSB) dan End of File (EOF)," *Jurnal Informatika Mulawarman*, vol. 12, no. 2, 2017.
- [8] G. W. Sasmito, "Penerapan Metode Waterfall Pada Desain Sistem Informasi Geografis Industri Kabupaten Tegal," vol. 2, no. 1, 2017, [Online]. Available: <http://www.tegalkab.go.id>,
- [9] G. W. Bhaudhayana and I. M. Widiartha, "Implementasi Algoritma Kriptografi AES 256 dan Metode Steganografi LSB pada Gambar Bitmap," *Jurnal Ilmiah Ilmu Komputer Universitas Udayana*, vol. 8, no. 2, 2015.
- [10] G. Badshah, S. C. Liew, J. M. Zain, and M. Ali, "Watermark Compression in Medical Image Watermarking Using Lempel-Ziv-Welch (LZW) Lossless Compression Technique," *Journal of Digital Imaging*, vol. 29, no. 2, pp. 216–225, Apr. 2016, doi: 10.1007/s10278-015-9822-4.