Performance Analysis of 3DES Image Encryption Algorithm for Image Transmission Between Two Computers Over a Wifi Network

Muhammad Ilham Nurcahyono[1], Nugroho Suharto[2*], Rizky Ardiansyah[3], Rieke Adriati Wijayanti[4]

[1,3,4]Digital Telecommunication Network Study Program, Department of Electrical Engineering, State Polytechnic of Malang, 65141, Indonesia.

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

[1]m.ilhamnurcahyono@gmail.com, [2]nugroho.suharto@polinema.ac.id, [3]rizkyardiansyah@polinema.ac.id, [4]riekeaw@polinema.ac.id

Abstract— **In today's digital era, data security, especially digital images, has become very important because of the risk of interception and unauthorized access. This research focuses on the use of the 3DES encryption algorithm to secure digital images. Taking inspiration from previous research examining image encryption based on Advanced Encryption Standard (AES) and Rivest–Shamir–Adleman (RSA) algorithms, this research develops and tests the implementation of the 3DES algorithm in the context of digital image storage and transmission. The main focus is on analyzing the performance of the 3DES algorithm in sending images between two computers over a Wi-Fi network, evaluating its effectiveness in maintaining data security. In addition, performance aspects such as encryption time, decryption time, and transmission overhead are also observed to assess feasibility. This research makes an important contribution to the development of reliable data security solutions in the rapidly changing digital era, particularly for secure wireless image communication systems.**

Keywords*— *3DES Algorithm, Data Security, Digital Image Encryption, Image Sending, Wifi Network.

I. INTRODUCTION

In the age of rapid technological development like today, many documents or personal data that were originally in printed form, become digital forms[1]. This of course makes it easier for humans to access documents and also saves storage space[2]. One of these documents can be in the form of images or digital images. A digital image is an image composed of pixels, where each pixel represents a color (or gray level for black and white images) at a specific point in the image[3].

Digital images can be stored on a variety of offline and online storage media, such as flash disks, hard disks, and also cloud storage services, such as Google Drive[4]. However, documents in the form of digital images are not completely secure. Digital images are, in principle, easily accessible to unauthorized parties, especially when they are shared over the internet[5]. Therefore, the application of encryption is a solution to overcome this problem[6]. Encryption is a process that converts data into a form that cannot be read or understood without using the right encryption key[7]. It is a security method applied to protect sensitive information from unauthorized access or attempts by spies who seek to access or steal data[8].

This research is a development of previous research entitled "Performance Analyses of AES and 3DES Algorithms for Encryption of Satellite Images". The research discussed the comparison between AES and 3DES encryption algorithms by testing the quality of encryption using histogram parameters, correlation coefficient, NPCR, UACI, PSNR, and computation time. There is no real implementation of the encryption algorithm when the image is stored and transmitted. Therefore, the research that will be made, examines the direct implementation of sending images over a wifi network. The consideration for choosing wifi as a tool in this research is that wifi is often used and very easy to find[9]. However, without realizing it, wifi has weaknesses, especially in the field of security. These weaknesses include hacker attacks, theft of company data, consumer data, and password data[10].

The research to be conducted discusses the performance of the 3DES image encryption algorithm with an implementation in the form of sending encrypted images over a wifi network between two computers. The 3DES image encryption algorithm was developed in 1998 as a replacement for the DES algorithm, which is vulnerable to brute force attacks because it uses a relatively short key. 3DES has a block size of 64 bits and uses a key of 196 bits, three times the size of the DES key[11]. On this basis, the 3DES encryption algorithm is considered more effective than DES for securing images.

II. METHOD

The research that has been done is a development of previous research. However, this research is differentiated in terms of the encryption method used, testing parameters, and implementation. The data used in this research is a .jpg image.

*Corresponding author

The image is processed using software that has been created using the Python programming language.

A. System Methodology

The process begins by importing the image to be encrypted and then entering the encryption key which must be 16 or 24 bytes long. If the key matches one of the required lengths, the image is encrypted using the 3DES algorithm. After encryption, the system calculates the correlation coefficient to evaluate the encryption and records the time taken for the encryption process. The encrypted image is then stored and ready to be sent to the server.

On the receiver side, the encrypted image is received by the server and then decrypted using 3DES to restore the image to its original form. The output of this decryption process is the initial image and the decryption time, indicating that the image has been successfully restored to its original form and the encryption-decryption process has been completed. This process is iterative and can be repeated by importing a new image and starting the process all over again. Are some of the terms used.

1. Encryption Time: Encryption time is the time taken by the system to convert plaintext into ciphertext. Encryption time depends on the key size, size, and mode of the plaintext block. Encryption Time impacts the performance of the system. The faster the time taken by the system for encryption, the better [12]. In this research, encryption time is measured using units of seconds.
2. Decryption Time: Decryption time is the time taken by the system to recover plaintext to ciphertext [13]. The shorter the time the system takes for decryption, the better. In this research, decryption time is measured using units of seconds.
3. Correlation Coefficient: The correlation coefficient between the original image and the resulting image is a measurement formula that is often used to assess the changes that occur between the image before and after the encryption process. The quality of encryption is considered good if the correlation coefficient between the original image and the resulting image is closer to zero (0). In addition to the comparison between the result image and the original image, the correlation coefficient can also be applied to measure the correlation between adjacent pixels. The purpose of image encryption is to reduce the correlation between adjacent pixels in the encrypted image so that the correlation coefficient becomes weak or close to zero [14].
4. Histogram: An image histogram is a graph that depicts the distribution of pixel intensity values of an image. The histogram also shows the brightness and contrast of an image. Histograms are thus an important tool in image processing, both qualitatively and quantitatively. A good image has a histogram that fills the gray degree region with an even distribution of each pixel intensity value. The following is the shape of the histogram with image characteristics [15].

B. Application Design

Application design is an important stage in system or software development. It involves the process of detailing how the application will operate and look, as well as how users will interact with the application.

1. Encryption Application Design: The encryption application consists of two segments, namely the upper segment and the lower segment. The upper segment contains some important components, namely four main image viewers. The first image viewer is on the top left, which is the input image viewer. The input image viewer is a button for inserting the original image (plain image) into the user interface (UI). The second image viewer on the top right is the input image histogram viewer. The input image histogram viewer is a button to save the input image histogram. The third image viewer is located just the input image button, which is the encrypted (grayscale) image viewer. The encrypted (grayscale) image viewer is a button to save the encrypted image. The fourth image viewer is located just the save input image histogram button, which is the encrypted image histogram viewer. The encrypted image histogram viewer is a button to save the encrypted image histogram as illustrated in Fig. 1.

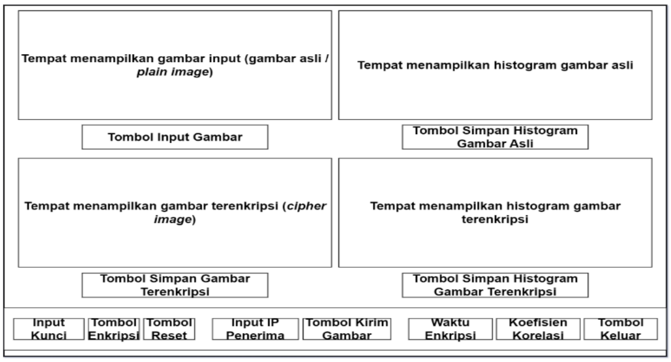

Tempat menampilkan gambar input (gambar asli / *plain image*)
Tempat menampilkan histogram gambar asli
Tombol Input Gambar
Tombol Simpan Histogram Gambar Asli
Tempat menampilkan gambar terenkripsi (*cipher image*)
Tempat menampilkan histogram gambar terenkripsi
Tombol Simpan Gambar Terenkripsi
Tombol Simpan Histogram Gambar Terenkripsi
Input Kunci
Tombol Enkripsi
Tombol Reset
Input IP Penerima
Tombol Kirim Gambar
Waktu Enkripsi
Koefisien Korelasi
Tombol Keluar

Figure 1. Encryption Application Design (Sender)

The bottom segment contains several entries and buttons in a row, on the far left there is an entry for entering the encryption key. To the right of the key entry is an encryption button to start the image encryption process. To the right of the encryption button is a reset button to empty the data in each component after the encryption process is complete. To the right of the reset button is an entry for the IP input of the receiving computer. To the right of the recipient's IP entry is a send encrypted image button to send the encrypted image to the recipient's computer (server). To the right of the send encrypted image button is the encryption time viewer text. To the right of the encryption time display text is the correlation coefficient display text. On the far right of the bottom segment, there is an exit button to close the UI.

2. Decryption Application Design: The decryption application consists of two segments, the upper segment and the lower segment. The upper segment consists of two columns, the left column contains the input image directory viewer and the encrypted image input button (cipher image) to insert the image into the UI. While the right column contains the decrypted image viewer, like shown in Fig. 2.

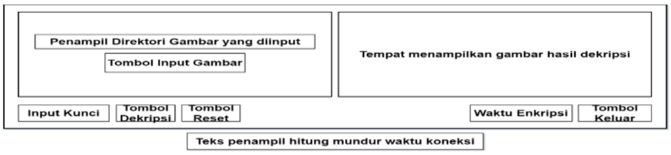

Penampil Direktori Gambar yang diinput
Tombol Input Gambar
Tempat menampilkan gambar hasil dekripsi
Input Kunci
Tombol Dekripsi
Tombol Reset
Waktu Enkripsi
Tombol Keluar
Teks penampil hitung mundur waktu koneksi

Figure 2. Decryption Application Design (Receiver)

In the lower segment, there are two sides, the left side and the right side. On the left side, the leftmost part is filled with a key entry to enter the decryption key (the same key as the encryption key). To the right of the key entry is a decryption button to start the decryption process. To the right of the decryption button is a reset button to clear the data in each component (directory viewer, decrypted image viewer, key entry, and decryption time) after the decryption process is complete. On the right side, the leftmost part is filled with the decryption time text, and to the right of the decryption time text, there is an exit button to exit the UI.

III. RESULTS AND DISCUSSION

A. Software Design Results

This section describes the results of the software design that has been designed previously. The purpose of this section is to provide an understanding of how the application works technically. The results of the software design are shown in Fig. 3 and Fig. 4.

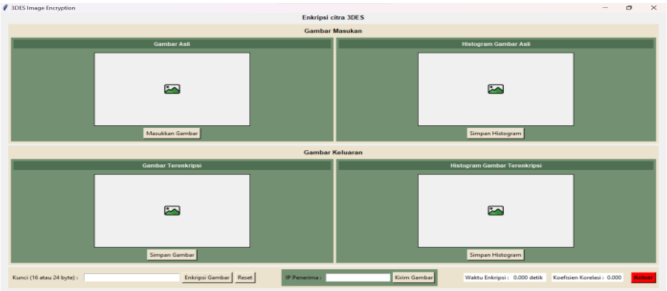


Figure 3. Encryption User Interface

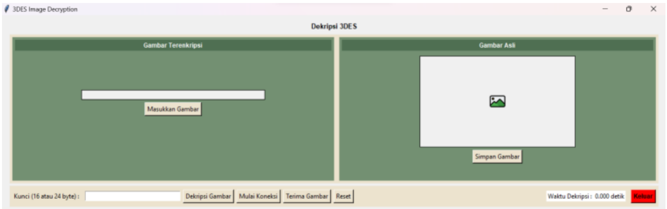
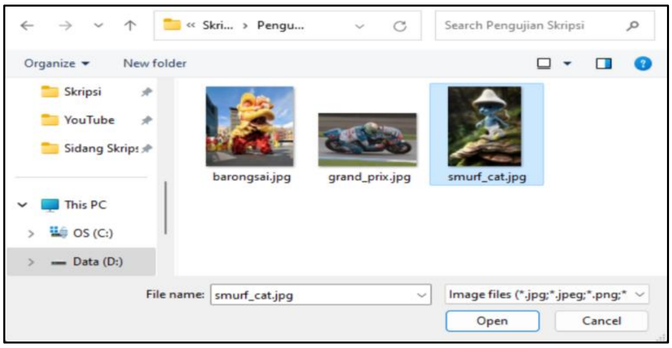


Figure 4. Decryption User Interface

B. Encryption Testing

Encryption testing begins with pressing the “Masukkan Gambar” button to enter an image into the UI. The user is asked to enter an image with a predetermined format (for example .jpg) and then click “Open” as shown in Fig. 5. For example, we use a smurf_cat.jpg image with dimensions of 716 x 955 pixels with a file size of 95.1 KB (kilobytes) shown in Fig. 6.



File name: smurf_cat.jpg

Image files (*.jpg;*.jpeg;*.png;*

Figure 5. Image Input Display

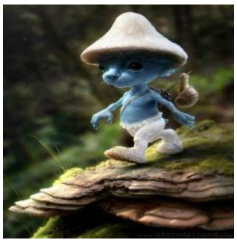
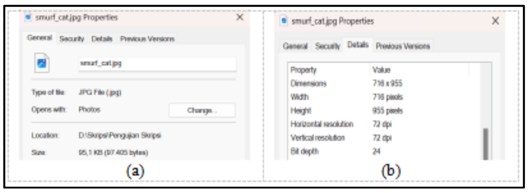


Figure 6. smurf_cat.jpg

Fig.7 shows the metadata or information about the image. Information displayed such as file name, file type, file directory, file size, dimensions, width, height, and others.



smurf_cat.jpg

Type of file: JPG File (.jpg)

Opens with: Photos

Location: D:\Skripsi\Pengujian Skripsi

Size: 95,1 KB (97.405 bytes)

Property	Value
Dimensions	716 x 955
Width	716 pixels
Height	955 pixels
Horizontal resolution	72 dpi
Vertical resolution	72 dpi
Bit depth	24

Figure 7. (a) Image Metadata smurf_cat.jpg (General) and (b) Image Metadata smurf_cat.jpg (Details)

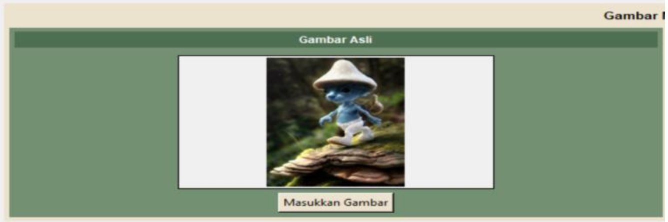


Figure 8. Original Image Display

When the image is finished being inserted, the UI display will change as shown in Fig. 8. Simultaneously, the system also displays the original image histogram just to the right of the input image as shown in Fig. 9. In this case, the user can press the “Simpan Histogram” button to save the original image histogram.

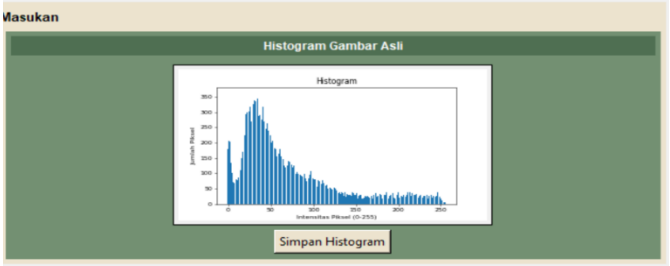
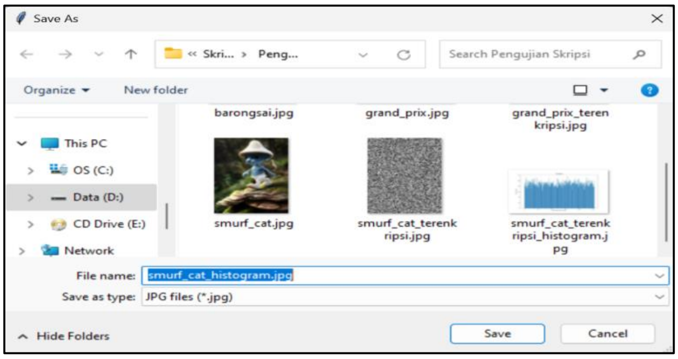


Figure 9. Display of Original Image Histogram

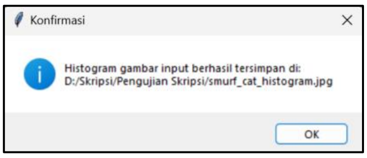
After the “Simpan Histogram” button is pressed, the user is asked to specify the directory and file name of the histogram as shown in Fig. 10. In this case, the histogram of the input image is named “smurf_cat_histogram”. Press “Save” to save the histogram. After the histogram has been successfully saved, a notification window will appear containing the image directory. The notification window is shown in Fig.11.



File name: smurf_cat_histogram.jpg

Save as type: JPG files (*.jpg)

Figure 10. Input Image Histogram Save Window



Histogram gambar input berhasil tersimpan di:
D:/Skripsi/Pengujian Skripsi/smurf_cat_histogram.jpg

Figure 11. Notification of Input Image Histogram has Successfully Saved

Fig. 12 shows the histogram of the smurf_cat.jpg image. In the image, the histogram has a distribution of pixel intensities that tends to be centered on the left. This indicates that the image is dominated by dark pixels.

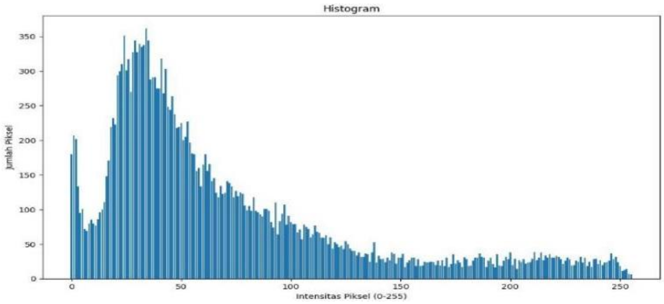
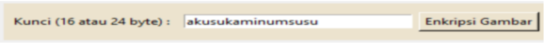


Figure 12. smurf_cat_histogram.jpg

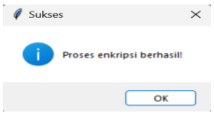
The next step is to enter the encryption key. The encryption key is 16 or 24 bytes (characters). In this case, the key used is 16 bytes, for example, “akusukaminumsusu”. This process is shown in Fig. 13.



Kunci (16 atau 24 byte) : akusukaminumsusu

Figure 13. Display of Encryption Key Input

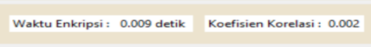
After the encryption key is entered into the entry, the next step is to press the “Enkripsi Gambar” button to start the encryption process. After the encryption process is complete, a notification window will appear that the image has been successfully encrypted as shown in Fig. 14.



Proses enkripsi berhasil!

Figure 14. Confirmation Window

In addition, the encrypted image, encryption time, correlation coefficient, and histogram of the encrypted image appear on the UI. The smurf_cat image with dimensions of 716 x 955 pixels in .jpg format was successfully encrypted with an encryption time of 0.009 seconds with a correlation coefficient of 0.002. The UI display of encryption time and correlation coefficient is shown in Fig. 15.



Waktu Enkripsi : 0.009 detik

Koefisien Korelasi : 0.002

Figure 15. UI Display of Encryption Time and Correlation Coefficient

At this stage, the encrypted image and the histogram of the encrypted image can both be saved by the user. The encrypted image can be saved by pressing the “Simpan Gambar” button just the encrypted image viewer. The “Simpan Gambar” button is shown in Fig. 16.

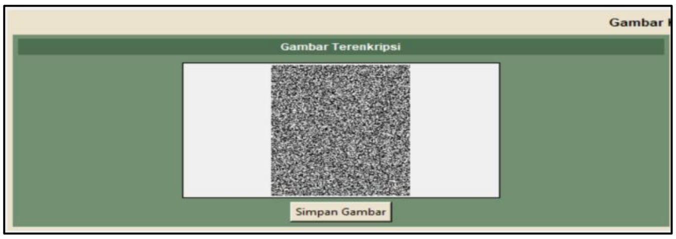
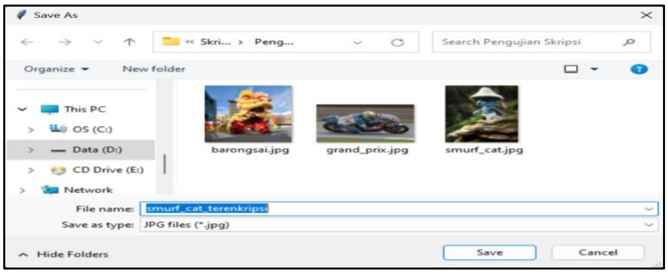


Figure 16. Display of Encrypted Image

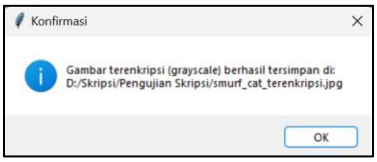
When the “Simpan Gambar” button is pressed, the saving window will appear as shown in Fig. 17. The user is then asked to specify the directory and enter the desired image file name. In this case, the file name "smurf_cat_encrypted" is used for the encrypted image. If the directory and image file name have been specified, the user can press the “Save” button to save the image.



File name: smurf_cat_terenkripsi
Save as type: JPG files (*.jpg)

Figure 17. Encrypted Image Save Window

After the encrypted image is successfully saved, a notification window will appear showing confirmation that the image was successfully saved in the desired directory. The notification window is shown in Fig. 18.



Gambar terenkripsi (grayscale) berhasil tersimpan di: D:/Skripsi/Pengujian Skripsi/smurf_cat_terenkripsi.jpg

Figure 18. Saved Image Notification Window

To ensure that the encrypted image has been saved perfectly, search for the file in the file explorer according to the previously specified directory. Fig. 19 shows that the encrypted image was successfully saved.

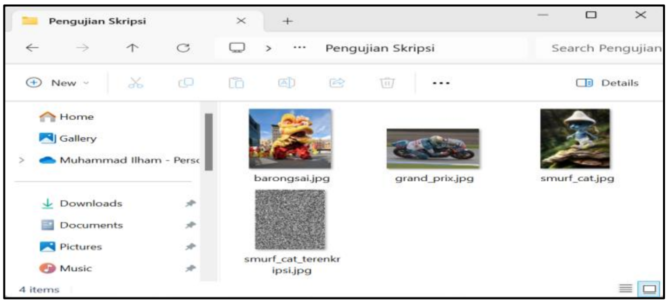


Figure 19. Encrypted Image Successfully Saved

When the image is opened with an image viewer application, it will appear as shown in Fig. 20.

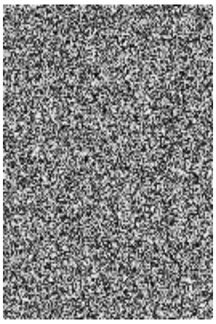


Figure 20 . smurf_cat_encrypted.jpg

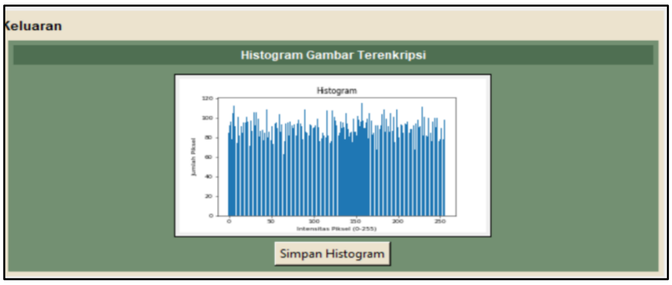
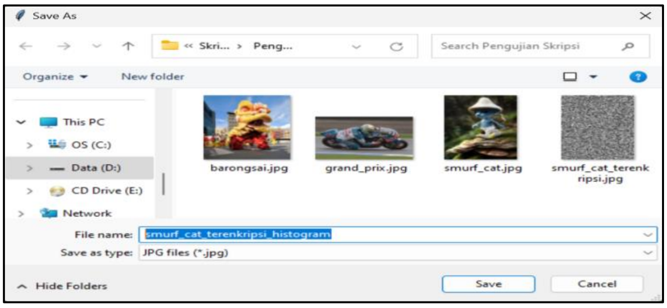


Figure 21. Display of Encrypted Image Histogram

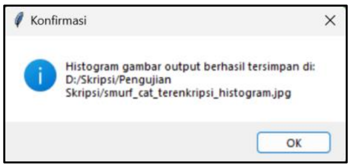
The encrypted image histogram saving stage is not much different from the previous encrypted image saving stage. The user presses the "Simpan Histogram" button as in Fig. 21, then a storage window appears to specify the directory and image file name as in Fig. 22. In this case, the histogram file is named "smurf_cat_encrypted_histogram". After that, the user presses the "Save" button to save the histogram.



File name: smurf_cat_terenkripsi_histogram
Save as type: JPG files (*.jpg)

Figure 22. Encrypted Image Histogram Save Window

A notification window appears that the histogram was successfully saved to the desired directory. The notification window is shown in Fig. 23.



Histogram gambar output berhasil tersimpan di: D:/Skripsi/Pengujian Skripsi/smurf_cat_terenkripsi_histogram.jpg

Figure 23. Saved Histogram Notification Window

To ensure that the encrypted image histogram has been saved perfectly, search for the file in the file explorer according to the previously specified directory. Fig. 24 shows that the encrypted image was successfully saved.

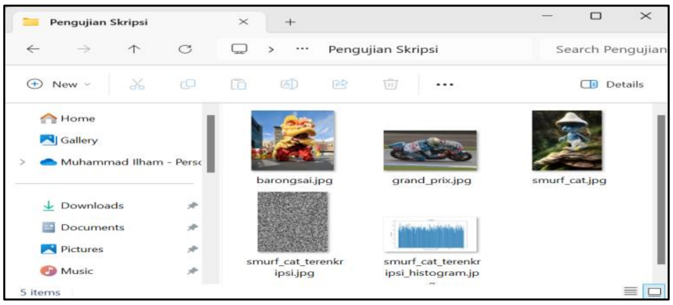


Figure 24. Histogram of Encrypted Images has Successfully Saved

Fig. 25 is a histogram of the encrypted image. The histogram tends to have an even distribution of pixel intensities.

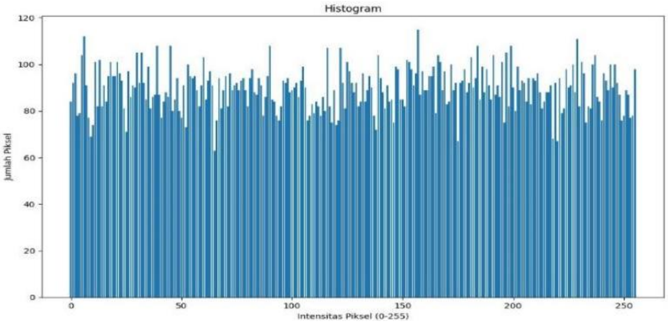
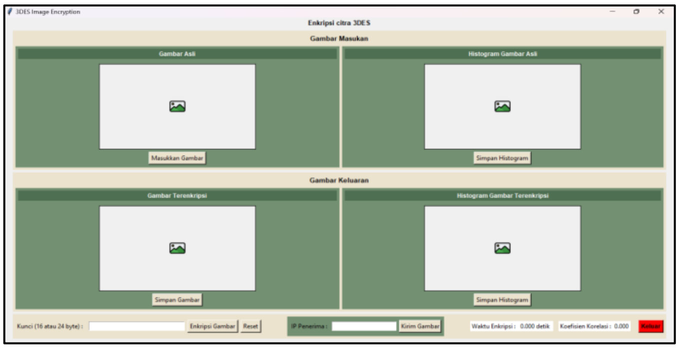


Figure 25. smurf_cat_encrypted_histogram.jpg

Furthermore, if you want to re-encrypt (with a different image), the user can press the "Reset" button. After the button is pressed, the original image viewer field, original image histogram, encrypted image, encrypted image histogram, key entry, IP address entry, encryption time, and correlation coefficient will be emptied again as shown in Fig. 26.



Waktu Enkripsi : 0.000 detik

Koefisien Korelasi : 0.000

Figure 26. Display after Reset

C. *Image Sending and Receiving Test*

Before starting the sending and receiving process, both computers (client and server) must first be connected to the same network. In this case, a wifi with the name “Uti Sugeng” is used which is shown in Fig. 27 and Fig. 28.

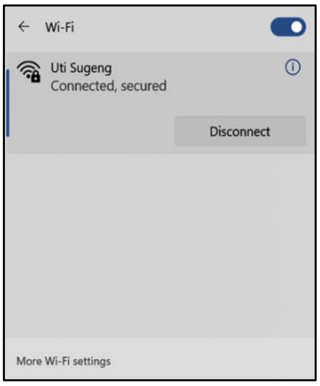


Figure 27. Wifi Connection on Sender Computer (Client)

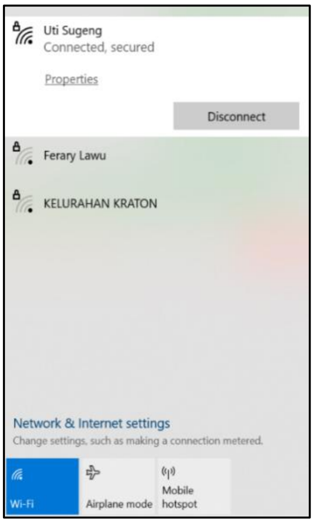
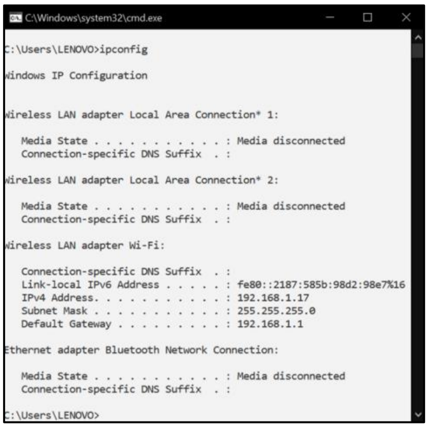


Figure 28. Wifi Connection on Receiver Computer (Server)

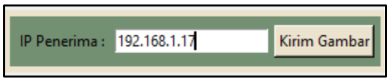
After both computers are connected to the same network, on the sending computer (client), the user enters the IP of the receiving computer (server) in the IP entry field. The IP address of the server computer can be found by typing the command “ipconfig” at the command prompt (cmd) on the server computer itself. In Fig. 29, the IP address is shown on the line labeled “IPv4 Address”. So the IP of the server computer is 192.168.1.17.



```
C:\Users\LENOVO>ipconfig

Windows IP Configuration


Wireless LAN adapter Local Area Connection* 1:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::2187:585b:98d2:98e7%16
   IPv4 Address. . . . . . . . . . . : 192.168.1.17
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.1.1

Ethernet adapter Bluetooth Network Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\Users\LENOVO>
```

Figure 29. IP address of Server Computer

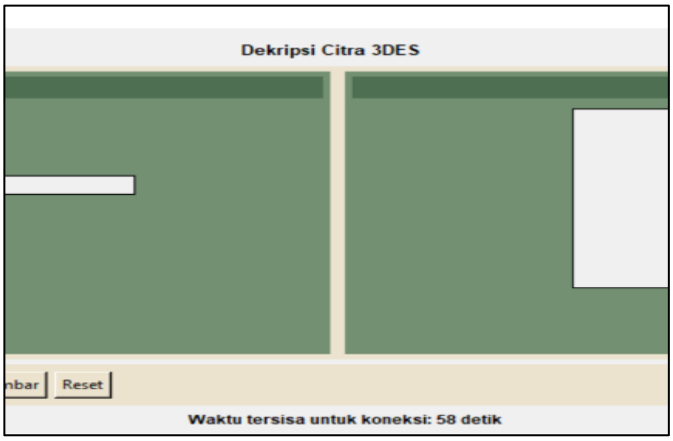
After knowing the IP of the server computer, enter the IP address of the server computer into the IP entry as shown in Fig. 30.



IP Penerima : 192.168.1.17

Figure 30. Server Computer IP Input (Client Side)

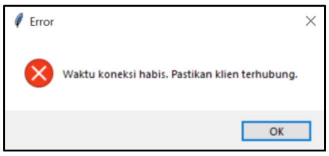
Ensure that the program on the server computer is running and shows the connection countdown as shown in Fig. 31. The connection countdown will run for 60 seconds and ensure that the server computer is ready to receive the image file.



Waktu tersisa untuk koneksi: 58 detik

Figure 31. Connection Time Countdown

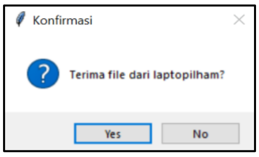
If the server computer connection time has expired, a notification window will appear as shown in Fig. 32.



Waktu koneksi habis. Pastikan klien terhubung.

Figure 32. Connection Timeout Window

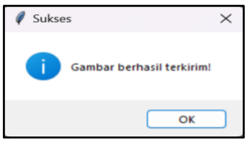
After the server computer is ready to receive the image file, the next step is for the client computer user to press the “Kirim Gambar” button to start the sending process according to the IP address of the destination computer. On the server computer side, a window will appear containing information on the name of the sending computer. Press “Yes” to receive the image file. The window is shown in Fig. 33.



Terima file dari laptopilham?

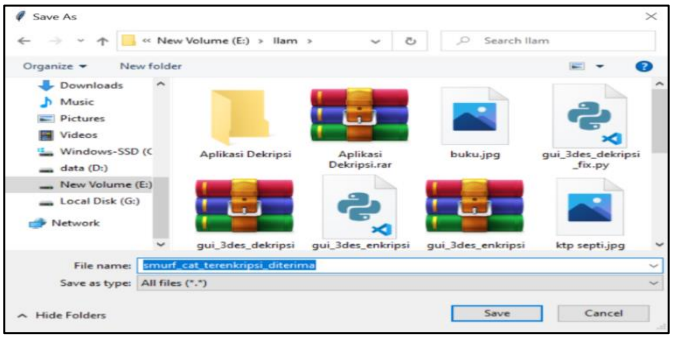
Figure 33. Image File Acceptance Confirmation Window

After the image file has been received, on the client's computer side, the image notification window will appear once the image has been successfully sent as in Fig. 34. Meanwhile, on the server computer. The user is asked to specify the directory and file name as shown in Fig. 35. In this case, the file is named “smurf_cat_encrypted_received”. Press “Save” to save the image.



Gambar berhasil terkirim!

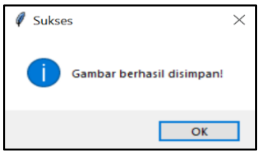
Figure 34. Notification of The Image has Successfully Sent (Client Side)



File name: smurf_cat_terenkripsi_diterima
Save as type: All files (*.*)

Figure 35. Saving Window (Server Side)

After the encrypted image is successfully stored on the server computer, a notification of a successfully stored image will appear, as shown in Fig. 36. The encrypted image received does not have an extension, so it cannot be opened with any application. However, it will still be decrypted later.



Gambar berhasil disimpan!

Figure 36. Notification of Successfully Saved Image (Server Side)

To ensure that the encrypted image has been received and stored perfectly on the server computer, search for the file in the file explorer according to the previously specified directory. Fig. 37 shows that the encrypted image was successfully saved.

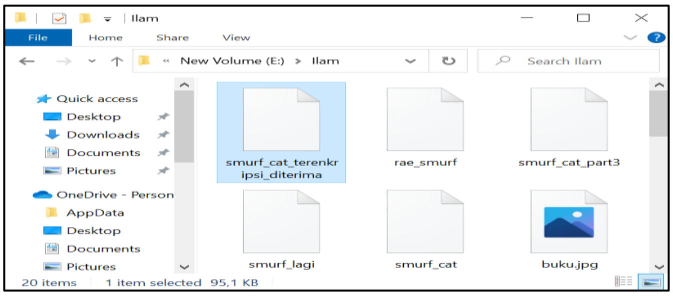
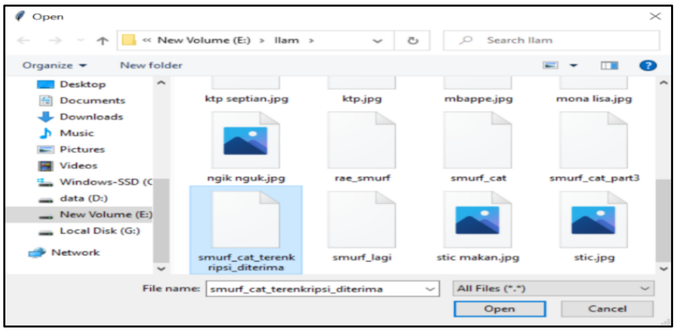


Figure 37. Encrypted Image Received by Server Computer

D. *Decryption Testing*

At this stage, the decryption process starts by pressing the “Masukkan Gambar” button to insert the image to be decrypted (in this case the file named “smurf_cat_encrypted_received” into the UI. The image selection window will open, select the desired image, then click “Open” to continue. This step is shown in Fig. 38.

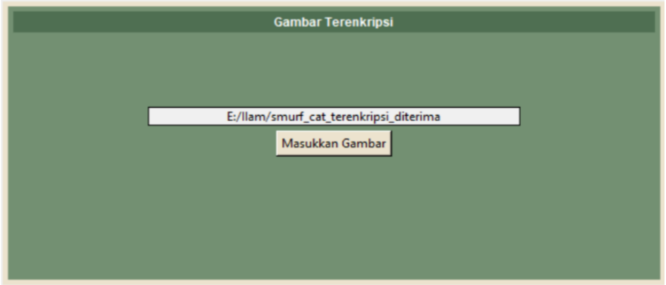


File name: smurf_cat_terenkripsi_diterima

All Files (*.*)

Figure 38. Enter the Image to be Decrypted

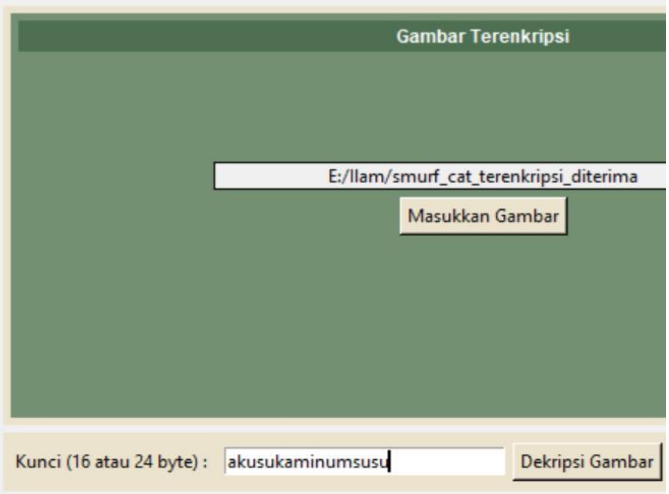
Once the image has been inserted, the directory viewer box will display the directory of the selected image. Fig. 39 shows the directory viewer.



E:/Ilam/smurf_cat_terenkripsi_diterima

Figure 39. Directory Viewer After Image Input

After the image has been entered, the next step is to enter the decryption key (the same key used during the encryption process). In this case, the encryption key is "akusukaminumsusu" which means that the decryption process also uses that key. Once the key is entered, the decryption process can be started by pressing the "Decrypt Image" button. The process is shown in Fig. 40.

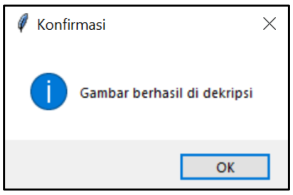


E:/Ilam/smurf_cat_terenkripsi_diterima

Kunci (16 atau 24 byte) : akusukaminumsusu

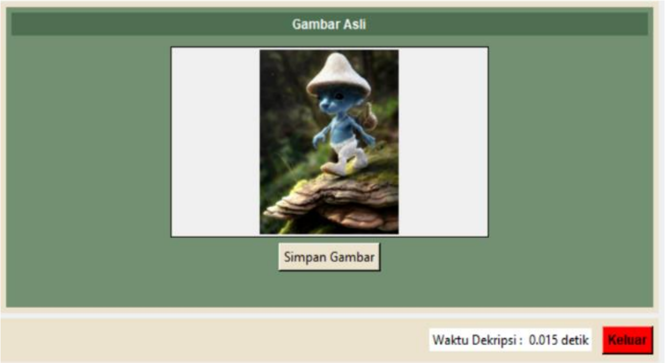
Figure 40. Decryption Key Input

After the decryption button is pressed, a notification window appears that the decryption process was successful as shown in Fig. 41. In addition, the UI also displays the original image (plain image) and the decryption time shown in Fig. 42. The decryption time required is 0.015 seconds.



Gambar berhasil di dekripsi

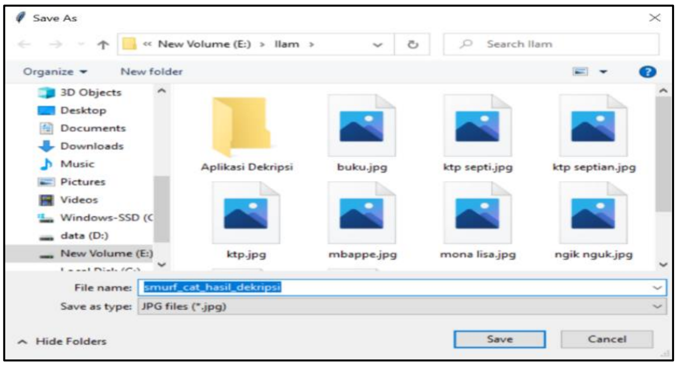
Figure 41. Successful Decryption Notification Window



Waktu Dekripsi : 0.015 detik

Figure 42. UI Display after Decryption Completed

The decrypted image can be saved by pressing the “Simpan Gambar” button. After that, a window will appear to specify the directory and file name (in this case the name "smurf_cat_hasil_dekripsi.jpg" is used), then press “Save” to save the image on the computer. The process is shown in Fig. 43.

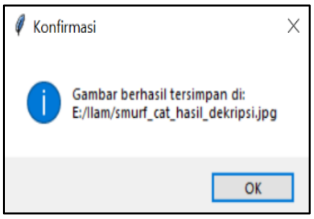


File name: smurf_cat_hasil_dekripsi

Save as type: JPG files (*.jpg)

Figure 43. Save Decrypted Image Window

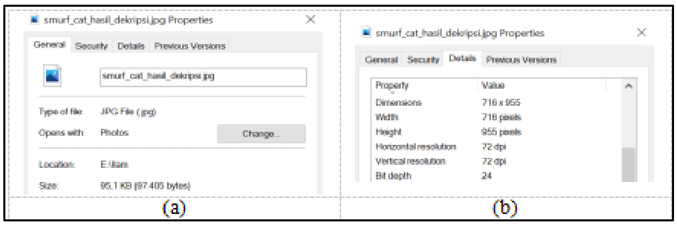
After the image is successfully saved, a notification window will appear where the image file is successfully saved, as shown in Fig. 44. Users can exit the UI by pressing the “OK” button.



Gambar berhasil tersimpan di:
E:/Ilam/smurf_cat_hasil_dekripsi.jpg

Figure 44. Notification Window To Save The Decrypted Image

As can be seen in Fig. 7 and Fig. 45, the image metadata before and after the encryption/sending process remains the same. This indicates that there is no addition or subtraction of any image element during the encryption, decryption, and delivery process.



smurf_cat_hasil_dekripsi.jpg

Type of file: JPG File (.jpg)

Opens with: Photos

Location: E:\Ilam

Size: 95,1 KB (97.405 bytes)

Property	Value
Dimensions	716 x 955
Width	716 pixels
Height	955 pixels
Horizontal resolution	72 dpi
Vertical resolution	72 dpi
Bit depth	24

Figure 45. (a) Metadata smurf_cat_hasil_dekripsi.jpg (General) and (b) Metadata smurf_cat_hasil_dekripsi.jpg (Details)

The image resulting from the decryption process when opened using an image viewer application will look identical to the initial image. The decrypted image can be seen in Fig. 46.

Figure 46. smurf_cat_result_decryption.jpg

IV. CONCLUSION

The image encryption process using the 3DES method begins by determining the image to be processed. Images can be in .jpg, .png, or .gif format. After the image is selected, then determine the external key used for encryption. The external key is 16 or 24 characters long. During processing, the image is divided into data blocks and each data block is encrypted by the internal key created from the previous external key. After that, the image is converted to binary form and the output is an encrypted image (looks random in grayscale format). The performance results of the 3DES image encryption algorithm produce a good performance, the encryption and decryption time is very short, the correlation coefficient is close to zero (0) and the encrypted image histogram is quite uniform and has a significant difference from the original image histogram. The image delivery process utilizes the client-server principle by connecting to the same network. The server must first open a connection for the client to send an image delivery request. The client must also specify the exact IP of the server so that the image can be transmitted and stored properly.

REFERENCES

[1] A. P. Sukma, W. B. Nugroho, and N. Zuryani, “Digitalisasi Al-Quran: Meninjau Batasan Antara yang Sakral dan yang Profan pada Aplikasi ‘Muslim Pro,’” J. Ilm. Sosiol., vol. 1, no. 1, pp. 1–15, 2019.

[2] S. M. Rosmaniah, B. Santoso, and S. A. Muhidin, “Digitalisasi Arsip Statis Pada Masa Pandemi Covid-19 Di Perguruan Tinggi,” J. Pendidik. Manaj. Perkantoran, vol. 7, no. 2, pp. 214–224, 2022.

[3] I. Aulia, “Implementasi Teknik Watermarking Pada Citra Digital Dengan Menggunakan Metode Fractal dan Discrete Cosine Transform (DCT),” Inf. dan Teknol. Ilm., vol. 6, no. 2, pp. 235–240, 2019.

[4] A. F. Yana, “Implementasi Pengolahan Citra Digital Pada Penghitungan Anak Burung Puyuh Menerapkan Metode Blob,” J. Inf. Syst. Res., vol. 1, no. 4, pp. 237–245, 2020.

[5] A. Azanuddin, S. Yakub, and J. Prayudha, “Implementasi Keamanan Citra Menggunakan Algoritma AES-128 Dengan Aplikasi Client-Server,” Jurasik (Jurnal Ris. Sist. Inf. dan Tek. Inform., vol. 7, no. 1, p. 51, 2022.

[6] T. Darmanto and E. Gunawan, “Penerapan Algoritma Simetris Pada Aplikasi Kriptografi Pesan Berbasis Android,” J. InTekSis, vol. 8, no. 1, p. 53.

[7] D. Hulu, B. Nadeak, and S. Aripin, “Implementasi Algoritma AES (Advanced Encryption Standard) Untuk Keamanan File Hasil Radiologi di RSU Imelda Medan,” KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer), Volume 4, Nomor 1 vol. 4, pp. 78–86, 2020.

[8] A. M. Ujung, M. Irwan, and P. Nasution, “Pentingnya Sistem Keamanan Database Untuk Melindungi Data Pribadi,” JISKA J. Sist. Inf. Dan Inform., vol. 1, no. 2, p. 44, 2023.

[9] D. P. Rustianti and T. S. Reza, “Pengaruh Strategi Pemasaran Dan Inovasi Produk Terhadap Volume Penjualan Produk Wifi.Id Di Pt. Telkom Indonesia Wilayah Jakarta Pusat,” J. Adm. Bisnis, vol. 2, no. 2, pp. 193–207, 2022.

[10] R. Sahara, S. Abdullah, and R. Saputra, “Analisis Ancaman Sniffing pada Jaringan WiFi di PT. Stepa Wirausaha Adiguna,” Pros. Semin. Nas. Ris. Inf. Sci., vol. 4, no. 2, pp. 224–230, 2022.

[11] Y. Ortakci and M. Y. Abdullah, “Performance Analyses of AES and 3DES Algorithms for Encryption of Satellite Images,” Lect. Notes Networks Syst., vol. 183, no. February 2021, pp. 877–890, 2021.

[12] S. Permatasari, A. Aminudin, and S. Arifianto, “Modifikasi Enkripsi dan Dekripsi AES dengan Polybius Chiper dalam Pengamanan Data,” JRST (Jurnal Ris. Sains dan Teknologi, vol. 4, no. 1, p. 41, 2020.

[13] R. Perangin-angin, I. Kelana Jaya, B. Rumahorbo, and B. Juni R Marpaung, “Analisa Alokasi Memori dan Kecepatan Kriptograpi Simetris Dalam Enkripsi dan Dekripsi,” Journal Information System Development, vol. 4, no. 1, 2019.

[14] M. Haris, M. S. Lydia, and S. Sutarman, “Pengamanan Pada Citra Digital dengan Menggunakan Modifikasi Blok Data Algoritma AES-Rijndael,” J. Media Inform. Budidarma, vol. 7, no. 1, pp. 444–453, 2023.

[15] R. Gonydjaja, Pengantar Pengolahan Citra Digital, 978th-623rd–81sted. Lombok Tengah: Pusat Pengembangan Pendidikan Penelitian Indonesia, 2023.