Meningkatkan Keamanan File DICOM melalui Enkripsi AES: Implementasi dan Analisis

Aloysius Arya Wibisono 18329925
Program Studi Teknik Biomedis
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): aloysiusaryawibisono@gmail.com

***Abstract*— This paper discusses the implementation of AES encryption on DICOM files in the context of medical data security and privacy. Using the AES encryption method, this study examines the effectiveness of encryption on DICOM images through histogram analysis, PSNR, and NPCR. The analysis results indicate that encryption affects changes in the encrypted images, but the decrypted images successfully maintain the integrity of information with NPCR approaching zero and high PSNR. This indicates that encrypted images can still be used accurately in patient diagnosis and treatment. In conclusion, AES encryption is an effective choice for securing medical image data without compromising information quality. (*Abstract*)**

Keywords—AES; DICOM; Medical Data, PSNR, NPCR, Histogram

I. INTRODUCTION (*HEADING 1*)

Salah satu Sistem yang penting di rumah sakit adalah sistem PACS (Picture Archiving and Communication System) yakni sistem yang mengatur bagaimana data citra dari modalitas - modalitas tertentu dikirimkan dan disimpan untuk dianalisis. Dengan adanya sistem ini, rumah sakit tidak memerlukan penyimpanan citra medis secara manual yakni menggunakan film – film hasil pengolahan di kamar gelap yang memiliki banyak kelemahan diantaranya adalah kerusakan film pada proses penyimpanan dikarenakan tidak adanya standar. Selain menghindari kerusakan pada saat penyimpanan, sistem PACS juga mempercepat analisis citra medis dikarenakan dokter radiologi bisa memperoleh data secara langsung lewat klien (komputer, laptop, dan sejenisnya) tanpa harus menunggu pengantaran film citra medis.

Untuk standardisasi pertukaran dan penyimpanan citra – citra digital digunakan standar DICOM (Digital Imaging and Communications in Medicine) yang mencakup format data penyimpanan, protocol komunikasi, dan transfer antar file. DICOM sendiri berisikan data – data pribadi pasien dan juga citra dari modalitas terkait. Seiring berkembanganya penggunaan DICOM sebagai file medis muncul masalah – masalah sekuritas yang perlu dihadapi, dikutip dari Departemen Health and Human Services terdapat lebih dari 3000 kebocoran data yang melibatkan lebih dari 500 riwayat medis di Amerika. Untuk mengatasi hal tersebut salah satu solusi yang dilakukan adalah melakukan enkripsi dan dekripsi data, enkripsi sendiri berfungsi untuk menerjemahkan data ke sesuatu yang tampak acak dan dekripsi adalah bagaimana data acak tersebut bisa diterjemahkan kembali, teknik untuk melakukan kedua hal ini disebut sebagai kriptografi. Walaupun standar DICOM yang dikeluarkan oleh NEMA (National Electrical Manufacturers Association) sendiri tidak menyatakan kebutuhan akan enkripsi, sudah banyak yuridiksi yang mengatur akan kewajiban enkripsi data terutama data medis salah satunya adalah aturan oleh HIPAA (Health Insurance Portability and Accountability Act).

Algoritma yang sering digunakan pada enkripsi data diantaranya adalah AES (Advanced Encryption Standard), Triple DES, Blowfish, RSA, dan Twofish. AES merupakan salah satu algoritma yang sangat cocok untuk enkripsi file citra, termasuk file DICOM. Sebagaimana yang dinyatakan oleh Kobayashi (2009) dalam jurnalnya, AES memiliki keunggulan dalam hal keamanan yang sangat tinggi. Selain itu, menurut penelitian terbaru oleh Qamar dkk (2023), meskipun terdapat anggapan bahwa kecepatan enkripsi AES mungkin menjadi kendala, teknologi dan optimasi modern telah membuat AES menjadi lebih efisien dan cepat dalam memproses enkripsi citra. Dengan peningkatan performa ini, AES dapat menyelesaikan proses enkripsi dalam waktu yang lebih singkat, sehingga sesuai dengan kebutuhan dokter yang memiliki waktu terbatas. Keamanan dan efisiensi yang ditawarkan oleh AES menjadikannya pilihan yang sangat baik untuk enkripsi data citra dengan tingkat kerumitan dan kebutuhan keamanan yang tinggi.

Makalah ini akan melakukan perancangan perangkat lunak dengan DICOM Viewer untuk memproses file DICOM serta protokol dasar DICOM untuk transfer data dari klien ke server dengan fasilitas enkripsi dari file DICOM itu sendiri menggunakan AES dengan harapan makalah ini dapat membantu mengembangkan pilihan yang tepat dalam dunia kriptografi di bidang medis, mempercepat serta mengamankan proses transfer file DICOM.

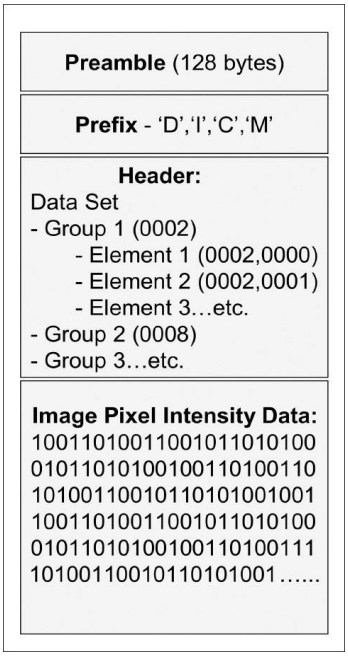
II. STUDI PENDAHULUAN

A. Standar DICOM, Format DICOM

DICOM (Digital Imaging and Communications in Medicine) adalah sebuah standar yang mencakup format data penyimpanan, protocol komunikasi, dan struktur file untuk file citra biomedis. DICOM sendiri bertujuan untuk menentukan bagaimana sebuah citra medis dapat disimpan dan dikirim, karena DICOM merupakan standar untuk seluruh citra yang

ada di rumah sakit, sumber file DICOM terdiri dari bermacam – macam modalitas rumah sakit. Modalitas rumah sakit adalah alat yang digunakan untuk membantu pasien dalam penyembuhannya, bisa berupa CT-scan, MRI, X-ray dan berbagai alat imaging lainnya.

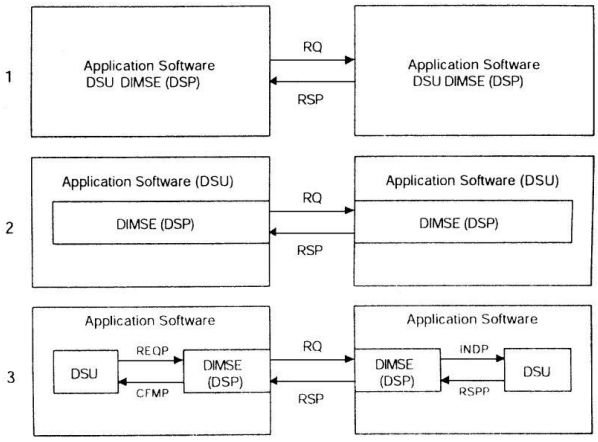
Isi dari file DICOM bukan hanya sebuah citra tetapi melainkan berupa grup informasi disimpan menjadi satu file, karena file DICOM sendiri bertujuan untuk mengubah objek di dunia nyata menjadi dunia 'DICOM' atau dengan kata lain mengubah informasi pasien menjadi data digital. File DICOM terdiri dari header yang berisikan data pasien dan data citra seperti NIK, Nama pasien, dan dimensi citra. Kemudian bagian kedua dari DICOM adalah dataset citra yakni berisi informasi tentang intensitas piksel dari citra yang berupa angka 1 dan 0.



Gambar II. 1. Struktur File DICOM

Prefix sendiri hanya terdiri dari kumpulan string DICM yang berfungsi untuk menentukan apakah file tersebut merupakan file DICOM atau bukan. Sedangkan preamble berfungsi untuk keperluan implementasi atau profil aplikasi, biasanya disediakan sebesar 128 byte, jika tidak digunakan maka preamble diset ke 00H, preamble sendiri tidak diperuntukan sebagai penentuan file tersebut merupakan file DICOM atau tidak oleh pembaca file di aplikasi terkait. Tanpa adanya header, komputer tidak bisa menentukan apakah studi sudah dilakukan atau belum dan pemilik citra dari file DICOM ini. Karena header dalam file merupakan bagian penting DICOM dan memiliki informasi privat pasien, pada saat proses publikasi header file DICOM pada bagian 0008 (informasi terkait studi) dan 0010 (informasi pasien) dihilangkan tetapi tetap mempunyai header sehingga komputer bisa membaca file DICOM tersebut.

Selain mengatur format sebuah file, DICOM juga mengatur bagaimana proses transfer file tersebut, disebut sebagai DICOM Protocol. Terdapat beberapa implementasi dari protocol DICOM, berikut merupakan beberapa aplikasi dari DICOM protocol:

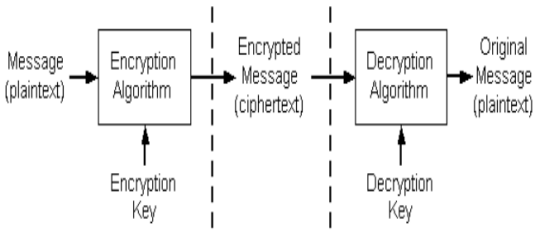


Gambar II. 2. Contoh Implementasi Protokol DICOM

Dalam skematik satu ditunjukkan bahwa perangkat lunak mengeluarkan kueri request (RQ) dan kueri respons (RSP) antara perangkat lunak satu dengan lainnya, pada skematik dua ditunjukkan bahwa DIMSE (DICOM Message Service Element) merupakan bagian terpisah dari perangkat lunak dan bertugas sebagai penghasil kueri, bukan aplikasi yang memberikan kueri secara langsung, DIMSE sendiri merupakan elemen yang bertugas dan mengatur bagaimana kueri diberikan ke perangkat lunak lain. Skematik yang ketiga memisah semua bagian komunikasi dan memiliki kuerinya sendiri – sendiri, diantara lainnya adalah REQP (Request Primitive), CFMP (Confirmation Primitive), INDP (Indication Primitive), dan RSPP (Response Primitive). Protokol DICOM dapat diintregasikan dengan TCP (Transmission Control Protocol) dan IP (Internet Protocol) yang membuat DICOM bisa berkomunikasi dengan internet serta dilengkapi dengan TLS (Transport Layer Security) agar data DICOM tetap aman.

B. Kriptografi

Kriptografi merupakan proses menyembunyikan atau mengacak informasi sehingga hanya orang yang memiliki akses akan data tersebut dapat membacanya. Kriptografi sendiri terdiri dari dua proses yakni enkripsi dan dekripsi, enkripsi merupakan proses untuk menyembunyikan informasi pada data untuk melindungi isinya, sedangkan dekripsi adalah proses mengembalikan informasi yang telah terenkripsi menjadi seperti semula agar bisa dibaca lagi. Dalam kriptografi ada yang disebut sebagai cipher, cipher sendiri merupakan algoritma yang digunakan untuk melakukan enkripsi dan dekripsi.



Gambar II. 3. Proses Kriptografi

Dalam enkripsi terdapat dua jenis tipe yakni enkripsi kunci simetris dan enkripsi kunci non simetris. Perbedaan inti dari kedua tipe enkripsi ini adalah enkripsi kunci simetris menggunakan satu kunci untuk mengenkripsi dan juga dekripsi dari suatu data sedangkan enkripsi kunci asimetris adalah proses enkripsi yang menggunakan dua kunci berbeda yakni kunci publik dan kunci privat untuk proses enkripsi dan dekripsi. Kelebihan dari proses enkripsi kunci simetris adalah prosesnya yang lebih cepat dibandingkan dengan enkripsi kunci asimetris tetapi proses enkripsi kunci simetris kurang aman dibandingkan dengan proses enkripsi kunci asimetris. Contoh dari algoritma kunci simetris adalah AES, DES, Chaotic map Algorithm, dan Blow Fish, sedangkan contoh dari algoritma kunci asimetris adalah RSA, DSA, dan Diffie-Helman.

C. AES

AES (Advanced Encryption Standard) adalah standar yang digunakan untuk algoritma kriptografi yang dikeluarkan oleh NIST (National Institute of Standards and Technology). Algoritma ini dikenal juga sebagai Rijndael, yang dikembangkan oleh dua kriptografer Belgia, Joan Daemen dan Vincent Rijmen. Algoritma ini difasilitasi oleh standar DICOM yang dikeluarkan oleh NEMA. AES merupakan algoritma yang paling sering digunakan dikarenakan melibatkan proses Round Function, dimana AES melakukan proses iterasi transformasi berulang tergantung dengan jenis key AES, yakni AES 128, AES 192, dan AES 256. Proses enkripsi AES (Rijndael) terdiri dari beberapa tahap utama, yaitu:

Key Expansion: Pada tahap ini, kunci utama yang diberikan diperluas untuk menghasilkan serangkaian subkunci yang akan digunakan pada setiap round. Jumlah subkunci yang dihasilkan tergantung pada panjang kunci yang digunakan (AES 128 menghasilkan 10 subkunci, AES 192 menghasilkan 12 subkunci, dan AES 256 menghasilkan 14 subkunci).

Initial Round: Proses enkripsi dimulai dengan tahap ini yang melibatkan penambahan kunci awal (AddRoundKey) ke blok data yang akan dienkripsi. Ini adalah operasi XOR antara blok data awal dengan kunci pertama yang dihasilkan dari key expansion.

Main Rounds: Tahap ini terdiri dari serangkaian transformasi yang dilakukan berulang kali (10 kali untuk AES 128, 12 kali untuk AES 192, dan 14 kali untuk AES 256). Setiap round terdiri dari empat langkah utama:

SubBytes: Setiap byte dalam blok data digantikan oleh byte lain menggunakan tabel substitusi (S-box) yang telah ditentukan. Langkah ini bertujuan untuk meningkatkan kompleksitas dan keamanan.
ShiftRows: Proses ini mengubah posisi baris-baris dalam blok data dengan cara menggeser byte di setiap baris ke kiri dengan offset yang berbeda. Hal ini bertujuan untuk menyebarkan byte secara lebih merata.
MixColumns: Setiap kolom dari blok data diolah secara linear untuk mencampur byte dalam setiap kolom, yang meningkatkan difusi data.
AddRoundKey: Kunci sub-round yang sesuai ditambahkan ke blok data dengan operasi XOR.

Final Round: Pada ronde terakhir, langkah MixColumns tidak dilakukan. Tahap akhir ini hanya terdiri dari langkah SubBytes, ShiftRows, dan AddRoundKey, yang memberikan sentuhan terakhir pada blok data yang dienkripsi.

Setelah melalui semua ronde, blok data yang telah dienkripsi dihasilkan sebagai output. Proses dekripsi AES (Rijndael) merupakan kebalikan dari proses enkripsi dengan menggunakan kunci yang sama, tetapi urutan langkah-langkahnya dibalik.

AES (Rijndael) menawarkan tingkat keamanan yang sangat tinggi melalui penggunaan kombinasi transformasi yang kompleks dan berulang. Inilah mengapa AES sangat populer dan menjadi standar dalam berbagai aplikasi kriptografi, termasuk dalam pengolahan dan penyimpanan file DICOM. Keandalan dan keamanan yang ditawarkan oleh AES (Rijndael) menjadikannya pilihan utama dalam melindungi data sensitif di berbagai sektor

III. METODOLOGI

Penelitian ini bertujuan untuk menguji efektivitas enkripsi file DICOM menggunakan algoritma AES melalui analisis histogram, PSNR (Peak Signal-to-Noise Ratio), dan NPCR (Number of Pixel Change Rate). Metodologi yang digunakan dalam penelitian ini mencakup beberapa tahapan sebagai berikut:

A. Persiapan Data

Data diambil dari rumah sakit Carolus di Jakarta, data CT scan otak pada pasien *stroke*, karena data hanya berupa DICOM CT scan otak saja, data dibagi menjadi tiga bagian berdasarkan ukuran file, yakni 10 MB, 15 MB dan 20 MB. Sebelum data citra diproses, dilakukan *preprocessing*, yakni seperti yang sudah dijelaskan, DICOM terdiri dari *header* yang harus dipisahkan dulu dengan file citranya dikarenakan *header* ini memiliki beberapa hal – hal privasi pasien yang tidak boleh disebarluaskan.

B. Implementasi Enkripsi AES

Kunci AES yang digunakan adalah AES-256, AES akan digunakan untuk mengekripsi file DICOM dan dilakukan di MATLAB

C. Analisis Histogram

Analisis histogram dilakukan dengan cara menampilkan histogram citra sebelum dan sesudah dienkripsi kemudian histogram tersebut akan dihitung. Bandingkan histogram dari citra asli dan terenkripsi untuk mengevaluasi distribusi pixel.

Histogram citra terenkripsi yang baik harus mendekati distribusi seragam.

D. Analisis PSNR (Peak Signal-to-Noise Ratio)

Hitung PSNR antara citra asli dan citra terenkripsi untuk menilai kualitas enkripsi. PSNR yang lebih rendah mengindikasikan enkripsi yang lebih baik karena perbedaan antara citra asli dan terenkripsi lebih besar. Dengan rumus:

$$PSNR = 10. Log_{10} (MAX^2/MSE) \quad (1)$$

Dimana MAX adalah nilai maksimel pixel dan MSE adalah mean squared error antara citra asli dan citra terenkripsi.

E. Analisis NPCR (Number of Pixel Change Rate)

Hitung NPCR untuk mengukur persentase perubahan pixel antara citra asli dan citra terenkripsi. NPCR yang tinggi menunjukkan perubahan yang signifikan di seluruh citra, yang diinginkan dalam enkripsi yang kuat, akan tetapi ketika dekripsi, NPCR haruslah kecil bahkan tidak ada dikarenakan citra harus sama dengan citra awal karena ini merupakan file medis yang tidak boleh berubah banyak

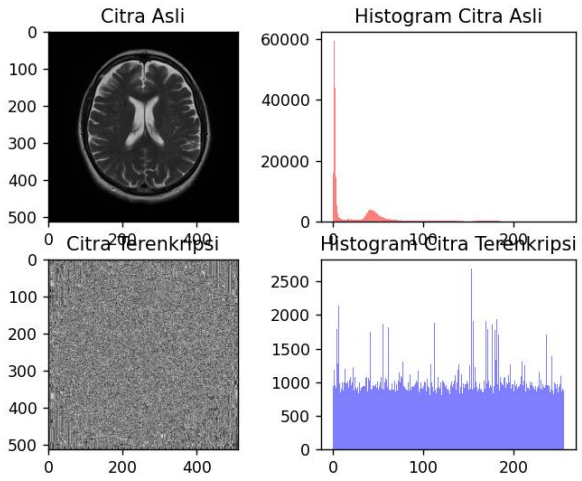
F. Source Code

Berikut adalah source code metodologi diatas:

```
from Crypto.Cipher import AES
from Crypto.Random import get_random_bytes
import numpy as np
import matplotlib.pyplot as plt

def AES_Encrypt(data, key):
    cipher = AES.new(key, AES.MODE_ECB)
    encrypted_data = cipher.encrypt(data)
    return encrypted_data

def AES_Decrypt(data, key):
    cipher = AES.new(key, AES.MODE_ECB)
    decrypted_data = cipher.decrypt(data)
    return decrypted_data

def calculate_NPCR(img1, img2):
    diff_pixels = np.sum(img1 != img2)
    total_pixels = img1.shape[0] * img1.shape[1]
    NPCR = (diff_pixels / total_pixels) * 100
    return NPCR

def calculate_PSNR(img1, img2):
    mse = np.mean((img1 - img2) ** 2)
    max_pixel = 255.0
    PSNR = 10 * np.log10((max_pixel ** 2) / mse)
    return PSNR

img = plt.imread('data2.jpg')
if img.ndim == 3:
    img = np.dot(img[...,:3], [0.2989, 0.5870, 0.1140])
img = img.astype(np.uint8)

img_vector = img.flatten()
key = get_random_bytes(32)
encrypted_img_vector = AES_Encrypt(img_vector.tobytes(), key)

encrypted_img = np.frombuffer(encrypted_img_vector, dtype=np.uint8).reshape(img.shape)

decrypted_img_vector = AES_Decrypt(encrypted_img_vector, key)
decrypted_img = np.frombuffer(decrypted_img_vector, dtype=np.uint8).reshape(img.shape)

NPCR_encrypted = calculate_NPCR(img, encrypted_img)
PSNR_encrypted = calculate_PSNR(img, encrypted_img)
NPCR_decrypted = calculate_NPCR(img, decrypted_img)
PSNR_decrypted = calculate_PSNR(img, decrypted_img)

plt.subplot(2, 2, 1)
plt.imshow(img, cmap='gray')
plt.title('Citra Asli')

plt.subplot(2, 2, 2)
plt.hist(img.flatten(), bins=256, range=[0,256], color='r', alpha=0.5)
plt.title('Histogram Citra Asli')

plt.subplot(2, 2, 3)
plt.imshow(encrypted_img, cmap='gray')
plt.title('Citra Terenkripsi')

plt.subplot(2, 2, 4)
plt.hist(encrypted_img.flatten(), bins=256, range=[0,256], color='b', alpha=0.5)
plt.title('Histogram Citra Terenkripsi')

print("NPCR (Citra Terenkripsi): {}%".format(NPCR_encrypted))
print("PSNR (Citra Terenkripsi): {} dB".format(PSNR_encrypted))
print("NPCR (Citra Terdekripsi): {}%".format(NPCR_decrypted))
print("PSNR (Citra Terdekripsi): {} dB".format(PSNR_decrypted))
plt.show()
```

IV. Analisis dan Pembahasan

Berikut adalah hasil enkripsi AES dari citra DICOM yang telah dilakukan:



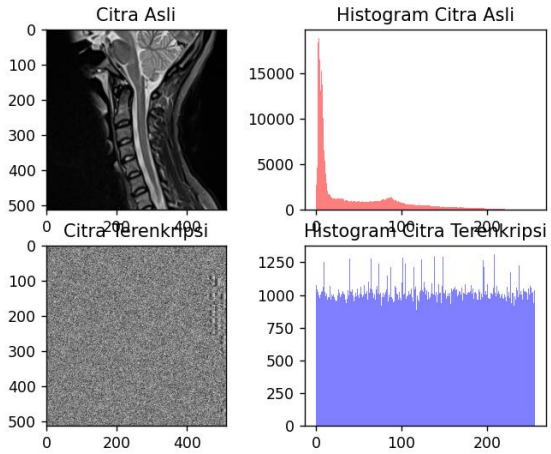
Gambar IV. 1. Hasil Enkripsi File DICOM 10MB

NPCR: 99.64141845703125%

PSNR: 27.834872079519894 dB

NPCR (Citra Terdekripsi): 0.0%

PSNR (Citra Terdekripsi): inf dB



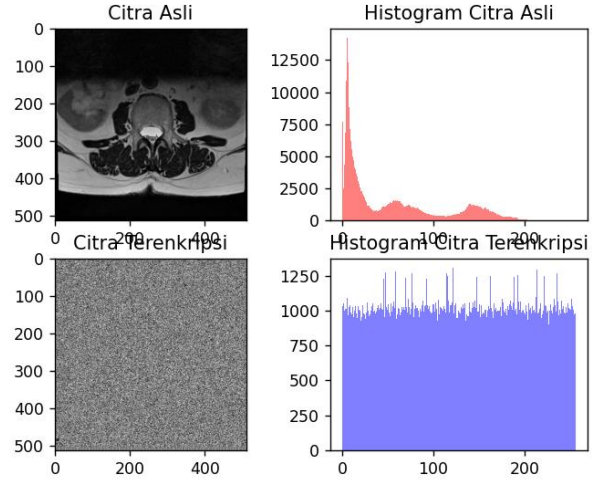
Gambar IV. 2. Hasil Enkripsi File DICOM 15MB

NPCR: 99.61433410644531%

PSNR: 27.91123803358817 dB

NPCR (Citra Terdekripsi): 0.0%

PSNR (Citra Terdekripsi): inf dB



Gambar IV. 3. Hasil Enkripsi File DICOM 20MB

NPCR: 99.61128234863281%

PSNR: 27.89079168112966 dB

NPCR (Citra Terdekripsi): 0.0%

PSNR (Citra Terdekripsi): inf dB

Dari hasil perhitungan yang telah dilakukan terhadap tiga citra DICOM yang telah dienkripsi, terdapat beberapa poin penting yang perlu diperhatikan dalam analisis:

Hasil PCR untuk ketiga citra DICOM menunjukkan persentase perubahan piksel yang signifikan setelah proses enkripsi. Secara khusus, nilai PCR mencapai sekitar 99.61% untuk semua citra, menandakan bahwa mayoritas piksel pada citra terenkripsi berbeda dengan piksel pada citra asli. Hal ini sesuai dengan ekspektasi dari proses enkripsi, di mana tujuannya adalah untuk menyembunyikan informasi asli dengan mengacak data.

PSNR adalah metrik yang mengukur kualitas rekonstruksi citra terenkripsi. Dalam hasil perhitungan, PSNR untuk ketiga citra DICOM cukup tinggi, berkisar antara 27.83 dB hingga 27.91 dB. Nilai PSNR yang tinggi menandakan bahwa citra terenkripsi masih mempertahankan kualitas informasi yang baik meskipun telah melalui proses enkripsi. Ini menunjukkan bahwa citra terenkripsi masih dapat diinterpretasikan dengan baik oleh pengguna meskipun telah diubah.

NPCR (Citra Terdekripsi) memberikan informasi tentang perubahan piksel setelah proses dekripsi. Hasil menunjukkan bahwa tidak ada perubahan pada piksel citra terdekripsi dibandingkan dengan citra asli. Ini sesuai dengan harapan, karena proses dekripsi seharusnya mengembalikan citra terenkripsi ke keadaan semula.

PSNR (Citra Terdekripsi) memiliki nilai "inf" (tak terhingga). Ini menunjukkan bahwa tidak ada perbedaan antara citra terdekripsi dan citra asli. Dalam konteks ini, ketika citra terdekripsi identik dengan citra asli, mse menjadi nol, sehingga PSNR menjadi tak terhingga.

Dalam konteks file citra medis, terutama citra DICOM, penting bagi citra terdekripsi untuk tidak berubah sepenuhnya

setelah proses dekripsi. Hal ini disebabkan oleh kebutuhan untuk memastikan bahwa citra medis yang didekripsi tetap dapat dianalisis oleh dokter dan profesional medis dengan akurasi yang tinggi. Analisis citra medis menjadi kritis dalam diagnosis, perencanaan pengobatan, dan pemantauan pasien.

Jika citra terdekripsi mengalami perubahan yang signifikan atau kehilangan informasi esensial selama proses enkripsi dan dekripsi, hal ini dapat mengakibatkan kesalahan dalam interpretasi citra dan penilaian kondisi pasien. Oleh karena itu, meskipun proses enkripsi diperlukan untuk menjaga keamanan dan privasi data medis, tetapi penting juga untuk memastikan bahwa proses dekripsi tidak merusak atau mengubah informasi penting dalam citra medis.

Dengan demikian, hasil analisis yang menunjukkan bahwa citra terdekripsi memiliki NPCR yang mendekati nol dan PSNR yang tinggi mengindikasikan bahwa citra terdekripsi berhasil mempertahankan integritas informasi dan dapat dipertimbangkan oleh dokter untuk keperluan diagnosis dan penanganan pasien secara akurat.

Berdasarkan hasil analisis yang telah dilakukan terhadap tiga citra DICOM yang dienkripsi, diperoleh kesimpulan bahwa proses enkripsi mempengaruhi perubahan pada citra terenkripsi. Hasil PCR yang tinggi menunjukkan bahwa mayoritas piksel pada citra terenkripsi mengalami perubahan signifikan, sejalan dengan tujuan enkripsi untuk menyembunyikan informasi asli dengan mengacak data. Meskipun mengalami enkripsi, citra terenkripsi masih mempertahankan kualitas informasi yang baik, seperti yang ditunjukkan oleh nilai PSNR yang tinggi. Ini menandakan bahwa citra terenkripsi masih dapat diinterpretasikan dengan baik oleh pengguna, meskipun telah melalui proses enkripsi.

Pentingnya memastikan integritas informasi pada citra terdekripsi juga disoroti dalam analisis ini. Hasil NPCR (Citra Terdekripsi) mendekati nol, menunjukkan bahwa tidak ada perubahan yang signifikan pada piksel citra terdekripsi setelah proses dekripsi. PSNR (Citra Terdekripsi) memiliki nilai tak terhingga, menandakan bahwa citra terdekripsi identik dengan citra asli. Hal ini penting dalam konteks citra medis, di mana citra terdekripsi harus mempertahankan integritas informasi agar tetap dapat dianalisis oleh dokter dan profesional medis dengan akurasi yang tinggi.

Dengan demikian, meskipun telah mengalami proses enkripsi, citra terdekripsi masih dapat digunakan untuk keperluan diagnosis dan penanganan pasien secara akurat. Ini menunjukkan bahwa metode enkripsi yang digunakan berhasil mengamankan data citra medis tanpa mengorbankan kualitas informasi.

REFERENCES

[1] Annadurai, Kannammal & Rani, S.. (2012). DICOM image authentication and encryption based on RSA and AES algorithms. 10.1007/978-3-642-35197-6_39.

[2] Kobayashi LO, Furuie SS. Proposal for DICOM multiframe medical image integrity and authenticity. J Digit Imaging. 2009 Mar;22(1):71-83. doi: 10.1007/s10278-008-9103-6.

[3] Natsheh, Q.; Sălăgean, A.; Zhou, D.; Edirisinghe, E. Automatic Selective Encryption of DICOM Images. Appl. Sci. 2023, 13, 4779. https://doi.org/10.3390/app13084779

[4] Kannammal, A., Subha Rani, S. (2012). DICOM Image Authentication and Encryption Based on RSA and AES Algorithms. In: Ponnambalam, S.G., Parkkinen, J., Ramanathan, K.C. (eds) Trends in Intelligent Robotics, Automation, and Manufacturing. IRAM 2012. Communications in Computer and Information Science, vol 330. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-35197-6_39R.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 12 Juni 2024

Aloysius Arya Wibisono 18320025