JURNAL TEKNIK INFORMATIKA, Vol.4 No.1 Febuari 2025, Halaman 197~207
E-ISSN: 2828-4445

ANALISIS PERFORMA ALGORITMA RSA DAN AES DALAM ENKRIPSI GAMBAR DIGITAL

PERFORMANCE ANALYSIS OF RSA AND AES ALGORITHMS IN DIGITAL IMAGE ENCRYPTION

Maulana Hadi Prasetyo[1], Harminto Mulyo [2*], Teguh Tamrin [3]

[1,2,3]Universitas Islam Nahdlatul Ulama Jepara
Email : [1*]201240001037@unisnu.ac.id, minto@unisnu.ac.id, teguh@unisnu.ac.id

Abstrak

Gambar digital merupakan aspek penting yang perlu diamati dalam hal keamanan. Selain digunakan sebagai media penyimpanan dan berbagi informasi, gambar digital seringkali mengandung data penting yang harus dijaga kerahasiaannya. Untuk melindungi gambar digital, penggunaan kriptografi menjadi metode yang efektif dengan mengenkripsi gambar tersebut. Dalam domain kriptografi, terdapat dua jenis algoritma utama: simetris dan asimetris. Dari berbagai algoritma yang ada, RSA dan AES merupakan dua algoritma yang dianggap unggul di bidangnya. Namun, pemilihan algoritma kriptografi tanpa mempertimbangkan performanya dapat menurunkan tingkat keamanan gambar serta kualitas gambar itu sendiri, bahkan mengurangi efektivitas sistem yang mengimplementasikan algoritma tersebut. Oleh karena itu, penelitian dilakukan untuk membandingkan performa algoritma RSA dan AES dalam mengenkripsi gambar digital. Penelitian ini melibatkan pengujian pada 16 gambar dari sistem presensi SIJUNA yang dienkripsi untuk mengevaluasi performa kedua algoritma. Hasil penelitian menunjukkan bahwa algoritma RSA memiliki keunggulan dalam hal waktu enkripsi gambar dan ukuran gambar setelah dienkripsi. Di sisi lain, algoritma AES lebih unggul dalam menjaga keamanan informasi pada gambar serta menjaga kualitas gambar setelah proses dekripsi. Dengan demikian, pemilihan antara RSA dan AES perlu dipertimbangkan dengan cermat sesuai dengan kebutuhan keamanan dan kualitas gambar yang diinginkan.

Kata Kunci: Analisis Performa Algoritma,Gambar Digital, Algoritma RSA, Algoritma AES

Abstract

Ensuring the security of digital images is crucial and requires careful attention. These images serve not only as a means of storing information but also as a way of sharing it, containing valuable data that must be safeguarded. One effective method to secure digital images is by encrypting them using cryptography. Within the realm of cryptography, there are two main types of algorithms: symmetric and asymmetric. Among the various algorithms available, RSA and AES stand out as superior algorithms in their respective fields. However, selecting a cryptographic algorithm without considering its performance can compromise the security level and quality of the images, even reducing the effectiveness of the system implementing the algorithm. Hence, research has been conducted to compare the performance of RSA and AES algorithms in encrypting digital images. This study involved testing 16 images obtained from the SIJUNA attendance system, which were encrypted to evaluate the performance of both algorithms. The research findings indicate that the RSA algorithm excels in terms of image encryption time and the size of images after encryption. On the other hand, the AES algorithm outperforms in maintaining information security within the images and preserving image quality after decryption. Therefore, the choice between RSA and AES should be carefully considered based on the desired security and image quality requirements.

***Keywords**: Algorithm Performance Analysis, Digital Image, RSA Algorithm, AES Algorithm*

PENDAHULUAN

Gambar digital menjadi media salah satu media berbagi informasi yang mempunyai peran yang penting. Gambar digital ini selain digunakan sebagai media penyimpanan informasi juga digunakan sebagai media pengiriman data (Engko M et al., 2022). Hal ini tentu membuat keamanan dari gambar digital tersebut menjadi perhatian tersendiri mengingat gambar digital tersebut mengandung berbagai informasi penting yang bersifat privasi. Salah satu cara untuk mengamankan gambar tersebut yaitu dengan mengenkripsinya menggunakan teknik kriptografi (Pruett & Babb, 2023).

Pada kriptografi terdapat berbagai algoritma yang dapat digunakan untuk mengamankan gambar tersebut. Secara umum algoritma pada kriptografi dapat dikelompokkan menjadi 2 yaitu algoritma simetris dan asimetris. Algoritma simetris yaitu algoritma yang menggunakan 1 kunci untuk mengenkripsi dan mendekripsi data, sedangkan algoritma asimetris yaitu algoritma yang menggunakan 2 kunci berbeda dimana 1 kunci digunakan untuk mengenkripsi data dan 1 kunci lainnya digunakan untuk mendekripsi data. Dari dua jenis algoritma tersebut terdapat 2 algoritma yang lebih unggul daripada algoritma lain sejenisnya yaitu algoritma RSA dan AES. Algoritma RSA lebih unggul daripada algoritma asimetris lainnya seperti El-Gammal dan ECC. Algoritma RSA ini mempunyai waktu enkripsi sekitar 0,1366 - 0.6250 detik sementara algoritma *El-Gammal* memerlukan waktu 7 detik (Fajrin et al., 2023) dan algoritma ECC memerlukan Waktu 0.2969 - 15.0882 detik (Adrian et al., 2022) untuk mengenkripsi data. Pada algoritma simetris algoritma AES juga memiliki keunggulan dari segi waktu enkripsi dari algoritma sejenisnya seperti algoritma DES, dan *Triple* DES, *Twofish* (Nino, 2023). Algoritma AES mempunyai waktu enkripsi 0,06 sampai 374 detik sedangkan algoritma *Twofish* memerlukan waktu enkripsi 1,34 detik untuk mengenkripsi data sederhana seperti nomor NIK. Sedangkan untuk algoritma DES dan *Triple* DES memerlukan waktu 389 dan 452 detik. Namun antara algoritma RSA dan AES tersebut memiliki performa yang berbeda antara satu sama lain. Jika algoritma RSA maupun AES tersebut digunakan tanpa memperhatikan performa dari algoritma itu selain dapat mengurangi keamanan dan kualitas gambar yang ada hal tersebut juga dapat mengurangi keefektifan dari sistem yang menerapkan algoritma tersebut.

Berbagai penelitian telah dilakukan untuk mengetahui performa dari algoritma RSA dan AES sebelumnya. pada penelitian yang dilakukan oleh Alif Khamsyar dan Muh. Basri membuktikan bahwa algoritma RSA mampu digunakan untuk mengenkripsi gambar dengan baik. Berdasarkan pengujian yang dilakukan pada 5 gambar, menunjukkan bahwa gambar yang dienkripsi menggunakan algoritma RSA mengalami perubahan ukuran namun pada penelitian tersebut tidak diukur seberapa jauh ukuran gambar mengalami perubahan setelah dienkripsi menggunakan algoritma RSA. Sedangkan dari segi waktu enkripsi menunjukkan bahwa waktu enkripsi dari algoritma RSA tergantung dari ukuran gambar yang dienkripsi, semakin besar gambar yang yang dienkripsi maka semakin lama proses enkripsi dari gambar tersebut (Khamsyar & Basri, 2022). Selain algoritma RSA algoritma AES juga bisa digunakan untuk mengenkripsi gambar digital, hal ini dibuktikan pada penelitian yang dilakukan oleh Angga Aditya Permana dan Luigi Ajeng Pratiwi. Pada penelitian tersebut dilakukan pengujian menggunakan algoritma AES untuk mengenkripsi gambar digital sebagai solusi untuk meningkatkan keamanan dari gambar digital tersebut. Pengujian dilakukan dengan mengacak *pixel* dari gambar tersebut menggunakan algoritma AES sehingga gambar yang terenkripsi tidak bisa dibuka dan mengalami *corrupt*. Dari pengujian yang dilakukan menunjukkan bahwa algoritma AES juga bisa digunakan untuk mengenkripsi gambar digital (Permana & Pratiwi, 2022). Penelitian lainnya yang dilakukan oleh Dalia Mubarak Alsaffar,DKK, menunjukkan bahwa algoritma AES lebih unggul dari algoritma RSA jika dilihat dari kualitas enkripsi yang dihasilkan. Pada penelitian tersebut menggunakan 3 gambar berbeda untuk menguji kualitas enkripsi dari masing-masing algoritma. Masing-masing gambar kemudian dienkripsi menggunakan masing-masing algoritma RSA dan AES untuk kemudian dihitung hubungan antara algoritma dan kualitas enkripsi menggunakan metode koefisien korelasi, semakin mendekati nol nilai yang didapat menandakan semakin baik enkripsi yang terjadi. Dari hasil perhitungan algoritma RSA mampu menghasilkan nilai korelasi sebesar -0,0280 sedangkan algoritma AES mampu menghasilkan nilai korelasi sebesar -0,084. Hal ini menandakan bahwa algoritma AES mempunyai kualitas enkripsi yang lebih baik dari algoritma RSA (Alsaffar et al., 2020).

Dari beberapa penelitian yang telah dilakukan dapat diketahui bahwa baik algoritma RSA maupun AES dapat digunakan untuk mengamankan gambar digital. Namun dari penelitian sebelumnya masih belum diketahui dengan jelas terkait performa dari algoritma tersebut. Oleh karena itu perlu dilakukan penelitian lebih lanjut untuk meneliti performa dari algoritma RSA dan AES secara lebih lengkap.

Gambar digital dengan Tingkat kepadatan *pixel* berbeda digunakan untuk mengukur performa algoritma yang lebih akurat. Performa yang diukur meliputi pengukuran waktu enkripsi, ukuran gambar setelah dienkripsi, kualitas enkripsi, dan kualitas gambar setelah didekripsi.

1. METODE PENELITIAN

A. Pengumpulan data gambar

Pengumpulan data gambar dilakukan Selama 1 bulan menggunakan metode *experimental*. Metode *experimental* merupakan metode untuk mengumpulkan data dengan cara mengumpulkan data secara langsung dari sistem presensi yang berjalan.

B. Proses enkripsi algoritma RSA dan AES

Data berupa gambar yang terkumpul kemudian digunakan untuk menguji performa dari masing-masing algoritma RSA dan AES. Proses pengujian dilakukan dengan mengenkripsi gambar tersebut menggunakan masing-masing algoritma RSA dan AES untuk kemudian hasil enkripsi digunakan untuk mengukur performa dari algoritma tersebut berdasarkan segi kecepatan enkripsi, ukuran gambar setelah dienkripsi dan tingkat keamanan informasi dari gambar yang telah dienkripsi.

C. Penghitungan waktu enkripsi

Setelah tahap enkripsi, tahap selanjutnya yaitu mengukur waktu enkripsi yang diperlukan oleh algoritma untuk mengenkripsi gambar. pengukuran waktu enkripsi bertujuan untuk mengetahui algoritma yang memiliki waktu enkripsi tercepat saat mengenkripsi gambar dengan tingkat kerapatan *pixel* tertentu. waktu enkripsi diukur dengan mengurangi selisih antara waktu awal dan waktu akhir dari proses enkripsi. Dalam penelitian ini untuk mengukur waktu enkripsi dari algoritma RSA dan AES dilakukan menggunakan modul *time* yang merupakan salah satu modul bawaan dari *python* yang memiliki kegunaan untuk menghitung waktu saat ini (*Time - Time Access and Conversions*, 2024).

D. Penghitungan ukuran gambar

Tahap berikutnya yaitu tahap penghitungan dari ukuran gambar setelah dienkripsi. Penghitungan ukuran gambar dilakukan untuk mengukur seberapa jauh ukuran gambar dengan tingkat kepadatan *pixel* tertentu mengalami perubahan setelah dienkripsi menggunakan algoritma RSA dan AES. Proses penghitungan ukuran dilakukan menggunakan fungsi *getSize* yang merupakan bagian dari modul *OS* untuk mengetahui ukuran file gambar sebelum dan setelah dienkripsi. Kemudian hasil pengukuran dari gambar sebelum dienkripsi dibandingkan dengan ukuran gambar setelah dienkripsi untuk mengetahui tingkat perubahan file yang terjadi. Selanjutnya hasil pembandingan oleh masing masing algoritma tersebut kemudian dibandingkan satu sama lain untuk mengetahui algoritma yang memiliki tingkat perubahan file terkecil pada gambar dengan tingkat kerapatan *pixel* tertentu (*Os - Miscellaneous Operating System Interfaces*, 2024).

E. Pengukuran tingkat keamanan gambar

Selanjutnya dilakukan pengukuran terkait tingkat keamanan gambar pada gambar setelah mengalami proses enkripsi menggunakan masing-masing algoritma RSA dan AES. Pengukuran dilakukan dengan menghitung nilai *matrix* MSE (Mean Squared Error) pada gambar asli dengan gambar setelah dienkripsi. Dari hasil pengukuran akan didapat nilai dari *matrix* MSE Dimana semakin besar nilai *matrix* MSE yang diperoleh berarti semakin teracak pixel yang ada pada gambar setelah dienkripsi. Hal ini berarti semakin baik keamanan informasi yang ada pada gambar yang terenkripsi tersebut.

F. Proses dekripsi algoritma RSA dan AES

Pada tahap ini gambar yang telah dienkripsi menggunakan masing-masing algoritma didekripsi kembali menggunakan algoritma RSA dan AES. Proses dekripsi dilakukan dengan tujuan selain

agar gambar yang dienkripsi menjadi bisa dibaca kembali juga agar dapat dilakukan pengukuran terkait tingkat kualitas gambar yang dihasilkan pada gambar setelah didekripsi.

G. Penghitungan kualitas gambar setelah didekripsi

Setelah gambar didekripsi kemudian dilakukan perhitungan kualitas gambar. proses perhitungan ini dilakukan dengan menggunakan *matrix* PSNR untuk menghitung tingkat kualitas gambar yang dihasilkan dari gambar yang telah didekripsi dengan gambar aslinya. semakin tinggi nilai yang dihasilkan maka semakin sedikit perubahan *pixel* yang terjadi pada gambar setelah didekripsi. Hal ini menandakan semakin baik algoritma tersebut dalam menjaga kualitas gambar setelah mengalami proses enkripsi dan dekripsi (Mido & Ujianto, 2022).

H. Membandingkan performa yang diperoleh dari algoritma RSA dan AES.

Tahap terakhir yaitu menganalisis data yang diperoleh, proses analisis menggunakan metode analisis dengan cara membandingkan hasil yang diperoleh oleh masing masing algoritma (Loanardo et al., 2022), hasil pengujian yang mencangkup waktu enkripsi, ukuran gambar setelah dienkripsi, tingkat keamanan informasi pada gambar yang terenkripsi, dan kualitas gambar pada gambar yang didekripsi tersebut kemudian dibandingkan antara satu sama lain menggunakan metode analisis deskriptif, Dimana hasil pengujian ditampilkan dalam tabel dan dikelompokkan berdasarkan tingkat kerapatan *pixel* yang dimiliki oleh gambar tersebut.

2. HASIL DAN PEMBAHASAN

A. Hasil

Berdasarkan hasil pengujian yang dilakukan terhadap waktu enkripsi, ukuran gambar, kualitas gambar, dan tingkat keamanan informasi pada gambar oleh algoritma RSA dan AES, terdapat perbedaan signifikan antara waktu enkripsi algoritma RSA dan AES. Algoritma AES memiliki waktu enkripsi lebih lama dibandingkan RSA. AES membutuhkan 12,71 detik untuk file 1,2 Kb dengan waktu rata-rata 3681,40 detik, sementara RSA hanya 0,54 detik dan 8,81 detik. RSA juga unggul dalam ukuran gambar setelah enkripsi, dengan peningkatan sekitar 1,6%, sedangkan AES sekitar 7,5%. Dalam keamanan informasi, RSA lebih baik, dengan nilai MSE yang lebih rendah. Namun, AES lebih baik dalam menjaga kualitas gambar setelah dekripsi. Pada gambar PNG, keduanya menghasilkan gambar yang sama setelah dekripsi.

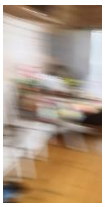
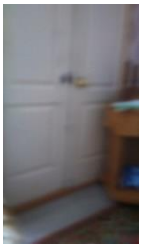
B. Pembahasan

1. Pengumpulan Data

Data yang digunakan pada penelitian diperoleh dari SIJUNA yang merupakan sistem presensi mengajar guru SMKN 1 Bangsri. Proses Pengumpulan data dilakukan dengan menggunakan metode experimental dengan mengumpulkan data secara langsung dari sistem yang berjalan. Dari hasil pengumpulan data diperoleh 16 gambar digital berupa foto presensi, sebagai pembanding ditambah 9 data gambar tambahan yang didapat dari dataset *kaggle*. Untuk memudahkan dalam pengujian algoritma nama file dari gambar yang diperoleh di ubah terlebih dahulu menjadi tesImage1, tesImage2 dan seterusnya sampai tesImage25. Berikut Sebagian data gambar yang telah terkumpul.

Tabel 3. 1 Sebagian Gambar yang akan digunakan

No	Nama File	Gambar	Format Gambar	Kepadatan Pixel (PPI)

1	tesImage3		JPG	3,885683481
2	tesImage4		JPG	3,885683481
3	tesImage17		PNG	3,607054016
4	tesImage18		PNG	4,169349803
5	tesImage19		JPG	5,269156418
6	tesImage20		JPG	4,733622303

7	tesImage22		JPG	3,592102448
8	tesImage23		JPG	3,927817079
9	tesImage24		JPG	4,186993457

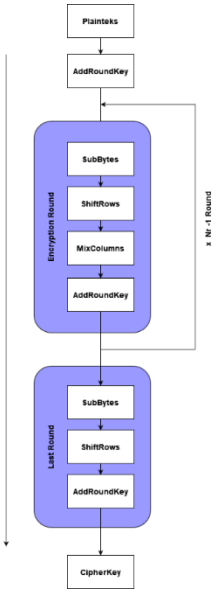
2. Proses enkripsi

Proses enkripsi pada penelitian ini dilakukan secara terpisah menggunakan masing - masing algoritma. Proses enkripsi algoritma RSA dimulai dengan pembangkitan kunci untuk nantinya 1 kunci digunakan untuk mengenkripsi gambar dan 1 kunci lainnya digunakan untuk mendekripsi. Proses pembangkitan kunci dimulai dengan memilih dua bilangan prima besar secara acak, yang akan disebut sebagai *p* dan *q*. Kedua bilangan prima ini sangat penting karena mereka akan digunakan untuk menghasilkan kunci. Selanjutnya, nilai *modulus (n)* dihitung dengan mengalikan *p* dan *q*. Setelah mendapatkan nilai *modulus (n)*, langkah berikutnya adalah menghitung nilai *totient* dari *n (Θn)*. *Totient* dari *n (Θn)* adalah jumlah bilangan bulat positif yang lebih kecil dari *n* dan relatif prima dengan *n*. Dalam kasus RSA, *Θn* dihitung sebagai *(p-1)(q-1)*. Setelah mendapatkan nilai *totient (Θn)*, kita perlu memilih bilangan bulat *e* yang merupakan *eksponen* enkripsi. Bilangan e haruslah relatif prima dengan *Θn*, yang berarti *e* dan *Θn* tidak memiliki faktor yang sama selain 1. Biasanya, bilangan *e* yang sering digunakan adalah bilangan prima kecil, seperti 65537, karena faktorisasinya mudah dan efisien dalam perhitungan. Setelah memilih bilangan *e*, langkah terakhir adalah menemukan nilai *d*, yang merupakan eksponen dekripsi. Nilai *d* dipilih sedemikian rupa sehingga *(e * d) % Θn = 1*. Dalam kata lain, nilai *d* adalah *invers modular* dari *e modulo Θn*. Ini dapat dihitung menggunakan algoritma *Extended Euclidean* atau algoritma pencarian *invers modular* lainnya.

Setelah proses pembangkitan kunci selesai, pasangan kunci *(e, n)* akan digunakan sebagai kunci publik, yang akan digunakan untuk mengenrkispi gambar. Sementara itu, pasangan kunci *(d, n)* akan menjadi kunci privat, yang digunakan untuk mendekripsi gambar. Dalam proses enkripsi, gambar digital kemudian dikonversi penjadi *pixel* RGB untuk kemudian dilakukan enkripsi pada masing-masing *pixel* RGB tersebut, untuk proses enkripsi sendiri dimulai memangkatkan nilai RGB tersebut dengan *eksponen* enkripsi (e) dan diambil *modulo* n. Hasilnya adalah *pixel* RGB yang terenkripsi. *Pixel* RGB tersebut kemudian dikonversi menjadi gambar yang terenkripsi.

Sementara itu untuk proses enkripsi algoritma AES dilakukan dengan melalui beberapa tahapan yang dimulai dengan pembangkitan kunci dan dilanjut dengan proses enkripsi. Proses enkripsi algoritma AES terdiri dari beberapa tahapan yang diulang selama beberapa putaran tergantung dari jenis algoritma AES yang dipakai, pada penelitian ini menggunakan algoritma AES-128 yang menggunakan kunci 128 bit dan menggunakan 10 *round* dalam proses enkripsi maupun dekripsinya. Sebelum mulai enkripsi pertama gambar yang akan dienkripsi dikonversi dulu menjadi

pixel RGB gambar, kemudian *pixel* gambar yang ada dikonversi menjadi kumpulan *matrix* 4x4, baru kemudian dilakukan enkripsi per *matrix* yang didapat.



Gambar 3. 1 Alur Enkripsi Algoritma AES

Untuk proses enkripsi sendiri dimulai dengan menghitung *XOR* dari *plainteks* dengan *roundkey* ke-0, hasilnya kemudian digunakan untuk menghitung nilai *subbytes subbytes* merupakan proses mengganti setiap elemen pada blok dengan elemen dari tabel *S-Box*.

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	xe	xf
0x	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1x	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2x	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3x	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4x	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5x	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6x	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7x	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8x	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9x	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
ax	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
bx	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
cx	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
dx	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
ex	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
fx	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Gambar 3. 2 Tabel S-Box

Selanjutnya dilakukan pergeseran baris yang disebut dengan proses *Shiftrows*. Pada proses ini baris pada *matrix* digeser ke kiri berdasarkan aturan yang ada. Baris pertama tidak mengalami pergeseran. Baris ke 2 digeser 1 kali ke kiri, baris ke 3 digeser 2 kali ke kiri, dan baris ke 4 digeser 3 kali ke kiri. Hasil dari *Shiftrows* kemudian di kalikan dengan *matrix State* yang ada. Tahap ini dinamakan *MixColumn.*

02	03	01	01
01	02	03	01
01	01	02	03
03	01	01	02

Matriks

MixColumns

Gambar 3. 3 Matrix MixColumns

Matrix yang diperoleh kemudian digunakan untuk menghitung *addRoundKey* dengan cara menghitung XOR dari *matrix* yang diperoleh dengan *matrix roundkey* pada *round* ini, keempat tahapan mulai dari *shiftRow* hingga *addRoundKey* diulangi sebanyak 10 kali *round* , namun pada *round* terakhir tidak dilakukan perhitungan *MixColumn.* Setelah semua *matrix* sudah dienkripsi kemudian dikonversi menjadi *pixel* RGB baru kemudian dikonversi menjadi gambar yang terenkripsi.

3. Penghitungan Waktu Enkripsi

Berdasarkan pengukuran kecepatan enkripsi masing-masing algoritma terhadap 25 gambar digital yang ada menunjukkan terdapat perbedaan yang signifikan antara waktu enkripsi diperlukan algoritma RSA dengan waktu enkrispi algoritma AES dalam mengenkripsi gambar digital. Berdasarkan pengujian waktu enkripsi yang didapat, algoritma RSA menghasilkan waktu enkripsi yang lebih cepat daripada algoritma AES.

Untuk mengenkripsi gambar dengan tingkat kepadatan gambar sebesar 3,59 - 3,92 PPI algoritma RSA hanya memerlukan waktu sekitar 4 detik, untuk mengenkripsi gambar dengan kepadatan *pixel* 4,16 - 4,73 PPI membutuhkan rata-rata waktu 22,79 detik untuk mengenkripsi gambar, dam untuk mengenkripsi gambar digital dengan tingkat kepadatan *pixel* 5,26 - 5,37 PPI memerlukan waktu rata-rata 33,20 untuk mengenkripsi gambar. Kemudian pada algoritma AES mampu mengenkripsi gambar dengan kepadatan *pixel* 3,59 - 3,92 PPI dalam 2858,25 detik. Kemudian untuk mengamankan data dengan tingkat kepadatan 4,16 - 4,73 PPI algoritma AES memerlukan 6781,59 detik untuk mengenkripsi gambar tersebut, dan membutuhkan 7262,58 detik untuk mengenkripsi gambar dengan kerapatan *pixel* sebesar 5,26 - 5,37. Selain itu dapat disimpulkan bahwa tingkat kepadatan gambar maupun ukuran gambar dapat mempengaruhi waktu enkripsi dari algoritma tersebut, semakin padat *pixel* dalam gambar itu maka akan semakin lama waktu yang diperlukan algoritma itu untuk mengenkripsi gambar tersebut.

4. Penghitungan Ukuran Gambar

Perbedaan yang signifikan juga terjadi pada penghitungan ukuran gambar digital setelah dienkripsi. Dari hasil pengujian, setelah masing-masing gambar tersebut dienkripsi ukuran dari gambar mengalami perubahan berupa kenaikan ukuran dari gambar yang dienkripsi, setelah gambar dienkripsi menggunakan algoritma RSA ukuran gambar dengan tingkat kerapatan *pixel* 3,59 – 3,92 PPI yang sebelumnya berukuran antara 1,28 - 419 Kb naik menjadi 1,28 – 688 Kb, pada gambar dengan kerapatan *pixel* 4,16 – 4,73 PPI mengalami kenaikan menjadi 160 – 1214 Kb dari yang sebelumnya berukuran 60 – 615 Kb. Dan gambar dengan tingkat kerapatan *pixel* 5,26 - 5,37 PPI naik dari yang sebelumnya 387 – 532 Kb menjadi 1211-1151 Kb. Hal yang serupa juga terjadi pada gambar yang dienkripsi menggunakan algoritma AES, gambar dengan kerapatan *pixel* 3,59 – 3,92 PPI naik menjadi 1,80 – 739 Kb, sedangkan gambar dengan kerapatan *pixel* 4,16 – 4,73 PPI mengalami kenaikan 0,52% sampai 0,98% dari ukuran semula, namun ada juga yang mengalami penurunan sebesar 0,26 % dari ukuran semula. Penurunan ukuran juga terjadi pada gambar dengan kepadatan *pixel* 5,26 – 5,37 PPI yang dienkripsi menggunakan algoritma AES, Dimana ukuran gambar yang dienkripsi menurun hingga 0,51% dari gambar aslinya. Dari hasil pengujian

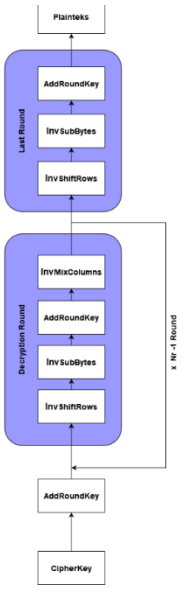
yang dilakukan diperoleh bahwa algoritma RSA dapat bekerja dengan baik apabila digunakan untuk mengenkripsi gambar digital dengan tingkat kepadatan *pixel* antara 3,59 sampai 3,92 PPI sedangkan algoritma AES cocok untuk mengenkripsi gambar dengan tingkat kepadatan *pixel* antara 4,16 sampai 5,37 PPI.

5. Pengukuran Tingkat Keamanan Gambar

Pada pengukuran tingkat keamanan informasi pada gambar yang dienkripsi menggunakan masing-masing algoritma RSA dan AES menunjukkan bahwa Algoritma AES lebih unggul dalam mengamankan informasi yang terdapat dalam gambar yang telah dienkripsi dari pada algoritma RSA. Hal ini dapat dilihat berdasarkan nilai MSE yang diperoleh. Berdasarkan hasil pengukuran terhadap nilai MSE pada gambar yang telah dienkripsi dengan masing-masing algoritma RSA dan AES, menunjukkan bahwa nilai dari *matrix* MSE pada algoritma RSA mampu menghasilkan nilai MSE hingga 36580,32 pada gambar dengan Tingkat kepadatan *pixel* sebesar 3,59 hingga 3,92 PPI, Selain itu algoritma ini mampu menghasilkan nilai MSE hingga 24049,57 pada gambar dengan Tingkat kepadatan *pixel* 4,16 - 4,7 PPI dan pada gambar dengan kepadatan *pixel* 5,26 - 5,37 PPI, algoritma ini mampu menghasilkan nilai MSE hingga 22156,63. Dilain sisi algoritma AES mampu menghasilkan nilai MSE 46030,59 pada Tingkat kepadatan *pixel* 3,59 - 3,92 PPI, untuk gambar dengan Tingkat kepadatan *pixel* 4,16 - 4,7 PPI algoritma ini mampu menghasilkan nilai MSE hingga 26573,01 dan pada gambar dengan Tingkat kepadatan *pixel* 5,26 - 5,37 PPI algoritma AES dapat menghasilkan nilai MSE hingga 22763,34.

6. Proses Dekripsi

Sama dengan proses enkripsi, proses dekripsi yang dilakukan juga dilakukan secara terpisah. Dimana 1 gambar digital yang telah dienkripsi akan didekripsi dengan algoritma yang sama saat proses enkripsi. Algoritma RSA menggunakan kunci *privat* yang telah dibuat pada proses enkripsi sebelumnya. Gambar yang telah dienkripsi dikonversi menjadi *pixel* RGB baru kemudian dilakukan enkripsi *perpixel* RGB menggunakan algoritma RSA, *pixel* RGB dipangkatkan dengan nilai *d* dan dihitung *mod n* dari kunci *public*. Selanjutnya *pixel* RGB dikonversi kembali menjadi gambar digital.



Gambar 3. 4 Alur Dekripsi Algoritma AES

Pada proses dekripsi algoritma AES dilakukan dengan membalik Langkah pada proses enkripsi. Gambar yang terenkripsi dikonversi menjadi *pixel* RGB dan dikonversi lagi menjadi *matrix* 4x4. Selanjutnya proses dekripsi dimulai dari menghitung *addRoundKey* dengan menghitung XOR dari *cipherkey* dengan *roundKey* 10. Kemudian dilakukan perhitungan *Inverse ShiftRows* yaitu proses menggeser baris kearah kanan sesuai aturan yang berlaku, pada baris pertama tidak digeser, pada baris ke dua digeser 1 kali ke kanan, pada baris ke tiga digeser 2 kali ke kanan, dan pada baris ke empat digeser 3 kali kearah kanan. Tahapan selanjutnya yaitu tahap *inverse Subbytes* dengan cara

menukarkan masing-masing elemen pada *matrix* dengan nilai dari *S-box* secara terbalik. Hasil yang diperoleh kemudian *dixorkan* dengan *roundKey* pada *round* itu secara terbalik, untuk *round* ke satu menggunakan *roundkey* ke Sembilan untuk *round* ke dua menggunakan *roundKey* ke delapan, untuk *round* ke 3 menggunakan *roundKey* ke 7 dan Seterusnya hingga *round* ke 10 yang menggunakan *roundKey* 0. Selanjutnya hasil dari *addRoundKey* kemudian digunakan untuk menghitung *inverse MixColumns*, proses ini dilakukan dengan mengalikannya dengan *matrix inverse mixColumns*.

0E	0B	0D	09
09	0E	0B	0D
0D	09	0E	0B
0B	0D	09	0E

Matriks Inverse MixColumns

Proses *Inverse Shiftrow* sampai dengan *inverse MixColumns* kemudian dilakukan kembali hingga mencapai 1 *round* sebelum *round* terakhir, pada *round* terakhir prosesnya sama namun pada *round* ini tidak dilakukan *inverse mixColumns*. Kumpulan *matrix* yang telah didekripsi kemudian dikonversi ke *pixel* RGB dan dikonversi lagi menjadi gambar yang telah terdekripsi.

7. Penghitungan Kualitas Gambar

Dari hasil pengujian terhadap gambar digital yang ada, algoritma AES menghasilkan nilai PSNR antara 40,81 *dB* sampai 48,41 *dB* sedangkan algoritma RSA menghasilkan nilai PSNR antara 30,82 *dB* sampai 41,50 *dB*. Pada gambar yang memiliki kepadatan *pixel* 3,59 PPI hingga 3,92 PPI algoritma AES mampu menghasilkan nilai *matrix* PSNR hingga 48,41 *dB* sedangkan algoritma RSA hanya mampu menghasilkan nilai *matrix* PSNR hingga 41,71. Pada gambar dengan Tingkat kepadatan *pixel* mencapai 4,16 - 4,73 PPI algoritma RSA mampu menghasilkan nilai *matrix* PSNR hingga 44,05 *dB* sedangkan algoritma RSA hanya mampu menghasilkan 40,02 *dB*, dan pada gambar dengan Tingkat kepadatan *pixel* 5,26 - 5,37 PPI algoritma AES mampu memperoleh nilai 40,45 *dB* pada *matrix* PSNR dan algoritma RSA mampu menghasilkan nilai 38,57 *dB*. Kemudian saat dilakukan pengujian menggunakan gambar dengan tipe PNG baik algoritma RSA maupun AES mampu menghasilkan nilai *infinite* dari *matrix* PSNR, hal ini berarti kedua algoritma tersebut dapat menghasilkan gambar yang persis sama setelah didekripsi dengan gambar sebelum dienkripsi.

3. KESIMPULAN

Berdasarkan hasil pengujian yang sudah dilakukan, algoritma RSA mempunyai performa yang berbeda dari algoritma AES. Algoritma RSA cenderung memiliki waktu enkripsi yang lebih cepat dibandingkan dengan algoritma AES terlepas dari tingkat kerapatan pixel yang dimilki oleh gambar tersebut. Selain itu ukuran gambar yang dienkripsi menggunakan algoritma RSA dan AES sama-sama mengalami perubahan, namun gambar dengan kepadatan pixel 3,59 – 3,92 PPI dapat dienkripsi dengan baik oleh algoritma RSA sehingga ukuran gambar setelah dienkripsi hanya mengalami sedikit perubahan. Sedangkan gambar dengan kepadatan pixel 4,16 – 5,37 PPI lebih cocok dienkripsi menggunakan algoritma AES, karena gambar yang dienkripsi akan lebih sedikit mengalami kenaikan ukuran bahkan berpotensi bisa mengurangi ukuran gambar setelah dienkripsi. Dari segi keamanan informasi dalam gambar algoritma AES sedikit lebih unggul daripada algoritma RSA. Hasil yang serupa juga didapat dari pengujian kualitas gambar setelah didekripsi, dimana algoritma RSA terbukti tidak begitu bisa menjaga kualitas gambarnya, sedangkan algoritma AES lebih unggul dalam menjaga kualitas gambar setelah didekripsi. Kesimpulannya algoritma RSA mempunyai performa yang lebih baik dalam mengenkripsi gambar digital dibandingkan algoritma AES terlepas dari tingkat kepadatan pixel yang dimiliki oleh gambar tersebut.

DAFTAR PUSTAKA

Adrian, Y., Friscilla, C., Suardiman, N., Wijaya, A., & Sudimanto. (2022). Analisa Perbandingan Waktu Enkripsi dan Dekripsi pada Algoritma ECC dan RSA. *Media Informatika*, *21*(2), 124–132. https://doi.org/10.37595/mediainfo.v21i2.95

Alsaffar, D. M., Sultan Almutiri, A., Alqahtani, B., Alamri, R. M., Fahhad Alqahtani, H., Alqahtani, N. N., Mohammed Alshammari, G., & Ali, A. A. (2020). Image Encryption Based on AES and RSA Algorithms. *ICCAIS 2020 - 3rd International Conference on Computer Applications and Information Security*, 1–5. https://doi.org/10.1109/ICCAIS48893.2020.9096809

Engko M, G. Y. P., Id Hadiana, A., & Nurul Sabrina, P. (2022). Kriptografi Untuk Enkripsi Ganda Pada Gambar Menggunakan Algoritma AES (Advanced Encryption Standard) Dan RC5 (Rivest Code 5). *Informatics and Digital Expert (INDEX)*, *4*(1), 25–32. https://doi.org/10.36423/index.v4i1.884

Fajrin, A. M., Benedict, J. R., & Kusuma, H. J. (2023). Analisis Performa dari Algoritma Kriptografi RSA dan ElGamal dalam Enkripsi dan Dekripsi Pesan. *Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK)*, *8*(1), 91–98. https://tunasbangsa.ac.id/ejurnal/index.php/jurasik

Khamsyar, A., & Basri, M. (2022). Aplikasi Enkripsi Gambar Menggunakan Metode (Rivest Shamir Adleman) Rsa. *Jurnal Sintaks Logika*, *2*(3), 39–45. https://jurnal.umpar.ac.id/index.php/sylog

Loanardo, R., Hidayat Koniyo, M., & Hadjaratie, L. (2022). Analisis kualitas website perpustakaan Fakultas Teknik Universitas Negeri Gorontalo. *Diffusion: Journal of System and Information Technology*, *2*(1), 107–114.

Mido, A. R., & Ujianto, E. I. H. (2022). Analisis Pengaruh Citra Terhadap Kombinasi Kriptografi RSA dan STEGANOGRAFi LSB. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *9*(2), 279. https://doi.org/10.25126/jtiik.2022914852

Nino, B. E. (2023). Perbandingan Performa Algoritma AES dan Twofish Menggunakan Metode Strict Avalanche Criterion pada Nomor Induk Kependudukan Indonesia. *Jurnal Teknologi Informasi*, *9*(1), 19–29. https://doi.org/10.52643/jti.v9i1.2994

os - Miscellaneous operating system interfaces. (2024). https://docs.python.org/3/library/os.html

Permana, A. A., & Pratiwi, L. A. (2022). Implementation of The Advanced Encryption Standard (AES) Algorithm for Digital Image Security. *Jurnal Teknik Informatika*, *15*(1), 44–51. https://doi.org/10.15408/jti.v15i1.25735

Pruett, M. K., & Babb, B. A. (2023). Data Breach Report 2022. In *Family Court Review* (Vol. 61, Issue 1). https://doi.org/10.1111/fcre.12690

time - Time access and conversions. (2024). https://docs.python.org/3/library/time.html