JURNAL TIMES
TECHNOLOGY INFORMATICS & COMPUTER SYSTEM

Volume XIII No 1, Juni 2024
*p*ISSN : 2337 – 3601
*e*ISSN : 2549 – 015X

Tersedia *online* di http://ejournal.stmik-time.ac.id

IMPLEMENTASI KRIPTOGRAFI DALAM PENGAMANAN DATA GAMBAR MENGGUNAKAN ALGORITMA RSA

Christian Repi[1)], Jullia Titaley[2)], Eliasta Ketaren[3)]
Sistem Informasi
Universitas Sam Ratulangi
Bahu, Kec. Malalayang, Kota Manado, Sulawesi Utara
email: christianrepi106@student.unsrat.ac.id[1)], july_titaley@unsrat.ac.id[2),] eliasketaren@unsrat.ac.id[3)]

Abstrak
Penelitian ini bertujuan untuk mengatasi masalah kejahatan siber yang sering terjadi di era digital, khususnya pencurian data yang dapat menyebabkan kerugian bagi korban dengan memanfaatkan data yang dicuri, seperti gambar berisi informasi sensitif seperti KTP. Dengan mempertimbangkan risiko yang ada, penelitian ini menerapkan teknik kriptografi menggunakan Algoritma RSA untuk mengamankan data gambar. Metode penelitian mencakup pengembangan aplikasi yang mampu mengenkripsi data gambar sehingga, jika terjadi pencurian, data tersebut tidak dapat langsung digunakan oleh peretas. Hasil dari penelitian ini adalah aplikasi yang dapat mengenkripsi data gambar, meningkatkan keamanan, dan mencegah penyalahgunaan data, sehingga diharapkan dapat membantu mengurangi risiko kerugian akibat pencurian data di era digital.
Kata Kunci: Kejahatan Siber, Gambar, Kriptografi, Algoritma RSA.

1. Pendahuluan

Perkembangan Teknologi Informasi dan Komunikasi telah mempermudah pencarian dan pengiriman data, memungkinkan manusia untuk melakukan berbagai aktivitas informasi kapan saja dan di mana saja. Dengan perangkat seperti handphone yang terhubung ke internet, seseorang bisa mengirim dan menerima data secara instan, yang sangat berharga bagi perkembangan teknologi (Ahmad, 2012) [1].

Namun, perkembangan ini juga membawa risiko seperti pencurian data. Orang yang tidak bertanggung jawab bisa mengeksploitasi celah keamanan untuk mencuri data penting dan sensitif, seperti informasi akun atau dokumen identitas, yang kemudian dapat disalahgunakan untuk berbagai tujuan, termasuk pendaftaran pinjaman online atas nama korban (Yusuf et al., 2022) [2]. Insiden seperti ini menimbulkan kekhawatiran akan keamanan penggunaan teknologi.

Untuk mencegah pencurian data, diperlukan langkah-langkah keamanan seperti kriptografi. Kriptografi, dengan enkripsi dan dekripsinya, dapat melindungi data dari akses tidak sah. Algoritma RSA adalah salah satu metode yang efektif untuk mengamankan data gambar dan foto, memastikan hanya pihak yang memiliki kunci enkripsi yang bisa mengakses data tersebut (Sasongko, 2005) [3]. Penelitian sebelumnya, seperti oleh Fahreza dan Arif (2019) [4], menunjukkan efektivitas algoritma RSA dalam pengamanan data, meski penulis menggunakan bahasa pemrograman yang berbeda.

2. Landasan Teori

Kriptografi

Kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Dalam ilmu kriptografi, terdapat dua buah proses yaitu melakukan enkripsi dan dekripsi. Pesan yang akan dienkripsi disebut sebagai plaintext(teks biasa). Disebut demikian karena informasi ini dengan mudah dapat dibaca dan dipahami oleh siapa saja. Algoritma yang dipakai untuk mengenkripsi dan mendekripsi sebuah plaintext melibatkan penggunaan suatu bentuk kunci. Pesan plaintext yang telah dienkripsi (atau dikodekan) dikenal sebagai ciphertext (teks sandi) (Munir, 2019) [5].

Algoritma RSA

Sandi RSA merupakan algoritma kriptografi kunci publik (asimetris). Ditemukan pertama kali pada tahun 1977 oleh Ron Rivest, Adi Shamir, dan Len Adleman. Nama RSA sendiri diambil dari ketiga penemunya tersebut. Sebagai algoritma kunci publik, RSA mempunyai dua kunci, yaitu kunci publik dan kunci rahasia. RSA mendasarkan proses enkripsi dan dekripsinya pada konsep bilangan prima dan aritmetika modulo. Baik kunci enkripsi maupun dekripsi keduanya merupakan bilangan bulat. Kunci enkripsi tidak dirahasiakan dan diberikan kepada umum (sehingga disebut dengan kunci publik), namun kunci untuk dekripsi bersifat rahasia (kunci privat). Untuk menemukan kunci dekripsi, dilakukan dengan memfaktorkan suatu bilangan bulat menjadi faktor-faktor primanya. Kenyataannya, memfaktorkan bilangan bulat menjadi faktor primanya bukanlah pekerjaan yangmudah. Karena belum ditemukan algoritma yang efisien untuk melakukan pemfaktoran. Cara yang bisaa digunakan dalam

pemfaktoran adalah dengan menggunakan pohon faktor. Jika semakin besar bilangan yang akan difaktorkan, maka semakin lama waktu yang dibutuhkan. Jadi semakin besar bilangan yang difaktorkan, semakin sulit pemfaktorannya, semakin kuat pula algoritma RSA (Ginting et al., 2015) [6].

Langkah – langkah proses pengamanan algoritma RSA dapat berupa:
1. Pemilihan dua bilangan prima acak (p dan q).
2. Menghitung nilai n yang akan digunakan sebagai modulus p dan q
$$n = p \times q \quad (1)$$
Keterangan:
n = Nilai Modulus
p = Bilangan Prima 1
q = Bilangan Prima 2

3. Menghitung Nilai Euler untuk menentukan kunci privat dan kunci publik
$$\phi(n) = (p-1) \times (q-1) \quad (2)$$
Keterangan:
$\phi(n)$ = Nilai Euler
4. Menentukan nilai eksponen (e) yang relatif prima dengan nilai euler $\phi(n)$
5. Menentukan kunci privat dengan menggunakan rumus
$$d \equiv e^{-1} \ (mod\ \phi(n)) \quad (3)$$
Keterangan:
e = Eksponen Publik
d = Eksponen Privat
n = Nilai Modulus
6. Menentukan pasangan kunci privat (d,n) dan kunci publik (e,n)
7. Proses enkripsi menggunakan kunci publik
$$C = P^e \ mod\ n \quad (4)$$
Keterangan:
C = *Ciphertext*
P = *Plaintext*
e = Eksponen Publik
n = Nilai Modulus
8. Proses dekripsi menggunakan kunci privat
$$P = C^d \ mod\ n \quad (5)$$
Keterangan:
C = *Ciphertext*
P = *Plaintext*
d = Eksponen Privat
n = Nilai Modulus

3. Metode Penelitian

Penelitian ini menggunakan metode penelitian Agile Secure Development yang merupakan metode penelitian Agile yang lebih berfokus pada keamanan sistem aplikasi yang memiliki tahapan – tahapan berupa, Requirement,s Design, Development, Secure, Deployment, dan Review.
Tahapan penelitian dapat berupa :
1. *Requirement*
2. *Design*
3. *Development*
4. *Secure Testing*
5. *Deployment*
6. *Review*

Gambar 1. Diagram Alur Penelitian (Veracode, 2023)

4. Hasil Penelitian

Pembuatan Kunci

Pembuatan kunci RSA dilakukan menggunakan kaidah serta aturan pembuatan kunci RSA dengan langkah – langkah sebagai berikut :

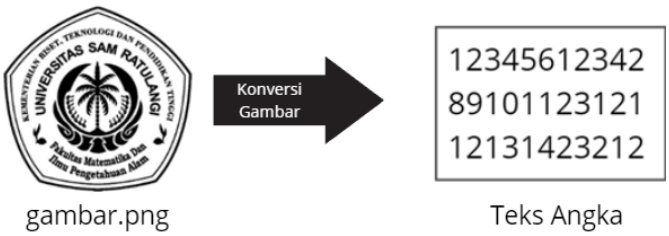
a. Pemilihan Bilangan Prima
   Pemilihan bilangan prima menjadi landasan pembuatan kunci dimana semakin besar bilanngan prima yang dipilih maka akan semakin kuat kunci yang dibuat. Pada aplikasi terdapat dua pilihan pemilihan bilangan prima yaitu memasukkan bilangan prima secara manual dan juga dibuat acak oleh sistem.
b. Perhitungan Hasil Kali Bilangan Prima
   Hasil kali bilangan prima diperlukan untuk perhitungan penentuan kunci apakah bisa digunakan atau tidak.
   $$n = p \times q \qquad (2.1)$$
c. Penentuan Fungsi Euler (n)
   Penentuan fungsi euler n (ϕn)diperlukan untuk menentukan kelayakan dari kunci yang dibuat apakah bisa terpakai atau tidak.
   $$\phi n = (p-1)(q-1) \qquad (2.2)$$
d. Penentuan Ekponen Publik (e)
   Eksponen publik(e) menjadi bagian penting dalam proses algoritma RSA dimana eksponen publik (e) akan digunakan pada bagian enkripsi sebagai salah satu dari kunci privat dan juga penentuan kunci publik yang akan digunakan dalam proses enkripsi dan dekripsi. Penentuan eksponen publik (e) dilakukan dengan memilih bilangan bulat yang tidak relatif prima dengan Fungsi Euler (ϕn).
e. Penentuan Eksponen privat (d)
   Perhitungan eksponen privat (d) dapat dilakukan dengan :
   $$d \times e \equiv 1 \; mod \; \phi n \qquad (2.3)$$
   Eksponen Privat dicari menggunakan Nilai euler dan Eksponen Publik yang sebelumnya telah didapat.
f. Penentuan Kunci Privat dan Kunci Publik
   Kunci privat didapat dari hasil kali bilangan prima (n) dan Eksponen Publik (e) yang akan digunakan pada proses dekripsi sementara untuk kunci publik didapat dari hasil kali bilangan prima (n) dan Eksponen Privat (d) yang akan digunakan pada proses dekripsi.
   $$Kunci\ Privat = (n, d)$$
   $$Kunci\ Publik = (n, e)$$

   Dalam aplikasi yang telah dibuat, pembuatan kunci dilakukan pada file generateKey.py

Proses Enkripsi

a. Konversi Gambar

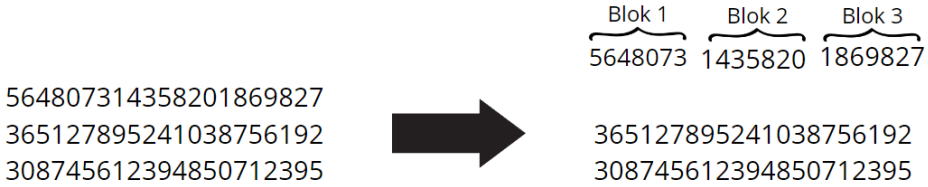
Dalam proses enkripsi hal pertama yang dilakukan dapat berupah mengkoversi gambar yang awalnya berbetuk pixel menjadi text. Proses konversi melibatkan image processing dengan menggunakan base64 untuk mengubah gambar yang sebelumnya adalah bentuk pixel menjadi bentuk text dan dilanjutkan kebentuk teks angka dengan format ascii.



Gambar 2. Ilustrasi Konversi Gambar

b. Membagi Blok Teks Angka

Proses ini dapat berupa pembagian blok atau bagian – bagian pada gambar yang telah dikonversi dalam bentuk teks menjadi beberapa bagian. Proses ini bertujuan untuk membatasi proses enkripsi agar lebih efesien dimana pembagian blok teks dapat membuat proses enkripsi menjadi lebih ringan dikarenakan tidak perlu menkripsi teks secara penuh . Selain itu proses ini juga dapat meningkatkan keamanan data enkripsi dikarenakan teks yang dienkripsi telah dibagi secara terpisah dikarenakan setiap bagian yang dibagi memiliki perhitungan yang berbeda.



Gambar 3. Ilustrasi Pembagian Blok Angka

c. Proses Enkripsi

Proses enkripsi dilakukan dengan mengenkripsi blok-blok teks angka yang telah dipisahkan. Enkripsi dilakukan menggunakan kunci publik yang terdiri dari eksponen publik dan hasil kali dari bilangan prima. Dengan menggunakan metode RSA, setiap blok data dienkripsi untuk menghasilkan ciphertext yang informasi rahasia gambar yang dienkripsi.

Proses enkripsi menggunakan rumus enkripsi RSA yaitu:

$$\boldsymbol{C = P^e \ mod \ n} \quad (2.4)$$

Sebagai contoh proses enkripsi dapat berupa:

Misalkan:

Bilangan Prima

$p = 17$

$q = 27$

Modulus

$n = 17 \times 27$ (2.1)

$n = 629$

Nilai Euler

$\phi n = (17 - 1)(37 - 1)$ (2.2)

$\phi n = 576$

Eksponen Publik

$e = 5$ (Tidak relatif prima dengan 576)

Eksponen Privat

$d \times 5 \equiv 1 \ mod \ 576$ (2.3)

$d = 461$

Kunci Publik = (629, 5)

Kunci Privat = (629,461)

P = $\underbrace{123}_{\text{blok 1}} \ \underbrace{345}_{\text{blok 2}} \ \underbrace{456}_{\text{blok 3}}$

P1 = 123

P2 = 345

P3 = 456

Proses Enkripsi

$\boldsymbol{C = P^e \ mod \ n}$ (2.4)

C1 = $123^5 \ mod \ 629 = 599$

C2 = $345^5 \ mod \ 629 = 303$

C3 = $456^5 \ mod \ 629 = 488$

C = $\underbrace{599}_{\text{blok 1}} \ \underbrace{303}_{\text{blok 2}} \ \underbrace{488}_{\text{blok 3}}$

Proses Dekripsi

a. Proses Dekripsi

Dekripsi file teks dilakukan dengan menggunakan kunci privat yang bersifat rahasia, dimana dengan menggunakan rumus dekripsi RSA maka dapat dilakukanlah proses dekripsi terhadap blok – blok yang telah dipisahkan sebelumnya.

Proses Dekripsi menggunakan rumus dekripsi RSA yaitu:

$$\boldsymbol{P = C^d \ mod \ n} \quad (2.5)$$

Sebagai contoh perhitungan:

Dengan menggunakan perhitungan dari proses enkripsi sebelumnya

Kunci Privat = (629,461)

$$P = C^d \ mod \ n \qquad (2.5)$$

C = $\underbrace{599}_{\text{blok 1}} \ \underbrace{303}_{\text{blok 2}} \ \underbrace{488}_{\text{blok 3}}$

C1 = 599
C2 = 303
C3 = 488

Proses Dekripsi
P1 = $599^{461} \ mod \ 629$ = 123
P2 = $303^{461} \ mod \ 629$ = 345
P3 = $488^{461} \ mod \ 629$ = 456

P = $\underbrace{123}_{\text{blok 1}} \ \underbrace{345}_{\text{blok 2}} \ \underbrace{456}_{\text{blok 3}}$

b. Pengabungan Blok Dekripsi

Setelah proses dekripsi blok dilakukan maka bagian – bagian yang telah dipisahkan tersebut akan digabungkan kembali menjadi file teks angka yang sudah memiliki informasi asli (*plaintext*) dari gambar yang telah dienkripsi meskipun masih dalam bentuk teks angka.

c. Konversi Pesan Teks

Proses terakhir yaitu konversi plaintext hasil dekripsi ke bentuk gambar semula dengan menggunakan konversi ascii dan base64.

Hasil Pengamatan Data

Hasil pengamatan data diperoleh dari uji coba aplikasi pengamanan data gambar menggunakan Algoritma RSA. Uji coba ini menilai aspek kecepatan waktu enkripsi dan dekripsi, serta perbandingan ukuran data asli dengan data yang telah diamankan. Dalam pengujian, diamati bahwa Algoritma RSA memberikan perlindungan terhadap data gambar dengan memperhatikan waktu yang dibutuhkan untuk proses enkripsi dan dekripsi, serta dampaknya terhadap perubahan ukuran data.

a. Ukuran Gambar

Pengamatan ini bertujuan untuk mengevaluasi hasil pengamanan data yang memperhitungkan kinerja terhadap gambar yang memiliki ukuran yang berbeda.

Tabel 1. Uji Ukuran

Plaintext (Gambar)	Ukuran	Panjang Bit Kunci	Kecepatan Enkripsi	Jumlah Blok	Ukuran Ciphertext	Kecepatan Dekripsi
GambarBesar.png	549kb (805x805)	512	5.2 Detik	14913	2.21 Mb	17 Detik
GambarSedang.jpg	270kb (537x537)	512	2.7	7349	1.09Mb	9.4 Detik
GambarKecil.jpg	162kb (375x375)	512	1.2 Detik	3416	519kb	5.4 Detik

Dari hasil uji coba pada tabel, terlihat bahwa terdapat perbedaan dalam kecepatan serta ukuran dari hasil pengamanan data yang dilakukan terhadap gambar yang sama dengan ukuran yang berbeda. Semakin kecil ukuran gambar, semakin cepat proses enkripsi dan dekripsi dapat dilakukan.

b. Warna

Pengujian ini berfokus pada hasil pengamanan data gambar terhadap gambar dengan ukuran yang sama tetapi memiliki warna yang berbeda.

Tabel 2. Uji Warna

Plaintext (Gambar)	Ukuran	Panjang Bit Kunci	Jumlah Blok	Ukuran Ciphertext
Original.jpg	175kb (1200x1588)	512	6460	725kb

Plaintext (Gambar)	Ukuran	Panjang Bit Kunci	Jumlah Blok	Ukuran Ciphertext
Grayscale.jpg	173kb (1200x1588)	512	4673	711kb

Tabel 3. Waktu Uji Warna

Plaintext (Gambar)	Uji 1 (Detik)		Uji 2 (Detik)		Uji 3 (Detik)		Uji 4 (Detik)		Rata – Rata (Detik)	
	E	D	E	D	E	D	E	D	E	D
Original.jpg	0.9	4.5	0.98	4.4	0.94	4.3	0.92	4.3	0.93	4.3
HitamPutih.jpg	0.89	4.2	0.87	4.2	0.9	4.1	0.89	4.2	0.88	4.17

Dalam pengujian ini dilakukan beberapa uji coba terhadap dua gambar yang memiliki resolusi yang sama namun memiliki intensitas warna yang berbeda dimana terdapat gambar berwarna dan gambar *grayscale*. Pengujian ini memperoleh hasil berupa gambar dengan tingkat pixel warna yang lebih variatif dan banyak diperlukan waktu lebih dalam proses pengaman data dibandingkan gambar yang memiliki tingkat pixel warna yang lebih rendah seperti hitam putih.

c. Kunci

Pengujian ini berfokus pada besar kunci RSA yang digunakan dalam proses pengamanan data dimana dilakukan uji coba terhadap gambar yang sama namun memiliki panjang kunci RSA yang berbeda.

Tabel 4. Uji Kunci

Plaintext (Gambar)	Ukuran	Panjang Bit Kunci	Kecepatan Enkripsi	Jumlah Blok	Jumlah Digit	Ukuran Ciphertext	Kecepatan Dekripsi
Lena.jpg	400kb (400x400)	1024	2.2 Detik	5321	309	1.57 Mb	21 Detik
Lena.jpg	400kb (400x400)	512	2 Detik	10783	155	1.6 Mb	10 Deik
Lena.jpg	400kb (400x400)	128	1.9 Detik	45226	38	1.78 Mb	6.4 Detik

Pengujian ini memperoleh hasil berupa semakin kecil bit bilangan kunci yang dipakai dalam proses enkripsi maka blok angka yang dihasilkan semakin banyak, ini dapat terjadi dikarenakan dalam proses pembagian blok, panjang digit angka maksimal dalam satu blok ditentukan dari besar kunci yang dipakai dalam proses enkripsi. Hal ini dapat mempengaruhi keamanan data yang di amankan dimana dapat terlihat bahwa semakin kecil kunci yang digunakan maka kecepatan dekripsi akan semakin cepat yang tentunya hal tersebut kurang baik dalam keamanan data.

5. Kesimpulan

Berdasarkan hasil penelitian yang dilakukan, dalam proses pengamanan data gambarmenggunakan algoritma RSA terdapat beberapa kesimpulan yang dapat diambil Segi Keamanan Aplikasi sudah dapat berjalan dengan baik dan dapat melakukan tugasnya yaitu mengamankan data gambar dengan baik, namun aplikasi belum dapat berjalan dengan baik terhadap data gambar yang berukuran besar. Segi Keefesienan Semakin besar ukuran serta resolusi gambar yang dienkripsi maka semakin lama juga serta semakin berat hasil enkripsi dari gambar tersebut. Gambar dengan variasi pixel warna yang banyak akan lebih lama terenkripsi dan terdekripsi dibandingkan gambar yang memiliki pixel warna yang sedikit seperti grayscale. Panjang kunci mempengaruhi keamanan data dimana semakin besar kunci yang digunakan maka akan menambah kompleksitas dari hasil enkripsi membuat keamanan gambar semakin terjaga. Proses pengamanan data akan lebih efesien jika menggunakan perangkat keras yang memenuhi.

6. Daftar Pustaka

[1] Ahmad, A. 2012. Perkembangan Teknologi Komunikasi dan Informasi Akar Revolusi dan Berbagai Standarnya. Jurnal Dakwah Tabligh. 13.

[2] Yusuf, M., Suryadi. dan Hamid, R. 2022. Analisis Kejahatan Hacking Sebagai Bentuk Cyber Crime Dalam Sistem Hukum yang berlaku di Indonesia. Jurnal Pendidikan dan Konseling. Universitas Pahlawan Tuanku Tambusai.

[3] Sasongko, J. 2005. Pengamanan Data Informasi menggunakan Kriptografi Klasik. Jurnal Teknologi Informasi DINAMIK. 160-167.

[4] Fahreza, A. M., dan Harbani, A. 2019. Aplikasi Keamanan Data Gambar Menggunakan Algoritma RSA (Rivest Shamir Adleman) Berbasis Desktop. Program Studi Teknik Informatika STIKOM Binaniaga Bogor. Jurnal Ilmiah Teknologi, 1-9.
[5] Munir, R. (2019). Kriptografi. Pernerbit Informatika Bandung.
[6] Ginting, A., Isnanto, R. R., dan Windasari, I. P. 2015. Implementasi Algoritma Kriptografi RSA untuk Enkripsi dan Dekripsi Email. Jurnal Teknologi dan Sistem Komputer.
[7] Anwar, B., Nugroho, B. N., Prayudha, J., dan Azanuddin. Implementasi Algortima RSA Terhadap Keamanan Data Simpan Pinjam. STMIK Triguna Dharma. 18(1). 30-34.
[8] Sutejo. 2021. Implementasi Algoritma Kriptografi RSA (Rivest Shamir Adleman) Untuk Keamanan Data Rekam Medis Pasien. Journal of Information Technology and Computer Science. Universitas Lancang Kuning. 4(1).
[9] Hermawan, A., dan Ujianto, E. I. H. 2021. Implementasi Enkripsi Data Menggunakan Kombinasi AES dan RSA. Jurnal Nasional Informatika Dan Teknologi Jaringan.
[10] Das, R., dan Goldsztein, G. 2023. Mathematics Behind the RSA Algorithm. Journal of Student Research, 12(1).
[11] Kim, Gyu-Choi., Jong, Yong-Bok. 2022. Fast signing method in RSA with high speed verification. Kim Chaek University of Technology.