Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) DOI: 10.25126/jtiik.202295388
Vol. 9, No. 6, Desember 2022, hlm. 1203-1210 p-ISSN: 2355-7699
Akreditasi KEMENRISTEKDIKTI, No. 36/E/KPT/2019 e-ISSN: 2528-6579

PENGAMANAN PESAN E-COMPLAINT FASILITAS DAN KINERJA CIVITAS AKADEMIKA MENGGUNAKAN ALGORITMA RSA

Dwi Yuny Sylfania[*1], Fransiskus Panca Juniawan[2], Laurentinus[3], Hengki[4]

[1,2,3,4] Institut Sains dan Bisnis Atma Luhur, Pangkal Pinang
Email: [1]dysylfania@atmaluhur.ac.id, [2]fransiskus.pj@atmaluhur.ac.id.ac.id, [3]laurentinus@atmaluhur.ac.id.ac.id, [4]hengki@atmaluhur.ac.id.ac.id
*Penulis Korespondensi

(Naskah masuk: 04 Agustus 2021, diterima untuk diterbitkan: 16 Desember 2022)

Abstrak

Keluhan merupakan suatu bentuk pengaduan yang muncul karena ketidakpuasan produk atau layanan perusahaan. ISB Atma Luhur tidak terlepas dari adanya keluhan mengenai fasilitas kampus, serta kinerja dosen dan karyawan. Meskipun keluhan identik dengan hal negatif dan sulit ditangani, namun harus diselesaikan karena menyangkut kredibilitas perusahaan. Kebanyakan mahasiswa memilih untuk tidak melaporkan keluhan, khususnya mengenai kinerja dosen dan karyawan, karena takut terjadi kebocoran pada keluhan yang disampaikan. Hal tersebut bisa mengakibatkan pengurangan nilai akademik dan pelayanan yang buruk ketika mengurus administrasi perkuliahan. Penggunaan teknik kriptografi dengan algoritma RSA dijadikan sebagai solusi untuk mencegah kebocoran pesan. Kriptografi digunakan untuk menerjemahkan pesan ke dalam bentuk kode sehingga isi pesan sebenarnya tidak dapat diketahui secara langsung. Pemilihan RSA sebagai algoritma kriptografi karena sulitnya pemfaktoran bilangan prima dalam pembuatan kunci. Pemfaktoran tersebut akan menghasilkan kunci publik untuk proses enkripsi pesan, dan kunci privat untuk proses dekripsi pesan. Proses enkripsi menghasilkan *ciphertext* dan proses dekripsi menghasilkan *plaintext*. Pengujian *brute force* menghasilkan bahwa dengan panjang kunci sebesar 24 bit, maka waktu yang dibutuhkan untuk menemukan kunci privat sebanyak 16.777.216 adalah selama 16,77 detik/10^6 percobaan. Berdasarkan pengujian tersebut, peluang untuk menemukan kunci privat yang tepat membutuhkan waktu yang lama, sehingga kebocoran pesan pada aplikasi e – complaint dapat dicegah dengan baik.

Kata kunci: keluhan, kriptografi, *ciphertext*, *plaintext*, algoritma RSA, *brute force*

MESSAGES SECURITY OF E-COMPLAINT ON THE FACILITY AND PERFORMANCE OF THE ACADEMIC COMMUNITY USING RSA ALGORITHM

Abstract

A grievance is a complaint that arises because of the emergence of the company's products or services. ISB Atma Luhur is inseparable from complaints about campus facilities and the performance of lecturers and employees. Because complaints can be company problems, so must be correct. Most students choose not to report their grievances, especially regarding lecturers and employees, for fear of leaks in the complaints submitted. However, it can decrease academic grades and poor service when taking care of lecture administration. This study proposes the application of cryptography, which converts messages into code, so the actual message is not known. The use of cryptographic techniques with the RSA algorithm is used as a solution to prevent message leakage. The choice of RSA as a cryptographic algorithm is due to the difficulty of factoring prime numbers in key generation. This factoring will generate a public key for the message encryption process and a private key for the message decryption process. The encryption process produces ciphertext, and the decryption process produces plaintext. Brute force testing results that with a key length of 24 bits, the time required to find the private key of 16,777,216 is 16.77 seconds/106 attempts. Based on these tests, the opportunity to find the right private key in a long time so that there is no message on the e-complaint application can result in good.

Keywords: *complaint, cryptography, ciphertext, plaintext, RSA algorithm, brute force*

1. PENDAHULUAN

Keluhan merupakan ungkapan ketidakpuasan pelangggan atas produk, jasa, maupun fasilitas yang diberikan perusahaan (Atmodjo, 2020). Meskipun keluhan tersebut identik dengan hal negatif dan sulit untuk ditangani, namun harus diselesaikan karena menyangkut kredibilitas perusahaan di mata

masyarakat sebagai usaha memberikan kualitas pelayanan yang baik (Afriansyah, 2019). Begitu pula dengan ISB Atma Luhur yang acap kali menerima pengaduan dari mahasiswa mengenai fasilitas kampus, bahkan kinerja dosen dan karyawan. Namun, sebagian mahasiswa memilih untuk tidak melakukan pengaduan, khususnya mengenai kinerja dosen dan karyawan. Hal tersebut dikarenakan adanya rasa takut terjadinya kebocoran pesan yang disampaikan yang berakibat pengurangan nilai akademik dan pelayanan yang kurang baik ketika mengurus administrasi perkuliahan. Padahal, pengaduan tersebut akan memberikan *feed back* positif bagi kampus untuk selalu terpacu meningkatkan kualitas dan pelayanan yang baik bagi civitas akademika.

Berdasarkan permasalahan di atas, penulis akan membuat aplikasi pengaduan yang menggabungkan teknik kriptografi dengan menggunakan algoritma RSA, yang tidak sekedar untuk menampung keluhan tetapi juga mampu menjaga kerahasiaan dan keakuratan isi pesan. Kriptografi adalah salah satu seni pengamanan data melalui suatu saluran, yang mana data disandikan sebelum dikirimkan, sehingga ketika data dicuri oleh pihak yang tidak berhak, tidak dapat mengetahui isi data sebenarnya (Jamaludin, 2020). Algoritma RSA dipublikasikan pada tahun 1977 oleh Ronald Rivest, Adi Shamir, dan Leonard Adleman. Algoritma ini memperoleh keamanan dari sulitnya memfaktorkan nilai-nilai besar (Siahaan and Sianipar, 2020). Selain itu, aplikasi ini berbasis android, sehingga mahasiswa bisa melaporkan keluhan tanpa terikat ruang dan waktu. Android merupakan platform pemrograman yang dikembangkan oleh Google untuk ponsel cerdas dan perangkat seluler lainnya, misalnya tablet (Herlinah and KH, 2019).

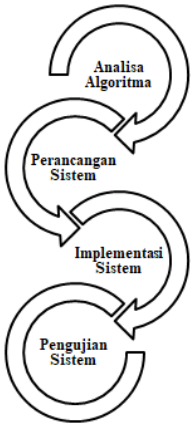
Adapun penelitian terdahulu yang dijadikan sebagai referensi antara lain penerapan algoritma RSA pada aplikasi *e-voting* BEM berbasis Android (Juniawan, 2016). Waktu komputasi yang dibutuhkan kombinasi algoritma RSA dan El Gamal relatif pendek dibandingkan dengan algoritma RSA asli, namun dari segi keamanan belum dapat dibuktikan (Iswari, 2017). Efisiensi algoritma AES dalam proses enkripsi dan dekripsi *medical image* lebih baik dibandingkan dengan algoritma RSA (Santhosh Kumar, Roshni and Nair, 2018). Penelitian selanjutnya menghasilkan bahwa kombinasi algoritma AES dan RSA mampu mengamankan komunikasi *via email,* namun untuk tingkat keamanan baru sebatas simulasi karena adanya kendala yang belum dapat terselesaikan (Ye Liu, 2018). Enkripsi optik virtual dengan menggunakan teknik *phase shifted digital holography* dan algoritma RSA dinilai sederhana, cepat, akurat dan aman (Chatterjee, 2018). Waktu pelaksanaan proses penulisan dan pembacaan data pada sistem berbasis NFC dengan algoritma DES lebih cepat dibandingkan 3DES (Ratnadewi et al., 2018). Kinerja dan keamanan algoritma ECC lebih baik dibandingkan algoritma El Gamal dan RSA (Mallouli, 2019). Selain menyelesaikan masalah keamanan, algoritma RSA – IOT juga mengurangi *overhead* dan *delay* di jaringan, namun proses pembentukan kunci dengan kombinasi algoritma ini cukup sulit karena dibatasi oleh teknologi (Luo, 2019). Implementasi algoritma RSA dengan panjang kunci 16 bit pada aplikasi pengaduan kecurangan pilkada dapat menjaga kerahasiaan dan keakuratan isi pesan, namun perlu peningkatan ukuran panjang kunci untuk meminimalisir terjadinya kebocoran (Sylfania, 2019). Penerapan kriptografi visual pada kartu kredit dapat mencegah *phising* dan pencurian data pribadi, serta autentikasi, otorisasi, kerahasiaan, dan integritas (keaslian) data lebih terjamin, namun kenyamanan *user* dan load *server* bertambah dikarenakan adanya proses kriptografi visual (Trihastuti, 2020). Dari segi proses enkripsi dan dekripsi, algoritma Blowfish memiliki kecepatan lebih tinggi dibandingkan algoritma RSA, namun belum ada pengujian untuk keamanan algoritma tersebut (Sylfania, 2020). Modifikasi Caesar Cipher dengan 256 karakter ASCII menghasilkan aplikasi keamanan data informasi yang aman, namun beberapa karakter tidak terbaca dengan baik, karena keterbatasan pembacaan karakter pada media *smartphone* berbasis android (Triana, 2020). Aplikasi mampu menampung keluhan mahasiswa dan civitas akademika secara terpusat dan terintegrasi, namun perlu pengembangan lebih lanjut dengan menerapkan teknik kriptografi untuk sisi keamanan informasi (Sylfania, Perkasa and Juniawan, 2020). Efisiensi SoRSA untuk IOT di sisi *cloud server* terbukti lebih unggul dibandingkan dengan ExpSOS (Zhang et al., 2020). Kombinasi metode Affine Cipher dan Knapsack Merkle Hellman dapat digunakan untuk proses enkripsi dan dekripsi data, yang memperkecil kemungkinan pihak-pihak yang tidak berkepentingan untuk mengetahui suatu data rahasia (Fadlan, 2017). Proses pengamanan data menggunakan metode gifshuffle, algoritma AES, dan kompresi Huffman berhasil diintegrasikan dan diterapkan dengan hasil yang baik, namun penambahan *noise* ke dalam *stegoimage* menjadikan kualitas citra kurang baik (Darwis, 2018). Implementasi algoritma blowfish pada aplikasi mampu mengubah teks asli file video ke dalam bentuk *ciphertext*, dan mengubah kembali ke dalam bentuk *plaintext*, namun penerima pesan harus memiliki aplikasi yang serupa agar dapat menerjemahkan *ciphertext* (Fahriani, 2019). Algoritma kriptografi 3DES dan steganografi LSB sanggup memberikan keamanan ganda karena pesan bukan hanya tersembunyi dalam gambar tetapi juga terenkrip menggunakan algoritma kriptografi 3DES (Rantelinggi, 2020). Algoritma RSA mampu mengenkripsi dan mendekripsi data file teks dengan baik, namun perlu pengujian enkripsi data file dengan ukuran tiga hingga ratusan digit (Sihotang et al., 2020). Proses enkripsi dan dekripsi gambar dengan algoritma RSA dapat ditingkatkan sebesar 14% dan

22% (Al-Kadei, Mardan and Minas, 2020). Selain menghasilkan kualitas visual dan kapasitas penyimpanan yang efektif, kombinasi antara algoritma RSA untuk enkripsi, dan algoritma Huffman dan DWT untuk teknik kompresi teks dan gambar, menghasilkan tingkat keamanan dan daya tahan yang tinggi terhadap serangan (Wahab et al., 2021).

Berdasarkan rumusan masalah dan penelitian terdahulu, peneliti akan mengembangkan aplikasi, terutama dari segi panjang kunci enkripsi dan dekripsi, dari 16 bit menjadi 24 bit dan melakukan pengujian *brute force attack* untuk mengukur resistansi terhadap serangan yang memungkinkan.

2. METODE PENELITIAN

Penelitian ini menggunakan model *waterfall*. Model *waterfall* adalah proses hidup perangkat lunak yang memiliki sebuah proses yang linear dan sekuensial (Sylfania, 2019). Pada Gambar 1 menunjukkan tahapan penelitian.



Gambar 1. Model Penelitian

2.1. Analisa Algoritma

Algoritma asimetris ialah algoritma yang memiliki dua buah kunci yaitu kunci privat dan kunci publik. Kunci privat digunakan untuk melakukan proses dekripsi, sedangkan kunci publik digunakan untuk melakukan proses enkripsi.

Algoritma RSA merupakan algoritma asimetris, yang mana tingkat kesulitan dalam memfaktorkan bilangan yang besar menjadi faktor – faktor prima, menjadi keamanan bagi algoritma tersebut (Rivest, 1977). Algoritma RSA memiliki 3 tahapan, ialah sebagai berikut:

2.1.1. Pembentukan Kunci

Tahap ini dilakukan untuk memperoleh kunci privat dan kunci publik, yang akan digunakan dalam proses dekripsi dan enkripsi. Pada tahap ini akan menghasilkan kunci privat, yaitu d dan n, dan kunci publik yaitu e dan n.

Tahap pertama dalam pembentukan kunci adalah memilih dua buah bilangan prima secara sembarang yaitu p dan q, dimana p ≠ q. Tahap kedua adalah menghitung nilai n menggunakan rumus 1.

$$n = pq \quad (1)$$

dengan n adalah hasil perkalian antara dua buah bilangan prima yaitu p dan q.

Tahap ketiga adalah hitung ϕn menggunakan rumus 2.

$$\phi n = (p - 1)(q - 1) \quad (2)$$

dengan ϕn adalah hasil perkalian antara bilangan p dikurangi 1 dan bilangan q dikurangi 1.

Tahap selanjutnya adalah menentukan kunci publik, e, yakni bilangan prima acak menggunakan rumus 3.

$$1 < e < \phi n, FPB(\phi n, e) = 1 \quad (3)$$

dengan e, adalah bilangan yang relatif prima terhadap ϕn. FPB (ϕn, e) = 1, adalah faktor pembagi terbesar ϕn dan e adalah 1.

Tahap terakhir adalah menghitung kunci privat, d, dengan menggunakan rumus 4.

$$d = \frac{1 + k^{\Phi(n)}}{e} \quad (4)$$

dimana d adalah kunci dekripsi, k adalah bilangan bulat yang merupakan nilai eksperimen.

2.1.2. Enkripsi

Proses enkripsi adalah proses mengubah pesan (plaintext) ke dalam bentuk kode rahasia (ciphertext). Enkripsi dilakukan oleh si pengirim dengan menggunakan kunci publik (e, n), sehingga pesan tidak dapat dibaca oleh orang yang tidak memiliki kunci privat.

Tahap awal dalam proses enkripsi adalah mengkonversikan tiap karakter pesan (M) ke bilangan desimal, 65 – 90, dimana A = 65, B = 66, …, Z = 90, dan 97 – 122, dimana a = 97, b = 98, …, z = 122. Tahap selanjutnya adalah menghitung C menggunakan rumus 5.

$$C = Me \bmod n \quad (5)$$

dimana C adalah ciphertext, M adalah karakter yang telah dikonversikan ke dalam bilangan desimal, dengan ketentuan $0 \leq M < n$.

2.1.3. Dekripsi

Proses dekripsi adalah proses mengubah pesan yang berisi kode rahasia (ciphertext) ke dalam bentuk pesan asli (plaintext). Dekripsi dilakukan oleh si

penerima dengan menggunakan kunci privat (d, n), sehingga makna pesan dapat dimengerti.

Tahap awal dalam proses dekripsi adalah blok ciphertext didekripsi kembali dengan menggunakan rumus 6.

$$M = Cd \bmod n \quad (6)$$

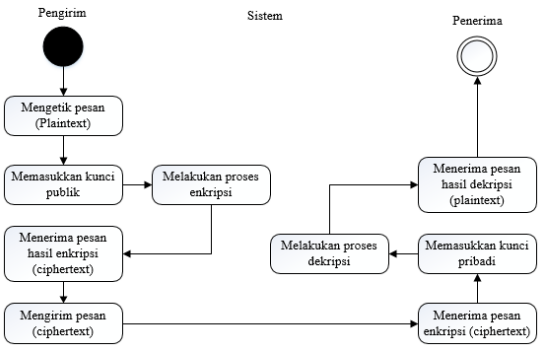
dengan M adalah karakter yang masih berbentuk bilangan desimal, C adalah ciphertext.

Tahap selanjutnya adalah mengkonversikan M ke dalam bentuk karakter.

2.2. Perancangan Sistem

Tahap ini mengambarkan penerapan algoritma RSA pada proses enkripsi dan dekripsi pesan. Pada Gambar 2 menjelaskan, pengirim (civitas akademika) mengetik pesan (plaintext) yang berisi keluhan, selanjutnya memasukkan kunci publik agar sistem melakukan proses enkripsi. Setelah itu, sistem akan menghasilkan pesan dalam bentuk ciphertext, dan pengirim akan mengirimkan pesan tersebut ke penerima.

Selanjutnya, setelah menerima pesan (ciphertext), si penerima (admin) akan memasukkan kunci pribadi agar sistem melakukan proses dekripsi. Hasil dekripsi berupa pesan (plaintext) yang dapat dipahami arti sebenarnya.



Gambar 2. Activity Diagram Proses Enkripsi dan Dekripsi

2.3. Implementasi Sistem

Tahap ini akan membahas cara kerja algoritma RSA dalam melakukan proses enkripsi dan dekripsi pesan. Tahap pertama adalah proses pembentukan kunci yang dilakukan dengan cara membangkitkan bilangan prima acak. Semakin besar bilangan prima yang dipilih maka semakin sulit untuk memecahkan kunci. Tahap ini menghasilkan pasangan kunci pribadi dan pasangan kunci publik.

Selanjutnya adalah melakukan proses enkripsi, dengan menggunakan pasangan kunci publik yang akan menghasilkan *ciphertext*. Tahap terakhir adalah melakukan proses dekripsi dengan menggunakan pasangan kunci privat yang akan menghasilkan *plaintext*.

2.4. Pengujian Sistem

Pada tahap ini akan dilakukan dua jenis pengujian. Pengujian pertama adalah pengujian sistem dengan menggunakan *blackbox* untuk menguji kesesuaian antara input yang diberikan dengan output yang dihasilkan.

Pengujian selanjutnya adalah pengujian dengan menggunakan *brute force attack* untuk menguji kerentanan sistem terhadap serangan, khususnya serangan untuk memecahkan kunci privat dan kunci publik yang digunakan dalam proses enkripsi dan dekripsi.

3. HASIL DAN PEMBAHASAN

Proses enkripsi dan dekripsi tidak dapat dilakukan tanpa menggunakan kunci. Oleh karena itu, tahap awal yang dilakukan adalah melakukan proses pembentukan kunci yang menghasilkan kunci publik dan kunci privat.

Proses enkripsi menggunakan pasangan kunci publik yang bersifat umum, bisa disebarluaskan ke siapapun. Sedangkan, kunci privat digunakan untuk proses dekripsi, yang bersifat rahasia dan hanya diketahui oleh si pemilik kunci saja.

3.1. Pembentukan Kunci

Tahap awal yang dilakukan untuk melakukan proses enkripsi dan dekripsi pesan ialah proses pembentukan kunci. Tahap ini menghasilkan kunci publik untuk melakukan proses enkripsi, dan kunci privat untuk melakukan proses dekripsi.

Adapun tahapan dalam pembentukan kunci adalah nilai p dan q ialah bilangan prima, dimana p ialah 19, dan q ialah 17. Selanjutnya, berdasarkan persamaan (1), nilai n didapat dari hasil perkalian p dan q, sehingga, nilai n = 19 * 17 = 323. Langkah selanjutnya, berdasarkan persamaan (2), menentukan nilai $\phi n = (p - 1)(q - 1)$ sehingga didapat $\phi n = (19 - 1)(17 - 1) = 288$. Kemudian dicari nilai e seperti pada rumus persamaan (3), sehingga nilai akhirnya adalah 1.

Tabel 1. Perhitungan Pencarian Nilai e

e	FPB (288, e)	Hasil
20	FPB (288,20)	4
21	FPB (288,21)	3
22	FPB (288,22)	2
23	FPB (288,23)	1
24	FPB (288,24)	24
25	FPB (288,25)	1
26	FPB (288,26)	2
27	FPB (288,27)	9

Untuk mendapatkan nilai e, langkah awal yang harus dilakukan adalah mencari nilai faktorial dari ϕn yang bersifat bilangan prima. Langkah ini bertujuan agar nilai e bukan nilai faktorial dari ϕn, sehingga,

$\phi n = 288$

$= 2^5 \times 3^2$

Berdasarkan Tabel 1, nilai e yang memenuhi syarat yaitu ketika e bernilai 23, dan 25, karena FPB (288, 23), dan FPB (288, 25) = 1. Oleh karena itu, ditentukan nilai e = 25.

Langkah terakhir adalah mencari nilai d dengan persamaan (4) sehingga didapat hasil seperti pada Tabel 2.

Tabel 2 .Perhitungan Pencarian Nilai k

k	$d = \frac{1 + k*(\phi(n))}{e}$	Hasil
20	$d = \frac{1 + 20*\phi(288)}{25}$	230,44
21	$d = \frac{1 + 21*\phi(288)}{25}$	241,96
22	$d = \frac{1 + 22*\phi(288)}{25}$	253,48
23	$d = \frac{1 + 23*\phi(288)}{25}$	265
24	$d = \frac{1 + 24*\phi(288)}{25}$	276,52
25	$d = \frac{1 + 25*\phi(288)}{25}$	288,04
26	$d = \frac{1 + 26*\phi(288)}{25}$	299,56
27	$d = \frac{1 + 27*\phi(288)}{25}$	311,08
28	$d = \frac{1 + 28*\phi(288)}{25}$	322,6
29	$d = \frac{1 + 29*\phi(288)}{25}$	334,12

Berdasarkan Tabel 2, nilai k merupakan nilai eksperimen (bilangan integer) = 20, 21, 22, …, dst. Nilai d merupakan bilangan bulat, sehingga dipilih nilai k adalah 23, sehingga hasil dari perhitungan persamaan d adalah 265. Oleh karena itu, diperoleh kunci publik (25, 323) dan kunci privat (265, 323).

3.2. Enkripsi Pesan

Sebelum melakukan proses enkripsi, konversikan terlebih dahulu tiap huruf ke dalam bentuk desimal, kemudian dengan menggunakan kunci publik (e, n) yang sudah didapatkan dari proses pembentukan kunci sebelumnya. Proses enkripsi dari pesan akan menghasilkan ciphertext yang akan dikirimkan kepada penerima.

Berdasarkan Tabel 3, kolom P berisi "AC ruang LPPM mati" merupakan pesan (plaintext) yang akan dikirimkan. Kolom M merupakan hasil konversi tiap kata ke bilangan desimal. Selanjutnya, proses enkripsi dilakukan dengan menggunakan kunci publik (25, 323) dan menggunakan persamaan matematis $C = M^e \bmod n$, yang menghasilkan ciphertext 122, 186, 219, 209, 287, 90, 127, 103, 219, 76, 158, 158, 77, 219, 231, 90, 71, 167.

Tabel 3. Proses Enkripsi Pesan

P	M	$C = M^e \bmod n$
A	65	$C = 65^{25} \bmod 323 = 122$
C	67	$C = 67^{25} \bmod 323 = 186$
Spasi	32	$C = 32^{25} \bmod 323 = 219$
r	114	$C = 114^{25} \bmod 323 = 209$
u	117	$C = 117^{25} \bmod 323 = 287$
a	97	$C = 97^{25} \bmod 323 = 90$
n	110	$C = 110^{25} \bmod 323 = 127$
g	103	$C = 103^{25} \bmod 323 = 103$
Spasi	32	$C = 32^{25} \bmod 323 = 219$
L	76	$C = 76^{25} \bmod 323 = 76$
P	80	$C = 80^{25} \bmod 323 = 158$
P	80	$C = 80^{25} \bmod 323 = 158$
M	77	$C = 77^{25} \bmod 323 = 77$
Spasi	32	$C = 32^{25} \bmod 323 = 219$
m	109	$C = 109^{25} \bmod 323 = 231$
a	97	$C = 97^{25} \bmod 323 = 90$
t	116	$C = 116^{25} \bmod 323 = 71$
i	105	$C = 105^{25} \bmod 323 = 167$

3.3. Dekripsi Pesan

Pesan yang diterima berupa ciphertext. Untuk mengetahui isi dari pesan, maka penerima melakukan proses dekripsi. Adapun proses dekripsi menggunakan kunci privat (d, n) yang sudah didapatkan dari proses pembentukan kunci sebelumnya.

Tabel 4. Proses Dekripsi Pesan

C	$M = C^d \bmod n$	P
122	$M = 122^{265} \bmod 323 = 65$	A
186	$M = 186^{265} \bmod 323 = 67$	C
219	$M = 219^{265} \bmod 323 = 32$	Spasi
209	$M = 209^{265} \bmod 323 = 114$	r
287	$M = 287^{265} \bmod 323 = 117$	u
90	$M = 90^{265} \bmod 323 = 97$	a
127	$M = 127^{265} \bmod 323 = 110$	n
103	$M = 103^{265} \bmod 323 = 103$	g
219	$M = 219^{265} \bmod 323 = 32$	Spasi

C	M = C^d mod n	P
76	M = 76^{265} mod 323 = 76	L
158	M = 158^{265} mod 323 = 80	P
158	M = 158^{265} mod 323 = 80	P
77	M = 77^{265} mod 323 = 77	M
219	M = 219^{265} mod 323 = 32	Spasi
231	M = 231^{265} mod 323 = 109	m
90	M = 90^{265} mod 323 = 97	a
71	M = 71^{265} mod 323 = 116	t
167	M = 167^{265} mod 323 = 105	i

Berdasarkan Tabel 4, kolom C berisi ciphertext 122, 186, 219, 209, 287, 90, 127, 103, 219, 76, 158, 158, 77, 219, 231, 90, 71, 167, yang akan didekripsi dengan menggunakan kunci privat (265, 323) dan menggunakan persamaan matematis P = C^d mod n. Dari persamaan tersebut, hasil dekripsi berupa bilangan desimal yang perlu dikonversikan terlebih dahulu untuk mengetahui isi dari pesan, sehingga menghasilkan plaintext "AC ruang LPPM mati"

3.4. Pengujian

Pada tahapan ini, penulis menggunakan dua jenis pengujian, yaitu pengujian *blackbox* untuk menguji fungsionalitas sistem, dan pengujian dengan teknik *brute force attack* untuk mengukur resistansi sistem terhadap serangan.

Tabel 5. Pengujian Blackbox

Pengujian	Hasil yang diharapkan	Hasil
Membuat pesan pengaduan	Menampilkan struktur dari pesan	Sesuai
Melakukan enkripsi pesan	Pesan berhasil dienkripsi sesudah pengirim menginput kunci publik	Sesuai
Melakukan pengiriman pesan	Pesan (berupa ciphertext) dikirimkan ke tujuan	Sesuai
Melakukan dekripsi pesan	Pesan berhasil didekripsi sesudah penerima menginput kunci privat	Sesuai

Berdasarkan Tabel 5, fungsionalitas sistem seperti proses pembuatan pesan, proses enkripsi, dan proses dekripsi berjalan dengan baik sesuai dengan fungsionalitasnya.

Tabel 6. Pengujian Brute Force Attack

Kunci Privat	Ukuran Kunci	Jumlah Kemungkinan Kunci	Lama Waktu Untuk 10^6 Percobaan
265	24 bit	16.777.216	16,77 detik

Berdasarkan Tabel 6, dapat diketahui bahwa kunci privat yang terbentuk memiliki panjang sebesar 24 bit, dan jumlah kemungkinan kunci sebanyak 16.777.216 kunci. Untuk memecahkan kunci privat, si penyerang harus melakukan sebanyak 10^6 percobaan atau setara dengan 1.000.000 percobaan dalam waktu 16,77 detik.

4. KESIMPULAN

Adapun kesimpulan dari penelitian ini menghasilkan bahwa algoritma RSA pada e-complaint mampu diterapkan dengan baik. Proses enkrpsi dan dekripsi pesan keluhan menghasilkan ciphertext dan plaintext yang tepat dan sesuai.

Berdasarkan pengujian Blackbox, fungsionalitas sistem telah berjalan dengan baik, karena input yang dimasukkan sesuai dengan output yang diharapkan. Selain itu, berdasarkan pengujian *brute force attack*, dengan menggunakan panjang kunci sebesar 24 bit, maka waktu yang dibutuhkan untuk memecahkan 16.777.216 kunci, selama 16,77 detik/10^6 percobaan. Berdasarkan pengujian tersebut, peluang untuk menemukan kunci privat yang tepat membutuhkan waktu yang lama, sehingga kebocoran pesan pada aplikasi e-complaint dapat dicegah dengan baik.

Untuk pengembangan penelitian selanjutnya, agar panjang kunci dapat ditingkatkan hingga 1024 bit, sehingga peluang terjadinya kebocoran pesan menjadi sangat kecil.

DAFTAR PUSTAKA

AFRIANSYAH, J.Y., 2019. Dear Customer, I Love You! Jakarta Pusat: Elex Media Komputindo.

AL-KADEI, F.H.M.S., MARDAN, H.A. AND MINAS, N.A., 2020. Speed Up Image Encryption by Using RSA Algorithm. In: 2020 6th International Conference on Advanced Computing and Communication Systems, ICACCS 2020. pp.1302–1307. DOI: 10.1109/ICACCS48705.2020.9074430

ATMODJO, M.W., 2020. Manajemen Komplain Pada Industri Jasa Pelayanan Makanan - Minuman. Yogyakarta: ANDI.

CHATTERJEE, A., DHANOTIA, J., BHATIA, V. AND PRAKASH, S., 2018. Virtual Optical Encryption Using Phase Shifted Digital Holography and RSA Algorithm. In: 2018 3rd International Conference on Microwave and Photonics, ICMAP 2018. pp.1–2. DOI: 10.1109/ICMAP.2018.8354560

DARWIS, D., PRABOWO, R. AND HOTIMAH, N.,

2018. Kombinasi Gifshuffle, Enkripsi AES dan Kompresi Data Huffman untuk Meningkatkan Keamanan Data. Jurnal Teknologi Informasi dan Ilmu Komputer, 5(4), pp.389–394. DOI: 10.25126/jtiik.201854727

FADLAN, M. AND HADRIANSA, H., 2017. Rekayasa Aplikasi Kriptografi dengan Penerapan Kombinasi Algoritma Knapsack Merkle Hellman dan Affine Cipher. Jurnal Teknologi Informasi dan Ilmu Komputer, 4(4), pp.268–274. DOI: 10.25126/jtiik.201744468

FAHRIANI, N. AND ROSYID, H., 2019. Implementasi Teknik Enkripsi dan Dekripsi di File Video Menggunakan Algoritma Blowfish. Jurnal Teknologi Informasi dan Ilmu Komputer, 6(6), pp.697–702. DOI: 10.25126/jtiik.2019661465

HERLINAH AND KH, M., 2019. Pemrograman Aplikasi Android dengan Android Studio, Photoshop dan Audition. Jakarta: PT Elex Media Komputindo.

ISWARI, N.M.S., 2017. Key Generation Algorithm Design Combination of RSA and ElGamal algorithm. Proceedings of 2016 8th International Conference on Information Technology and Electrical Engineering: Empowering Technology for Better Future, ICITEE 2016. DOI: 10.1109/ICITEED.2016.7863255

JAMALUDIN AND ROMINDO, 2020. Kriptografi: Teknik Hybrid Cryptosystem Menggunakan Kombinasi Vigenere Cipher dan RSA. 1st ed. Yayasan Kita Menulis. Yayasan Kita Menulis.

JUNIAWAN, F.P., 2016. RSA Implementation for Data Transmission Security in BEM Chairman E-voting Android Based Application. In: Proceedings - 2016 1st International Conference on Information Technology, Information Systems and Electrical Engineering, ICITISEE 2016. pp.93–98. DOI: 10.1109/ICITISEE.2016.7803054

LUO, G., ZHANG, M., LI, H. AND LI, G., 2019. A Real-Time Perception Information Security Algorithm in Internet of Things. In: Proceedings - 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics, CISP-BMEI 2019. pp.0–5. DOI: 10.1109/CISP-BMEI48845.2019.8965730

MALLOULI, F., HELLAL, A., SHARIEF SAEED, N. AND ABDULRAHEEM ALZAHRANI, F., 2019. A Survey on Cryptography: Comparative Study between RSA vs ECC Algorithms, and RSA vs El-Gamal Algorithms. In: Proceedings - 6th IEEE International Conference on Cyber Security and Cloud Computing, CSCloud 2019 and 5th IEEE International Conference on Edge Computing and Scalable Cloud, EdgeCom 2019. pp.173–176. DOI: 10.1109/CSCloud/EdgeCom.2019.00022

RANTELINGGI, P.H. AND SAPUTRA, E., 2020. Algoritma Kriptografi Triple DES Dan Steganografi LSB Sebagai Metode Gabungan Dalam Keamanan Data. 7(4), pp.661–666. DOI: 10.25126/jtiik.202071838

RATNADEWI, ADHIE, R.P., HUTAMA, Y., SALEH AHMAR, A. AND SETIAWAN, M.I., 2018. Implementation Cryptography Data Encryption Standard (DES) and Triple Data Encryption Standard (3DES) Method in Communication System Based Near Field Communication (NFC). Journal of Physics: Conference Series, 954(1). DOI: 10.1088/1742-6596/954/1/012009

SANTHOSH KUMAR, B.J., ROSHNI, R.V.K. AND NAIR, A., 2018. Comparative Study on AES and RSA Algorithm for Medical Images. Proceedings of the 2017 IEEE International Conference on Communication and Signal Processing, ICCSP 2017, 2018-Janua, pp.501–504. DOI: 10.1109/ICCSP.2017.8286408

SIAHAAN, V. AND SIANIPAR, R.H., 2020. Java Untuk Keamanan Data. Medan: Balige Publishing.

SIHOTANG, H.T., EFENDI, S., ZAMZAMI, E.M. AND MAWENGKANG, H., 2020. Design and Implementation of Rivest Shamir Adleman's (RSA) Cryptography Algorithm in Text File Data Security. Journal of Physics: Conference Series, 1641(1). DOI: 10.1088/1742-6596/1641/1/012042

SYLFANIA, D.Y., JUNIAWAN, F.P. AND AGUSTI, L., 2019. Implementasi Sistem Informasi Akademik Berbasis Android pada SMA Negeri 1 Tempilang. Jurnal Edukasi dan Penelitian Informatika (JEPIN), 5(3), p.301. DOI: 10.26418/jp.v5i3.33276

SYLFANIA, D.Y., JUNIAWAN, F.P., LAURENTINUS, L. AND PRADANA, H.A., 2019. SMS Security Improvement using RSA in Complaints Application on Regional Head Election's Fraud. Jurnal Teknologi dan Sistem Komputer, 7(3), pp.116–120. DOI: 10.14710/jtsiskom.7.3.2019.116-120

SYLFANIA, D.Y., JUNIAWAN, F.P., LAURENTINUS and PRADANA, H.A., 2020. Blowfish–RSA Comparison Analysis of the Encrypt Decrypt Process in Android-Based Email Application. pp.113–119. DOI: 10.2991/aisr.k.200424.017

SYLFANIA, D.Y., PERKASA, E.B. AND

JUNIAWAN, F.P., 2020. Implementasi E-Complaint Mahasiswa dan Civitas Akademika Berbasis Client Server. Jurnal Informatika, 7(2), pp.205–210. DOI: 10.31294/ji.v7i2.8919

TRIANA, F., ENDRI, J. AND SALAMAH, I., 2020. Implementasi Teknik Kriptografi Caesar Cipher untuk Keamanan Data Informasi Berbasis Android. RESTI (Rekayasa Sistem dan Teknologi Informasi), [online] 4(4), pp.627–634. Available at: <http://jurnal.iaii.or.id/index.php/RESTI/article/view/1984>. DOI: 10.29207/resti.v4i4.1984

TRIHASTUTI, Y. AND KRESNA, I., 2020. Metode Pembayaran Elektronik yang Aman pada Online Shopping. Rekayasa Sistem dan Teknologi Informasi (RESTI), 1(10), pp.319–328. DOI: 10.29207/resti.v4i2.1732

WAHAB, O.F.A., KHALAF, A.A.M., HUSSEIN, A.I. AND HAMED, H.F.A., 2021. Hiding Data Using Efficient Combination of RSA Cryptography, and Compression Steganography Techniques. IEEE Access, 9, pp.31805–31815. DOI: 10.1109/ACCESS.2021.3060317

YE LIU; WEI GONG; WENQING FAN, 2018. Application of AES and RSA Hybrid Algorithm in E-mail. In: International Conference on Computer and Information Science (ICIS). Singapore: IEEE.pp.701–703.DOI: 10.1109/ICIS.2018.8466380

ZHANG, H., YU, J., TIAN, C., TONG, L., LIN, J., GE, L. and WANG, H., 2020. Efficient and Secure Outsourcing Scheme for RSA Decryption in Internet of Things. IEEE Internet of Things Journal, 7(8), pp.6868–6881. DOI: 10.1109/JIOT.2020.2970499