Analisis Pengembangan Aplikasi Menggunakan Algoritma RSA dan *El-Gamal* Pada Teknik Steganografi Dengan Metode *Least Significant Bit (LSB)*

1st Nashwa Abdiel Fauzi
Fakultas Informatika
Universitas Telkom
Purwokerto Indonesia
nashwaabdielfauzi@student.telkomuniveristy.ac.id

2nd Wahyu Adi Prabowo, S.Kom., M.B.A., M.Kom.
Fakultas Informatika
Universitas Telkom
Purwokerto Indonesia
wahyup@telkomuniversity.ac.id

3rd Alon Jala Tirta Segara, S.Kom., M.Kom
Fakultas Informatika
Universitas Telkom
Purwokerto Indonesia
alonhs@telkomuniveristy.ac.id

Abstrak Dalam era digital modern, internet telah menjadi penunjang utama dalam berbagai aktivitas, seperti komunikasi dan pengiriman data, yang sebelumnya dilakukan secara konvensional. Namun, perkembangan ini harus diimbangi dengan sistem keamanan yang baik agar aktivitas digital dapat berjalan lancar. Salah satu upaya untuk menjaga keamanan dan integritas data adalah kombinasi kriptografi dan steganografi. Penelitian ini menganalisis perbandingan algoritma kriptografi RSA dan El-Gamal, yang merupakan algoritma kunci publik, dikombinasikan dengan metode steganografi Least Significant Bit (LSB) serta hash function MD5 untuk sistem keamanan data. Hasil penelitian menunjukkan bahwa algoritma RSA lebih cepat dalam proses enkripsi, sedangkan El-Gamal lebih unggul dalam proses dekripsi. Dari segi akurasi dekripsi, RSA lebih baik karena kepadatan isi file yang lebih tinggi memungkinkan hasil yang lebih sempurna dibandingkan El-Gamal. Untuk validasi sistem, dilakukan pengujian black-box testing yang berfokus pada fungsionalitas sistem, dengan hasil seluruh pengujian berhasil. Dari hasil pengujian, dapat disimpulkan bahwa sistem telah berhasil dibangun sesuai tujuan, yaitu pengamanan data. RSA terbukti lebih efektif dalam menjaga integritas data setelah melalui proses enkripsi dan dekripsi dibandingkan El-Gamal.

Kata kunci*— Kriptografi, Steganografi, RSA, *El-Gamal*, *Hash* MD5, *Least Significant Bit

I. PENDAHULUAN

Kemudahan dalam membagikan dan mendapatkan media digital seperti gambar, audio, video dan teks semakin pesat. Karena adanya perkembangan teknologi ini membuat sebuah informasi menjadi kebutuhan pokok bagi masyarakat atau organisasi. Informasi merupakan suatu hal yang penting bagi masyarakat modern saat ini maupun sebuah perusahaan, karena dengan adanya informasi ini kehidupan manusia modern dapat berjalan dengan baik. Dengan pesatnya perkembangan teknologi informasi, terutama internet, terjadi peningkatan signifikan dalam kasus cybercrime. Salah satu tantangan utama yang dihadapi dalam berbagi dan mentransmisikan berbagai jenis informasi melalui saluran publik adalah masalah keamanan data. Ancaman seperti penyadapan, pencurian, dan pemalsuan informasi melalui jaringan komputer dapat memberikan kerugian besar bagi pemilik informasi[1]. Menurut laporan dari Surfshark, perusahaan *virtual private network* (VPN) asal Belanda, selama Januari 2020 - Januari 2024 ada sekitar 3,96 miliar akun digital yang mengalami kebocoran data[2]. Angka tersebut merupakan estimasi berdasarkan riset Surfshark terhadap kasus kebocoran data di 250 negara. Kebocoran data atau data breach merupakan keadaan dimana data suatu akun digital dapat diakses secara illegal oleh pihak selain pemilik. Data yang bocor tersebut berupa data pribadi, seperti nama lengkap, jenis kelamin, lokasi geografis, alamat email, kata sandi akun, nomor telepon dsb. Oleh karena itu, upaya untuk melindungi informasi yang dikirimkan terhadap seorang penyadap dan pihak yang tidak bertanggungjawab menjadi sebuah kebutuhan. Seperti Kriptografi dan Steganografi sangat disarankan sebagai bentuk upaya keamanan tersebut. Kriptografi adalah seni dan ilmu menjaga kerahasiaan data. Pada kriptografi, data asli diubah menjadi bentuk lain yang tidak dapat dibaca menggunakan suatu algoritma yang menghasilkan suatu key. Steganografi adalah seni dan ilmu menyembunyikan data pada media lain sebagai cover (misalnya citra) sehingga terlihat samar. Steganografi membutuhkan dua properti, yaitu pesan dan media penampung. Media penampung yang umumnya digunakan sekarang dapat berupa teks, suara, gambar, atau video. Sedangkan pesan yang disembunyikan dapat berupa teks, gambar, atau pesan lainnya. Penggabungan steganografi dan kriptografi secara bersamaan dapat meningkatkan pengamanan data.

II. KAJIAN TEORI

A. Keamanan Data

Data merupakan sekumpulan informasi mengenai suatu hal yang perlu dijaga keamanannya. Aspek kemanan suatu data terdiri dari integritas dan autentikasi seiring dengan kemajuan zaman dan teknologi diikuti akan adanya peningkatan ancaman terhadap keamanan. Ancaman bukan hanya terhadap fisik (pencurian, penipuan) namun juga terjadi untuk hal yang sifatnya maya, seperti pesan yang hanya boleh diketahui orang tertentu.

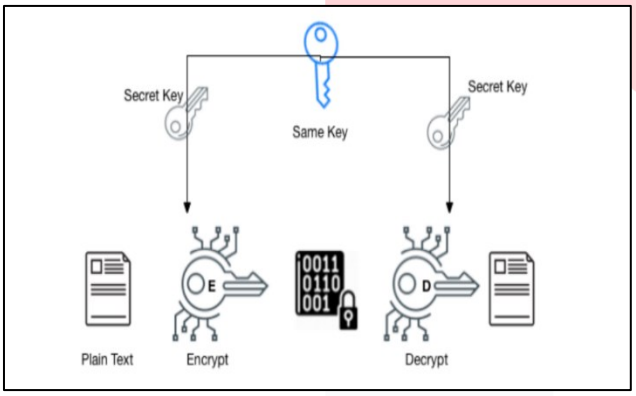
B. Kriptografi

Kriptografi *(Cryptography)* berasal dari Bahasa Yunani, terdiri dari dua suku kata yaitu "kripto" dan "*graphia*" yang mana "kripto" artinya menyembunyikan sedangkan "*graphia*" berarti tulisan[3]. Kriptografi ialah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi, seperti kerahasiaan data,

keabsahan data, integritas data serta autentikasi data. Untuk menjaga suatu data/pesan, maka pesan tersebut diubah menjadi kode yang tidak dapat dimengerti oleh pihak lain. Proses pengubahan pesan yang dapat dimengerti (*plaintext*) menjadi sebuah kode sandi (*ciphertext*) disebut enkripsi. Se dangkan proses pembalikan pengubahan *ciphertext* menjadi *Plaintext* disebut dekripsi. Berdasarkan jenis algoritmanya terdapat dua jenis algoritma kriptografi yaitu algoritma simetri dan algoritma asimetri.

1. Algortima Simetri

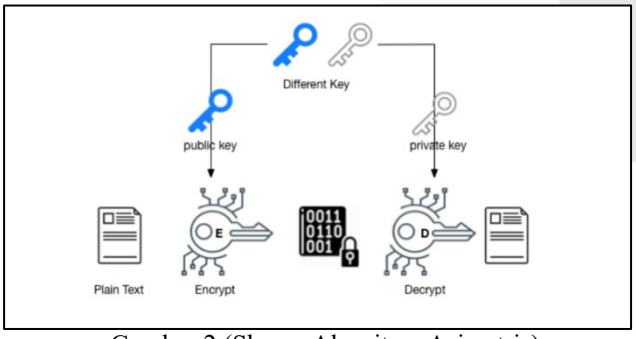
Algoritma simetri merupakan algoritma kriptografi yang menggunakan kunci enkripsi yang sama dengan kunci dekripsinya[4]. Algoritma ini mengharuskan pengirim dan penerima membuat kesepakatan suatu kunci tertentu sebelum mereka memulai untuk saling berkomunikasi/meng irimkan pesan. Kekuatan dari algoritma ini terletak pada bagian kuncinya, apabila terjadi kebocoran kunci maka pihak diluar antara pengirim dan penerima dapat mengetahui isi pesan yang dikirimkan.



GAMBAR 1 (SKEMA ALGORITMA SIMETRI)

2. Algoritma Asimetris

Kebalikan dari algoritma simetris, algoritma asimetris atau biasa disebut algoritma kunci public menggunakan kunci yang berbeda pada proses enkripsi dan dekripsi[5]. Pada algoritma ini kunci enkripsinya dapat dilihat secara umum yang disebut *public key* sedangkan kunci dekripsinya tidak dapat dilihat secara umum atau bersifat rahasia yang disebut *private key*[6].



Gambar 2 (Skema Algoritma Asimetris)

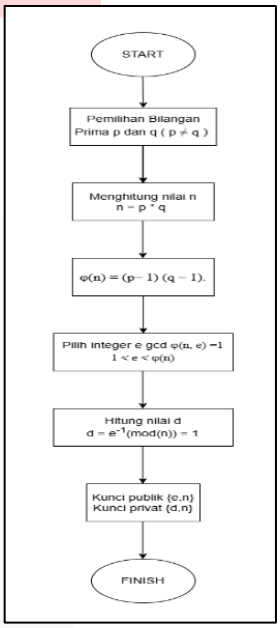
C. Algoritma RSA (River, Shamir, Alderman)

Algoritma kriptografi RSA merupakan algoritma kriptografi kunci public (nirsimetris). Diciptakan pertama kali pada tahun 1977 oleh R. Rivest, A. Shamir dan L. Adleman. Nama RSA diambil dari ketiga nama penciptanya tersebut. Karena termasuk dalam klasifikasi algoritma kunci public maka RSA mempunyai dua kunci, yaitu kunci publik (*public key*) dan kunci rahasia (*private key*). Keamanan Algoritma RSA terletak pada sulitnya dalam memfaktorkan bilangan-bilangan prima besar dari proses pembangkitan sepasang kunci[7]. Dalam tahapan penggunaan algoritma RSA terdiri dari tiga tahap, yaitu:

1. Pembangkitan Kunci

   Berikut adalah tahapan dalam pembangkitan kunci pada Algoritma RSA:

   a) Memilih dua bilangan prima dengan pelabelan variable sebagai p dan q, dimana $p \neq q$.
   b) Hitung nilai n = p . q ($p \neq q$, jika p = q maka nilai n = p^2 sehingga nilai p dapat diperoleh dengan menarik akar pangkat dua dari n).
   c) Dihitung fungsi *Euler's totient* (n) = (p – 1) (q – 1). (1)
   d) Pilih sebuah bilangan bulat sebagai *public key* sebagai variable e yang mana relative prima terhadap (n) artinya faktor pembagi terbesar keduanya adalah 1.
   e) Pembangkitan kunci privat dengan menggunakan persamaan:

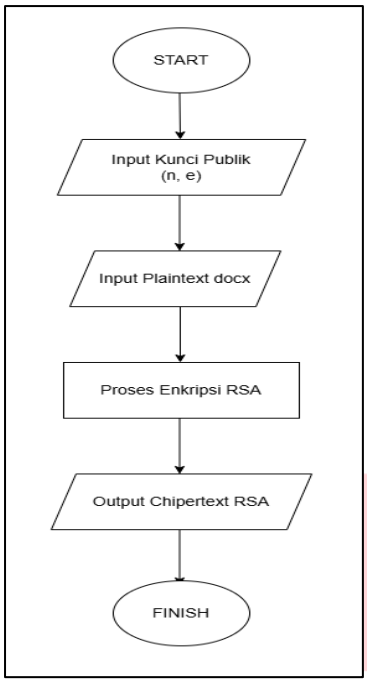


Gambar 3 (Skema Pembangkitan Kunci RSA)

2. Proses Enkripsi RSA

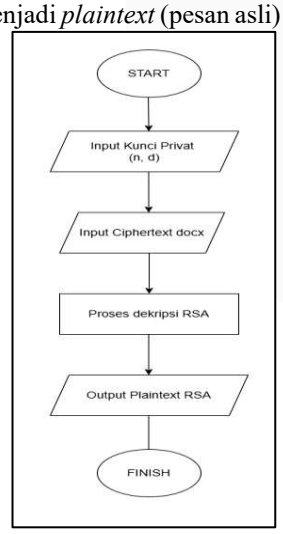
   Berikut adalah tahapan proses enkripsi Algoritma RSA:

   a) Pilih kunci public penerima pesan e dan modulus n (e, n) dimana nilai e harus memenuhi syarat berupa 1 < e < φ(n), φ(n) merupakan fungsi Euler dari modulus n, dan nilai e harus *relative prima* dengan φ(n).
   b) Pilih *plaintext* m dan ubah isi pesan m menjadi pesan dengan nilai bilang ASCII lalu gabungkan nilai m tersebut sebagai bilangan bulat besar.
   c) Bagi pesan menjadi blok pesan m_1, m_2, m_3,…. Yang mana nilai tiap blok adalah $0 \leq m \leq n-1$.
   d) Hitung tiap blok m dengan $c_1 = m_1^e$ mod n
   e) Susun kembali nilai c dari hasil enkripsi berurut c_1 $c_2 c_3$ ……. c_n sebagai perolehan hasil *ciphertext* dari pesan m.



GAMBAR 4 (SKEMA ENKRIPSI RSA)

3. Proses Dekripsi RSA
   Pada proses dekripsi dengan menggunakan algoritma RSA terdapat beberapa tahapan berikut:
   a) Susun pesan *ciphertext* yang telah diterima.
   b) Gunakan kunci rahasia (d) dan modulus (n) atau (d, n).
   c) Potong/bagi pesan menjadi tiap bagian c_1 $c_2 c_3$ ……. c_n dengan nilai tiap bagiannya $0 \leq c \leq n-1$.
   d) Hitung tiap blok/bagian dengan $m = c_1^d$ mod n
   e) Susun kembali nilai m hasil dekripsi beurut $m_1, m_2, m_3, \ldots m_n$ sehingga tersusun kembali menjadi *plaintext* (pesan asli) dari cipherteks.

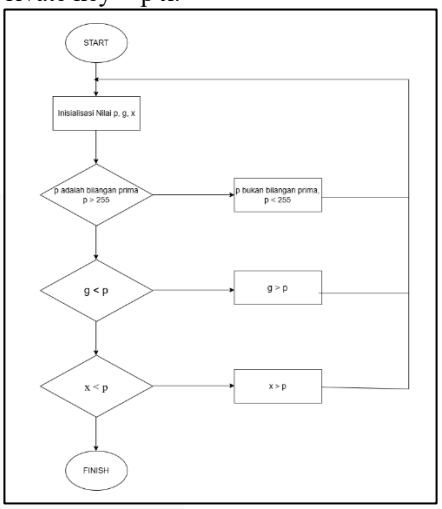


GAMBAR 5 (SKEMA DEKRIPSI RSA)

D. Algoritma El-Gamal

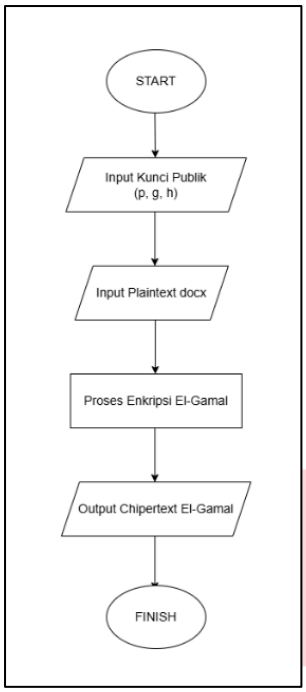
Algoritma *El-Gamal* merupakan algoritma kriptografi asimetris yang menggunakan 2 jenis kunci, yaitu *public key* dan *private key*. Algoritma ini ditemukan oleh Taher *El-Gamal* pada tahun 1985. Algoritma ini menjadi metode alternatif untuk cipher kunci public Algoritma RSA. Perbedaan mendasar Algoritma *El-Gamal* dengan Algoritma RSA terletak pada algoritma RSA keamanannya bergantung pada kesulitan faktorisasi bilangan prima besar, sedangkan Algoritma *El-Gamal* tingkat keamanannya terletak pada kesulitan perhitungan modulus logaritmik diskrit dari bilangan prima besar. Keunggulan dari Algoritma *El-Gamal* juga ada pada pesan teks yang sama mampu menghasilkan pesan teks rahasia unik yaitu pesan rahasia akan selalu berbeda setiap di-enkripsi[8]. Dalam penerapannya, terdapat 3 tahapan Algoritma El-Gamal yaitu:

1. Pembangkitan Kunci
   Berikut adalah tahapan dalam pembangkitan kunci pada Algoritma El-Gamal:
   a) Memilih bilangan prima dengan pelabelan variable sebagai p, p adalah bilangan prima, p > 255.
   b) Memilih dua buah bilangan acak (g dan x), dengan syarat bahwa nilai g < p, x<p. Dengan persamaan: Y = g^x mod p (2)
   c) Public key = y g p.
   d) Private key = p x.



GAMBAR 6 (SKEMA PEMBANGKITAN KUNCI EL-GAMAL)

2. Proses Enkripsi El-Gamal
   Berikut adalah tahapan proses enkripsi Algoritma El-Gamal:
   a) Bagi plaintext menjadi masing-masing blok/bagian m_(1,) m_(2,) m_(3,)….m_n.
   b) Ubah nilai blok kedalam nilai bilangan ASCII.
   c) Tentukan bilangan acak (k), dengan $1 \leq k \leq p - 2$. Nilai k digunakan untuk mencari nilai a dan b.
   d) Enkripsi tiap blok m dengan persamaan:
      y = g^k mod p
      Δ = y^k. m mod p (3)
   e) Susun kembali blok hasil enkripsi yang sudah menjadi *ciphertext* berurutan $a_1, b_1, a_1\ b_2, a_3, b_3, \ldots\ldots a_n, b_n$.



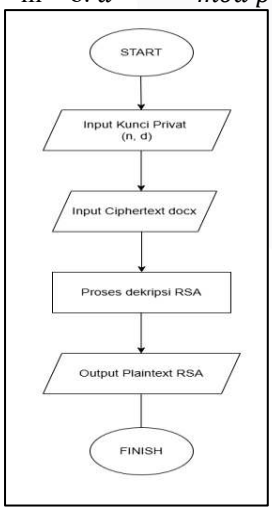
GAMBAR 7 (SKEMA ENKRIPSI EL-GAMAL)

3. Proses Dekripsi El-Gamal

Pada proses dekripsi dengan menggunakan algoritma El-Gamal terdapat beberapa tahapan berikut:

a) Susun pesan *ciphertext* yang telah diterima.
b) Gunakan kunci rahasia (d) dan modulus (n) atau (d, n).
c) Potong/bagi pesan menjadi tiap bagian $c_1\, c_2 c_3$ ……. c_n dengan nilai tiap bagiannya $0 \leq c \leq$ n-1.
d) Hitung tiap blok/bagian dengan $m = c_1^d$ mod n
e) Susun kembali nilai m hasil dekripsi beurut m_1, m_2, m_3,….m_n sehingga tersusun kembali menjadi *plaintext* (pesan asli) dari cipherteks.

$$m = b.\ a^{(p-1-x)}\ mod\ p \qquad (4)$$

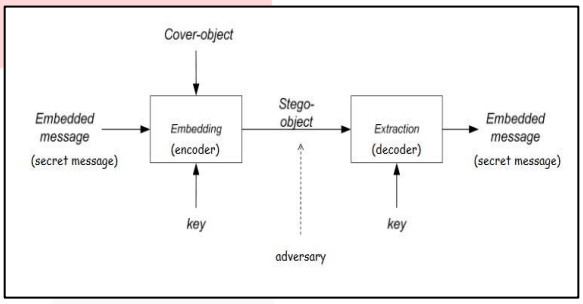


GAMBAR 8 SKEMA DEKRIPSI EL-GAMAL

E. Steganografi

Steganografi merupakan ilmu dan seni penyembunyian pesan kedalam pesan lain sehingga keberadaan pesan awal tidak diketahui kebereadaannya. Steganografi berasal dari Bahasa Yunani "Steganos" yang berarti tulisan tersembunyi. Steganografi berbeda dengan kriptografi, karena kriptografi hanya mengubah bentuk pesan asli menjadi pesa rahasia sedangkan steganografi memindahkan keberadaan pesan asli kedalam pesan lain sebagai penampung. Namun dua hal ini dapat menjadi suatu hal yang membantu dalam proses keamanan karena pada dasarnya keamanan akan semakin baik ketika kedua metode ini digabungkan, yaitu pesan dienkripsi terlebih dahulu kemudia disembunyikan kedalam media lain sehingga tidak diketahui keberadaannya. Steganografi memiliki beberapa properti untuk prosesnya dapat berjalan, yaitu:

a) *Embedded message* (*Hidden Text*): pesan rahasia yang akan disembunyikan pada media. Pesan ini dapat berupa text, gambar, audio, video, dll.
b) *Cover object* (cover text): pesan yang digunakan sebagai media penampung pesan rahasia. Sama halnya dengan *Hidden Text*, pesan pada cover text pun dapat berupa text, gambar, audio, video, dll.
c) *Stego object* (stego text): pesan yang didalamnya sudah terdapat pesan rahasia/*Embedded message*.



GAMBAR 9 (SKEMA STEGANOGRAFI)

F. *Least Significant Bit (LSB)*

Dalam praktiknya penggunaan metode steganografi audio terdapat beberapa macam diantaranya *LSB, Phase Encoding, Spread Spectrum dan Echo Hiding.* LSB merupakan salah satu metode steganografi yang sering diterapkan, karena tingkat kompleksitasnya yang tidak terlalu tinggi dan mudah untuk diimplementasikan. Mekanisme dari metode ini ialah memodifikasi bit-bit yang termasuk bit LSB atau memiliki nilai potensial paling sedikit, biasanya terletak pada bit yang posisinya paling kanan. Setelah bit potensial sudah ditemukan, nantinya pesan rahasia akan mengisi bit-bit tersebut dan menggantikannya sebagai tempat persembunyian sehingga tidak dapat diketahui secara langsung. Untuk proses ekstraksi dilakukan dengan mengembalikkan nilai bit yang sebelumnya digunakan untuk menggantikan pesan rahasia atau data, pada proses ekstaksi bit tersebut dikembalikan dengan bit awal dari data sehingga pesan dapat kembali terbaca[9].

G. File (*Docx)

Merupakan format ekstensi file yang digunakan di aplikasi Microsoft word. Terdapat perkembangan dengan format serupa yang awalnya berformat *docs* yang digunakan Microsoft word 2003 dan versi sebelumnya, sedangkan format *(*docx)* digunakan pada Microsoft word 2007 dan versi sesudahnya. Salah satu kendala terletak pada kompatibilitas untuk pengguna Microsoft word versi 2003 dan sebelumnya pada saat menerima file *(*docx)* dimana file tidak dapat dibuka namun sebaliknya pengguna Microsoft word versi 2007 dan setelahnya dapat membuka file *docs*[10].

H. *Waveform (WAV)*

Merupakan bentuk format file yang fleksibel untuk menyimpan semua kombinasi audio baik rates maupun bitrates. Hal ini menyebabkan format ini sangat disarankan dan layak untuk menyimpan dan mengarsipkan rekaman asli. File *WAV* menggunakan struktur standar RIFF dengan mengelompokan isi file kedalam bagian-bagian seperti format WAV dan data digital audio. Struktur RIFF (*Resource Interchange File Format*) yaitu struktur yang biasa digunakan untuk data multimedia dalam *windows*, struktur file ini mengatur data dalam file ke dalam bagian-bagian yang masing-masing memiliki *header* dan ukuran yang disebut sebagai *chunk*[11].

I. *Message Digest 5 (MD5)*

Fungsi *hash* (prosedur terdefinisi yang mengubah variabel dari suatu data yang berukuran besar menjadi lebih sederhana) *MD5* merupakan salah satu fungsi *hash* yang digunakan untuk keperluan kriptografi. *MD5 (Message-Diggest Algorithm 5)* ialah fungsi *hash* yang digunakan secara luas dengan nilai *hash* 128-bit. *MD5* didesain oleh Ronald Rivest, salah satu pencipta Algoritma RSA untuk menggantikan fungsi *hash* sebelumnya yaitu *MD4. MD5* merupakan fungsi *hash* yang memproses pesan dengan panjang variabel dan mengubahnya menjadi output dengan panjang tetap sebesar 128 bit. Algoritma ini bekerja dengan membagi pesan ke dalam blok-blok berukuran 512 bit, yang kemudian diproses melalui empat putaran. Setiap putaran secara bergantian mengolah 16 sub-blok, di mana masing-masing sub-blok memiliki ukuran 32 bit[12]. Enkripsi dengan menerapkan metode *MD5* dianggap kuat karena enkripsi yang dihasilkan bersifat *"one way hash"*.

J. *Python*

Python adalah salah satu bahasa pemrograman yang banyak dimanfaatkan oleh perusahaan besar serta para *developer* untuk mengembangkan berbagai jenis aplikasi, baik berbasis desktop, web, maupun *mobile*. Bahasa pemrograman ini dikembangkan oleh *Guido van Rossum* di Belanda pada tahun 1990, dan namanya terinspirasi dari acara televisi favorit Guido, yaitu *Monty Python's Flying Circus*. *Python* dipakai menjadi bahasa pemrograman yang dipakai secara luas dalam industri dan Pendidikan karena ringkas, sintaks sintuitif dan memiliki Pustaka atau *library* yang luas[13].

K. *Peak Signal to Ratio (PSNR)*

PSNR merupakan parameter perbandingan antara nilai maksimum dari sinyal yang diukur dengan besarnya derau yang berpengaruh pada sinyal dalam satuan desibel (dB). Semakin besar nilai PSNR maka semakin mirip dengan citra asli. Secara umum besaran PSNR berada pada kisaran dibawah 30 dB termasuk dalam *low quality*[14]. Untuk menentukan nilai PSNR digunakan persamaan berikut:

$$\text{PSNR} = 10\ Log_{10}\left(\frac{P1^2}{P1^2+P0^2-2P1P0}\right) \quad (5)$$

Ket:

P1 = kekuatan sinyal berkas audio setelah proses penyisipan pesan

P0 = kekuatan sinyal awal

L. *Signal to Noise Ratio (SNR)*

Ukuran perbandingan pada tingkat kekuatan sinyal yang diinginkan dengan tingkat kebisingan *(noise)* yang tidak diinginkan dalam suatu komunikasi analog dan digital disebut *Signal to Noise Ratio*. SNR biasanya dinyatakan dalam nilai numerik dengan satuan deibel (dB), rasio yang nilainya lebih tinggi menunjukkan kualitas sinyal lebih baik[15]. Untuk mendapatkan nilai SNR diterapkan persamaan berikut:

$$\text{SNR} = 10\ Log_{10}\left(\frac{Psinyal}{Pderau}\right) \quad (6)$$

Ket:

P sinyal = nilai daya dari sinyal yang diinginkan

P derau = nilai daya dari derau/bising

M. *Bit Error Rate (BER)*

Nilai jumlah kesalahan bit per unit waktu atau jumlah kesalahan bit dibagi dengan jumlah bit yang dikirimkan selama interval waktu yang dilakukan. BER tidak memiliki ukuran satuan, namun seringkali diekspresikan dalam persentase. Nilai dari BER yang lebih rendah menunjukkan kualitas sinyal yang lebih baik dan menjadi parameter penting untuk mengevaluasi kinerja sistem komunikasi[15]. Persamaan BER adalah sebagai berikut:

$$BER = \frac{Jumlah\ bit\ salah}{jumlah\ bit\ yang\ diterima} \quad (7)$$

N. *Blackbox Testing*

Blackbox testing yaitu teknik pengujian suatu perangkat lunak yang berfokus pada kinerja fungsionalitas dari perangkat yang diuji. Pengujian ini tergolong pengujian yang sederhana namun penting untuk dilakukan. Pengembangan suatu perangkat lunak perlu dilakukan pengujian terlebih dahulu guna memvalidasi dan memverifikasi bahwa program yang dibuat sudah sesuai dengan kebutuhan. Salah satunya dengan menerapkan metode *blackbox testing*. Mekanisme pengujian dilakukan dengan cara memberikan input *user* yang akan menghasilkan keluaran yang sesuai dengan fungsi perangkat. Keunggulan dari penerapan blackbox testing adalah penguji tidak memerlukan pemahaman yang mendalam tentang bahasa pemrograman tertentu. Pengujian dilakukan dari perspektif pengguna, sehingga dapat membantu mengidentifikasi inkonsistensi dalam persyaratan sistem.

III. METODE

Metode yang digunakan pada penelitian ini yaitu menerapkan metode blackbox testing sebagai teknik pengujian sistem yang telah dibuat. Dimana aplikasi yang telah dibangun dilakukan pengujian fungsionalitas pada fitur yang ada didalamnya. Kemudian juga dilakukan pengujian kinerja sistem yang berfokus pada teknik atau algoritma yang digunakan berdasarkan parameter-parameter yang telah ditentukan.

IV. HASIL DAN PEMBAHASAN

Hasil pada penelitian yaitu merupakan aplikasi kriptografi dan steganografi yang bertujuan untuk mengamankan data dengan menerapkan Algoritma RSA dan *El-Gamal* serta metode *Least Significant Bit (LSB)* untuk menyisipkan pesan. Objek yang digunakan sebagai pengujian ini yaitu file (*docx) sebagai file pesan rahasia yang enkripsi dengan algoritma RSA dan El-Gamal. Kemudian cover audio atau file media penampung dari pesan rahasian untuk menyembunyikan pesan rahasia yaitu file audio berformat

(*wav). Namun untuk menvalidasi bahwa system telah berjalan sesuai dengan tujuan awal yaitu mengamankan data dilakukan pengujian dengan blacbox testing dan pengujian kinerja system. Berikut tampilan dari aplikasi yang telah dibuat:

A. Tampilan Utama Aplikasi

GAMBAR 10 (TAMPILAN UTAMA APLIKASI)

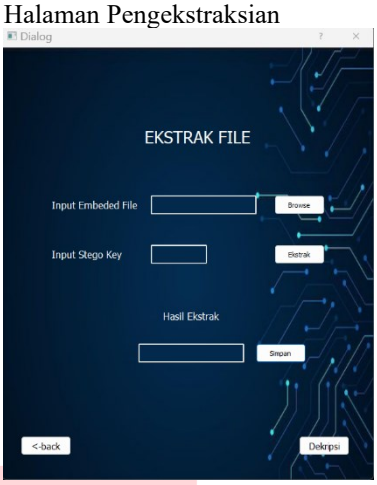
Pada halaman utama ini *user* dapat memilih untuk langsung melakukan *embed* atau penyisipan tanpa dienkripsi dengan algoritma kriptografi dahulu atau tidak. Jika *user* memilih tombol ekstraksi, nantinya *user* akan diarahkan pada pemilihan algoritma kriptografi yang ingin diterapkan.

B. Halaman Penyisipan

GAMBAR 11 (TAMPILAN EMBED APLIKASI)

Pada halaman ini, pengguna dapat memulai proses embedding atau penyisipan file sebagai langkah untuk mengamankan data dengan menyembunyikannya di dalam suatu *cover object*. Proses ini bertujuan untuk menjaga kerahasiaan informasi yang disisipkan, sehingga tidak mudah terdeteksi oleh pihak yang tidak berwenang. Metode yang digunakan dalam tahapan ini adalah steganografi *Least Significant Bit (LSB),* di mana data disisipkan pada bit-bit paling tidak signifikan dalam representasi biner dari *cover object,* sehingga perubahan yang terjadi tidak terlihat secara kasat mata dan tidak memengaruhi kualitas keseluruhan dari file yang digunakan sebagai media penyisipan.

C. Halaman Pengekstraksian



GAMBAR 12 (TAMPILAN EKSTRAK APLIKASI)

Pada Gambar 12 ini proses ekstraksi dimulai. Tahapan ini bertujuan untuk mengeluarkan file hasil penyisipan yang sudah dilakukan sebelumnya. Proses diawali dengan menginputkan *Stego audio* yaitu file audio yang sudah disisipkan. Kemudian menginputkan kunci yang sama seperti yang diinputkan pada proses penyisipan sebelumnya.

D. Halaman Kriptografi RSA

GAMBAR 13 (TAMPILAN HALAMAN KRIPTOGRAFI RSA)

Pada Gambar 13 yaitu halaman pengguna akan diberikan beberapa pilihan fungsionalitas utama, yaitu Enkripsi, Ekstrak, dan Dekripsi, yang dirancang untuk memberikan kemudahan dalam mengelola dan memproses data sesuai dengan kebutuhan pengguna. Setiap pilihan memiliki tujuan dan fungsi yang berbeda, yang memungkinkan pengguna untuk memilih sesuai dengan tujuan spesifik mereka. Fungsi Enkripsi memungkinkan pengguna untuk mengamankan data dengan mengubahnya menjadi bentuk yang tidak dapat dibaca tanpa kunci yang tepat. Fungsi Ekstrak memberikan kemudahan bagi pengguna untuk mengekstraksi data atau informasi tertentu dari file yang sudah terenkripsi atau diproses. Sementara itu, fungsi Dekripsi memungkinkan pengguna untuk mengembalikan data terenkripsi ke bentuk aslinya, sehingga dapat dibaca dan digunakan kembali. Penerapan ketiga pilihan fungsionalitas ini bertujuan untuk meningkatkan fleksibilitas aplikasi, sehingga pengguna dapat memilih proses yang sesuai dengan kebutuhan mereka, baik untuk mengamankan data, mengambil informasi yang diperlukan, atau mengembalikan data ke bentuk yang dapat dibaca.

E. Halaman Kriptografi El-Gamal

GAMBAR 14 (TAMPILAN HALAMAN KRIPTOGRAFI RSA)

Pada Gambar 14 halaman ini menjelaskan *user* akan diberikan beberapa pilihan yaitu Enkripsi, Ekstrak dan Dekripsi. Penerapan beberapa pilihan ini bertujuan untuk memudahkan *user* dan meningkatkan nilai fleksibilitas aplikasi bergantung pada tujuan *user*.

F. Pengujian Blacbkox Testing

Tahapan pengujian ini menerapkan salah satu metode pengujian yaitu *blackbox testing*. Dimana pengujian dilakukan kepada dosen atau experts dalam bidang kriptografi steganografi, dalam penelitian ini melakukan uji coba dengan dosen sebagai tester. Berikut beberapa kasus pengujian:

TABEL 1 (PENGUJIAN BLACBOX TESTING HALAMAN KRIPTOGRAFI RSA)

No	Test Case	Hasil yang Diharapkan	Hasil Pengujian
1.	Menginputkan bilangan prima pada halaman "pembangkitan kunci RSA"	Sistem akan generate kunci publik dan privat dari inputan bilangan prima	Berhasil
2.	Melanjutkan proses enkripsi setelah pembangkitan kunci RSA	Tombol enkripsi pada halaman pembangkitan kunci akan otomatis membawa pada halaman enkripsi beserta kunci publik tercantum	Berhasil
3.	Menginputkan file yang ingin dienkripsi berformat *(*docx)* pada halaman "Enkripsi RSA"	Sistem akan mengenkripsi file *(*docx)* yang diinputkan menjadi *ciphertext*	Berhasil
4.	Menyimpan hasil enkripsi RSA	Sistem akan menampilkan jendela untuk menuliskan nama file hasil enkripsi sesuai yang diinginkan dalam format *(*docx)* dan menyimpan file	Berhasil
5.	Mendekripsi pesan *ciphertext* RSA pada halaman "Dekripsi RSA"	Sistem akan memberikan kesempatan *user* untuk menginputkan file *ciphertext* RSA berformat *(*docx)*	
6.	Menginputkan kunci privat dengan variabel "n" dan "d"	Sistem akan menampilkan kunci privat hasil inputan *user* untuk melanjutkan proses dekripsi pada kolom input kunci rivat dan modulus n	Berhasil
7.	Menyimpan hasil dekripsi RSA	Sistem akan menampilkan jendela untuk menuliskan nama file hasil dekripsi sesuai yang diinginkan dalam format *(*docx)* dan menyimpan file	Berhasil

Pada Tabel 1 diatas menjelaskan sistem yang diuji memenuhi semua kebutuhan fungsional berdasarkan skenario uji yang dirancang. Implementasi algoritma RSA berjalan dengan baik, stabil, dan sesuai dengan spesifikasi yang diharapkan. Sistem dapat diandalkan untuk pembangkitan kunci, enkripsi, dekripsi, serta pengelolaan file secara efektif.

TABEL 2 (PENGUJIAN BLACBOX TESTING HALAMAN KRIPTOGRAFI EL-GAMAL)

No	Test Case	Hasil yang Diharapkan	Hasil Pengujian
1.	Menginputkan sepasang bilangan prima pada halaman "pembangkitan kunci *El-Gamal*"	Sistem akan generate kunci publik dan privat dari inputan bilangan prima	Berhasil
2.	Melanjutkan proses enkripsi setelah pembangkitan kunci *El-Gamal*.	Tombol enkripsi pada halaman pembangkitan kunci akan otomatis membawa pada halaman enkripsi beserta kunci publik tercantum	Berhasil
3.	Menginputkan file yang ingin dienkripsi berformat *(*docx)* pada halaman "Enkripsi *El-Gamal*".	Sistem akan mengenkripsi file *(*docx)* yang diinputkan menjadi *cipher text*	Berhasil
4.	Menyimpan hasil enkripsi *El-Gamal*.	Sistem akan menampilkan jendela untuk menuliskan nama file hasil enkripsi sesuai yang diinginkan	Berhasil

No	Test Case	Hasil yang Diharapkan	Hasil Pengujian
		dalam format *(*docx)* dan menyimpan file	
5.	Mendekripsi pesan *ciphertext* RSA pada halaman "Dekripsi *El Gamal*".	Sistem akan memberikan kesempatan *user* untuk menginputkan file *ciphertext ElGamal* berformat *(*docx)*	Berhasil
6.	Menginputkan kunci privat dengan variabel "p" dan "x".	Sistem akan menampilkan kunci privat hasil inputan *user* untuk melanjutkan proses dekripsi pada kolom input kunci privat dengan variable "p" dan "x"	Berhasil
7.	Menyimpan hasil dekripsi *El-Gamal*.	Sistem akan menampilkan jendela untuk menuliskan nama file hasil dekripsi sesuai yang diinginkan dalamformat *(*docx)* dan menyimpan file	Berhasil

Pada Tabel 2 berisi hasil pengujian sistem terkait dengan implementasi algoritma kriptografi ElGamal, mulai dari pembangkitan kunci, enkripsi, hingga dekripsi data. Semua test case berhasil dijalankan dengan hasil yang sesuai dengan harapan, yang menunjukkan sistem berfungsi dengan baik dan stabil tanpa kendala.

TABEL 3 PENGUJIAN BLACBOX TESTING HALAMAN STEGANOGRAFI

No	Test Case	Hasil yang Diharapkan	Hasil Pengujian
1.	Menginputkan file pesan rahasia berformat *(*docx)* pada "halaman *embed*"	Sistem akan menampilkan jendela untuk memilih file *(*docx)* yang diinginkan untuk proses penyisipan.	Berhasil
2.	Menginputkan file audio berformat wav sebagai *cover audio* pada "halaman *embed*"	Sistem akan menampilkan jendela untuk memilih file audio berformat wav yang diinginkan sebagai *cover audio* untuk proses penyisipan.	Berhasil
3.	Menginputkan *Stego Key* berupa digit angka random pada "halaman *embed*"	Sistem akan memvalidasi *Stego Key* dengan memberikan nilai *hash MD5* untuk proses pengekstraksian file	Berhasil
4.	Menekan tombol *embed* setelah inputan terpenuhi	Sistem akan menampilkan nilai dari PSNR dan SNR audio pada proses penyisipan dan memberikan keterangan "proses *embedding* berhasil"	Berhasil
5.	Menyimpan file hasil penyisipan	Dengan menekan simpan maka sistem akan menampilkan jendela untuk menuliskan nama file hasil *embed* sesuai yang diinginkan dan berhasil disimpan	Berhasil
6.	Menginputkan file *stego audio* berformat wav pada "halaman ekstrak"	Sistem akan menampilkan jendela untuk memilih file audio hasil penyisipan berformat wav	Berhasil
7.	Menginputkan *Stego Key* yang digunakan pada proses penyisipan pada "halaman ekstrak"	Ketika menekan ekstrak maka system akan menampilkan hasil *hash Stego Key* untuk validasi kunci dan nilai BER audio hasil ekstraksi	Berhasil

Pada Tabel 3 ini berisi hasil pengujian terkait aplikasi steganografi berbasis file audio yang melibatkan proses embedding dan ekstraksi file berformat (*DOCX). Setiap test case berhasil dijalankan dengan hasil yang sesuai harapan, menunjukkan bahwa aplikasi dapat menangani setiap skenario dengan baik tanpa kendala berarti.

G. Pengujian Kinerja Sistem

Tahapan pengujian sistem ini merupakan tahapan yang bertujuan untuk menilai tingkat keberhasilan dan keefektifan dari penerapan algoritma dan metode pada system aplikasi yang telah dirancang. Dimana pengujian dilakukan kepada dosen atau experts dalam bidang kriptografi steganografi, dalam penelitian ini melakukan uji coba dengan expert sekaligus dosen sebagai *tester*. Faktor-faktor keberhasilan dari aplikasi yang diharapkan adalah integrasi file yang kuat, kualitas file yang baik setelah melalui beberapa tahapan proses yang diterapkan pada sistem dan keaslian file yang diproses. Media yang digunakan adalah file audio berformat (*wav) dan file berformat (*docx) sebagai Plaintext dan ciphertext. Berikut rinciannya:

Tahapan pengujian steganografi ini mengacu pada hasil dari proses penyisipan dan ekstraksi. Pengujian dari steganografi ini menerapakan beberapa parameter pengujian, yaitu PSNR, SNR dan BER.

TABEL 4 BAHAN FILE AUDIO

Nama File	Durasi (s)	Ukuran File (bytes)
audio pertama.wav	00.01.15	13316236
audio kedua.wav	00.00.15	2646094
audio ketiga.wav	00.00.43	610446
audio keempat.wav	00.00.30	5435470

TABEL 5 (BAHAN FILE DOCX)

1) Pengujian Parameter PSNR Audio *Embed* RSA dan *El-Gamal*

Pengujian Peak Signal-to-Noise Ratio (PSNR) ini dilakukan selama proses penyisipan ciphertext ke dalam cover audio untuk mengukur kualitas sinyal setelah proses embedding. Semakin tinggi nilai PSNR yang diperoleh, maka semakin baik kualitas stego audio yang dihasilkan, karena perbedaan antara cover audio asli dan stego audio menjadi lebih kecil, sehingga perubahan akibat proses penyisipan data tidak terlalu terlihat atau terdengar. Secara umum, nilai PSNR yang lebih besar menunjukkan bahwa kualitas audio tetap terjaga meskipun telah mengalami proses embedding. Sebaliknya, apabila nilai PSNR berada di bawah 30 dB, maka kualitas stego audio dapat dikategorikan sebagai rendah atau low quality, yang berarti terdapat degradasi signifikan dalam kualitas suara akibat proses penyisipan ciphertext.

Tabel 6 (Pengujian PSNR File Audio RSA)

No	Cover Audio	Stego Audio	Ciphertext	Ukuran Files (Bytes)		PSNR(dB)
				Cover audio	*Stego audio*	
1	audio pertama.wav	audio pertama pesan pertama.wav	pesan pertama *encrypted.(*docx)*	13316236	13316236	67.69
2	audio pertama.wav	audio pertama pesan kedua.wav	pesan *backup encrypted.(*docx)*	13316236	13316236	67.05
3	audio pertama.wav	audio pertama pesan ketiga.wav	pesan lirik *encrypted.(*docx)*	13316236	13316236	67.20
4	audio pertama.wav	audio pertama pesan kelima.wav	pesan kelima *encrypted.(*docx)*	13316236	13316236	67.61
5	audio kedua.wav	audio kedua pesan pertama.wav	pesan pertama *encrypted.(*docx)*	2646094	2646094	60.69
6	audio kedua.wav	audio kedua pesan kedua.wav	pesan *backup encrypted.(*docx)*	2646094	2646094	60.10
7	audio kedua.wav	audio kedua pesan ketiga.wav	pesan lirik *encrypted.(*docx)*	2646094	2646094	60.26
8	audio kedua.wav	audio kedua pesan kelima.wav	pesan kelima *encrypted.(*docx)*	2646094	2646094	60.60
9	audio ketiga.wav	audio ketiga pesan pertama.wav	pesan pertama *encrypted.(*docx)*	7610446	7610446	65.28
10	audio ketiga.wav	audio ketiga pesan *backup*.wav	pesan *backup encrypted.(*docx)*	7610446	7610446	64.69
11	audio ketiga.wav	audio ketiga pesan lirik.wav	Pesan lirik *encrypted.(*docx)*	7610446	7610446	64.83
12	audio ketiga.wav	audio ketiga pesan kelima.wav	pesan kelima *encrypted.(*docx)*	7610446	7610446	65.20
13	audio keempat.wav	audio keempat pesan pertama.wav	pesan pertama *encrypted.(*docx)*	5435470	5435470	63.78
14	audio keempat.wav	audio keempat pesan *backup*.wav	pesan *backup encrypted.(*docx)*	5435470	5435470	63.22
15	audio keempat.wav	audio keempat pesan lirik.wav	pesan lirik *encrypted.(*docx)*	5435470	5435470	63.37
16	audio keempat.wav	audio keempat pesan kelima.wav	pesan kelima *encrypted.(*docx)*	5435470	5435470	63.69

Pada Tabel 6 di atas ditampilkan nilai Peak Signal-to-Noise Ratio (PSNR) untuk setiap file audio cover yang telah disisipi pesan rahasia. Nilai PSNR ini digunakan sebagai indikator kualitas audio setelah proses penyisipan, di mana semakin tinggi nilai PSNR, semakin kecil tingkat distorsi yang terjadi pada file audio hasil embedding dibandingkan dengan file audio aslinya. Dari data yang ditampilkan dalam tabel, rentang nilai PSNR yang diperoleh berada antara 60.10 dB hingga 67.69 dB. Rentang ini menunjukkan bahwa kualitas audio setelah proses embedding masih berada dalam kategori yang baik, dengan perbedaan yang relatif kecil antara audio yang telah disisipi pesan rahasia dan versi aslinya.

Nama File	Ukuran File (bytes)
Pesan pertama.*(*docx)*	12265
Pesan *backup*.*(*docx)*	15242
Pesan lirik.*(*docx)*	14554
Pesan kelima.*(*docx)*	12762

TABEL 7 (PENGUJIAN PSNR FILE AUDIO EL-GAMAL)

No	Cover Audio	Stego Audio	Ciphertext	Ukuran Files (Bytes)		PSNR(dB)
				Cover audio	*Stego audio*	
1	audio pertama.wav	audio pertama pesan pertama.wav	pesan pertama *encrypted.(*docx)*	13316236	13316140	67.40
2	audio pertama.wav	audio pertama pesan *backup*.wav	pesan *backup* elgamal *encrypted.(*docx)*	13316236	13316140	66.91
3	audio pertama.wav	audio pertama pesan lirik.wav	pesan lirik elgamal *encrypted.(*docx)*	13316236	13316140	67.18
4	audio pertama.wav	audio pertama pesan kelima.wav	pesan kelima *encrypted.(*docx)*	13316236	13316140	67.61
5	audio kedua.wav	audio kedua pesan pertama.wav	pesan pertama *encrypted.(*docx)*	2646094	2646060	60.69
6	audio kedua.wav	audio kedua pesan *backup*.wav	pesan *backup* elgamal *encrypted.(*docx)*	2646094	2646060	59.87
7	audio kedua.wav	audio kedua pesan lirik.wav	pesan lirik elgamal *encrypted.(*docx)*	2646094	2646060	60.17
8	audio kedua.wav	audio kedua pesan kelima.wav	pesan kelima *encrypted.(*docx)*	2646094	2646060	60.60
9	audio ketiga.wav	audio ketiga pesan pertama.wav	pesan pertama *encrypted.(*docx)*	7610446	7610412	65.28
10	audio ketiga.wav	audio ketiga pesan *backup*.wav	pesan *backup* elgamal *encrypted.(*docx)*	7610446	7610412	64.46
11	audio ketiga.wav	audio ketiga pesan lirik.wav	pesan lirik elgamal *encrypted.(*docx)*	7610446	7610412	64.75
12	audio ketiga.wav	audio ketiga pesan kelima.wav	pesan kelima *encrypted.(*docx)*	7610446	7610412	65.20
13	audio keempat.wav	audio keempat pesan pertama.wav	pesan pertama *encrypted.(*docx)*	5435470	5435436	63.78
14	audio keempat.wav	audio keempat pesan *backup*.wav	pesan *backup* elgamal *encrypted.(*docx)*	5435470	5435436	63.00
15	audio keempat.wav	audio keempat pesan lirik.wav	pesan lirik *encrypted.(*docx)*	5435470	5435436	63.30
16	audio keempat.wav	audio keempat pesan kelima.wav	pesan kelima *encrypted.(*docx)*	5435470	5435436	63.69

Pada Tabel 7 di atas, ditampilkan nilai Peak Signal-to-Noise Ratio (PSNR) untuk setiap file audio cover yang telah disisipi pesan rahasia. Nilai PSNR ini berfungsi sebagai ukuran kualitas audio setelah proses embedding, di mana semakin tinggi nilainya, semakin kecil tingkat distorsi yang terjadi akibat penyisipan pesan. Berdasarkan tabel tersebut, rentang nilai PSNR yang diperoleh berada antara 59.87 dB hingga 67.40 dB. Hal ini menunjukkan bahwa file audio hasil embedding masih tergolong dalam klasifikasi audio dengan kualitas yang baik, di mana perbedaan antara audio asli dan audio hasil steganografi masih berada dalam batas yang dapat diterima, sehingga tidak terlalu mengganggu kualitas pendengaran.

2) Pengujian Parameter SNR Audio *Embed* RSA dan *El-Gamal*

Pengujian SNR berlangsung bersamaan dengan pengujian PSNR, karena object yang diukur adalah stego audio. Semakin besar nilai dari SNR maka stego audio dapat diklasifikasikan memiliki nilai SNR yang baik, secara umum besaran SNR apabila berada di nilai > 40 maka dapat digolongkan stego audio memiliki kualitas yang baik.

TABEL 8 (PENGUJIAN SNR FILE AUDIO RSA)

No	Cover Audio	Stego Audio	Ciphertext	Ukuran Files (Bytes)		SNR(dB)
				Cover audio	*Stego audio*	
1	audio pertama.wav	audio pertama pesan pertama.wav	pesan pertama *encrypted.(*docx)*	13316236	13316236	63.52
2	audio pertama.wav	audio pertama pesan *backup*.wav	Pesan *backup encrypted.(*docx)*	13316236	13316236	62.88
3	audio pertama.wav	audio pertama pesan lirik.wav	pesan lirik *encrypted.(*docx)*	13316236	13316236	63.03
4	audio pertama.wav	audio pertama pesan kelima.wav	pesan kelima *encrypted.(*docx)*	13316236	13316236	63.44
5	audio kedua.wav	audio kedua pesan pertama.wav	pesan pertama *encrypted.(*docx)*	2646094	2646094	56.82
6	audio kedua.wav	audio kedua pesan *backup*.wav	Pesan *backup encrypted.(*docx)*	2646094	2646094	56.23
7	audio kedua.wav	audio kedua pesan lirik.wav	pesan lirik *encrypted.(*docx)*	2646094	2646094	56.39
8	audio kedua.wav	audio kedua pesan kelima.wav	pesan kelima *encrypted.(*docx)*	2646094	2646094	56.73
9	audio ketiga.wav	audio ketiga pesan pertama.wav	pesan pertama *encrypted.(*docx)*	7610446	7610446	60.65
10	audio ketiga.wav	audio ketiga pesan *backup*.wav	pesan *backup encrypted.(*docx)*	7610446	7610446	60.79
11	audio ketiga.wav	audio ketiga pesan lirik.wav	pesan lirik *encrypted.(*docx)*	7610446	7610446	61.16
12	audio ketiga.wav	audio ketiga pesan kelima.wav	pesan kelima *encrypted.(*docx)*	7610446	7610446	61.16
13	audio keempat.wav	audio keempat pesan pertama.wav	pesan pertama *encrypted.(*docx)*	5435470	5435470	58.72
14	audio keempat.wav	audio keempat pesan *backup*.wav	pesan *backup encrypted.(*docx)*	5435470	5435470	58.16
15	audio keempat.wav	audio keempat pesan lirik.wav	pesan lirik *encrypted.(*docx)*	5435470	5435470	58.31
16	audio keempat.wav	audio keempat pesan kelima.wav	pesan kelima *encrypted.(*docx)*	5435470	5435470	58.63

Pada Tabel 8 di atas, disajikan nilai Signal-to-Noise Ratio (SNR) untuk setiap cover audio yang telah mengalami proses penyisipan pesan rahasia. Dari hasil pengujian, diperoleh rentang nilai Peak Signal-to-Noise Ratio (PSNR) antara 56.23 dB hingga 63.52 dB, yang menunjukkan bahwa kualitas audio setelah proses embedding masih tergolong sangat baik. Nilai PSNR yang tinggi menandakan bahwa perubahan yang terjadi akibat penyisipan pesan rahasia ke dalam cover audio tidak menyebabkan degradasi yang signifikan terhadap kualitas suara. Dengan kata lain, stego audio yang dihasilkan tetap memiliki karakteristik yang sangat mirip dengan file audio aslinya, sehingga tidak mudah terdeteksi adanya modifikasi atau penyisipan data. Hal ini membuktikan bahwa metode steganografi yang diterapkan dalam penelitian ini mampu menjaga kualitas audio secara optimal meskipun telah mengalami proses embedding data rahasia. Dengan mempertahankan nilai PSNR dalam rentang yang tinggi, metode ini dapat dikategorikan sebagai pendekatan yang efektif untuk menyisipkan informasi tanpa menyebabkan penurunan kualitas audio yang dapat terdeteksi secara signifikan oleh pendengar atau alat analisis sinyal.

TABEL 9 (PENGUJIAN SNR FILE AUDIO EL-GAMAL)

No	Cover Audio	Stego Audio	Ciphertext	Ukuran Files (Bytes)		SNR(dB)
				Cover audio	*Stego audio*	
1	audio pertama.wav	audio pertama pesan pertama.wav	pesan pertama *encrypted.(*docx)*	13316236	13316140	67.69
2	audio pertama.wav	audio pertama pesan *backup*.wav	pesan *backup* elgamal *encrypted.(*docx)*	13316236	13316140	62.74
3	audio pertama.wav	audio pertama pesan lirik.wav	pesan lirik elgamal *encrypted.(*docx)*	13316236	13316140	63.01
4	audio pertama.wav	audio pertama pesan kelima.wav	pesan kelima *encrypted.(*docx)*	13316236	13316140	67.61
5	audio kedua.wav	audio kedua pesan pertama.wav	pesan pertama *encrypted.(*docx)*	2646094	2646060	60.69
6	audio kedua.wav	audio kedua pesan *backup*.wav	pesan *backup* elgamal *encrypted.(*docx)*	2646094	2646060	55.99
7	audio kedua.wav	audio kedua pesan lirik.wav	pesan lirik elgamal *encrypted.(*docx)*	2646094	2646060	56.30
8	audio kedua.wav	audio kedua pesan kelima.wav	pesan kelima *encrypted.(*docx)*	2646094	2646060	60.60
9	audio ketiga.wav	audio ketiga pesan pertama.wav	pesan pertama *encrypted.(*docx)*	7610446	7610412	65.28
10	audio ketiga.wav	audio ketiga pesan *backup*.wav	pesan *backup* elgamal *encrypted.(*docx)*	7610446	7610412	60.42
11	audio ketiga.wav	audio ketiga pesan lirik.wav	pesan lirik elgamal *encrypted.(*docx)*	7610446	7610412	60.71
12	audio ketiga.wav	audio ketiga pesan kelima.wav	pesan kelima *encrypted.(*docx)*	7610446	7610412	65.20
13	audio keempat.wav	audio keempat pesan pertama.wav	pesan pertama *encrypted.(*docx)*	5435470	5435436	63.78
14	audio keempat.wav	audio keempat pesan *backup*.wav	pesan *backup* elgamal *encrypted.(*docx)*	5435470	5435436	57.94
15	audio keempat.wav	audio keempat pesan lirik.wav	pesan lirik *encrypted.(*docx)*	5435470	5435436	58.24
16	audio keempat.wav	audio keempat pesan kelima.wav	pesan kelima e ncrypted.*(*docx)*	5435470	5435436	63.69

Pada Tabel 9 di atas memberikan nilai SNR untuk tiap tiap cover audio yang disisipi pesan rahasia, memiliki rentang nilai PSNR adalah 55.99 – 67.61 db. Hal ini menunjukkan bahwa file audio masih tergolong pada klasifikasi audio yang baik setelah disisipkan pesan rahasia.

3) Pengujian Parameter BER Audio *Embed* RSA dan *El-Gamal*

Pengujian BER berlangsung pada proses ekstraksi, karena pengujian ini menghitung kesalahan bit dari *stego audio*. Semakin kecil nilai BER maka semakin baik kualitas file.

TABEL 10 (PENGUJIAN BER FILE AUDIO RSA)

No	Cover Audio	Stego Audio	Ciphertext	Ukuran Files (Bytes)		BER(%)
				Cover audio	*Stego audio*	
1	audio pertama.wav	audio pertama pesan pertama.wav	pesan pertama *encrypted.(*docx)*	13316236	13316236	0
2	audio pertama.wav	audio pertama pesan *backup*.wav	pesan *backup* *encrypted.(*docx)*	13316236	13316236	0
3	audio pertama.wav	audio pertama pesan lirik.wav	pesan lirik *encrypted.(*docx)*	13316236	13316236	0
4	audio pertama.wav	audio pertama pesan kelima.wav	pesan kelima *encrypted.(*docx)*	13316236	13316236	0
5	audio kedua.wav	audio kedua pesan pertama.wav	pesan pertama *encrypted.(*docx)*	2646094	2646094	0
6	audio kedua.wav	audio kedua pesan *backup*.wav	Pesan *backup* *encrypted.(*docx)*	2646094	2646094	0
7	audio kedua.wav	audio kedua pesan lirik.wav	pesan lirik *encrypted.(*docx)*	2646094	2646094	0
8	audio kedua.wav	audio kedua pesan kelima.wav	pesan kelima *encrypted.(*docx)*	2646094	2646094	0
9	audio ketiga.wav	audio ketiga pesan pertama.wav	pesan pertama *encrypted.(*docx)*	7610446	7610446	0
10	audio ketiga.wav	audio ketiga pesan *backup*.wav	pesan *backup* *encrypted.(*docx)*	7610446	7610446	0
11	audio ketiga.wav	audio ketiga pesan lirik.wav	pesan lirik *encrypted.(*docx)*	7610446	7610446	0
12	audio ketiga.wav	audio ketiga pesan kelima.wav	pesan kelima *encrypted.(*docx)*	7610446	7610446	0
13	audio keempat.wav	audio keempat pesan pertama.wav	pesan pertama *encrypted.(*docx)*	5435470	5435470	0
14	audio keempat.wav	audio keempat pesan *backup*.wav	pesan *backup* *encrypted.(*docx)*	5435470	5435470	0
15	audio keempat.wav	audio keempat pesan lirik.wav	pesan lirik *encrypted.(*docx)*	5435470	5435470	0
16	audio keempat.wav	audio keempat pesan kelima.wav	pesan kelima *encrypted.(*docx)*	5435470	5435470	0

PADA TABEL 10 DI ATAS DITAMPILKAN PADA TABEL PENGUJIAN NILAI BIT ERROR RATE (BER) DI ATAS, DAPAT DISIMPULKAN BAHWA FILE STEGO AUDIO YANG DIHASILKAN SETELAH PROSES EMBEDDING CIPHERTEXT MENGGUNAKAN ALGORITMA RSA TIDAK MENGALAMI KESALAHAN BIT SETELAH DIEKSTRAKSI. HAL INI DIBUKTIKAN DENGAN HASIL PENGUJIAN BER YANG MENUNJUKKAN NILAI 0, YANG BERARTI TIDAK TERDAPAT PERBEDAAN BIT ANTARA FILE HASIL EKSTRAKSI DAN FILE ASLI SEBELUM DISISIPKAN KE DALAM AUDIO. DENGAN KATA LAIN, PROSES EMBEDDING DAN EKSTRAKSI BERHASIL DILAKUKAN TANPA MENYEBABKAN KERUSAKAN ATAU KEHILANGAN DATA PADA FILE STEGO AUDIO, SEHINGGA MEMASTIKAN INTEGRITAS PESAN TETAP TERJAGA SETELAH PROSES EKSTRAKSI SELESAI.

TABEL 11 (PENGUJIAN BER FILE AUDIO EL-GAMAL)

No	Cover Audio	Stego Audio	Ciphertext	Ukuran Files (Bytes)		BER(%)
				Cover audio	*Stego audio*	
1	audio pertama.wav	audio pertama pesan pertama.wav	pesan pertama *encrypted.(*docx)*	13316236	13316140	0
2	audio pertama.wav	audio pertama pesan *backup*.wav	pesan *backup* elgamal *encrypted.(*docx)*	13316236	13316140	0
3	audio pertama.wav	audio pertama pesan lirik.wav	pesan lirik elgamal *encrypted.(*docx)*	13316236	13316140	0
4	audio pertama.wav	audio pertama pesan kelima.wav	pesan kelima *encrypted.(*docx)*	13316236	13316140	0
5	audio kedua.wav	audio kedua pesan pertama.wav	pesan pertama *encrypted.(*docx)*	2646094	2646060	0
6	audio kedua.wav	audio kedua pesan *backup*.wav	pesan *backup* elgamal *encrypted.(*docx)*	2646094	2646060	0
7	audio kedua.wav	audio kedua pesan lirik.wav	pesan lirik elgamal *encrypted.(*docx)*	2646094	2646060	0
8	audio kedua.wav	audio kedua pesan kelima.wav	pesan kelima *encrypted.(*docx)*	2646094	2646060	0
9	audio ketiga.wav	audio ketiga pesan pertama.wav	pesan pertama *encrypted.(*docx)*	7610446	7610412	0
10	audio ketiga.wav	audio ketiga pesan *backup*.wav	pesan *backup* elgamal *encrypted.(*docx)*	7610446	7610412	0
11	audio ketiga.wav	audio ketiga pesan lirik.wav	pesan lirik elgamal *encrypted.(*docx)*	7610446	7610412	0
12	audio ketiga.wav	audio ketiga pesan kelima.wav	pesan kelima *encrypted.(*docx)*	7610446	7610412	0
13	audio keempat.wav	audio keempat pesan pertama.wav	pesan pertama *encrypted.(*docx)*	5435470	5435436	0
14	audio keempat.wav	audio keempat pesan *backup*.wav	pesan *backup* elgamal *encrypted.(*docx)*	5435470	5435436	0
15	audio keempat.wav	audio keempat pesan lirik.wav	pesan lirik *encrypted.(*docx)*	5435470	5435436	0
16	audio keempat.wav	audio keempat pesan kelima.wav	pesan kelima *encrypted.(*docx)*	5435470	5435436	0

Berdasarkan informasi yang tercantum pada Tabel 11 mengenai hasil pengujian nilai Bit Error Rate (BER) di atas, dapat disimpulkan bahwa file stego audio yang dihasilkan setelah proses embedding ciphertext menggunakan algoritma RSA, ketika diekstraksi kembali, tidak mengalami kesalahan bit sedikit pun. Hal ini dibuktikan dengan hasil pengujian BER yang menunjukkan nilai sebesar 0, yang mengindikasikan bahwa tidak ada perubahan atau kerusakan pada data yang telah diekstraksi dari file stego audio tersebut. Dengan kata lain, proses embedding dan ekstraksi yang dilakukan terhadap file stego audio tetap mempertahankan keutuhan data, sehingga kualitas serta keakuratan informasi yang tersimpan di dalamnya tetap terjaga tanpa mengalami degradasi atau kehilangan bit selama proses berlangsung.

V. KESIMPULAN

Berdasarkan hasil pengujian dan analisis sistem yang telah dilakukan, dapat disimpulkan bahwa sistem aplikasi pengamanan data berhasil dirancang dengan mengimplemen tasikan algoritma RSA dan El-Gamal serta metode *Least Significant Bit (LSB)* untuk mendukung tujuan sistem. Pengujian menggunakan metode *blackbox testing* menunjuk kan bahwa sistem memiliki tingkat kelayakan yang baik, dengan hasil sempurna pada 22 kasus pengujian fungsionalitas, yang membuktikan bahwa sistem telah berjalan sesuai dengan kebutuhan. Selain itu, pengujian kinerja menunjukkan bahwa algoritma RSA lebih unggul dalam proses enkripsi yang lebih cepat, sedangkan algoritma El-Gamal lebih cepat dalam proses dekripsi, sehingga pemilihan algoritma dapat disesuaikan dengan kebutuhan sistem.

REFERENSI

[1] Septian R., “Desain Algoritma Steganografi dengan Metode Spread Spectrum Berbasis PCMK (Permutas Chaotic Multiputaran Mengecil dan Membesar) yang Tahan Terhadap Gangguan,” 2018.

[2] A. Ahdiat, “Indonesia Masuk 10 Negara dengan Kebocoran Data Terbesar,” *Katadata Media Network*, Jun. 28, 2024. [Online]. Available: https://databoks.katadata.co.id/teknologi-telekomunikasi/statistik/cc5473708a4f8dc/indonesia-masuk-10-negara-dengan-kebocoran-data-terbesar

[3] Direktorat Operasi Keamanan Siber, “Lanskap Keamanan Siber Indonesia 2023,” BSSN, Jakarta Selatan, Id-SIRTII/CC, 2023.

[4] D. Lutfiana. P, “Bjorka Muncul Kembali, Diduga Bocorkan 19 Juta Data BPJS Ketenagakerjaan,” *Kompas.com*, Mar. 14, 2023. [Online]. Available: https://www.kompas.com/tren/read/2023/03/14/091500565/bjorka-muncul-kembali-diduga-bocorkan-19-juta-data-bpjs-ketenagakerjaan?page=all

[5] A. M. Fajrin, J. R. Benedict, and H. J. Kusuma, “Analisis Performa dari Algoritma Kriptografi RSA dan ElGamal dalam Enkripsi dan Dekripsi Pesan,” vol. 8, 2023.

[6] I. Gunawan, “Penggunaan Algoritma Kriptografi Steganografi Least Significant Bit Untuk Pengamanan Pesan Teks dan Data Video,” *J-SAKTI J. Sains Komput. Dan Inform.*, vol. 2, no. 1, p. 57, Mar. 2018, doi: 10.30645/j-sakti.v2i1.48.

[7] T. S. Jaya, “Pengujian Aplikasi dengan Metode Blackbox Testing Boundary Value Analysis (Studi Kasus: Kantor Digital Politeknik Negeri Lampung),” *J. Inform. J. Pengemb. IT*, vol. 3, no. 1, pp. 45–48, Jan. 2018, doi: 10.30591/jpit.v3i1.647.

[8] K. Khairani and M. Z. Siambaton, “Pengamanan Data Teks Menggunakan Algoritma Kriptografi Elgamal dan XOR dari Serangan Hacker,” *Sudo J. Tek. Inform.*, vol. 2, no. 4, pp. 176–187, Dec. 2023, doi: 10.56211/sudo.v2i4.401.

[9] M. F. Syawal, D. C. Fikriansyah, and N. Agani, “Implementasi Teknik Steganografi Menggunakan Algoritma Vigenere Cipher Dan Metode LSB,” 2016.

[10] “Mengenal Format File Doc Dan Docx Di Microsoft Word,” *Termasmedia*. [Online]. Available: https://www.termasmedia.com/aplikasi/microsoft-office/office-word/486-mengenal-lebih-jauh-format-file-doc-dan-(*docx)-di-microsoft-office-word.html

[11] S. Santoso, A. Arisman, and W. Sentanu, “Steganografi Audio (WAV) Menggunakan Metode LSB (Least Significant Bit),” *CCIT J.*, vol. 9, no. 2, pp. 214–224, 2016, doi: 10.33050/ccit.v9i2.500.

[12] I. Rahim, N. Anwar, A. M. Widodo, K. Karsono Juman, and I. Setiawan, “Komparasi Fungsi Hash Md5 Dan Sha256 Dalam Keamanan Gambar Dan Teks,” *Ikraith-Inform.*, vol. 7, no. 2, Nov. 2022, doi: 10.37817/ikraith-informatika.v7i2.2249.

[13] M. Romzi and B. Kurniawan, “Pembelajaran Pemrograman Python Dengan Pendekatan Lpgika Algoritma,” vol. 3, no. 2, 2020.

[14] S. Y. Doo, S. Tena, and V. M. Ndolu, “Implementaasi Pengamanan Data Menggunakan Metode Kriptografi Hill Cipher dan Steganografi Least Significant Bit (LSB) Pada Media Citra Digital,” *J. Media Elektro*, pp. 93–99, Oct. 2019, doi: 10.35508/jme.v0i0.1778.

[15] W. Widyastuti, “Kinerja Sandi Koreksi Kesalahan LDPC pada Transmisi Citra,” 2023.