Jurnal Publikasi Sistem Informasi dan Manajemen Bisnis
Volume 5, Nomor 3, September 2026
E-ISSN : 2808-8980 ; P-ISSN : 2808-9383, Hal. 128-143
DOI: https://doi.org/10.55606/jupsim.v5i3.7827
Tersedia: https://journalcenter.org/index.php/jupsim

Implementasi Kriptografi Algoritma Rivest Shamir Adleman (RSA) pada Pengamanan Pesan Suara

(Studi Kasus: Pada File WAV)

Las Dorenta Hutasoit[1*], Nelson Nainggolan[2], Eliasta Ketaren[3]
[1]Sistem Informasi, Universitas Sam Ratulangi, Indonesia
[2]Matematika, Universitas Sam Ratulangi, Indonesia
[3]Sistem Informasi, Universitas Sam Ratulangi, Indonesia
**Penulis Korespondensi: lasdorenta221@gmail.com*

Abstract*. Voice message security has become increasingly important in digital communication due to the risk of unauthorized access and eavesdropping during data transmission. Ensuring the confidentiality of voice data requires effective cryptographic techniques capable of protecting information from unauthorized parties. This study aims to implement the RSA cryptographic algorithm to secure voice messages through encryption and decryption processes. The research employed a software development approach by designing and developing a Python-based desktop application integrated with the RSA algorithm. Voice messages were encrypted using a public key and subsequently restored using the corresponding private key. The effectiveness of the implementation was evaluated through functional testing and signal analysis using waveform and Mel-spectrogram visualizations to compare the original, encrypted, and decrypted voice signals. The results demonstrate that the RSA algorithm successfully transforms voice messages into encrypted data that cannot be interpreted without the appropriate private key. Furthermore, the decrypted voice messages closely match the original recordings, confirming the correctness of the decryption process. The waveform and Mel-spectrogram analyses also reveal significant differences between the original and encrypted signals, while the decrypted signals exhibit characteristics similar to the original voice data. These findings indicate that RSA can be effectively applied to enhance the confidentiality and security of voice messages in digital communication systems, making it a reliable solution for protecting sensitive voice information from unauthorized access.*

Keywords*: Cryptography; Data Security; Mel-Spectogram; RSA; Voice Messages.*

Abstrak. Keamanan pesan suara merupakan aspek penting dalam komunikasi digital karena data suara rentan terhadap penyadapan dan akses oleh pihak yang tidak berwenang. Oleh karena itu, diperlukan mekanisme pengamanan yang mampu menjaga kerahasiaan informasi selama proses penyimpanan maupun pengiriman. Penelitian ini bertujuan untuk mengimplementasikan algoritma kriptografi RSA sebagai metode pengamanan pesan suara melalui proses enkripsi dan dekripsi. Metode penelitian yang digunakan meliputi perancangan dan pengembangan aplikasi desktop berbasis Python, implementasi algoritma RSA, serta pengujian fungsional terhadap proses enkripsi dan dekripsi pada file pesan suara. Selain itu, dilakukan analisis menggunakan visualisasi Waveform dan Mel-Spectrogram untuk membandingkan karakteristik sinyal suara asli, sinyal terenkripsi, dan sinyal hasil dekripsi. Hasil penelitian menunjukkan bahwa algoritma RSA mampu mengenkripsi pesan suara sehingga isi pesan tidak dapat dipahami tanpa menggunakan kunci privat yang sesuai. Proses dekripsi juga berhasil mengembalikan data suara ke bentuk semula tanpa mengubah informasi yang terkandung di dalamnya. Perbedaan karakteristik pada visualisasi Waveform dan Mel-Spectrogram antara sinyal asli dan sinyal terenkripsi, serta kemiripan antara sinyal asli dan hasil dekripsi, menunjukkan bahwa implementasi RSA berjalan dengan baik. Temuan ini menunjukkan bahwa algoritma RSA dapat diterapkan secara efektif untuk meningkatkan keamanan dan kerahasiaan pesan suara dalam komunikasi digital.

Kata Kunci: Keamanan Data; Kriptografi; Mel-Spectogram; Pesan Suara; RSA.

1. LATAR BELAKANG

Perkembangan teknologi informasi dan komunikasi telah mengubah cara masyarakat dalam bertukar informasi melalui media digital. Selain pesan teks, penggunaan pesan suara (*voice message*) semakin meningkat karena dinilai lebih praktis, efisien, dan mampu menyampaikan informasi secara lebih jelas. Pesan suara saat ini banyak dimanfaatkan pada

Naskah Masuk: 18 Juni 2026; Revisi: 16 Juli 2026; Diterima: 10 Agustus 2026; Terbit: 01 September 2026

berbagai bidang, seperti komunikasi pribadi, pendidikan, layanan publik, hingga aktivitas bisnis. Namun, meningkatnya penggunaan media digital juga diikuti oleh berbagai ancaman keamanan, seperti penyadapan, pencurian, manipulasi, maupun akses tidak sah terhadap data yang dikirim melalui jaringan internet. Oleh karena itu, perlindungan terhadap kerahasiaan pesan suara menjadi salah satu aspek penting dalam menjaga keamanan informasi digital.

Berdasarkan laporan (Statista, 2025), jumlah pengguna aplikasi pesan instan di dunia telah melampaui 3 miliar pengguna dan diperkirakan terus meningkat setiap tahunnya. Di sisi lain, (IBM Corporation, 2024) melaporkan bahwa rata-rata kerugian akibat kebocoran data secara global mencapai USD 4,88 juta per insiden. Selain itu, Badan Siber dan Sandi Negara (BSSN) juga melaporkan masih tingginya jumlah serangan siber yang menargetkan berbagai bentuk data digital di Indonesia (Rosdiana, Reja, & Setiawan, 2026). Kondisi tersebut menunjukkan bahwa perlindungan terhadap data digital, termasuk pesan suara, menjadi kebutuhan yang semakin mendesak. Salah satu pendekatan yang banyak digunakan untuk menjaga kerahasiaan informasi adalah kriptografi, yaitu teknik yang mengubah data asli (*plaintext*) menjadi data yang tidak dapat dipahami (*ciphertext*). Salah satu algoritma kriptografi yang memiliki tingkat keamanan tinggi adalah RSA (*Rivest–Shamir–Adleman*), yaitu algoritma kriptografi asimetris yang menggunakan pasangan kunci publik dan kunci privat sehingga hanya pihak yang memiliki kunci privat yang dapat mengembalikan data ke bentuk semula.

Selain ancaman kebocoran data secara umum, pesan suara memiliki karakteristik yang berbeda dibandingkan data teks karena mengandung informasi biometrik, intonasi, dan identitas pembicara. Apabila pesan suara berhasil disadap, informasi pribadi maupun informasi sensitif dapat diketahui secara langsung oleh pihak yang tidak berwenang. Oleh karena itu, diperlukan mekanisme pengamanan yang mampu menjaga kerahasiaan data suara selama proses penyimpanan maupun transmisi.

Berbagai penelitian telah membahas penerapan algoritma RSA pada pengamanan data digital. Rivest, Shamir, dan Adleman (1978) memperkenalkan RSA sebagai algoritma kriptografi kunci publik yang hingga kini masih menjadi salah satu standar dalam sistem keamanan informasi. Penelitian oleh (Diarse & Bendi, 2016) menunjukkan bahwa RSA mampu mengenkripsi dan mendekripsi file audio MP3 sehingga kerahasiaan data dapat dipertahankan, sedangkan penelitian (Tobing, 2024) membuktikan bahwa RSA efektif dalam mengamankan data rekam medis melalui mekanisme enkripsi dan dekripsi menggunakan pasangan kunci publik dan kunci privat. Meskipun demikian, sebagian besar penelitian lebih berfokus pada pengamanan data teks, dokumen, atau file digital tertentu, sedangkan penelitian

mengenai implementasi RSA secara langsung pada pesan suara masih relatif terbatas. Selain itu, penelitian terdahulu umumnya hanya mengevaluasi keberhasilan proses enkripsi dan dekripsi tanpa melakukan analisis terhadap karakteristik sinyal audio maupun performa sistem secara lebih komprehensif.

Berdasarkan hasil kajian penelitian terdahulu, masih terdapat kesenjangan penelitian (*research gap*) pada implementasi RSA untuk pengamanan pesan suara. Penelitian sebelumnya belum banyak mengintegrasikan proses perekaman suara, pembangkitan kunci RSA, enkripsi, dekripsi, dan pemutaran hasil audio dalam satu aplikasi desktop yang dilengkapi dengan analisis visual terhadap perubahan sinyal audio. Selain itu, evaluasi terhadap integritas data hasil dekripsi, perubahan *waveform*, representasi *Mel-Spectrogram*, waktu komputasi enkripsi dan dekripsi, serta ukuran *ciphertext* masih jarang dilakukan. Padahal, analisis tersebut penting untuk memastikan bahwa algoritma RSA tidak hanya mampu menjaga kerahasiaan (*confidentiality*), tetapi juga mempertahankan keutuhan (*integrity*) data suara setelah proses dekripsi. Kebaruan penelitian ini terletak pada implementasi RSA untuk pengamanan pesan suara berbasis file WAV yang dipadukan dengan visualisasi sinyal audio dan pengujian performa sistem secara menyeluruh dalam satu aplikasi desktop berbasis Python.

Berdasarkan uraian tersebut, penelitian ini bertujuan mengimplementasikan algoritma RSA pada sistem pengamanan pesan suara berbasis aplikasi desktop menggunakan Python. Sistem yang dikembangkan mampu melakukan perekaman suara maupun unggah file WAV, pembangkitan pasangan kunci RSA, proses enkripsi dan dekripsi, pemutaran audio sebelum dan sesudah proses kriptografi, serta visualisasi *waveform* dan *Mel-Spectrogram*. Selain itu, penelitian ini mengevaluasi waktu proses enkripsi dan dekripsi, ukuran *ciphertext*, serta integritas data hasil dekripsi. Hasil penelitian diharapkan dapat memberikan kontribusi dalam pengembangan sistem keamanan informasi berbasis kriptografi, khususnya pada pengamanan pesan suara, serta menjadi referensi bagi penelitian selanjutnya di bidang keamanan multimedia.

2. KAJIAN TEORITIS

Kriptografi

Kriptografi merupakan ilmu yang mempelajari teknik untuk mengamankan informasi melalui proses enkripsi dan dekripsi sehingga data hanya dapat diakses oleh pihak yang memiliki kunci yang sesuai. Tujuan utama kriptografi meliputi menjaga kerahasiaan *(confidentiality)*, integritas *(integrity)*, autentikasi *(authentication)*, dan *non-repudiation*.

Dalam penelitian ini, kriptografi digunakan untuk melindungi pesan suara agar tidak dapat dibaca atau dimodifikasi oleh pihak yang tidak berwenang (Hidayat et al., 2023).

Algoritma Rsa

RSA (Rivest-Shamir-Adleman) merupakan algoritma kriptografi kunci publik (asimetris) yang menggunakan pasangan kunci publik dan kunci privat. Proses enkripsi dilakukan menggunakan kunci publik, sedangkan proses dekripsi menggunakan kunci privat. Keamanan RSA bergantung pada sulitnya memfaktorkan bilangan bulat yang merupakan hasil perkalian dua bilangan prima berukuran besar. Pada penelitian ini, RSA diterapkan untuk mengenkripsi data audio berformat WAV sehingga isi pesan tetap terlindungi selama penyimpanan maupun pengiriman (Fauji et al., 2016).

Format Audio Wav

WAV adalah format audio standar Microsoft dan IBM untuk personal computer (PC), biasanya menggunakan coding PCM *(Pulse Code Modulation),* WAV adalah data tidak terkompres sehingga seluruh sample audio disimpan semuanya di *harddisk*. Software yang dapat menciptakan WAV dari analog sound misalnya adalah *Windows Sound Recorder* (Santoso & Fakhriza, 2018).

Penelitian Terdahulu

Tabel 1. Penelitian Terdahulu.

Penelitian	Objek	Algoritma	Kekurangan	Kebaruan Penelitian
(Yousif, 2020)	Enkripsi dan Dekripsi audio	RSA	Implementasi dilakukan menggunakan simulasi MATLAB dan lebih menitikberatkan pada analisis kualitas sinyal audio. Penelitian belum menyediakan aplikasi dengan antarmuka pengguna maupun fitur pengelolaan file audio secara langsung.	Penelitian ini mengembangkan aplikasi desktop berbasis Python dengan fitur rekam suara, upload file WAV, pembangkitan kunci RSA, enkripsi, dekripsi, serta visualisasi hasil proses sehingga lebih mudah digunakan sebagai media pengamanan pesan suara.
(Fuad & Winata, 2017)	Keamanan File audio WAV	RSA	Enkripsi hanya dilakukan pada data sampel audio, sedangkan header file tetap dibiarkan sehingga masih terdapat informasi	Penelitian ini tidak hanya melakukan enkripsi dan dekripsi file WAV, tetapi juga menyediakan aplikasi desktop untuk merekam suara,

			struktur file yang tidak diamankan. Selain itu, penelitian hanya membahas implementasi dasar RSA tanpa analisis performa yang lebih mendalam	menghasilkan kunci RSA secara otomatis, serta menampilkan visualisasi waveform, spektrogram, dan data hasil enkripsi maupun dekripsi.
(Joshua, Godwin, & Francis, 2019)	Keamanan data audio pada jaringan komunikasi	RSA	Penelitian berfokus pada simulasi menggunakan MATLAB dan mengevaluasi kualitas audio menggunakan parameter CD, LPC, dan SSSNR, sehingga belum menghasilkan aplikasi yang dapat digunakan langsung oleh pengguna umum.	Penelitian ini mengimplementasikan RSA dalam bentuk aplikasi desktop berbasis Python yang dapat digunakan secara langsung untuk mengamankan pesan suara hasil rekaman maupun file WAV, sehingga lebih aplikatif.
(Azhari, Fauzi, & Sembiring, 2025)	Pengamanan file audio MP3 menggunakan metode super enkripsi	RSA + ElGamal	Menggunakan dua algoritma sehingga proses enkripsi dan dekripsi menjadi lebih kompleks serta waktu pemrosesan lebih lama. Penelitian berfokus pada file MP3, bukan pesan suara hasil perekaman secara langsung.	Penelitian ini hanya menggunakan algoritma RSA untuk mengamankan pesan suara hasil rekaman (.wav) sehingga implementasi lebih sederhana, mudah diterapkan, dan dapat menganalisis kinerja RSA secara spesifik

3. METODE PENELITIAN

Gambar 1. Tahapan Penelitian.

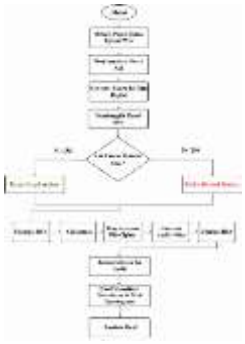
Bagian Penelitian ini menggunakan metode eksperimen dengan pendekatan pengembangan aplikasi untuk mengimplementasikan algoritma kriptografi Rivest Shamir Adleman (RSA) pada pengamanan pesan suara berformat WAV. Aplikasi dikembangkan menggunakan bahasa pemrograman Python dengan antarmuka desktop untuk melakukan proses perekaman suara, pembangkitan pasangan kunci RSA, enkripsi, dekripsi, pemutaran audio, serta visualisasi sinyal audio.

Data penelitian berupa pesan suara hasil perekaman secara langsung maupun file audio berformat WAV. Seluruh data audio direpresentasikan dalam format PCM 16-bit dengan sampling rate 22.050 Hz sebagai masukan pada proses enkripsi dan dekripsi menggunakan algoritma RSA.

Tahapan penelitian meliputi identifikasi masalah, studi literatur, perancangan sistem, implementasi algoritma RSA, pengujian sistem, dan analisis hasil. Pada tahap implementasi, data suara dikonversi menjadi representasi byte untuk diproses menggunakan pasangan kunci publik dan kunci privat RSA. Setelah proses dekripsi selesai, data direkonstruksi kembali menjadi file audio WAV.

Pengujian dilakukan untuk mengevaluasi keberhasilan proses enkripsi dan dekripsi, waktu komputasi, ukuran ciphertext, serta integritas data hasil dekripsi. Selain itu, analisis karakteristik sinyal dilakukan menggunakan visualisasi Waveform dan Mel-Spectrogram untuk membandingkan sinyal audio asli, terenkripsi, dan hasil dekripsi. Hasil pengujian kemudian dianalisis secara deskriptif untuk menilai efektivitas implementasi algoritma RSA dalam menjaga kerahasiaan dan keutuhan pesan suara. proses pengembangan aplikasi yang akan digunakan dalam mendukung proses penelitian, seperti perangkat keras, perangkat lunak diantaranya sebagai berikut: Perangkat lunak, Visual Studio Code, Python 3.13.7, File WAV. Library (customtkinter, tkinter, numpy, sounddevice, soundfile, Crypto.Util.number, matplotlib, math, threading, os, time), Perangkat keras, Intel Core i3, RAM 8 GB.

Diagram Alur Penelitian



Gambar 2. Flowchart Penelitian.

Penelitian diawali dengan perekaman pesan suara atau Upload WAV yang kemudian disimpan dan dikonversi menjadi data digital. Selanjutnya dilakukan pembangkitan kunci RSA dan pemeriksaan nilai modulus (*N*). Jika *N* > 255, proses enkripsi dan dekripsi dapat dilakukan secara lossless, sedangkan jika *N* < 256 terdapat risiko distorsi akibat operasi modulo. Setelah itu, data Suara dienkripsi menjadi ciphertext, disimpan sebagai file cipher, kemudian didekripsi dan dikonversi kembali menjadi Suara. Tahap terakhir adalah melakukan visualisasi menggunakan Waveform dan Mel-Spectrogram serta menganalisis hasil untuk mengevaluasi keberhasilan implementasi algoritma RSA dalam mengamankan pesan suara.

Input Data Suara Real Time

Input data suara real time merupakan tahap awal dalam proses pengamanan pesan suara.

$x(n),\ n = 0,1,2, \dots, N-1$.....(i)

dimana : $x(n)$: Ampiltudo sinya suara pada indeks ke-n, n : Indeks sampel, N: Jumlah total sampel suara. Data suara hasil perekaman kemudian disimpan dalam format WAV untuk mempertahankan kualitas suara sebelum diproses lebih lanjut.

Konversi Sinyal Suara PCM ke Segmentasi Byte (8-bit)

Sinyal Suara asli direpresentasikan sebagai himpunan sampel 16-bit berurutan. Setiap sampel S_i diekstrak dan dipecah menjadi dua buah blok data 8-bit, yaitu *Least Significant Byte* (LSB) dan *Most Significant Byte* (MSB), menggunakan operasi modulo dan pembagian integer dengan rumus:

$B_{LSB} = S_i(mod\ 256)$.....*(ii)*

dimana : S_i : Nilai amplitudo dari sampel Suara ke-i dalam format PCM 16-bit. B_{LSB}:Byte paling rendah (LSB) dengan rentang nilai 0-255. B_{MSB} : Byte paling tinggi (MSB) dengan rentang nilai 255. Setelah dikonversi menjadi deretan desimal berukuran 8-bit, setiap nilai desimal kemudian dikonversi ke dalam bentuk biner menggunakan representasi biner standar basis-2 sebelum dienkripsi, dengan fungsi konversi:

$B_k = \sum_{j=0}^{7} b_j \cdot 2^j$ .....(iii)

dimana : B_k: Nilai byte ke-k, b_j: Nilai bit (0 atau 1) pada posisi ke-j, j: Posisi bit 0 – 7, 2^j: Nilai bobot masing-masing bit pada sistem bilangan biner.

Aritmatika Modulo RSA

Dalam Penerapan *Teorema Euler* pada perumusan algoritma RSA Coding sangat dibutuhkan pemahaman tentang modulo. Modulo sendiri berarti sisa hasil bagi. Misalkan a adalah bilangan bulat dan m adalah bilangan bulat dimana a dan m lebih besar dari 0. Maka

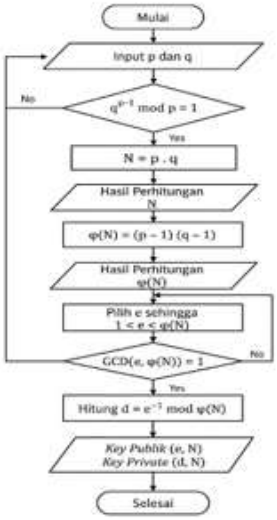
operasi a mod m (dibaca “a modulo m”) memberikan sisa jika a dibagi dengan m. Bilangan m disebut modulus atau modulo, dan hasil modulo m terletak di dalam himpunan $\{0,1,2, \dots m-1\}$ (Ginting, Isnanto, & Windasari, 2015). Contoh : Diambil a = 20 dan m = 6. Karena 20 dibagi 6 adalah 3 bersisa 2, maka diperoleh a mod m ≡ 20 mod 6 ≡ 2.

Pembangkitan Kunci RSA

Pembangkit Kunci menunjukkan alur pembuatan kunci RSA yang akan digunakan untuk enkripsi dan dekripsi. Dua bilangan bulat yang membentuk input adalah p, q.

Enkripsi Pesan Suara

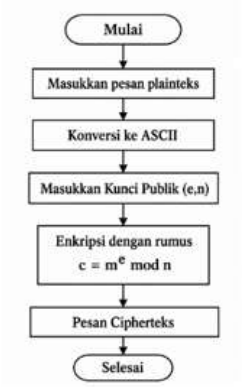
Untuk mengenkripsi pesan apapun, algoritma mengubah pesan yang diberikan menjadi sebuah bilangan bulat (*integer)* menggunakan skema padding yang sesuai. Berikut pada Gambar 4 flowchart enkripsi RSA.



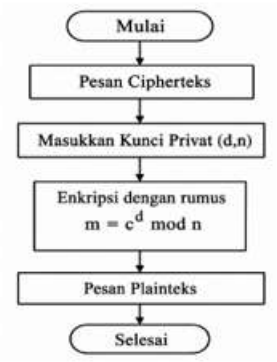
Gambar 3.Flowchart Pembangkit Kunci RSA

Dekripsi Pesan Suara

Proses dekripsi merupakan tahapan untuk mengembalikan pesan suara yang telah terenkripsi (ciphertext) menjadi pesan suara asli (plaintext) menggunakan kunci privat RSA. Pada proses ini, sistem menerima masukan berupa ciphertext dan pasangan kunci privat yang terdiri atas nilai d dan modulus$n$.Alur proses dekripsi pada sistem dapat dilihat pada Gambar 5.



Gambar 4. Flowchart Enkripsi RSA.



Gambar 5. Flowchart Dekripsi RSA.

Rekonstruksi Sinyal Suara

Setelah proses dekripsi, data byte dikonversi kembali menjadi sinyal suara berformat PCM 16-bit dan direkonstruksi menjadi file WAV. Hasil dekripsi menunjukkan bahwa data suara dapat dikembalikan ke bentuk semula, sehingga integritas data tetap terjaga setelah proses enkripsi dan dekripsi menggunakan algoritma RSA.

Ekstraksi Fitur dengan Mel-Spectogram

Setelah proses perekaman atau upload selesai, sinyal suara diekstraksi menggunakan metode Mel-Spectrogram Dengan menggunakan rumus *Short-Time Fourier Transform* (STFT)

$X(m,k) = \sum_{n=0}^{N-1} x(n)\, w(n-m) e^{-j\frac{2\pi}{N}kn}$ .....(iv)

dimana : $X(m,k)$: Hasil STFT pada frame waktu ke-m dan frekuensi ke-k. $x(n)$: Sinyal suara asli. $w(n-m)$: Fungsi Window yang diterapkan pada sinyal. N: Panjang Frame. k : Indeks frekuensi. m : Indeks waktu/Frame. $e^{-j\frac{2\pi}{N}kn}$: Komponen transformasi fourier.

Konversi Frekuensi ke Skala Mel

Skala Mel digunakan karena lebih sesuai dengan karakteristik pendengaran manusia dibandingkan skala frekuensi linear.

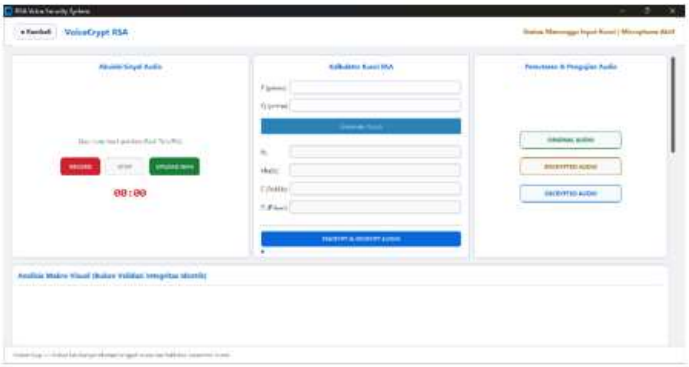
$$Mel(f) = 2595log_{10}\left(1 + \frac{f}{700}\right) \text{.....(v)}$$

dimana :$Mel(f)$: nilai frekuensi pada skala Mel$f$: frekuensi dalam Hertz (Hz).

4. HASIL DAN PEMBAHASAN

Bagian Data penelitian berupa pesan suara hasil perekaman langsung menggunakan mikrofon serta file audio berformat WAV yang digunakan sebagai data uji implementasi algoritma RSA.

Implementasi Sistem



Gambar 6. Halaman Menu Utama Aplikasi.

Halaman utama aplikasi terdiri atas beberapa menu utama, yaitu Record,Stop, Upload WAV, Generate Key RSA, Encrypt & Decrypt, Original Audio, Encrypted Audio, dan Decrypted Audio. Selain itu aplikasi menampilkan visualisasi Waveform dan Mel-Spectrogram serta informasi hasil pembangkitan kunci RSA berupa nilai n, φ(n), eksponen publik (e), dan eksponen privat (d).

Implementasi Algoritma RSA pada File Pesan Suara

Gambar 7. Halaman Menu Utama Aplikasi Lanjutan.

Input Suara/Upload WAV

Gambar 8. Record Suara Real Time/ Upload WAV.

Pembangkit Kunci RSA

Berikut adalah proses pembangkit kunci Manual, yaitu: Pilih bilangan prima, misalnya $p = 3\ dan\ q = 641$, Hitung $n = p\ x\ q = 3\ x\ 641 = 1923$, Hitung $\varphi(n) = (p - 1)\ (q - 1) = 2\ x\ 640 = 1280$, Pilih e yang relatif prima terhadap $gcd\ (e, \varphi(n)) = 1$ nilai e yang diambil adalah 3. $1280 = 3 \times 426 + 2$, $3 = 2 \times 1 + 1$, $2 = 1 \times 2 + 0$. Karena sisa terakhir yang bukan 0 adalah 1, maka gcd $(3,1280) =$ 1. Sehingga cari nilai $d \times e \equiv 1\ mod\ \varphi(n)$.

Gambar 9.Kalkulator Pembangkit Kunci RSA.

Mencari nilai, $d \times 3 \equiv 1\ mod\ 1280, 1280 + 1 = 1281, 1281 : 3 = 427, 427 \times 3 = 1281, 1280\ mod\ 1281 = 1$, jadi, $d = 427$. Bukti : $427 \times 3\ mod\ 1280 = 1$. Sehingga di dapatkan :Public key : $e(3), n\ (1923)$ dan Private key : $d(427), n(1923)$

Proses Enkripsi pada Pesan Suara

```
==========================================
 Sampel Analisis (30 Byte Ciphertext Hasil Enkripsi)
==========================================

 F-151552 | Desimal: 1000    | Biner: 0000001111101000
 F-151553 | Desimal: 694     | Biner: 0000001010110110
 F-151554 | Desimal: 1809    | Biner: 0000011100010001
 F-151555 | Desimal: 694     | Biner: 0000001010110110
 F-151556 | Desimal: 1725    | Biner: 0000011010111101
```

Gambar 10. Data Enkripsi Suara.

Menggunakan *Public key* (e, n). Setiap M di enkripsi menjadi C, dengan rumus

$C = M^e mod\ (N)$.....(vi)

Contoh proses enkripsi *Publik Key* dan *Private Key* sudah diketahui, maka selanjutnya dilakukan enkripsi pada plaintext misalnya penulis menghitung manual 1-10 byte pertama M = 10,253,183,253,252. $C1 = 10^3 mod\ 1923 = 1000$, $C2 = 253^3 mod\ 1923 = 694$, $C3 = 183^3 mod\ 1923 = 1809$, $C4 = 253^3 mod\ 1923 = 694$, $C5 = 252^3 mod\ 1923 = 1725$

Sehingga diperoleh hasil lima ciphertext pertama yaitu: 1000,694,1809,694,1725.

Proses Dekripsi pada Pesan Suara

```
========================================
 VERIFIKASI INTEGRITAS SIKLUS DEKRIPSI
========================================

 DEC   HEX   BIN        ASCII
 ---------------------------
 10    0A    00001010   .
 253   FD    11111101   .
 183   B7    10110111   .
 253   FD    11111101   .
 252   FC    11111100   .
```

Gambar 11. Data Dekripsi Suara.

Adapun langkah-langkah yang digunakan untuk melakukan proses dekripsi pada algorimta RSA adalah sebagai berikut. Menggunakan private key (d, n). Pilih ciphertext C

$P = C^d mod\ (N)$.....(vii)

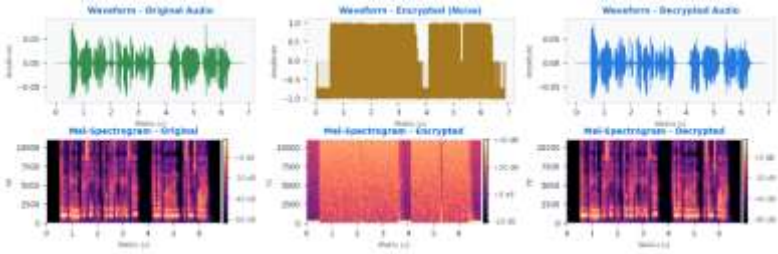
Contoh proses dekripsi, Ciphertext $(c) = 1000,694,1809,694,1725$ dalam karakter Desimal, $P1 = 1000^{427}\ mod\ 1923 = 10, P2 = 694^{427}\ mod\ 1923 = 253$, $P3 = 1809^{427}\ mod\ 1923 = 183, P4 = 694^{427} mod\ 1923 = 253, P5\ 1725^{427}\ mod\ 1923 = 252$. Maka, hasil gabungan $P1$ sampai $P5 = 10,253,183,253,252$.

Keterangan: Pada hasil dekripsi, nilai ASCII untuk byte ke-1 sampai byte ke-5 ditampilkan sebagai tanda titik (".") karena nilai desimal 10, 253, 183, 253, dan 252 bukan merupakan karakter ASCII yang dapat dicetak (*non-printable character* atau *extended byte*). Oleh karena itu, aplikasi menggunakan simbol titik (.) sebagai penanda bahwa byte tersebut bukan karakter teks, melainkan bagian dari data biner penyusun sinyal Suara.

Analisis Visualisasi Waveform &Mel-Spectogram Sinyal Suara

Waveform suara asli menunjukkan pola amplitudo yang teratur. Setelah proses enkripsi RSA, pola gelombang berubah menjadi acak sehingga informasi suara tidak dapat dikenali. Setelah dilakukan dekripsi menggunakan kunci privat, waveform kembali menyerupai sinyal asli, yang menunjukkan bahwa data suara berhasil dikembalikan ke bentuk semula.

Berdasarkan visualisasi Mel-Spectrogram, sinyal hasil enkripsi memiliki distribusi energi yang berbeda dari sinyal asli akibat perubahan data menjadi ciphertext. Setelah proses dekripsi, pola Mel-Spectrogram kembali mendekati sinyal asli, yang menunjukkan bahwa implementasi RSA mampu menjaga kerahasiaan data selama proses enkripsi serta mempertahankan integritas data setelah proses dekripsi. Visualisasi tersebut ditunjukkan pada Gambar 12.



Gambar 12. Visualisasi Waveform & Mel-Spectogram

Pengujian Data Suara

Pengujian Ukuran Data Suara

Pengujian ini dilakukan untuk mengevaluasi hasil pengamanan data yang memperhitungkan kinerja terhadap Suara yang memiliki besar ukuran resolusi.

Tabel 2. Tabel Uji Ukuran.

Plaintext (Suara)	SuaraPendek.wav	SuaraSedang.wav	SuaraPanjang.wav
Durasi	5.62 detik	15.51 detik	30.56 detik
Ukuran Suara	242 kb	668 kb	1,28 mb
Ciphertext (.bin)	484 kb	1,30 mb	2,57 mb
Jumlah blok (byte)	495.616 bytes	1.368.064 bytes	2.695.168 bytes
Waktu Enkripsi	0.1038 detik	0.1029 detik	0.2702 detik
Waktu dekripsi	0.2034 detik	0.3904 detik	0.8539 detik

RSA memiliki tingkat keamanan yang tinggi karena menggunakan pasangan kunci publik dan kunci privat. Namun, algoritma ini memerlukan waktu komputasi yang relatif lebih besar dibandingkan algoritma simetris, terutama ketika digunakan pada file suara berukuran besar. Selain itu, ukuran ciphertext yang dihasilkan lebih besar daripada plaintext sehingga kebutuhan ruang penyimpanan meningkat. Oleh karena itu, pada implementasi sistem komunikasi suara secara real-time, RSA lebih sesuai digunakan sebagai mekanisme pertukaran kunci, sedangkan proses enkripsi data suara dilakukan menggunakan algoritma simetris seperti AES (hybrid cryptography).

Berdasarkan Tabel 1, semakin besar ukuran dan durasi suara, semakin banyak byte yang diproses sehingga ukuran ciphertext dan waktu komputasi enkripsi maupun dekripsi turut meningkat. Meskipun demikian, seluruh data uji berhasil diproses dengan baik. Pada perangkat uji dengan RAM 8 GB, sistem bekerja optimal untuk durasi suara hingga sekitar 30 detik. Untuk durasi yang lebih panjang, waktu komputasi meningkat sehingga disarankan

menggunakan perangkat dengan spesifikasi yang lebih tinggi agar proses enkripsi dan dekripsi lebih efisien.

Pengujian Kunci RSA

Uji Kunci RSA. Pengujian dilakukan menggunakan satu sampel suara dengan tiga pasangan bilangan prima yang berbeda untuk mengevaluasi pengaruh nilai modulus Nterhadap proses enkripsi dan dekripsi. Berdasarkan Tabel 3, pasangan bilangan prima (50021, 65537) dan (61, 53) menghasilkan nilai modulus $N > 255$sehingga proses enkripsi dan dekripsi berhasil dilakukan. Sebaliknya, pasangan (61, 3) menghasilkan nilai $N = 183$, sehingga proses dekripsi tidak dapat mengembalikan data suara secara utuh. Hasil ini menunjukkan bahwa nilai modulus RSA harus lebih besar dari 255 agar seluruh nilai byte suara dapat direpresentasikan dengan benar dan proses dekripsi berlangsung secara lossless.

Tabel 3. Uji Kunci RSA pada Aplikasi.

Pengujian	Suara.wav	Suara.wav	Suara.wav
Bil Prima	50021, 65537	61,53	61, 3
N	3278226277	3233	183
φ(N)	3278110720	3120	120
e	3	7	7
d	21854007147	1783	103
Waktu enkripsi	0.1841 detik	0.1056 detik	0.1013 detik
Waktu dekripsi	1.8246 detik	0.2274 detik	0.1079 detik
Status	Berhasil	Berhasil	Gagal N=183<256

5. KESIMPULAN DAN SARAN

Penelitian ini berhasil mengimplementasikan algoritma RSA pada aplikasi desktop berbasis Python untuk mengamankan pesan suara berformat WAV. Hasil pengujian menunjukkan bahwa proses enkripsi mampu mengubah data suara menjadi ciphertext sehingga tidak dapat dikenali, sedangkan proses dekripsi berhasil mengembalikan data ke bentuk semula. Analisis Waveform dan Mel-Spectrogram menunjukkan perubahan karakteristik sinyal setelah enkripsi dan kemiripan dengan sinyal asli setelah dekripsi, sehingga membuktikan bahwa algoritma RSA mampu menjaga kerahasiaan dan integritas data pesan suara. Selain itu, waktu komputasi dipengaruhi oleh ukuran file dan parameter kunci RSA, di mana semakin besar data yang diproses, semakin lama waktu enkripsi dan dekripsi yang dibutuhkan.

Penelitian selanjutnya disarankan mengembangkan sistem pada platform web maupun mobile dengan dukungan berbagai format file suara, seperti MP3, OGG, dan FLAC. Selain itu, pembangkitan bilangan prima p dan q pada algoritma RSA dapat dilakukan secara otomatis serta dipertimbangkan penggunaan kriptografi hibrida (RSA–AES) untuk meningkatkan

efisiensi komputasi. Evaluasi sistem juga perlu diperluas menggunakan parameter kualitas suara, seperti SNR, PSNR, MSE, dan BER, disertai perbandingan kuantitatif antara file suara asli dan hasil dekripsi. Pengujian pada ukuran kunci RSA yang lebih besar, variasi ukuran dan durasi file suara, serta optimasi algoritma juga disarankan untuk meningkatkan kinerja sistem.

DAFTAR REFERENSI

Azhari, D. H., Fauzi, A., & Sembiring, H. (2025). Super enkripsi kriptografi pengamanan pesan file audio record MP3 dengan algoritma Rivest Shamir Adleman (RSA) dan ElGamal. *Merkurius: Jurnal Riset Sistem Informasi dan Teknik Informatika, 3*(6), 176–190. https://doi.org/10.61132/merkurius.v3i6.1160

Diarse, N. N., & Bendi, K. J. (2016). Penerapan algoritma RSA pada sistem kriptografi file audio MP3. *Jurnal HOAQ: Teknologi Informasi, 7*, 567–575.

Fauji, S. A., Pradana, M. S., & Azhari, N. A. (2016). Penerapan kode Huffman pada algoritma RSA (Rivest-Shamir-Adleman) untuk menyandikan password email. *UJMC (Unisda Journal of Mathematics and Computer Science), 2*(1), 41–49.

Fuad, R. N., & Winata, H. N. (2017). Aplikasi keamanan file audio WAV (Waveform) dengan terapan algoritma RSA. *Jurnal InfoTekJar (Nasional Informatika dan Teknologi Jaringan), 1*(2), 113–119.

Ginting, A., Isnanto, R. R., & Windasari, I. P. (2015). Implementasi algoritma kriptografi RSA untuk enkripsi dan dekripsi email. *Jurnal Teknologi dan Sistem Komputer, 3*, 253–258.

Hakim, A., Anggraini, Z., Sillfani, D., Ayuni, R., & Fauzi, A. (2025). Penerapan super enkripsi Hill Cipher dan RSA untuk pengamanan data file audio MP3. *Jurnal Sistem Informasi Kaputama (JSIK), 9*(1), 55–64. https://doi.org/10.59697/jsik.v9i1.959

Hidayat, M., Tahir, M., Sukriyadi, A., Sulton, A., S. A., C. A., & F. S. A. (2023). Penerapan kriptografi Caesar Cipher dalam pengamanan data. *JUKIM (Jurnal Ilmiah Multidisiplin), 2*(3), 35–41. https://doi.org/10.56127/jukim.v2i03.619

I, J. O., O, O. G., & A, O. F. (2019). Performance analysis of RSA algorithm for audio data security in communication networks. *International Journal of Latest Technology in Engineering, Management & Applied Science (IJLTEMAS), 8*, 48–52. https://www.ijltemas.in

IBM Corporation. (2024). *Cost of a data breach report*. IBM.

Paramita, C., & Sudibyo, U. (2021). Kriptografi audio MP3 menggunakan RSA dan transposisi kolom. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), 5*(3), 483–488. https://doi.org/10.29207/resti.v5i3.2996

Rosdiana, H., Reja, & Setiawan, B. H. (2026). Peran Badan Siber dan Sandi Negara (BSSN) dalam arsitektur keamanan siber Indonesia. *Jurnal Riset Rumpun Ilmu Sosial, Politik dan Humaniora, 5*(1), 872–882. https://doi.org/10.55606/jurrish.v5i1.8471

Santoso, H., & Fakhriza, M. (2018). Perancangan aplikasi keamanan file audio format WAV (Waveform) menggunakan RSA. *ALGORITMA: Jurnal Ilmu Komputer dan Informatika, 2*(1), 47–54. https://doi.org/10.0000/2422369ca3ff4b10a3a4b6dd84c6d819

Statista. (2025). *Most popular global mobile messenger apps as of October 2025, based on number of monthly active users*. https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps/

Tobing, D. T. (2024). Implementasi algoritma Rivest Shamir Adleman (RSA) untuk keamanan data rekam medik penyakit pasien rumah sakit. *Jurnal Kajian Ilmiah Teknologi Informasi dan Komputer*, *2*(2), 65–73. https://journal.grahamitra.id/index.php/jutik

Yousif, S. F. (2018). Encryption and decryption of audio signal based on RSA algorithm. *International Journal of Engineering Technologies and Management Research*, *5*(7), 57–64. https://doi.org/10.29121/ijetmr.v5.i7.2018.259