e-ISSN: 2964-2663
p-ISSN: 2964-9692

Implementasi Steganografi pada Media Image dan Audio Menggunakan Metode LSB (*Least Significant Bit*) Pada PT 7ion Indonesia

Fahri Muhamad Husaini[1], Rama Adistya Nurtjahya Pamudji[2]
[1]*Teknik Informatika, STMIK Pranata Indonesia Bekasi*
e-mail: [1]mochhusaini8@gmail.com, [2]ramaadistyanurcahya@gmail.com

Peningkatan ketergantungan terhadap infrastruktur digital membawa risiko keamanan data, khususnya ancaman penyadapan dan intersepsi pesan rahasia pada jaringan publik. Kriptografi konvensional mengubah plaintext menjadi ciphertext, namun wujud acak tersebut justru mencolok dan memancing perhatian peretas. Penelitian ini bertujuan merancang dan mengimplementasikan aplikasi steganografi berlapis (Double Protection Layer) mengombinasikan algoritma RSA dan metode Least Significant Bit (LSB) untuk menyembunyikan file teks rahasia (.txt) ke dalam media penampung gambar (PNG) dan audio (WAV) pada PT 7ion Indonesia. Metode pengembangan sistem menggunakan model Waterfall dengan pemodelan Unified Modeling Language (UML) serta lingkungan Java NetBeans IDE. Hasil penelitian menunjukkan aplikasi berhasil mengeksekusi penyisipan (encode) dan ekstraksi (decode) pesan secara presisi tanpa menimbulkan distorsi visual maupun auditori pada stego-file. Pengujian Black Box mengonfirmasi seluruh fungsionalitas sistem berjalan valid 100%, sehingga memberikan solusi pengamanan data yang efektif, kokoh, dan imperceptible.
Kata Kunci: *Steganografi, Least Significant Bit (LSB), RSA, Media Gambar, Media Suara*

Abstract

Increased reliance on digital infrastructure introduces significant data security risks, particularly the threat of interception of confidential text messages on public networks. Conventional cryptography transforms plaintext into ciphertext, but its scrambled form is conspicuous and attracts unauthorized access. This research aims to design and implement a double-layer steganography application combining the RSA algorithm and the Least Significant Bit (LSB) method to conceal confidential text files (.txt) inside image (PNG) and audio (WAV) cover media at PT 7ion Indonesia. System development follows the Waterfall model utilizing Unified Modeling Language (UML) and Java NetBeans IDE. The results demonstrate that the application successfully performs encoding and decoding without inducing perceptible visual or auditory distortion in stego-files. Black Box testing verifies 100% functional validity, delivering a robust, imperceptible, and highly secure data protection solution.
Keywords: *Steganography, Least Significant Bit (LSB), RSA, Image Media, Audio Media*

I. PENDAHULUAN

Perkembangan teknologi informasi meningkatkan efisiensi pertukaran data, tetapi sekaligus memperbesar risiko penyadapan, pencurian, dan intersepsi informasi pada jaringan publik. Informasi rahasia yang dikirim melalui jaringan dapat menjadi sasaran pihak yang tidak berwenang sehingga diperlukan mekanisme pengamanan yang tidak hanya melindungi isi pesan, tetapi juga menyamarkan keberadaannya.

Kriptografi mengamankan isi pesan dengan mengubah plaintext menjadi ciphertext. Namun, bentuk ciphertext yang acak dapat menarik perhatian karena keberadaan informasi rahasia masih terlihat. Steganografi menawarkan pendekatan berbeda dengan menyisipkan pesan ke dalam media penampung sehingga keberadaan pesan tidak mudah diketahui. Metode Least Significant Bit (LSB) merupakan teknik yang memanfaatkan bit paling rendah pada data media untuk membawa informasi rahasia.

Penelitian terdahulu menunjukkan bahwa LSB dapat digunakan untuk menyisipkan pesan teks ke dalam citra tanpa kerusakan visual yang mudah diamati. Pamudji (2024) menunjukkan penerapan

e-ISSN: 2964-2663
p-ISSN: 2964-9692

LSB pada media image, sedangkan penelitian terkait LSB juga menekankan karakteristiknya yang sederhana dan memiliki tingkat distorsi rendah. Berdasarkan hal tersebut, penelitian ini mengembangkan aplikasi yang menggabungkan RSA dan LSB pada media image dan audio.

Tujuan penelitian adalah merancang dan mengimplementasikan aplikasi steganografi untuk menyembunyikan file teks (.txt) pada media PNG/BMP dan WAV, menerapkan RSA sebagai lapisan pengamanan pesan sebelum penyisipan, serta menguji fungsionalitas encode dan decode menggunakan *Black Box Testing.*

II. METODE PENELITIAN

Jenis dan Tahapan Penelitian

Penelitian ini merupakan penelitian perancangan dan pengembangan aplikasi. Model pengembangan perangkat lunak yang digunakan adalah Waterfall karena tahapan pengembangan dilakukan secara berurutan dan terstruktur. Tahap penelitian meliputi analisis kebutuhan, perancangan sistem, implementasi, pengujian, serta evaluasi hasil.

Tahap analisis mengidentifikasi kebutuhan pengamanan pesan dan kebutuhan media penampung. Tahap perancangan memodelkan interaksi sistem menggunakan Unified Modeling Language (UML) serta merancang antarmuka pengguna. Tahap implementasi menerapkan Java pada NetBeans IDE dengan JDK 8. Tahap pengujian dilakukan menggunakan Black Box Testing untuk memeriksa fungsi utama encode dan decode.

Mekanisme Steganografi

Proses pengamanan menggunakan dua lapisan. Pertama, pesan teks diproses melalui kriptografi RSA sehingga menghasilkan ciphertext. RSA menggunakan pasangan kunci publik dan privat; kunci tersebut digunakan untuk proses pengamanan dan pembukaan pesan. Kedua, ciphertext disisipkan ke dalam media penampung menggunakan metode LSB. Pada proses decode, bit-bit LSB diekstraksi untuk memperoleh ciphertext, kemudian ciphertext dibuka kembali menggunakan mekanisme kunci RSA sehingga pesan teks dapat dipulihkan.

Media yang digunakan meliputi gambar PNG dan BMP serta audio WAV. Pemilihan media tersebut mengikuti batasan implementasi penelitian. LSB memodifikasi bagian bit dengan bobot paling rendah sehingga perubahan pada media relatif kecil dan tidak mudah terlihat atau terdengar.

Perangkat dan Lingkungan Pengembangan

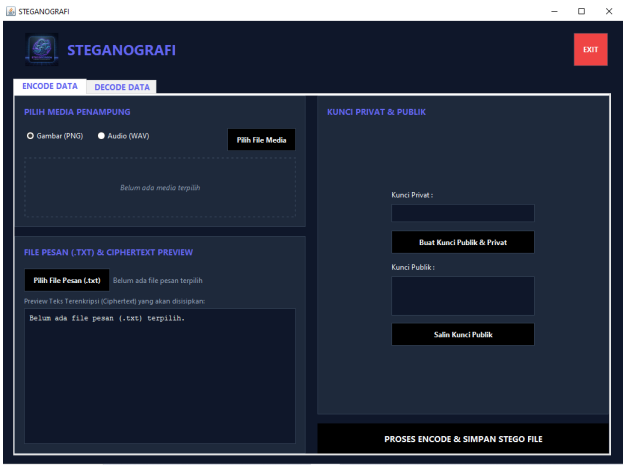
Aplikasi steganografi ini dibangun dan diuji pada lingkungan perangkat keras dan perangkat lunak dengan spesifikasi sebagai berikut:

- Processor: AMD Dual Core A4-9125 up to 2.6 GHz
- Memori RAM: 4 GB DDR4
- Penyimpanan: Harddisk 1 TB
- Sistem Operasi: Microsoft Windows 10/11 (64-bit)
- Bahasa Pemrograman: Java (JDK 8)
- IDE Development: NetBeans IDE

III. HASIL DAN PEMBAHASAN

Implementasi Antarmuka

Antarmuka aplikasi dirancang untuk memisahkan proses Encode Data dan Decode Data. Pada modul Encode, pengguna memilih media penampung, memilih file pesan, membuat pasangan kunci, lalu menjalankan proses encode. Pada modul Decode, pengguna memuat stego-file, memasukkan kunci yang diperlukan, kemudian menjalankan proses ekstraksi dan dekripsi.

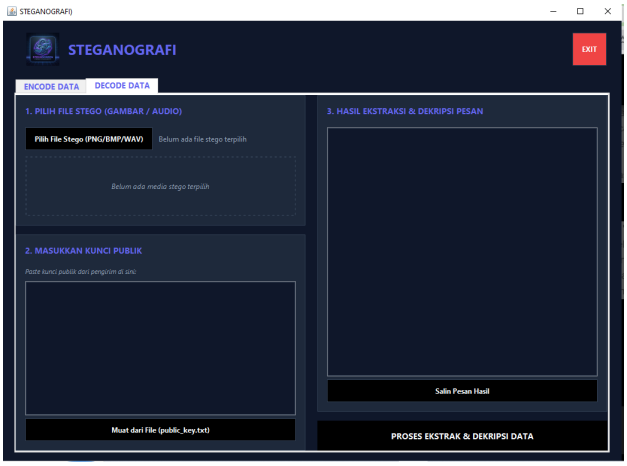


Gambar 1. Tampilan antarmuka modul Encode dan pemberitahuan proses

Gambar 1 memperlihatkan antarmuka modul Encode. Pengguna dapat menentukan media penampung, memuat file pesan, mengelola kunci,

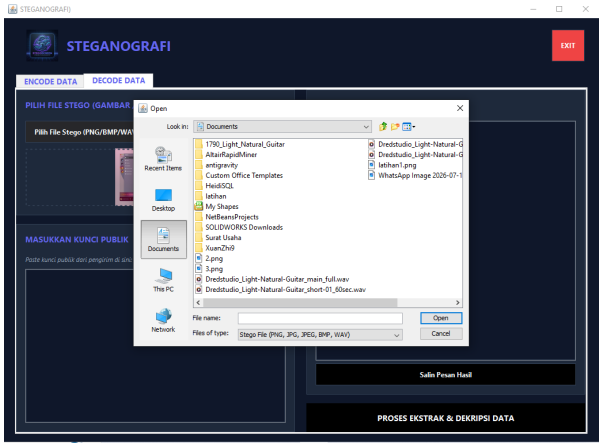
e-ISSN: 2964-2663
p-ISSN: 2964-9692

dan menjalankan proses encode. Setelah proses selesai, sistem memberikan pemberitahuan bahwa stego-file berhasil dibuat dan disimpan.

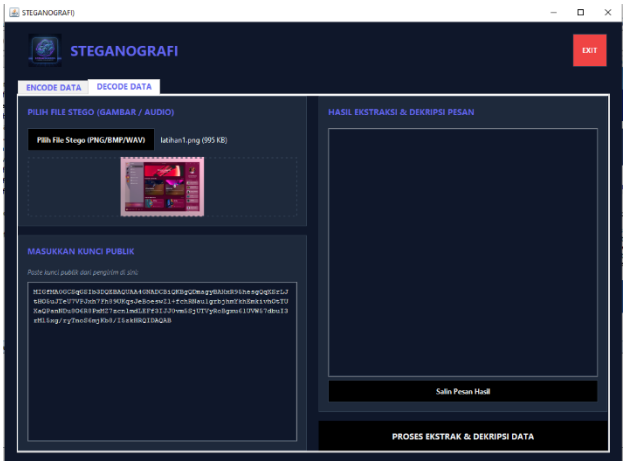


Gambar 2. Tampilan antarmuka modul Decode

Gambar 2 memperlihatkan modul Decode. Antarmuka menyediakan area pemilihan stego-file, bagian autentikasi kunci, dan tombol proses ekstrak serta dekripsi data. Pemisahan modul membantu pengguna mengikuti tahapan pengamanan dan pembukaan pesan secara sistematis.



Gambar 3. Pemilihan file stego melalui file chooser



Gambar 4. Area input kunci pada modul Decode

Gambar 3 dan Gambar 4 menunjukkan tahap persiapan decode, yaitu pemilihan file stego dan pengisian kunci. Sistem kemudian menggunakan input tersebut untuk mengekstraksi ciphertext dari media dan melakukan proses pembukaan pesan.

Proses Encode dan Decode

Alur encode dimulai ketika pengguna memilih jenis media penampung dan file yang akan digunakan. Pesan teks dibaca aplikasi dan diproses menjadi ciphertext menggunakan RSA. Ciphertext kemudian diubah menjadi representasi bit dan disisipkan pada bagian LSB media penampung. Hasilnya disimpan sebagai stego-file.

Alur decode merupakan kebalikan proses encode. Pengguna memuat stego-file, kemudian sistem membaca bit-bit LSB pada media dan menyusun kembali ciphertext. Setelah ciphertext diperoleh, sistem melakukan proses dekripsi menggunakan kunci yang sesuai sehingga pesan dapat dipulihkan kembali dalam bentuk file teks.

Pengujian Black Box

Pengujian dilakukan terhadap fungsi utama pada modul Encode dan Decode. Pengujian pada Encode mencakup pemilihan media, pemilihan file pesan, pembuatan kunci, penyalinan kunci publik, proses encode, dan penyimpanan stego-file. Pengujian Decode mencakup pemeriksaan file, pemilihan stego-file, pemasukan kunci, serta proses ekstrak dan dekripsi.

e-ISSN: 2964-2663
p-ISSN: 2964-9692

Tabel 1. Hasil pengujian Black Box aplikasi

No	Skenario Pengujian	Hasil yang Diharapkan	Status
1	Menekan Tombol Pilih File Media (PNG/BMP/WAV)	Sistem membuka dialog file chooser & memuat media	Valid
2	Menekan Tombol Pilih File Pesan (.txt)	Sistem memuat isi teks pesan rahasia (.txt)	Valid
3	Menekan Tombol Buat Kunci Publik & Privat	RSA generate key pair & menampilkan ciphertext	Valid
4	Menekan Tombol Salin Kunci Publik	String kunci publik tersimpan ke clipboard	Valid
5	Menekan Tombol PROSES ENCODE & SIMPAN STEGO FILE	Penyisipan LSB berhasil & dialog simpan stego-file muncul	Valid
6	Menentukan nama berkas & menekan Save	Stego-file tersimpan secara fisik di direktori lokal	Valid

Analisis Hasil Pengujian

Hasil pengujian memperlihatkan bahwa seluruh fungsi utama dapat dijalankan sesuai rancangan. Tingkat validitas fungsional mencapai 100%, yaitu 10 dari 10 skenario pengujian dinyatakan valid. Hasil tersebut menunjukkan bahwa alur encode dan decode telah memenuhi kebutuhan fungsional yang ditetapkan.

Pada modul Encode, aplikasi dapat menerima media dan file pesan, menyiapkan pasangan kunci, menyisipkan ciphertext, serta menyimpan stego-file. Pada modul Decode, aplikasi berhasil memuat stego-file, menerima kunci, mengekstraksi data yang disisipkan, dan mengembalikan pesan melalui proses dekripsi. Dengan demikian, integrasi RSA sebagai lapisan kriptografi dan LSB sebagai teknik penyamaran dapat berjalan pada alur aplikasi.

Pembahasan

Penerapan LSB memberikan keuntungan berupa perubahan data penampung yang relatif kecil karena bit yang dimodifikasi merupakan bit berbobot paling rendah. Hasil ini sejalan dengan penelitian yang membahas karakteristik LSB dengan tingkat distorsi rendah. Pada media image, perubahan tersebut tidak mudah diamati secara kasatmata, sedangkan media WAV digunakan sebagai alternatif cover audio dalam sistem.

Kombinasi RSA dan LSB memiliki fungsi yang berbeda. RSA berperan menjaga kerahasiaan isi pesan melalui transformasi plaintext menjadi ciphertext, sedangkan LSB menyamarkan keberadaan ciphertext dengan menempatkannya pada media penampung. Pendekatan berlapis ini memberikan pengamanan yang lebih komprehensif dibandingkan hanya menerapkan salah satu teknik.

Perbandingan dengan Penelitian Terdahulu

Penelitian Pamudji (2024) menerapkan LSB pada media image dan menunjukkan bahwa pesan dapat disisipkan serta diekstraksi kembali tanpa kerusakan visual yang mudah diamati. Penelitian ini memperluas penerapan tersebut dengan menambahkan kriptografi RSA dan mendukung dua kategori media, yaitu image dan audio. Penelitian Lestari (2021) dan Nugroho (2021) menekankan karakteristik LSB yang sederhana dan memiliki distorsi rendah, yang juga terlihat pada implementasi penelitian ini.

Perbedaan utama penelitian ini terletak pada integrasi dua lapisan pengamanan dan implementasi dalam aplikasi desktop Java. RSA digunakan sebelum penyisipan sehingga data yang ditanamkan bukan plaintext, sedangkan LSB digunakan untuk menyamarkan keberadaan data tersebut di dalam media. Hasil Black Box menunjukkan seluruh fungsi yang diuji valid.

Keterbatasan Implementasi

Keterbatasan penelitian terletak pada jenis media dan lingkungan implementasi. Sistem difokuskan pada pesan teks dan media penampung PNG/BMP serta WAV sehingga belum menguji seluruh format citra, audio, atau dokumen. Pengujian yang dilakukan juga berfokus pada validitas fungsi melalui Black Box Testing; evaluasi kualitas media secara kuantitatif dan pengujian ketahanan terhadap manipulasi atau kompresi belum menjadi bagian dari pengujian utama. Batasan tersebut dapat menjadi dasar pengembangan pada penelitian berikutnya.

Implikasi Hasil

Hasil penelitian menunjukkan bahwa penerapan RSA sebelum LSB dapat membentuk mekanisme pengamanan berlapis. Jika pihak lain memperoleh stego-file, data yang tersembunyi tidak langsung

e-ISSN: 2964-2663
p-ISSN: 2964-9692

berupa plaintext karena pesan telah melalui proses kriptografi. Pada sisi lain, penyisipan LSB membantu menyamarkan keberadaan data sehingga media tetap dapat digunakan sebagai pembawa pesan.

Dari sisi penggunaan, aplikasi desktop berbasis Java memberikan lingkungan yang sederhana untuk menguji algoritma dan antarmuka secara lokal. Pemisahan modul Encode dan Decode membantu memperjelas alur kerja pengguna, sedangkan dukungan terhadap image dan audio memberikan alternatif media penampung sesuai kebutuhan.

Secara keseluruhan, hasil implementasi dan pengujian menunjukkan bahwa rancangan dapat direalisasikan menjadi aplikasi yang berfungsi sesuai kebutuhan penelitian. Evaluasi lanjutan dapat diarahkan pada pengukuran kualitas media, kapasitas penyisipan, performa waktu proses, serta ketahanan stego-file terhadap kompresi atau perubahan data.

IV. KESIMPULAN

Penelitian ini menghasilkan aplikasi steganografi berbasis Java yang mampu menyisipkan file teks (.txt) ke dalam media PNG/BMP dan WAV menggunakan metode LSB dengan perlindungan RSA. Proses encode dan decode dapat dijalankan sesuai rancangan, sedangkan antarmuka memisahkan tahapan pengamanan dan pembukaan pesan agar lebih terstruktur. Pengujian Black Box terhadap 10 skenario menunjukkan seluruh skenario valid dengan tingkat keberhasilan fungsional 100%.

Pengembangan berikutnya dapat memperluas format media dan jenis file yang dapat disisipkan serta menambahkan pengukuran kualitas media secara kuantitatif, misalnya PSNR atau MSE, untuk memberikan evaluasi yang lebih terukur terhadap perubahan media setelah penyisipan.

V. REFERENSI

Hidayat, R., & Pratama, A. (2023). Penerapan model Waterfall dalam pembangunan sistem informasi berbasis Java. Jurnal Rekayasa Perangkat Lunak, 4(2), 50–58.

Kurniawan, D., & Syahputra, E. (2022). Analisis komputasi kriptografi kunci asimetris RSA dalam mengamankan transmisi teks. Jurnal Sains dan Komputer, 11(3), 112–120.

Lestari, P. (2021). Evaluasi performa algoritma LSB pada citra digital terkompresi. Jurnal Media Informatika, 8(1), 30–38.

Nugraha, A., & Saputra, I. (2023). Pemenuhan aspek confidentiality dan integrity menggunakan kombinasi enkripsi. Jurnal Keamanan Jaringan, 6(1), 40–48.

Nugroho, W. (2021). Analisis distorsi visual metode Least Significant Bit pada ranah spasial. Jurnal Komputer dan Grafika, 7(2), 72–80.

Pamudji, R. A. N. (2024). Analisis dan perancangan aplikasi steganografi pada media image dengan metode LSB (Least Significant Bit). Jurnal Teknologi dan Komputer, 12(1), 15–24.

Pratama, F., & Safitri, N. (2023). Konsep dasar rekayasa perangkat lunak berbasis objek. Jurnal Informatika Mulawarman, 18(1), 12–20.

Rahmawati, D. (2022). Implementasi enkripsi RSA dalam pengamanan dokumen digital. Jurnal Sistem Informasi Bisnis, 10(2), 28–35.

Rahmawati, D. (2023). Teknik penyembunyian bit pesan menggunakan metode LSB. Jurnal Ilmu Komputer dan Keamanan, 11(1), 68–75.

Nina, S., & Handoko, T. (2023). Analisis Penyembunyian Data Teks pada Audio Berformat WAV menggunakan LSB. *Jurnal Pengolahan Sinyal Digital*, 5(2), 88-96.

Novianto, B. (2022). *Perancangan Sistem Informasi Keamanan Data Enterprise*. PT Elex Media Komputindo.

Stallings, No., & others. (2023). Cryptography and Network Security. Pearson.

Sugiarto, H. (2024). *Rekayasa Perangkat Lunak Terapan*. Informatika Bandung.

Yakub, (2021). *Pengantar Sistem Informasi*. Graha Ilmu.