PENERAPAN STEGANOGRAFI VIDEO MENGGUNAKAN ALGORITMA *LEAST SIGNIFICANT BIT* (LSB) DAN *ADVANCED ENCRYPTION STANDARD* (AES) UNTUK KEAMANAN DATA

Disusun Oleh:

Nama : Muhammad Hafiz
NIM : 21523228

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
2026

HALAMAN PENGESAHAN DOSEN PEMBIMBING

PENERAPAN STEGANOGRAFI VIDEO MENGGUNAKAN ALGORITMA *LEAST SIGNIFICANT BIT* (LSB) DAN *ADVANCED ENCRYPTION STANDARD* (AES) UNTUK KEAMANAN DATA

TUGAS AKHIR

Disusun Oleh:

N a m a	:Muhammad Hafiz
NIM	:21523228

Yogyakarta, 15 Januari 2026

Pembimbing,

(Dr. Yudi Prayudi,S.Si., M.Kom.)

HALAMAN PENGESAHAN DOSEN PENGUJI

PENERAPAN STEGANOGRAFI VIDEO MENGGUNAKAN ALGORITMA *LEAST SIGNIFICANT BIT* (LSB) DAN *ADVANCED ENCRYPTION STANDARD* (AES) UNTUK KEAMANAN DATA

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 15 Januari 2026

Tim Penguji

Dr. Yudi Prayudi, S.Si., M.Kom.

Anggota 1

Moh. Idris, S.Kom., M.Kom.

Anggota 2

Taufiq Hidayat, S.T., M.C.S.

Mengetahui,
Ketua Program Studi Informatika – Program Sarjana
Fakultas Teknologi Industri
Universitas Islam Indonesia

(Dhomas Hatta Fudholi,S.T., M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama	:	Muhammad Hafiz
NIM	:	21523228

Tugas akhir dengan judul:

PENERAPAN STEGANOGRAFI VIDEO MENGGUNAKAN ALGORITMA *LEAST SIGNIFICANT BIT* (LSB) DAN *ADVANCED ENCRYPTION STANDARD* (AES) UNTUK KEAMANAN DATA

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 15 Januari 2026

(Muhammad Hafiz)

HALAMAN PERSEMBAHAN

Assalamualaikum warahmatullahi wabarakatuh.

Segala puji dan syukur saya panjatkan kepada Allah Swt. atas limpahan rahmat dan karunia-Nya, sehingga saya dapat menyelesaikan tugas akhir ini dengan lancar. Shalawat serta salam semoga senantiasa tercurah kepada Nabi Muhammad saw., panutan sepanjang masa. Terselesaikannya tugas akhir ini saya harap menjadi langkah awal menuju kehidupan yang lebih bermakna dan memberi manfaat bagi banyak orang.

Penelitian ini saya dedikasikan kepada keluarga tercinta yang selalu memberikan kasih sayang, doa, dan dukungan tanpa henti dalam setiap perjalanan hidup saya. Terima kasih yang sebesar-besarnya saya sampaikan kepada bunda saya serta seluruh kerabat atas segala pengorbanan, usaha, dan cinta yang tak ternilai. Doa dan semangat yang kalian berikan menjadi sumber kekuatan utama saya dalam menyelesaikan proses ini. Kehadiran keluarga adalah karunia terbesar yang selalu menyemangati saya untuk terus memberikan yang terbaik.

Saya juga mengucapkan terima kasih yang mendalam kepada dosen pembimbing Bapak Yudi Prayudi yang telah dengan penuh kesabaran membimbing dan memberikan arahan selama penyusunan penelitian ini. Ucapan terima kasih juga saya sampaikan kepada seluruh dosen dan staf pengajar di Program Studi Informatika Universitas Islam Indonesia atas ilmu, bimbingan, dan pengalaman berharga yang telah diberikan selama masa perkuliahan.

Ucapan terima kasih ini juga saya tujukan kepada seluruh teman seperjuangan di Program Studi Informatika UII, yang telah menjadi rekan diskusi, penyemangat, serta pendukung selama menempuh studi. Kebersamaan dan kerja sama kalian sangat berarti bagi saya dalam menghadapi berbagai tantangan akademik.

Akhir kata, saya menyampaikan terima kasih yang tulus kepada semua pihak yang telah membantu dan mendukung penyusunan skripsi ini, baik secara langsung maupun tidak langsung. Semoga hasil dari penelitian ini dapat memberikan manfaat yang luas bagi para pembaca dan masyarakat secara umum

HALAMAN MOTO

"You miss 100% of the shots you don't take."

(Wayne Gretzky)

"Dan bahwa manusia hanya memperoleh apa yang telah diusahakannya"

(QS. An Najm:39)

KATA PENGANTAR

Segala puji dan syukur saya panjatkan ke hadirat Allah Swt. atas limpahan rahmat, hidayah, dan karunia-Nya, sehingga saya dapat menyelesaikan penyusunan laporan tugas akhir ini dengan baik. Shalawat dan salam semoga senantiasa tercurah kepada Nabi Muhammad saw., sosok teladan utama bagi seluruh umat manusia.

Laporan tugas akhir ini disusun sebagai salah satu syarat kelulusan di Program Studi Informatika, Universitas Islam Indonesia. Adapun judul penelitian ini adalah "*Penerapan Steganografi Video Menggunakan Algoritma Least Significant Bit (LSB) dan Advanced Encryption Standard Untuk Keamanan Data*." Dalam proses penyusunannya, saya menghadapi berbagai hambatan, seperti keterbatasan waktu, tantangan dalam pengumpulan data, serta kendala teknis lainnya. Namun berkat bantuan dan dukungan dari banyak pihak, saya dapat menyelesaikan penelitian ini dengan sebaik-baiknya.

Dengan Penuh rasa hormat dan terimakasih , saya ingin menyampaikan apresiasi kepada:

1. Bunda saya Rosinah yang selalu senantiasa memberi doa dan dukungan kepada saya.
2. Keluarga besar saya yang selalu memberi dukungan dengan caranya sendiri.
3. Bapak Ir. Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D., selaku Ketua Program Studi Informatika Program Sarjana, atas arahan dan dukungan yang diberikan selama masa studi.
4. Bapak Dr. Yudi Prayudi, S.Si., M.Kom., selaku dosen pembimbing tugas akhir, yang dengan penuh kesabaran telah membimbing, memberikan arahan, serta masukan yang sangat membantu selama proses penelitian ini.
5. Bapak Galang Prihadi Mahardhika, S.Kom., M.Kom., dosen pembimbing akademik saya, atas segala bimbingan dan arahannya selama menjalani studi di Program Studi Informatika
6. Seluruh dosen dan staf di Program Studi Informatika UII yang telah membagikan ilmu, pengalaman, dan inspirasi yang begitu berharga selama masa kuliah.
7. M.Rasyid Baihaki, M.Irfan Abigail, Bagas Wahyu Herdiansyah, M.Javier Rasyadi, Fajar Juliyanto, yang telah menjadi teman seperjuangan, tempat berbagi semangat, kerja sama, dan kebersamaan yang memperkuat saya dalam menghadapi masa studi.
8. Teman Indekos Putra Tiara Citra yang memeberikan dukungan, semangat, serta kebersamaan

9. Dan seluruh pihak lainnya yang telah memberikan bantuan dan kontribusi, baik secara langsung maupun tidak langsung, dalam penyusunan tugas akhir ini.

Yogyakarta, 15 Januari 2026

(Muhammad Hafiz)

SARI

Penelitian ini mengintegrasikan teknik steganografi *Least Significant Bit* (LSB) dan kriptografi *Advanced Encryption Standard* (AES) untuk meningkatkan keamanan pertukaran informasi pada media video digital. Menggunakan bahasa pemrograman Python, sistem ini dirancang untuk menyembunyikan pesan teks yang telah terenkripsi ke dalam frame video secara imperseptibel. Pendekatan eksperimen terapan digunakan untuk mengevaluasi efektivitas sistem dalam menjaga kerahasiaan data tanpa menyebabkan degradasi visual yang signifikan pada media penampung (*cover*).

Hasil pengujian menunjukkan bahwa integrasi AES dan LSB berhasil diimplementasikan dengan tingkat akurasi ekstraksi pesan yang sempurna. Berdasarkan parameter *Peak Signal-to-Noise Ratio* (PSNR), video steganografi yang dihasilkan memiliki nilai rata-rata berkisar antara 61,67 dB hingga 106,60 dB, yang secara signifikan melampaui ambang batas standar kelayakan 40 dB. Hal ini mengindikasikan bahwa sistem mampu mempertahankan kualitas visual yang tinggi sehingga perubahan pada video tidak dapat dideteksi oleh indra penglihatan manusia. Selain itu, penggunaan enkripsi AES terbukti memberikan lapisan keamanan yang kuat, di mana pesan rahasia tetap terlindungi dan hanya dapat dipulihkan melalui kunci dekripsi yang tepat.

Kata kunci: Steganografi Video, Eksperimen Terapan, *Advanced Encryption Standard* (AES), *Least Significant Bit* (LSB), *Peak Signal-to-Noise Ratio* (PSNR).

GLOSARIUM

Bit	Unit informasi terkecil dalam sistem komputasi digital yang hanya memiliki dua kemungkinan nilai, yaitu 0 atau 1.
Cover Video	Berkas video asli yang berfungsi sebagai media penampung atau wadah untuk menyembunyikan pesan rahasia.
Dekripsi	Proses transformasi data yang telah terenkripsi (*ciphertext*) kembali menjadi data asli (*plaintext*)
Enkripsi	Proses mengamankan informasi dengan cara mengubah data asli (*plaintext*) menjadi format yang tidak dapat dibaca oleh pihak yang tidak berwenang (*ciphertext*)
Frame	Satuan gambar tunggal atau bingkai citra digital yang menyusun sebuah rangkaian video.
Piksel	Elemen terkecil dari citra digital yang mengandung informasi warna
Stego Video	Berkas video hasil akhir yang telah melalui proses penyisipan pesan.

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
PENDAHULUAN

1.1 Latar Belakang

Informasi merupakan sesuatu yang sangat berharga. Pentingnya menjaga kerahasiaan informasi telah menjadi perhatian tersendiri karena apabila jatuh ke pihak yang tidak berwenang dapat menimbulkan kerugian yang signifikan. Oleh karena itu, berbagai metode digunakan untuk merahasiakan informasi. Salah satu metode yang digunakan adalah steganografi, yaitu teknik penyembunyian pesan rahasia ke dalam media penampung. Media digital yang dapat digunakan dalam steganografi antara lain citra, suara, maupun video (Riadi et al., 2020).

Perkembangan teknologi informasi, jaringan, dan internet yang semakin pesat memberikan kemudahan dalam memperoleh informasi, baik yang bersifat umum maupun rahasia. Pada informasi yang bersifat rahasia, diperlukan upaya pengamanan sehingga hanya dapat diakses oleh pihak yang berwenang . Dalam konteks ini, video merupakan salah satu media yang efektif untuk menyembunyikan pesan atau informasi rahasia, yang kemudian dikenal dengan istilah steganografi video. Selain itu, kriptografi juga berperan penting sebagai ilmu dan seni menjaga keamanan pesan melalui enkripsi (mengubah *plaintext* menjadi *ciphertext*) dan dekripsi (mengembalikan *ciphertext* menjadi *plaintext*), sehingga informasi tidak dapat dipahami oleh pihak yang tidak berhak (Minarni et al., 2023).

Seiring dengan popularitas data multimedia, khususnya video digital, teknik penyembunyian data dalam video berkembang pesat. Video dinilai lebih unggul dibandingkan media digital lainnya, seperti citra, teks, maupun audio, karena memiliki kapasitas penyimpanan data yang lebih besar serta tingkat redundansi yang tinggi pada frame-frame berurutan (Fuad & Ernawan, 2020). Hal ini membuat video menjadi media yang lebih aman dan kompleks untuk penerapan steganografi dibandingkan dengan citra atau audio . Oleh sebab itu, metode penyembunyian data dalam file video semakin luas digunakan, baik untuk kebutuhan komunikasi rahasia, perlindungan hak cipta, maupun kepemilikan.

Steganografi sendiri bukanlah bidang baru. Sejak zaman kuno, manusia telah berupaya menyembunyikan pesan atau informasi yang dianggap berharga dengan berbagai cara. Catatan sejarah menunjukkan bahwa penggunaan steganografi pertama kali terjadi pada tahun 440 SM, ketika Demeratus, Raja Sparta, mengirimkan peringatan kepada rakyatnya dengan menulis pesan pada sebuah balok kayu kemudian menutupinya dengan lilin (Şahin et al., 2021). Hal ini

membuktikan bahwa konsep komunikasi tersembunyi telah ada sejak lama dan terus berevolusi mengikuti perkembangan zaman.

Penerapan steganografi video kini semakin meluas pada berbagai bidang. Di sektor intelijen dan militer, steganografi digunakan untuk menyamarkan komunikasi agar tidak terdeteksi pihak yang tidak berwenang, karena kebocoran informasi dapat mengancam keamanan nasional. Di sektor kesehatan, steganografi video dimanfaatkan untuk melindungi data pasien yang disimpan secara digital dan ditransfer melalui jaringan internet, sehingga privasi tetap terjaga meskipun data tersebut berpindah melalui saluran komunikasi yang rentan terhadap serangan siber (Kunhoth et al., 2023). Selain itu, teknologi ini juga digunakan dalam bidang multimedia untuk menjaga hak cipta serta pada sistem pengawasan (*surveillance*) untuk melindungi identitas individu yang terekam dalam video dengan cara menyembunyikan data pribadi ke dalam rangkaian frame video.

Metode *Least Significant Bit* (LSB) dikenal sebagai teknik yang ringan dan efisien dalam penerapan steganografi digital (Set et al., 2025). LSB merupakan salah satu teknik penyembunyian pesan yang bekerja dengan cara mengubah pesan ke dalam bentuk biner, kemudian menyisipkannya pada bit terendah dalam media digital, misalnya citra atau frame video. Teknik ini banyak digunakan karena pendekatannya yang sederhana dan relatif mudah diimplementasikan. Bahkan, dalam literatur internasional, LSB disebut sebagai salah satu metode steganografi yang paling populer digunakan hingga saat ini (Aslam et al., 2022).

Namun demikian, meskipun sederhana dan efektif, teknik LSB memiliki kelemahan. Bit terendah yang menjadi lokasi penyisipan pesan relatif mudah dianalisis dan dibongkar dengan teknik analisis statistik, sehingga kerahasiaan pesan dapat terancam apabila tidak disertai dengan sistem keamanan tambahan (Nirmala, 2020). Oleh karena itu, dibutuhkan mekanisme keamanan tambahan, seperti penggunaan algoritma kriptografi, agar data yang disisipkan tetap terlindungi meskipun metode LSB berhasil dibongkar.

Penggabungan steganografi dan kriptografi menjadi salah satu solusi dalam meningkatkan keamanan informasi. Keduanya memiliki perbedaan prinsip: pada steganografi, pesan disembunyikan ke dalam media digital sehingga keberadaannya tidak disadari, sedangkan pada kriptografi pesan diacak agar tidak bisa dibaca oleh pihak yang tidak berhak (Riadi et al., 2020). Kriptografi sendiri merupakan cara yang digunakan untuk menjaga keamanan pesan sehingga data atau dokumen yang diproses tidak dapat dibaca dengan mudah oleh pihak yang tidak berwenang. Metode kriptografi secara umum dibagi menjadi dua, yaitu metode simetris dan asimetris, berdasarkan jenis kunci yang digunakan. Beberapa algoritma

enkripsi yang umum digunakan antara lain DES, 3DES, AES, Blowfish, RC4, dan RSA (Malvi & Painem, 2020).

Dari berbagai algoritma tersebut, *Advanced Encryption Standard* (AES) dipilih karena beberapa alasan utama. Pertama, AES merupakan standar enkripsi simetris modern yang diakui secara internasional dengan tingkat keamanan tinggi serta efisiensi yang baik untuk data dalam jumlah besar. Kedua, AES memiliki struktur blok cipher yang kuat dan tahan terhadap serangan kriptanalisis yang umum, sehingga sangat sesuai untuk melindungi pesan rahasia sebelum disisipkan melalui metode LSB. Dengan demikian, kombinasi metode LSB dan algoritma AES memberikan solusi yang seimbang antara efisiensi penyisipan dan kekuatan keamanan data.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana mengimplementasikan metode steganografi video menggunakan algoritma LSB yang dikombinasikan dengan enkripsi AES?
2. Sejauh mana efektivitas dan keamanan kombinasi algoritma LSB dan AES dalam menjaga kerahasiaan informasi?

1.3 Pertanyaan Penelitian

1. Bagaimana cara algoritma LSB dan AES diintegrasikan dalam proses steganografi video?
2. Bagaimana pengaruh kombinasi LSB dan AES terhadap kualitas video dan keamanan data ?

1.4 Batasan Masalah

Dengan terdapatnya masalah yang ada, maka penelitian akan dilakukan dengan beberapa batasan masalah sebagai berikut :

1. Penelitian dibatasi pada penggunaan metode steganografi dengan algoritma LSB sebagai teknik penyisipan data dan AES sebagai algoritma enkripsi.
2. Jenis media yang digunakan terbatas pada video digital.
3. Parameter evaluasi difokuskan pada kualitas video hasil penyisipan dan tingkat keamanan data yang disembunyikan.

1.5 Tujuan dan Manfaat Penelitian

Tujuan dari penelitian ini adalah untuk mengembangkan metode steganografi video yang aman dan efisien melalui penggabungan algoritma *Least Significant Bit* (LSB) dengan enkripsi *Advanced Encryption Standard* (AES). Dengan mengombinasikan kedua metode tersebut, diharapkan dapat diperoleh suatu pendekatan yang tidak hanya mampu menyembunyikan pesan secara imperseptibel dalam video, tetapi juga memberikan perlindungan tambahan terhadap isi pesan melalui proses enkripsi.

Manfaat dari penelitian ini diharapkan dapat memberikan kontribusi nyata dalam bidang keamanan data dan komunikasi digital. Hasil penelitian dapat menjadi referensi dalam pengembangan metode steganografi modern yang lebih kuat, serta memberikan dasar bagi penelitian lanjutan yang berfokus pada integrasi antara steganografi dan kriptografi. Selain itu, penelitian ini juga diharapkan mampu mendukung kebutuhan praktis di berbagai sektor, seperti perlindungan informasi rahasia, keamanan komunikasi, maupun perlindungan hak cipta dalam konten multimedia digital.

1.6 Metode Penelitian

Metode yang digunakan ialah sebagai berikut :

a. Studi Literatur

   Studi literatur dilakukan dengan mempelajari buku, jurnal ilmiah, dan halama web yang berhubungan dengan ilmu steganografi, kriptografi, dan video digital.

b. Analisis Masalah

   Pada tahap ini perkerjaan yang dilakukan adalah menganalisis kebutuhan penelitian terkait penyisipan pesan/data steganografi terhadap video digital.

c. Pengumpulan data uji, dilakukan dengan menyiapkan sejumlah video yang akan digunakan sebagai sampel dalam proses penyisipan dan ekstraksi.

d. Perancangan Sistem, yaitu penyusunan rancangan sistem secara menyeluruh agar implementasi dapat dilakukan sesuai dengan kebutuhan yang telah dianalisis.

e. Implementasi Sistem, merupakan proses penerjemahan rancangan ke dalam bentuk kode program menggunakan bahasa pemrograman Python.

f. Pengujian dan Analisis Hasil, dilakukan untuk memastikan sistem berjalan sesuai harapan dan dapat digunakan untuk menyisipkan serta mengekstrak data dengan aman.

BAB II
LANDASAN TEORI

2.1 Keamanan Data

Keamanan data merupakan upaya untuk melindungi informasi dari ancaman seperti pencurian, perubahan tanpa izin, maupun penyalahgunaan oleh pihak yang tidak berwenang. Tujuan utama dari keamanan data adalah menjaga agar informasi tetap bersifat rahasia, utuh, dan selalu dapat digunakan saat dibutuhkan. Konsep dasar yang digunakan sebagai acuan dalam keamanan data adalah Triad CIA, yang terdiri dari tiga aspek utama, yaitu kerahasiaan (*confidentiality*), integritas (*integrity*), dan ketersediaan (*availability*) (Unggul Budi Astowo, 2024).

Kerahasiaan berkaitan dengan perlindungan agar data hanya dapat diakses oleh pihak yang memiliki hak. Integritas berfungsi untuk memastikan bahwa data tetap asli dan tidak mengalami perubahan tanpa izin. Sementara itu, ketersediaan menjamin bahwa data dapat digunakan oleh pihak yang berwenang pada saat diperlukan. Untuk mendukung ketiga aspek tersebut, digunakan dua teknik dalam keamanan data, yaitu kriptografi dan steganografi. Kriptografi berperan dalam mengamankan isi data melalui proses enkripsi, sedangkan steganografi berfungsi menyembunyikan keberadaan data agar tidak mudah terdeteksi. Perpaduan kedua teknik ini dapat memperkuat penerapan konsep Triad CIA dalam menjaga keamanan data secara menyeluruh. Pemahaman terhadap keamanan data selanjutnya perlu dilengkapi dengan pengetahuan mengenai media yang dapat digunakan dalam proses perlindungan informasi, salah satunya adalah video digital.

2.2 Video Digital

Video digital merupakan salah satu media penyimpanan informasi yang tersusun dari rangkaian gambar diam yang disebut frame. Setiap frame terdiri dari piksel-piksel yang merepresentasikan warna dan intensitas cahaya. Ketika frame ditampilkan secara berurutan dalam jumlah tertentu per detik—dikenal sebagai *Frame Per Second* (FPS)—video akan terlihat sebagai rangkaian gerakan yang halus. Semakin tinggi nilai FPS, semakin halus pergerakan yang ditampilkan, namun juga semakin besar kapasitas data yang dibutuhkan

Dalam konteks keamanan data, video digital menjadi media yang sangat potensial karena memiliki banyak ruang redundansi visual khususnya di tingkat piksel dan frame yang dapat dimanfaatkan untuk teknik penyembunyian informasi. Setiap frame menyimpan ribuan hingga jutaan piksel, dan perubahan kecil pada nilai piksel tertentu tidak akan terlihat secara signifikan

oleh penglihatan manusia. Hal ini menjadikan video digital sebagai media yang populer dalam penerapan steganografi, terutama pada metode berbasis bit terkecil seperti *Least Significant Bit* (LSB). Selain itu, karakteristik temporal pada video, yaitu keberadaan banyak frame, memberikan kapasitas penyisipan yang lebih besar dibandingkan gambar (Belyaev, 2023).

Oleh karena itu, pemahaman mengenai struktur dan karakteristik video digital sangat penting dalam pengembangan sistem keamanan data. Sebagaimana dijelaskan sebelumnya, keamanan data bertujuan menjaga kerahasiaan, integritas, dan ketersediaan informasi. Dengan memanfaatkan media video, teknik kriptografi dan steganografi dapat diterapkan secara bersamaan pesan dienkripsi untuk menjaga kerahasiaan, kemudian disisipkan ke dalam frame video untuk menyembunyikan keberadaannya. Hal ini menjadikan video digital salah satu media yang sangat mendukung implementasi konsep Triad CIA dalam upaya perlindungan informasi.

2.3 Steganografi

Steganografi berasal dari bahasa Yunani, yaitu *steganos* yang berarti tersembunyi atau terselubung, dan *graphein* yang berarti menulis. Secara terminologi, steganografi diartikan sebagai seni dan ilmu untuk berkomunikasi dengan cara menyembunyikan informasi sehingga keberadaannya tidak dapat dideteksi oleh pihak lain (Riadi et al., 2021). Dengan kata lain, steganografi merupakan teknik penyembunyian pesan yang memungkinkan informasi rahasia disampaikan secara tersembunyi di dalam suatu media digital tanpa menarik perhatian pihak ketiga.

Steganografi memanfaatkan keterbatasan sistem indera manusia, seperti penglihatan (*human visual system*) maupun pendengaran (*human auditory system*), sehingga pesan rahasia yang disisipkan tidak dapat disadari secara langsung oleh manusia. Teknik ini memungkinkan pesan yang tersembunyi tetap bertahan melalui berbagai proses pengolahan sinyal digital tanpa menurunkan kualitas media penampung hingga batas tertentu. Media yang dapat digunakan dalam steganografi sangat beragam, di antaranya teks, citra, audio, dan video.

Steganografi merupakan metode penyembunyian data rahasia pada media digital agar keberadaannya tidak diketahui orang lain. Dalam steganografi dikenal beberapa istilah penting, yaitu: (a) *hidden text* atau *embedded message*, yaitu pesan yang disembunyikan; (b) *cover text*, yaitu media yang digunakan sebagai penampung pesan; dan (c) *stego text* atau *stego object*, yaitu media yang telah berisi pesan tersembunyi (Nirmala, 2020). Proses penyisipan pesan disebut *encoding*, sedangkan proses pengambilan kembali pesan disebut *decoding*. Untuk

menjamin bahwa hanya pihak berwenang yang dapat melakukan proses *encoding* dan *decoding*, biasanya digunakan kunci rahasia (*stego key*).

Atoum dan Ibrahim mendefinisikan steganografi sebagai ilmu dan seni menyembunyikan informasi rahasia sehingga hanya pengirim dan penerima yang mengetahui keberadaan pesan tersebut (Riadi et al., 2020). Proses steganografi umumnya melibatkan kriptografi untuk memperkuat keamanan data. Tahapan yang dilakukan meliputi enkripsi pesan asli (*plain text*) menjadi *cipher text*, kemudian menyisipkannya ke dalam media digital berupa teks, citra, audio, atau video. Dengan demikian, steganografi dapat dilihat sebagai mekanisme yang memungkinkan data digital seperti gambar atau file tertentu disembunyikan di dalam file lain sehingga sulit dideteksi keberadaannya.

2.3.1 Steganografi Video

Video sebagai media digital memiliki karakteristik yang unik dibandingkan teks, citra, maupun audio. Video tersusun atas serangkaian frame citra yang ditampilkan secara berurutan sehingga membentuk ilusi gerakan. Redundansi data antar frame pada video menjadikan media ini memiliki kapasitas penyimpanan yang besar untuk menyisipkan pesan rahasia. Selain itu, kompleksitas struktur video juga membuat proses deteksi pesan tersembunyi menjadi lebih sulit dibandingkan media lain. Oleh karena itu, video dipandang sebagai salah satu media yang efektif untuk penerapan steganografi.

Proses steganografi video umumnya dilakukan dengan menyisipkan pesan pada frame tertentu, baik di domain spasial (piksel langsung) maupun di domain transformasi (misalnya *Discrete Cosine Transform*/DCT atau *Discrete Wavelet Transform*/DWT). Dalam praktiknya, frame video diperlakukan seperti citra digital, namun dengan perhatian khusus terhadap kesinambungan antar frame agar pesan yang disisipkan tetap tidak terlihat (*imperceptible*).

2.3.2 Metode Steganografi

Berikut beberapa metode yang paling umum digunakan untuk steganografi:

1. *Least Significant Bit* (LSB)

   Teknik paling sederhana yang menyisipkan pesan pada bit paling rendah dari data piksel citra atau frame video. LSB populer karena ringan dan mudah diimplementasikan, namun rentan terhadap analisis statistik.

2. *Masking and Filtering*

   Teknik ini menyamarkan pesan dengan memanfaatkan area tertentu dalam media digital, misalnya melalui pewarnaan atau pola tertentu pada citra.

3. *Transform Domain Techniques*

   Pesan disembunyikan dalam koefisien transformasi, misalnya DCT, DWT, atau FFT. Teknik ini lebih tahan terhadap kompresi dan manipulasi data dibandingkan metode spasial.

4. *Spread Spectrum*

   Menggunakan konsep komunikasi nirkabel, pesan disebarkan ke seluruh spektrum frekuensi sehingga lebih sulit dideteksi.

2.4 Algoritma *Least Significant Bit* (LSB)

Teknik steganografi dengan menggunakan metode modifikasi *Least Significant Bit* (LSB) merupakan salah satu teknik paling sederhana untuk menyisipkan informasi di dalam suatu citra digital atau *media cover* (Fitriani, 2020; Laksono et al., 2024). LSB bekerja dengan cara mengganti bit paling rendah atau paling kanan dalam representasi biner data, di mana perubahan hanya akan menggeser nilai byte satu tingkat lebih tinggi atau lebih rendah dari nilai sebelumnya, sehingga tidak memberikan perbedaan signifikan pada media asli. Misalnya, jika byte tersebut menyatakan warna merah dalam gambar, maka perubahan pada satu bit LSB tidak akan mengubah warna tersebut secara berarti, bahkan sulit dibedakan oleh mata manusia. Pada struktur biner, bit bagian kiri disebut *Most Significant Bit* (MSB), sedangkan bit bagian kanan adalah *Least Significant Bit* (LSB) yang menjadi bagian paling cocok untuk disisipi pesan karena nilainya paling kecil.

Dalam implementasinya, setiap piksel pada citra 24-bit (R=8-bit, G=8-bit, B=8-bit) dapat menyimpan hingga 3 bit pesan rahasia. Sebagai contoh, gambar berukuran 1.024 × 768 piksel memiliki kapasitas penyisipan hingga 2.359.296 bit informasi rahasia (Nirmala, 2020). Proses dalam steganografi dengan metode LSB umumnya melibatkan dua tahap, yaitu penyembunyian (*embedding*) dan ekstraksi data. Pada tahap *embedding*, bit-bit pesan rahasia disisipkan ke dalam bit LSB media cover, sedangkan pada tahap ekstraksi, bit-bit tersebut diambil kembali untuk memperoleh pesan asli (Simbolon & Nusantara, 2021). Dengan cara ini, data berukuran terbatas dapat disimpan dalam media digital lain, dan hasil modifikasi tetap terlihat serupa dengan media aslinya.

Sebagai ilustrasi sederhana, misalkan terdapat gambar yang berukuran 3X3 piksel dengan format 24 bit RGB, dimana setiap kanal warna (R,G,B), memiliki panjang 8 bit. Setiap piksel memiliki kombinasi warna yang membentuk gambar seperti berikut.

Gambar 2.1 Ilustrasi Kombinasi Kanal RGB dalam Citra 24-bit.

Misalkan pesan rahasia yang akan disisipkan adalah “TES”, yang dalam ASCII bernilai 116, 101, dan 115 atau dalam biner 01110100, 01100101, dan 01110011. Bit-bit pesan ini akan disisipkan ke bit paling tidak signifikan (LSB) dari tiap komponen warna secara berurutan. menghasilkan perubahan seperti berikut (hanya bagian terakhir tiap byte yang berbeda):

Gambar 2.2 Kombinasi Kanal RGB setelah LSB.

Perubahan ini hanya memodifikasi bit terakhir dari setiap komponen warna (LSB), sehingga perbedaan citra tidak terlihat oleh mata manusia. Meskipun tampilan citra masih sama persis, kini citra tersebut menyimpan 24 bit pesan rahasia, yaitu “TES”. Dengan demikian,

teknik *Least Significant Bit* (LSB) terbukti mampu menyembunyikan informasi kecil di dalam media gambar tanpa mengubah tampilan visualnya secara signifikan.

Konsep yang sama juga diterapkan pada media video, di mana setiap frame video diperlakukan seperti sebuah citra digital tempat pesan dapat disisipkan ke bit-bit paling rendahnya. Melalui pendekatan ini, pesan rahasia dapat didistribusikan ke banyak frame sehingga kapasitas penyimpanan meningkat dan deteksi menjadi lebih sulit dilakukan.

Namun demikian, penggunaan LSB dalam media video memiliki kelebihan dan kekurangan. Kelebihannya adalah metode ini sederhana, memiliki kapasitas penyimpanan besar, serta tidak menimbulkan perubahan visual yang signifikan sehingga sulit dideteksi oleh indera manusia. Selain itu, karena video memiliki banyak frame, penyisipan pesan dapat dilakukan secara tersebar sehingga meningkatkan keamanan (Nirmala, 2020; Simbolon & Nusantara, 2021).

Sedangkan kekurangannya, metode LSB sangat rentan terhadap serangan manipulasi atau kompresi *lossy* (misalnya MPEG atau H.264) yang dapat mengubah bit LSB sehingga pesan rahasia hilang atau rusak (Fitriani, 2020; Simbolon & Nusantara, 2021). Selain itu, teknik ini relatif mudah dilacak menggunakan metode analisis statistik (*steganalysis*), karena pola bit yang dimodifikasi dapat menimbulkan anomali tertentu pada distribusi data digital. Oleh sebab itu, metode LSB sering dipadukan dengan algoritma kriptografi seperti AES untuk meningkatkan keamanan dan kerahasiaan pesan.

2.5 Kapasitas *Payload*

Implementasi metode steganografi *Least Significant Bit* (LSB) pada media video digital secara langsung menentukan besaran ruang penyimpanan yang tersedia untuk data rahasia. Karena metode LSB bekerja dengan memanipulasi bit terendah pada setiap komponen warna *pixel*, maka total ketersediaan ruang simpan, atau yang disebut dengan kapasitas *payload*, berbanding lurus dengan dimensi resolusi dan durasi video tersebut. Secara teoritis, kapasitas *payload* maksimum dapat dihitung dengan mengintegrasikan variabel resolusi spasial (lebar dan tinggi frame) dengan resolusi temporal (jumlah frame per detik dan durasi). Rumus matematis untuk menghitung kapasitas total dalam satuan bit dinyatakan pada persamaan (2.1) berikut.

$$Kapasitas = Lebar \times Tinggi \times Kanal \times n \times fps \times Durasi \qquad (2.1)$$

Di mana Kanal adalah jumlah kanal warna (pada standar RGB nilai Kanal Adalah 3), n adalah jumlah bit LSB yang digunakan per kanal, serta fps dan durasi masing-masing mewakili *frame rate* dan durasi video dalam detik.

Sebagai ilustrasi perhitungan teoritis, dapat diambil contoh pada format video digital standar dengan resolusi 360p (640 x 360 *pixel*), frame rate 30 fps, dan durasi 10 detik dengan asumsi penggunaan 1-bit LSB. Dalam satu bingkai gambar (frame), terdapat total 230.400 *pixel* (640 x 360). Karena setiap *pixel* terdiri dari 3 kanal warna (*Red, Green, Blue*) dan setiap kanal menyumbangkan 1 bit ruang simpan, maka kapasitas per frame adalah sebesar 691.200 bit. Jika diakumulasikan selama durasi 10 detik (total 300 frame), maka kapasitas kumulatif yang dihasilkan secara matematis mencapai 207.360.000 bit atau setara dengan 25,92 Megabyte.

2.6 Kriptografi

Kriptografi merupakan suatu teknik yang digunakan untuk melindungi kerahasiaan serta keamanan data dengan cara mengonversi pesan asli ke dalam bentuk yang tidak dapat dipahami oleh pihak yang tidak memiliki hak akses. Proses tersebut dilakukan melalui dua tahapan utama, yaitu enkripsi sebagai proses pengubahan *plaintext* menjadi *ciphertext*, serta dekripsi sebagai proses untuk mengembalikan *ciphertext* ke bentuk semula dengan menggunakan kunci tertentu (Minarni et al., 2023). Dalam penerapannya, kriptografi tidak hanya berperan dalam menyembunyikan isi pesan, tetapi juga menjamin keutuhan data dan memastikan bahwa informasi hanya dapat diakses oleh pihak yang berwenang.

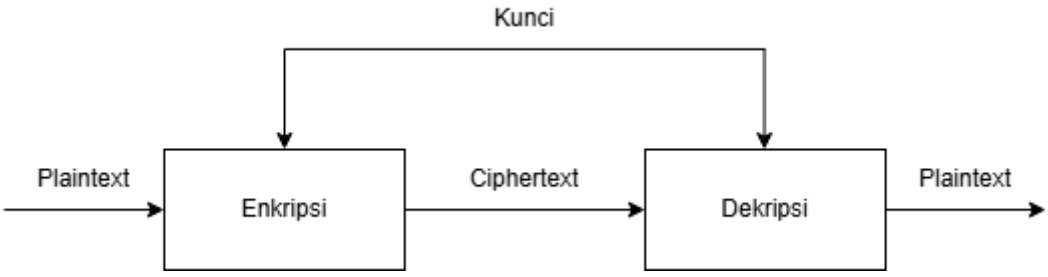
Secara etimologi, kata kriptografi berasal dari bahasa Yunani, yaitu *kryptós* yang berarti tersembunyi dan *gráphein* yang berarti tulisan, sehingga secara harfiah diartikan sebagai tulisan tersembunyi (Nirmala, 2020). Pengertian ini sejalan dengan tujuan utama kriptografi, yakni menyamarkan pesan melalui proses transformasi khusus. Sejumlah ahli menggolongkan kriptografi sebagai bagian dari ilmu matematika karena melibatkan proses perhitungan numerik, sementara sebagian lainnya memandangnya sebagai seni karena teknik penyandian dapat dilakukan dengan cara yang kreatif dan beragam. Secara umum, kriptografi dapat dipahami sebagai seni mengacak pesan agar tidak dapat dipahami oleh pihak yang tidak berhak (Riadi et al., 2020).

Dalam perkembangannya, kriptografi dibedakan menjadi dua kategori utama, yaitu kriptografi simetris dan kriptografi asimetris. Kriptografi simetris menggunakan satu kunci yang sama baik untuk proses enkripsi maupun dekripsi. Metode ini memiliki keunggulan dari sisi kecepatan dan efisiensi, namun memerlukan tingkat keamanan yang tinggi dalam pendistribusian kunci. Sementara itu, kriptografi asimetris menggunakan dua kunci yang

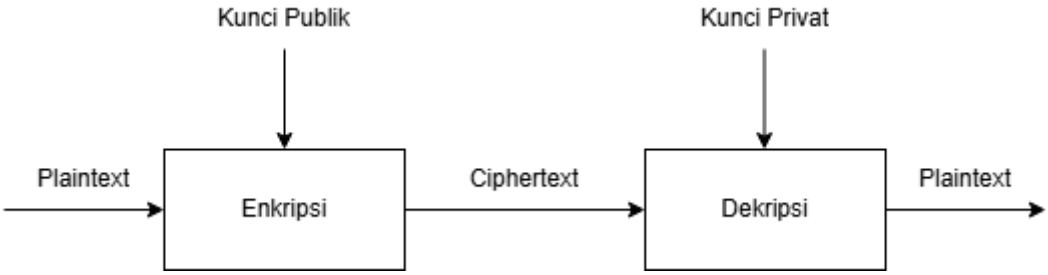
berbeda, yaitu kunci publik untuk enkripsi dan kunci privat untuk dekripsi. Model ini lebih aman dalam hal distribusi kunci, meskipun memiliki kelemahan dari sisi kecepatan akibat tingkat kompleksitas algoritmanya. Kedua pendekatan ini menjadi dasar utama dalam sistem pengamanan data modern, sebagaimana ditunjukkan pada ilustrasi kriptografi simetris dan asimetris.

Sebagai lanjutan dari penjelasan tersebut, mekanisme kerja kriptografi secara umum mengikuti alur yang relatif sederhana. Pengirim terlebih dahulu menyiapkan pesan asli yang akan diamankan, kemudian pesan tersebut diproses menggunakan algoritma tertentu beserta kunci yang telah ditentukan untuk menghasilkan *ciphertext*. *Ciphertext* inilah yang selanjutnya dikirimkan melalui media komunikasi yang digunakan. Apabila pesan tersebut disadap oleh pihak lain, isi pesan tidak dapat dipahami karena telah berubah menjadi bentuk tersandi. Setelah pesan diterima oleh pihak yang berwenang, proses dekripsi dilakukan menggunakan kunci yang sesuai sehingga *ciphertext* dapat dikembalikan menjadi *plaintext* yang dapat dibaca. Mekanisme ini berlaku pada kriptografi simetris maupun asimetris, dengan perbedaan utama terletak pada penggunaan dan pengelolaan kuncinya.

Dalam sistem kriptografi, kunci (*key*) memegang peranan yang sangat penting karena berfungsi sebagai parameter utama dalam proses enkripsi dan dekripsi. Kunci digunakan untuk mengendalikan proses perubahan *plaintext* menjadi *ciphertext* serta mengembalikannya kembali ke bentuk semula. Tanpa kunci yang sesuai, pesan yang telah terenkripsi tidak akan dapat dibuka meskipun algoritma yang digunakan diketahui. Pada kriptografi simetris, satu kunci yang sama digunakan oleh pengirim dan penerima, sehingga keamanan sistem sangat bergantung pada kerahasiaan kunci tersebut. Sementara itu, pada kriptografi asimetris digunakan sepasang kunci yang terdiri dari kunci publik dan kunci privat, di mana kunci publik digunakan untuk proses enkripsi dan kunci privat digunakan untuk proses dekripsi. Contoh algoritma kriptografi simetris yang menggunakan satu kunci adalah AES, sedangkan contoh algoritma kriptografi asimetris yang menggunakan pasangan kunci adalah RSA. Oleh karena itu, pengelolaan kunci yang baik menjadi faktor utama dalam menentukan tingkat keamanan suatu sistem kriptografi.



Gambar 2. 1 Kriptografi Simetris.



Gambar 2. 2 Kriptografi Asimetris.

2.7 Algoritma *Advanced Encryption Standard* (AES)

Advanced Encryption Standard (AES) merupakan salah satu algoritma kriptografi simetris yang terbukti aman, efisien, dan telah banyak diadopsi secara luas untuk melindungi data sensitif (Set et al., 2025). AES dipilih oleh National Institute of Standards and Technology (NIST) sebagai standar enkripsi melalui proses seleksi yang ketat dan ditetapkan dalam Federal Information Processing Standards (FIPS). Algoritma Rijndael akhirnya terpilih dari beberapa kandidat algoritma lain sebagai dasar dari AES, yang bekerja dalam bentuk block cipher (Nirmala, 2020).

Berbeda dengan DES yang menggunakan struktur Feistel, AES menggunakan struktur *Substitution-Permutation Network* (SPN) yang lebih efisien dan kuat. Pada AES, proses enkripsi melibatkan serangkaian transformasi yang dilakukan dalam blok 128-bit dengan panjang kunci 128-bit, 192-bit, atau 256-bit. Setiap proses enkripsi terdiri dari beberapa tahapan, yaitu: *AddRoundKey*, *SubBytes*, *ShiftRows*, *MixColumns*, dan kembali ke *AddRoundKey*. Untuk AES-128 terdapat 10 putaran (rounds), AES-192 memiliki 12 putaran, sedangkan AES-256 memiliki 14 putaran (Nirmala, 2020; Set et al., 2025).

Dasar-dasar enkripsi AES berfokus pada operasi substitusi, permutasi, dan pencampuran data yang dilakukan berulang kali dalam blok, sehingga menghasilkan *ciphertext* yang sulit diprediksi tanpa kunci yang sesuai. Keamanan AES bergantung pada kompleksitas matematisnya, ukuran kunci yang panjang, serta ketidakmungkinan serangan brute-force

dalam waktu yang efisien dengan teknologi komputasi saat ini. Hal ini menjadikan AES sebagai salah satu algoritma paling kuat dan andal dalam bidang keamanan data.

Dalam konteks steganografi, AES berperan penting untuk memberikan lapisan keamanan tambahan. Setelah data disisipkan ke dalam media penampung, misalnya citra atau video, data tersebut kemudian dienkripsi menggunakan AES agar lebih terlindungi. Proses ini memastikan bahwa meskipun pesan rahasia berhasil ditemukan di dalam media penampung, isinya tetap tidak dapat dibaca tanpa kunci enkripsi yang benar. Dengan demikian, kombinasi antara teknik penyembunyian (steganografi) dan teknik pengacakan (kriptografi AES) menghasilkan sistem keamanan berlapis yang mampu melindungi kerahasiaan serta integritas data.

2.7.1 Prinsip Kerja AES

Mekanisme kerja *Advanced Encryption Standard* (AES) didasarkan pada proses iterasi transformasi yang dilakukan terhadap sebuah matriks data dua dimensi. Unit data terkecil yang diproses dalam algoritme ini disebut dengan State, sebuah matriks persegi berukuran 4 X 4 byte. Pada awal proses enkripsi, blok input 128-bit akan disusun secara berurutan kolom demi kolom ke dalam matriks state. Sebagai algoritme berbasis *Substitution-Permutation Network* (SPN), AES mengombinasikan lapisan substitusi non-linear dan lapisan pergeseran untuk memastikan bahwa setiap bagian dari hasil enkripsi memiliki ketergantungan yang kompleks terhadap data asli dan kunci rahasia. (Oktavani et al., 2023)

Inti dari keamanan AES terletak pada siklus putaran (*rounds*) yang dilakukan secara berulang. Untuk AES-256 yang digunakan dalam penelitian ini, terdapat 14 putaran transformasi. Setiap putaran diawali dengan tahap SubBytes, di mana setiap byte dalam matriks state digantikan secara independen menggunakan sebuah tabel substitusi statis yang disebut S-Box. Transformasi ini berfungsi sebagai pertahanan utama untuk memutus hubungan linear antara data asli dengan kunci, sehingga menyulitkan upaya analisis pola oleh pihak yang tidak berwenang.

Setelah proses substitusi, dilakukan tahap *ShiftRows* yang memberikan pergeseran posisi pada tiap baris matriks secara horizontal. Baris pertama tetap pada posisinya, sementara baris kedua, ketiga, dan keempat digeser ke kiri dengan jumlah pergeseran yang berbeda-beda. Langkah ini memastikan bahwa byte yang awalnya berada dalam satu kolom yang sama akan tersebar ke kolom-kolom yang berbeda pada iterasi berikutnya. Proses ini dilanjutkan dengan tahap *MixColumns*, di mana setiap kolom pada matriks state diproses secara matematis untuk mencampurkan data dalam satu kolom tersebut. Kombinasi antara *ShiftRows* dan *MixColumns* bertujuan agar pengaruh dari satu byte data asli tersebar secara luas ke seluruh blok data hasil

enkripsi. Dengan demikian, perubahan kecil pada satu bit data asli akan mengakibatkan perubahan menyeluruh pada hasil akhir enkripsi.

Tahap terakhir dalam setiap putaran adalah *AddRoundKey*, yaitu proses penggabungan matriks state dengan sub-kunci (*round key*) menggunakan operasi bitwise XOR. Berbeda dengan tahap lainnya yang berfungsi untuk pengacakan posisi dan substitusi nilai, *AddRoundKey* adalah satu-satunya bagian yang secara langsung melibatkan kunci rahasia untuk memberikan keamanan pada data. Seluruh kunci yang digunakan dalam setiap putaran dikelola oleh unit *Key Expansion*, yang secara sistematis menghasilkan deretan sub-kunci dari satu kunci utama 256-bit. Melalui rangkaian transformasi yang terukur ini, AES mampu menjamin kerahasiaan data yang sangat tinggi dengan efisiensi komputasi yang optimal untuk media digital seperti video.

2.7.2 Mode Operasi Block Cipher

Setelah memahami bagaimana algoritma AES melakukan transformasi data di dalam satu matriks state, muncul tantangan teknis ketika data rahasia yang akan disisipkan memiliki ukuran yang jauh lebih besar daripada kapasitas satu blok standar (128-bit atau 16 byte). Dalam kondisi ini, algoritme AES memerlukan suatu mekanisme pengaturan yang disebut dengan mode operasi. Mode operasi block cipher menentukan prosedur yang digunakan untuk mengaplikasikan transformasi enkripsi terhadap serangkaian blok data yang saling berurutan. Berikut adalah beberapa mode operasi yang umum digunakan dalam standar kriptografi:

1. *Electronic Codebook* (ECB)

   Mode ECB merupakan metode operasi yang paling dasar, di mana setiap blok data asli (*plaintext*) dienkripsi secara mandiri menggunakan kunci yang sama untuk setiap bloknya. Secara teknis, jika terdapat dua atau lebih blok data yang memiliki nilai bit yang identik, maka hasil enkripsinya pun akan selalu identik. Karakteristik ini menjadi kelemahan utama dalam steganografi, karena pola atau struktur pada data rahasia masih dapat terlihat secara jelas pada hasil enkripsi, sehingga memberikan celah bagi penyerang untuk melakukan analisis statistis terhadap media penampung.

2. *Cipher Block Chaining* (CBC)

   Mode CBC, yang diterapkan dalam penelitian ini, merupakan pengembangan untuk mengatasi kelemahan pada mode ECB. Pada mode ini, setiap blok data rahasia terlebih dahulu digabungkan dengan hasil enkripsi dari blok sebelumnya melalui operasi bitwise XOR sebelum masuk ke tahap transformasi AES. Untuk memulai proses pada

blok pertama, digunakan sebuah *Initialization Vector* (IV) sebagai pemicu awal. Mekanisme perantaian ini memastikan bahwa blok data yang identik sekalipun akan menghasilkan hasil enkripsi yang jauh berbeda, sehingga struktur data asli menjadi sepenuhnya acak dan tidak berpola. Sifat ini sangat krusial dalam menyembunyikan eksistensi data rahasia di dalam video agar tidak menimbulkan anomali pada *pixel-pixel* yang digunakan.

3. *Cipher Feedback* (CFB) dan *Output Feedback* (OFB)

   Berbeda dengan CBC yang berorientasi pada blok, mode CFB dan OFB memungkinkan algoritme block cipher untuk bekerja menyerupai *stream cipher*. Pada mode CFB, hasil enkripsi dari blok sebelumnya digunakan sebagai input bagi proses enkripsi berikutnya untuk menghasilkan aliran kunci (*keystream*) yang kemudian digabungkan dengan data asli. Sementara itu, pada mode OFB, yang digunakan sebagai umpan balik adalah hasil keluaran dari enkripsi sebelumnya sebelum digabungkan dengan data asli. Kedua mode ini sering digunakan dalam transmisi data yang bersifat berkelanjutan (*streaming*), namun memiliki kompleksitas implementasi yang lebih tinggi dibandingkan CBC dalam hal sinkronisasi data.

4. *Counter* (CTR)

   Mode CTR bekerja dengan menggunakan nilai pencacah (*counter*) yang berbeda untuk setiap blok data. Nilai counter inilah yang dienkripsi oleh algoritma AES untuk menghasilkan aliran kunci yang unik bagi setiap blok. Keunggulan utama dari mode ini adalah kemampuannya untuk melakukan pemrosesan data secara paralel, karena enkripsi satu blok tidak bergantung pada blok lainnya. Meskipun menawarkan efisiensi waktu komputasi yang tinggi, mode CTR memerlukan manajemen nilai counter yang sangat ketat agar tidak terjadi pengulangan yang dapat merusak kerahasiaan data.

2.7.3 *Data Padding* (PKCS#7)

Penerapan mode operasi seperti *Cipher Block Chaining* (CBC) yang telah dibahas sebelumnya menuntut agar seluruh data rahasia diproses dalam bentuk blok-blok yang utuh dan simetris. Mengingat mekanisme AES bekerja secara eksklusif pada ukuran blok tetap sebesar 128-bit (16 byte), maka muncul sebuah batasan teknis di mana panjang data asli (*plaintext*) harus merupakan kelipatan tepat dari ukuran blok tersebut. Namun, pada

implementasi praktisnya, pesan rahasia yang akan disisipkan ke dalam media video seringkali memiliki ukuran yang bervariasi dan tidak selalu habis dibagi 16. Kondisi ketidaksesuaian ukuran ini mengharuskan adanya mekanisme pengisian byte tambahan agar blok terakhir mencapai panjang yang standar, yang dalam dunia kriptografi dikenal dengan istilah *padding*.

Tanpa adanya mekanisme *padding* yang terstandarisasi, algoritme enkripsi akan mengalami kegagalan proses atau error saat menemui blok terakhir yang tidak lengkap. Dalam penelitian ini, standar yang digunakan untuk menangani permasalahan tersebut adalah PKCS#7 (*Public-Key Cryptography Standards #7*). PKCS#7 bukan sekadar pengisi ruang kosong, melainkan sebuah protokol pengisian byte yang cerdas dan konsisten. Mekanismenya bekerja dengan menambahkan sejumlah byte pada akhir data asli, di mana nilai dari setiap byte yang ditambahkan tersebut identik dengan jumlah total byte yang ditambahkan. Sebagai contoh, apabila sebuah blok data kekurangan 3 byte untuk mencapai ukuran 16 byte, maka standar PKCS#7 akan menambahkan tiga byte dengan nilai 0x03 0x03 0x03 (dalam format heksadesimal).

Kelebihan utama dari penggunaan PKCS#7 terletak pada kemampuannya untuk memastikan proses dekripsi dan pembuangan *padding* (*unpadding*) berjalan secara akurat. Ketika data yang telah dienkripsi didekripsi kembali, sistem hanya perlu melihat nilai byte terakhir dari pesan tersebut untuk mengetahui secara pasti berapa banyak byte tambahan yang harus dibuang guna mendapatkan pesan asli yang utuh. Hal yang unik dari standar ini adalah perlakuannya terhadap data yang sudah memiliki panjang kelipatan blok. Jika data asli sudah tepat berukuran 16 byte atau kelipatannya, PKCS#7 tetap diwajibkan untuk menambahkan satu blok *padding* penuh (16 byte baru) yang berisi nilai 0x10 (desimal 16). Hal ini dilakukan untuk menghindari ambiguitas; tanpa penambahan blok penuh tersebut, sistem dekripsi mungkin akan salah menafsirkan byte terakhir dari data asli sebagai sebuah *padding* dan menghapusnya secara tidak sengaja.

2.8 PSNR

PSNR (*Peak Signal-to-Noise Ratio*) merupakan parameter yang digunakan untuk mengukur tingkat kemiripan antara dua data citra berdasarkan perbandingan antara nilai maksimum sinyal dengan tingkat kesalahan yang terjadi. PSNR dinyatakan dalam satuan desibel (dB) karena menggunakan skala logaritmik. Semakin tinggi nilai PSNR, maka semakin kecil perbedaan antara dua data yang dibandingkan, sehingga kualitas kemiripannya semakin baik (Hacimurtazaoglu & Tutuncu, 2022). Sebaliknya, nilai PSNR yang rendah menunjukkan bahwa perbedaan antara kedua data cukup besar.

Perhitungan PSNR didasarkan pada nilai *Mean Squared Error* (MSE). MSE merupakan nilai rata-rata dari selisih kuadrat antara nilai piksel data asli dengan nilai piksel data hasil pemrosesan. Nilai MSE menunjukkan tingkat kesalahan secara keseluruhan. Semakin kecil nilai MSE, maka semakin kecil pula perbedaan antara dua data yang dibandingkan

Secara matematis, nilai MSE dirumuskan pada persamaan (2.2) berikut.

$$MSE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{i=1}^{N} [I(i,j) - K(i,j)]^2 \qquad (2.2)$$

Keterangan Simbol MSE:

MSE : *Mean Squared Error*

I(i,j) : Nilai piksel data asli pada koordinat ke-(i,j)

K(i,j) : Nilai piksel data hasil pemrosesan pada koordinat ke-(i,j)

M : Jumlah baris piksel

N : Jumlah kolom piksel

i, j : Indeks posisi piksel

Nilai PSNR kemudian dihitung berdasarkan nilai MSE dengan persamaan (2.3) sebagai berikut.

$$PSNR = 10 \log_{10} \left(\frac{MAX^2}{MSE} \right) \qquad (2.3)$$

Keterangan Simbol PSNR:

PSNR : *Peak Signal-to-Noise Ratio* (dalam dB)

MAX : Nilai maksimum intensitas piksel

MSE : *Mean Squared Error*

Nilai MAX bergantung pada kedalaman bit citra yang digunakan. Sebagai contoh, untuk citra 8-bit nilai maksimum intensitas piksel adalah 255, sedangkan untuk citra 10-bit bernilai 1023. Hubungan antara MSE dan PSNR bersifat berbanding terbalik, yaitu semakin kecil nilai MSE, maka semakin besar nilai PSNR, yang menunjukkan bahwa kualitas kemiripan data semakin baik.

BAB III
METODE PENELITIAN

3.1 Jenis dan Pendekatan Penelitian

Penelitian ini termasuk dalam jenis penelitian eksperimen terapan (*applied experimental research*), karena penelitian dilakukan dengan cara menerapkan langsung metode steganografi dan kriptografi pada media video, kemudian menguji hasil penerapannya berdasarkan parameter tertentu.

Pendekatan yang digunakan dalam penelitian ini adalah pendekatan kuantitatif, karena hasil penelitian dianalisis menggunakan nilai numerik berupa nilai *Peak Signal to Noise Ratio* (PSNR) serta tingkat keberhasilan proses enkripsi, penyisipan, ekstraksi, dan dekripsi pesan.

3.2 Tahapan Pengembangan Sistem

Penelitian ini dilaksanakan melalui beberapa tahapan yang terstruktur, yaitu:

1. Studi Literatur, dilakukan untuk mengumpulkan berbagai referensi, teori, metode, serta hasil penelitian terdahulu yang relevan dengan topik steganografi dan kriptografi.
2. Analisis Kebutuhan, mencakup identifikasi kebutuhan sistem yang akan dibangun, meliputi input, proses, dan output.
3. Pengumpulan data uji, dilakukan dengan menyiapkan sejumlah video yang akan digunakan sebagai sampel dalam proses penyisipan dan ekstraksi.
4. Perancangan Sistem, yaitu penyusunan rancangan sistem secara menyeluruh agar implementasi dapat dilakukan sesuai dengan kebutuhan yang telah dianalisis.
5. Implementasi Sistem, merupakan proses penerjemahan rancangan ke dalam bentuk kode program menggunakan bahasa pemrograman Python.
6. Pengujian dan Analisis Hasil, dilakukan untuk memastikan sistem berjalan sesuai harapan dan dapat digunakan untuk menyisipkan serta mengekstrak data dengan aman.

3.3 Studi Literatur

Tahap ini bertujuan untuk memperoleh landasan teori yang kuat sebagai dasar dalam pengembangan sistem. Metode yang digunakan adalah metode kepustakaan, yaitu dengan menelaah berbagai sumber informasi yang relevan, seperti buku, jurnal ilmiah, prosiding, artikel daring, serta penelitian terdahulu yang membahas topik terkait steganografi, kriptografi, algoritma LSB, dan AES.

Melalui studi literatur, peneliti dapat memahami prinsip kerja dari algoritma yang digunakan, meninjau kelebihan dan kelemahannya, serta menentukan pendekatan terbaik dalam menggabungkan kedua metode tersebut untuk meningkatkan keamanan data pada media video.

3.4 Analisis Kebutuhan

Tahap ini dilakukan untuk menentukan kebutuhan sistem sebelum proses perancangan dan implementasi. Analisis kebutuhan membantu memastikan bahwa sistem yang dikembangkan sesuai dengan tujuan dan mampu berfungsi dengan baik.

3.4.1 Analisis Kebutuhan Masukan (Input)

Agar sistem dapat memproses penyisipan dan pengamanan data dengan benar, diperlukan tiga jenis masukan utama dengan spesifikasi sebagai berikut:

1. Media Penampung (*Cover Video*)

   Berupa berkas video digital yang akan digunakan sebagai wadah penyisipan pesan. Sistem dirancang untuk menerima format video .mp4 yang terdiri dari rangkaian frame gambar. Setiap frame harus diekstraksi menjadi komponen warna RGB (*Red, Green, Blue*) agar manipulasi bit dapat dilakukan pada setiap kanal warna.
2. Pesan Rahasia

   Data informasi berupa teks (*string*) yang ingin disembunyikan. Sebelum diproses, pesan ini akan dikonversi menjadi format biner (*byte stream*). Panjang pesan dibatasi oleh kapasitas maksimum yang tersedia pada *cover video* setelah dikurangi oleh header keamanan (*Salt* dan IV).
3. Kunci Sandi

   Berupa teks bebas yang dimasukkan oleh pengguna. Kunci sandi ini bukan kunci enkripsi final, melainkan bahan dasar yang akan diolah oleh sistem menjadi kunci kriptografis yang aman. Panjang dan kompleksitas kunci diserahkan kepada pengguna, namun sistem akan memprosesnya menjadi kunci tetap berukuran 256-bit.

3.4.2 Analisis Kebutuhan Output

Keluaran dari sistem yang dikembangkan adalah:

1. Video Steganografi (*Stego Video*)

   Merupakan berkas video hasil penyisipan yang secara fisik dan format identik dengan video asli (*cover video*). Secara visual, video ini tidak boleh menampakkan distorsi

atau noise yang dapat dilihat oleh mata manusia (*imperceptibility*). Struktur data internal video telah mengandung muatan data terenkripsi pada bit-bit tidak signifikan.

2. Pesan Asli (*Decrypted Message*)

   Pada proses ekstraksi yang berhasil, sistem akan menampilkan kembali pesan teks asli yang sama persis dengan pesan yang dimasukkan di awal.

3. Indikator Status/Error

   Jika proses dekripsi gagal misalnya akibat kesalahan memasukkan kunci atau kerusakan data pada video, sistem harus mampu memberikan keluaran berupa pesan penolakan akses. Hal ini penting untuk memastikan bahwa informasi sampah (*garbage output*) tidak dianggap sebagai pesan rahasia.

4. Nilai PSNR yang menunjukkan tingkat kualitas visual video setelah penyisipan.

3.4.3 Spesifikasi Kebutuhan Keamanan

Modul keamanan dirancang untuk menjamin aspek kerahasiaan (confidentiality) data. Sistem yang akan dibuat menerapkan spesifikasi kriptografi sebagai berikut:

1. Algoritma Enkripsi

   Algoritma enkripsi yang akan digunakan ialah AES (*Advanced Encryption Standard*) dengan panjang kunci 256-bit. Spesifikasi ini dipilih karena ketahanannya yang tinggi terhadap serangan *brute-force* dibandingkan varian 128-bit. Proses enkripsi diterapkan menggunakan mode operasi CBC (*Cipher Block Chaining*), di mana setiap blok *plaintext* akan di-XOR dengan blok *ciphertext* sebelumnya sebelum dienkripsi. Penggunaan mode CBC bertujuan untuk menghilangkan pola statistik pada *ciphertext*, sehingga pesan dengan karakter berulang tidak menghasilkan pola bit yang berulang pada bit LSB video.

2. *Initialization Vector* (IV) / *Nonce*

   Sistem akan membentuk IV acak sebesar 16-byte (128-bit) setiap kali proses enkripsi dilakukan. IV ini tidak bersifat rahasia tetapi harus unik (nonce), sehingga IV akan disisipkan bersamaan dengan *ciphertext* atau digabungkan di awal data agar dapat digunakan kembali saat proses dekripsi.

3. *Padding*

   Metode *padding* yang diterapkan menggunakan standar PKCS#7, yang digunakan untuk menyesuaikan panjang data agar sesuai dengan ukuran blok 16 byte, sehingga proses dekripsi dan penghapusan *padding* dapat memenuhi kelipatan 16 byte, memastikan proses dekripsi dan penghapusan berjalan akurat tanpa merusak data asli.

Penerapan mode *Cipher Block Chaining* (CBC) dipilih karena keunggulannya dalam memutus pola hubungan antara pesan asli dan hasil enkripsinya. Berbeda dengan mode operasi dasar yang seringkali masih menyisakan jejak pola visual, CBC menggunakan mekanisme perantaian di mana enkripsi setiap blok data akan sangat bergantung pada blok sebelumnya. Ditambah dengan penggunaan *Initialization Vector* (IV), pesan yang sama persis sekalipun akan menghasilkan kode enkripsi yang total berbeda setiap kali diproses. Hasil data yang menyerupai noise acak ini sangat menguntungkan bagi teknik steganografi, karena keberadaannya di dalam video menjadi tersamar dan jauh lebih sulit dicurigai oleh analisis keamanan (*steganalysis*) dibandingkan jika menggunakan mode enkripsi biasa.

3.5 Pengumpulanm Data Uji

Pemilihan data uji dalam penelitian ini menggunakan pendekatan *representative sampling*, yaitu teknik pengambilan sampel yang mewakili kondisi keseluruhan sistem tanpa harus menguji seluruh kemungkinan secara lengkap. Video uji yang digunakan memiliki variasi resolusi, yaitu 144p, 360p, dan 720p, yang masing-masing mewakili kondisi resolusi rendah, sedang, dan tinggi. Variasi resolusi ini digunakan untuk mengetahui pengaruh perbedaan jumlah piksel terhadap keberhasilan proses penyisipan pesan serta kualitas video hasil steganografi. Spesifikasi umum video uji adalah sebagai berikut:

1. Format file : MP4
2. Frame rate : 30 fps
3. Durasi video : 60 detik

3.6 Perancangan Sistem

Tahap perancangan dilakukan untuk menggambarkan alur kerja dan struktur sistem sebelum diimplementasikan. Rancangan sistem bertujuan agar setiap komponen memiliki fungsi yang jelas dan terintegrasi dengan baik

3.6.1 Perancangan Struktur Data *Payload*

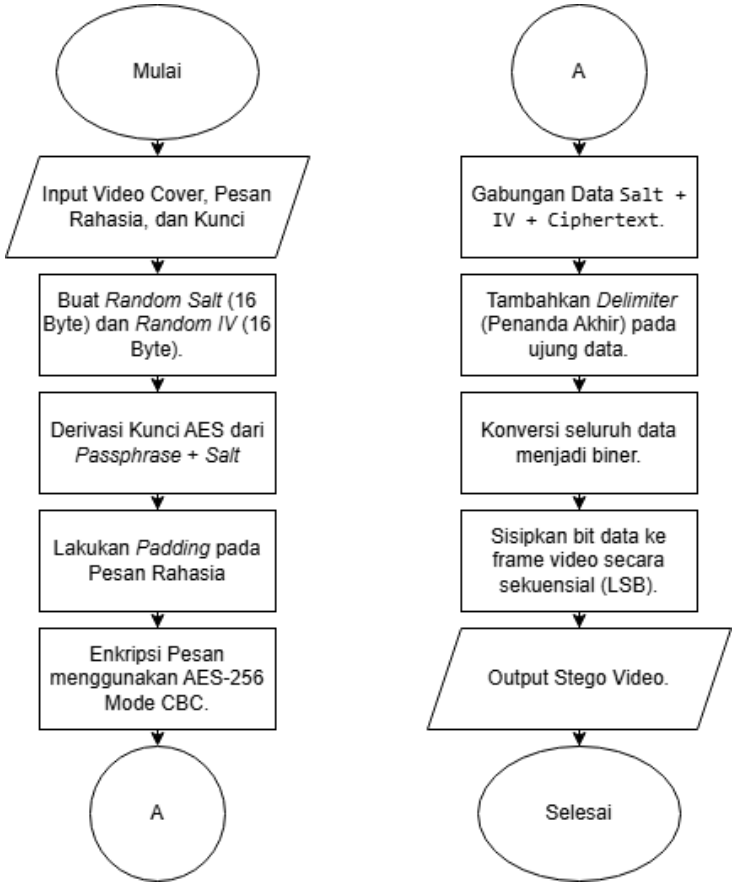
Sebelum data disisipkan ke dalam frame video, pesan rahasia harus melalui serangkaian transformasi agar aman dan memiliki integritas yang terjaga. Sistem ini merancang struktur data *payload* khusus menggunakan teknik penggabungan (*concatenation*) byte *array* secara sekuensial. Struktur paket data ini diawali dengan 16 byte data acak (*Salt*) yang berfungsi sebagai komponen pembentuk kunci, diikuti oleh 16 byte *Initialization Vector* (IV) yang digunakan untuk mode operasi CBC, dan dilanjutkan dengan blok *ciphertext* (pesan

terenkripsi). Sebagai penutup, sistem menambahkan serangkaian karakter unik atau *delimiter* pada bagian akhir paket data. Keberadaan *delimiter* ini sangat vital karena berfungsi sebagai penanda berhenti (*stop marker*) bagi sistem penerima, sehingga proses pembacaan bit LSB tidak perlu dilakukan hingga akhir durasi video jika pesan yang disisipkan berukuran kecil. Visualisasi struktur data gabungan ini dapat dilihat pada Gambar 3.1.

Gambar 3.1 Struktur Data *Payload*.

3.6.2 Perancangan Proses Enkripsi dan Penyisipan

Alur proses pengamanan dan penyisipan pesan dirancang sebagai satu kesatuan prosedur yang berjalan secara linear. Proses ini diawali dengan penerimaan input berupa video asli (*cover video*), pesan teks, dan kunci dari pengguna. Berdasarkan input tersebut, sistem pertama-tama membentuk nilai *Salt* secara acak dan menggunakannya bersama kunci untuk menurunkan kunci AES 256-bit melalui fungsi derivasi kunci (PBKDF2). Selanjutnya, sistem membentuk IV acak dan melakukan proses *padding* PKCS#7 pada pesan teks agar panjang datanya sesuai dengan kelipatan blok enkripsi. Pesan yang telah di-*padding* kemudian dienkripsi menggunakan algoritma AES-256 mode CBC. Hasil enkripsi tersebut kemudian digabungkan dengan *Salt*, IV, dan *delimiter* untuk membentuk satu kesatuan *payload* biner. Tahap terakhir adalah penyisipan bit-bit *payload* tersebut ke dalam bit paling tidak signifikan (*Least Significant Bit*) pada setiap kanal warna frame video secara berurutan, dimulai dari frame pertama hingga seluruh data tertampung. Alur logika proses ini digambarkan secara rinci melalui diagram alir (*flowchart*) pada Gambar 3.2.



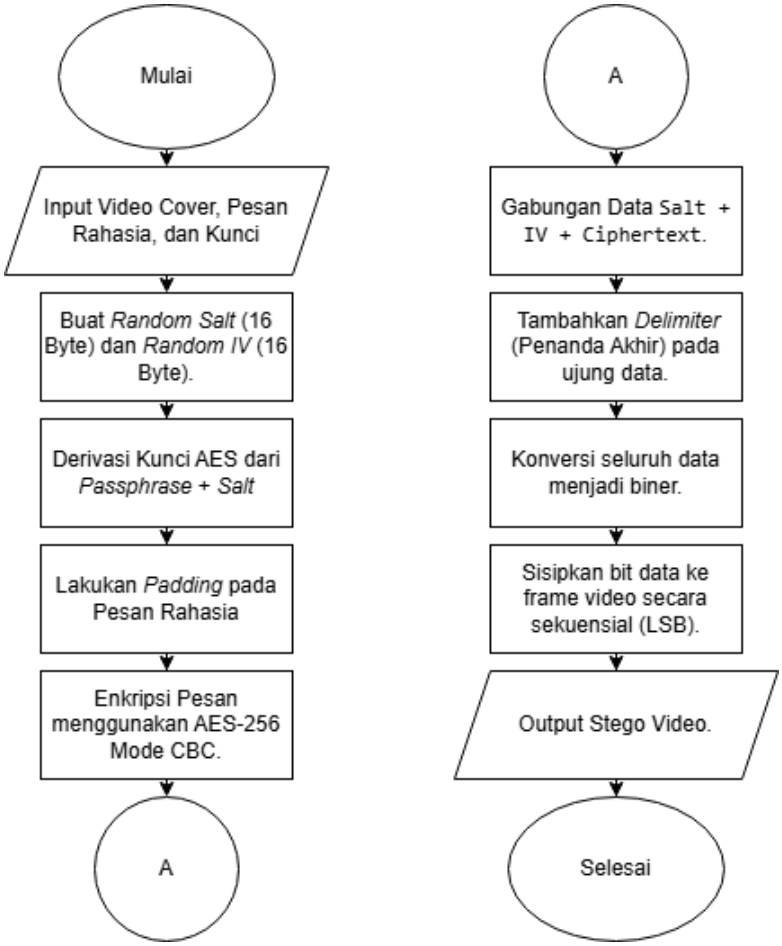
Gambar 3.2 Diagram Alir Proses Enkrips-Penyisipan.

3.6.3 Perancangan Proses Ekstraksi dan Dekripsi

Perancangan proses pemulihan pesan merupakan kebalikan dari proses penyisipan yang menuntut ketelitian tinggi dalam pembacaan data. Mekanisme ini dimulai dengan sistem membaca bit LSB dari setiap frame video secara berurutan dan mengumpulkannya menjadi aliran data biner. Proses pembacaan ini terus berlangsung hingga sistem mendeteksi pola bit yang sesuai dengan *delimiter* yang telah ditentukan. Setelah *delimiter* ditemukan, aliran data yang terkumpul akan dipisahkan berdasarkan struktur yang telah dirancang sebelumnya, yaitu 16 byte awal diidentifikasi sebagai *Salt*, 16 byte berikutnya sebagai IV, dan sisa data lainnya sebagai *ciphertext*.

Setelah data terpisah, sistem merekonstruksi kunci enkripsi dengan menggabungkan input kunci dari pengguna dan *Salt* yang baru saja diekstrak. Menggunakan kunci hasil rekonstruksi dan IV tersebut, sistem mendekripsi *ciphertext* dan melakukan validasi *padding* (PKCS#7 un*padding*). Jika proses *unpadding* berhasil dilakukan tanpa error, maka pesan dianggap valid dan ditampilkan ke pengguna. Sebaliknya, jika terjadi kegagalan pada saat penghapusan *padding*, sistem akan mengindikasikan bahwa kunci yang dimasukkan salah atau data telah

rusak, karena kunci yang tidak tepat akan menghasilkan data acak yang tidak mematuhi aturan format *padding*. Alur logika proses ekstraksi dan dekripsi ini dijelaskan pada diagram alir (*flowchart*) Gambar 3.3.



Gambar 3.3 Diagram Alir Proses Ekstraksi-Dekripsi.

3.7 Implementasi Sistem

Tahap implementasi merupakan proses penerjemahan perancangan sistem dan struktur data yang telah disusun ke dalam bahasa pemrograman yang dapat dieksekusi oleh komputer. Implementasi ini dilakukan dalam lingkungan pengembangan yang spesifik untuk memastikan seluruh pustaka (*library*) kriptografi, steganografi, dan pengolahan citra dapat berjalan optimal.

3.7.1 Lingkungan Pengembangan

Sistem steganografi video ini dikembangkan menggunakan perangkat keras (*hardware*) dan perangkat lunak (*software*) dengan spesifikasi sebagai berikut.

1. Perangkat Keras:

a. Prosesor Setara Intel Core i7
b. Memori (RAM) 8 GB
c. Penyimpanan SSD NVMe INTEL SSDPEKNW512G8

2. Perangkat Lunak:
a. Sistem Operasi Windows 11
b. Bahasa Pemrograman Python versi 3.13.7
c. IDE Visual Studio Code

3.7.2 Pustaka Pendukung (*Libraries*)

Dalam proses pengkodean, sistem memanfaatkan beberapa pustaka eksternal Python yang relevan dengan kebutuhan fungsi sistem, antara lain:

1. PyCryptodome (Crypto)
   Merupakan pustaka kriptografi utama yang digunakan untuk mengimplementasikan standar keamanan. Modul-modul spesifik yang digunakan meliputi:
   a. Crypto.Cipher.AES untuk melakukan proses enkripsi dan dekripsi menggunakan algoritme AES.
   b. Crypto.Protocol.KDF.PBKDF2 untuk menurunkan kunci enkripsi 256-bit dari kunci pengguna secara aman.
   c. Crypto.Random untuk membuat bilangan acak yang kuat secara kriptografis guna pembuatan *Salt* dan *Initialization Vector* (IV).
   d. Crypto.Util.*Padding* untuk menerapkan standar *padding* dan un*padding* (PKCS#7) pada blok data pesan.
2. OpenCV (cv2)
   Open Source Computer Vision Library digunakan sebagai mesin utama pengolahan media video. Pustaka ini berfungsi untuk membaca berkas video input, mengekstraksi video menjadi deretan frame gambar, serta menyusun kembali frame-frame yang telah disisipi pesan menjadi berkas video utuh.
3. Stegano (stegano.lsb)
   Pustaka ini digunakan untuk menangani operasi tingkat rendah (*low-level operation*) pada piksel gambar. Modul lsb pada pustaka ini berfungsi untuk menyisipkan data biner ke dalam *Least Significant Bit* media penampung serta mengekstraksinya kembali. Penggunaan pustaka ini memastikan proses manipulasi bit berjalan efisien pada setiap frame yang diekstraksi oleh OpenCV.

3.8 Skenario Pengujian

Tahap pengujian dirancang untuk memverifikasi apakah sistem yang dibangun telah memenuhi spesifikasi kebutuhan fungsional, keamanan, dan kualitas visual.

3.8.1 Skenario Pengujian Fungsional (*Black Box Testing*)

Pengujian fungsional dilakukan dengan metode *Black Box Testing* untuk memastikan setiap modul perangkat lunak memberikan keluaran yang sesuai dengan masukan yang diberikan. Skenario pengujian meliputi:

1. Uji Validasi Input

   Memastikan sistem menolak file non-video atau passphrase kosong.
2. Uji Proses Enkripsi-Penyisipan

   Memastikan sistem berhasil menghasilkan file video baru (*stego video*) yang dapat diputar oleh pemutar video standar.
3. Uji Proses Ekstraksi-Dekripsi

   Memastikan sistem dapat mengembalikan pesan rahasia secara utuh jika input kunci dan video benar.

3.8.2 Skenario Pengujian Keamanan dan Ketahanan

Pengujian ini dirancang untuk mengevaluasi sistem terhadap serangan dan mendeteksi anomali yang muncul akibat proses steganografi. Skenario pengujian dibagi menjadi empat kategori utama:

1. Pengujian Kerahasiaan

   Skenario ini mensimulasikan upaya akses ilegal oleh pihak ketiga. Pengujian dilakukan dengan mencoba mendekripsi video steganografi menggunakan kunci yang salah. Hal ini bertujuan untuk memastikan mekanisme integritas *padding* (PKCS#7) dan enkripsi AES berfungsi, sehingga sistem menolak mendekripsi dan tidak menampilkan informasi apapun (bahkan sampah informasi) kepada pengguna yang tidak terotorisasi.
2. Analisis Statistik Histogram (*Histogram Analysis*)

   Skenario ini mensimulasikan serangan steganalysis visual. Pengujian dilakukan dengan membandingkan grafik histogram (distribusi frekuensi nilai piksel RGB) antara frame video asli (*cover*) dan frame video hasil steganografi (*stego*). Hal ini bertujuan untuk membuktikan bahwa penyisipan pesan menggunakan metode LSB

tidak mengubah distribusi warna secara signifikan. Perubahan histogram yang minimal mengindikasikan bahwa pesan sulit dideteksi secara visual maupun statistik.

3. Analisis Ukuran Berkas (*File Size Analysis*)

   Skenario ini dilakukan untuk mendeteksi anomali forensik pada berkas hasil. Ukuran berkas video cover dan video stego akan dibandingkan. Hal ini bertujuan untuk memastikan bahwa teknik steganografi yang diterapkan bersifat substitusi (mengganti bit) dan bukan append (menambah bit), sehingga tidak terjadi peningkatan ukuran file yang mencurigakan yang dapat memicu kecurigaan pihak ketiga.

4. Uji Ketahanan Terhadap *Frame Drop Attack*

   Skenario ini mensimulasikan gangguan transmisi atau serangan aktif di mana sebagian frame video hilang atau dihapus secara sengaja. Pengujian dilakukan dengan memotong atau menghapus beberapa frame dari video steganografi, kemudian mencoba melakukan ekstraksi pesan kembali. Hal ini bertujuan untuk menguji batas toleransi sistem terhadap kerusakan media (*robustness*). Mengingat metode LSB bersifat rapuh (*fragile*), pengujian ini bertujuan untuk mengetahui apakah pesan masih dapat diselamatkan jika frame yang hilang berada di luar area penyisipan pesan, atau mendeteksi kegagalan sistem jika frame yang hilang memuat *payload* data.

3.8.3 Skenario Pengujian Kualitas Video (PSNR)

Pengujian kualitas visual bertujuan untuk mengukur tingkat degradasi atau penurunan kualitas citra pada video steganografi dibandingkan dengan video asli menggunakan parameter Peak Signal-to-Noise Ratio (PSNR). Sesuai dengan landasan teori yang telah dipaparkan pada Bab II, perhitungan nilai PSNR didasarkan pada nilai *Mean Square Error* (MSE) antar piksel yang akan dilakukan secara otomatis oleh sistem pada setiap pasang frame video asli dan video steganografi.

Hasil pengukuran nilai PSNR tersebut kemudian dianalisis berdasarkan standar pengolahan citra digital untuk menentukan kelayakan metode yang diterapkan. Kualitas penyisipan dikategorikan sangat baik apabila nilai PSNR berada di atas 40 dB karena perbedaan visual sangat sulit dilihat oleh mata manusia. Rentang nilai antara 30 hingga 40 dB dikategorikan baik meskipun terdapat distorsi yang mungkin terlihat namun masih dalam batas wajar, sedangkan nilai di bawah 30 dB dianggap memiliki kualitas buruk karena kerusakan visual terlihat jelas. Berdasarkan standar tersebut, penelitian ini menetapkan target keberhasilan berupa pencapaian nilai rata-rata PSNR di atas 40 dB yang mengindikasikan bahwa video steganografi memiliki tingkat ketahanan visual atau *imperceptibility* yang tinggi.

BAB IV
HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

Implementasi sistem merupakan tahapan realisasi dari rancangan perangkat lunak yang telah didefinisikan pada bab sebelumnya ke dalam bentuk kode program yang dapat dieksekusi. Pada tahap ini, spesifikasi kebutuhan fungsional sistem, struktur data, serta alur logika algoritma kriptografi *Advanced Encryption Standard* (AES) dan metode steganografi *Least Significant Bit* (LSB) diterjemahkan menggunakan bahasa pemrograman Python.

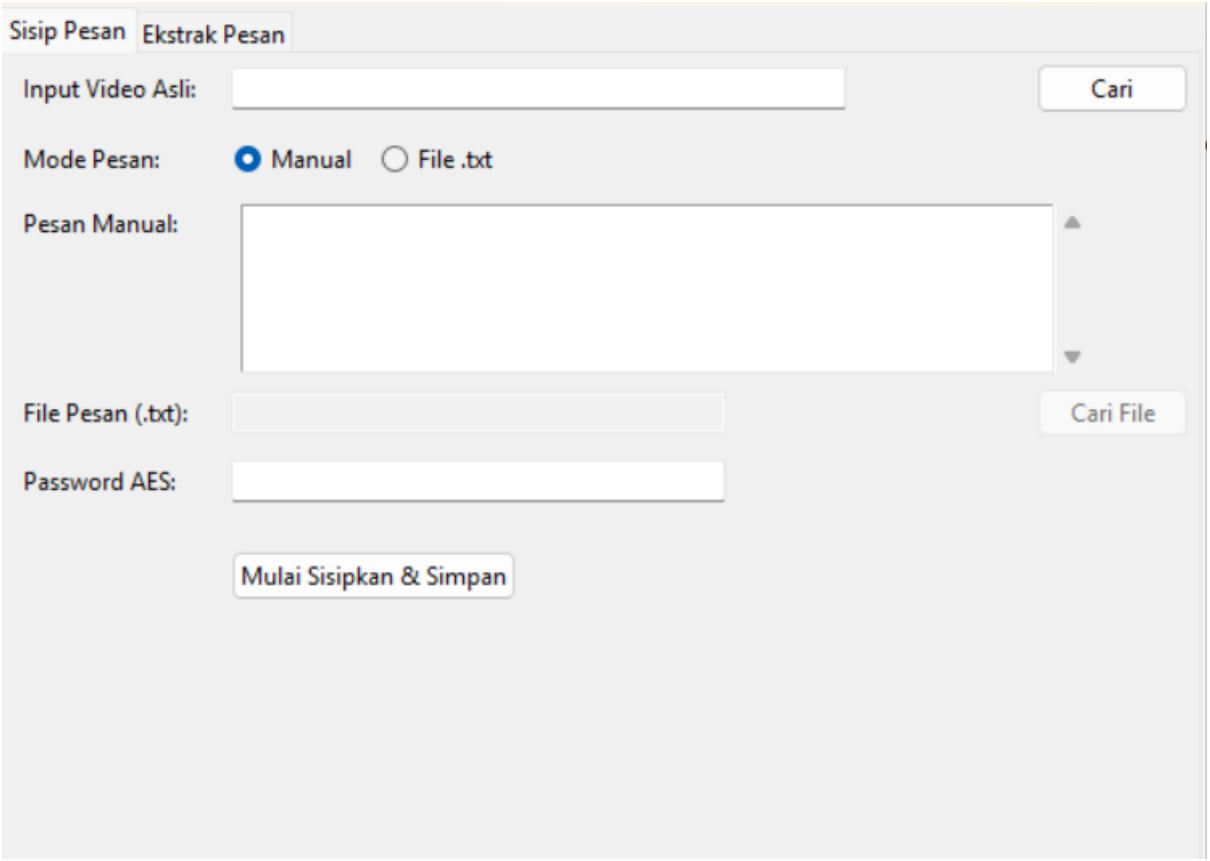
4.1.1 Implementasi Antarmuka Pengguna (*User Interface*)

Antarmuka pengguna (*User Interface*) dikembangkan untuk memberikan kemudahan akses bagi pengguna dalam mengoperasikan fitur-fitur kriptografi dan steganografi tanpa perlu berinteraksi langsung dengan kode sumber. Antarmuka ini terdiri dari dua menu utama, yaitu menu penyisipan pesan (*Embedding Interface*) dan menu ekstraksi pesan (*Extraction Interface*).

1. Halaman Menu Penyisipan Pesan

   Halaman ini berfungsi sebagai gerbang utama untuk memproses video *cover* dan menyisipkan pesan rahasia. Seperti yang ditunjukkan pada Gambar 4.1, desain antarmuka disusun secara sistematis dengan komponen-komponen input sebagai berikut:

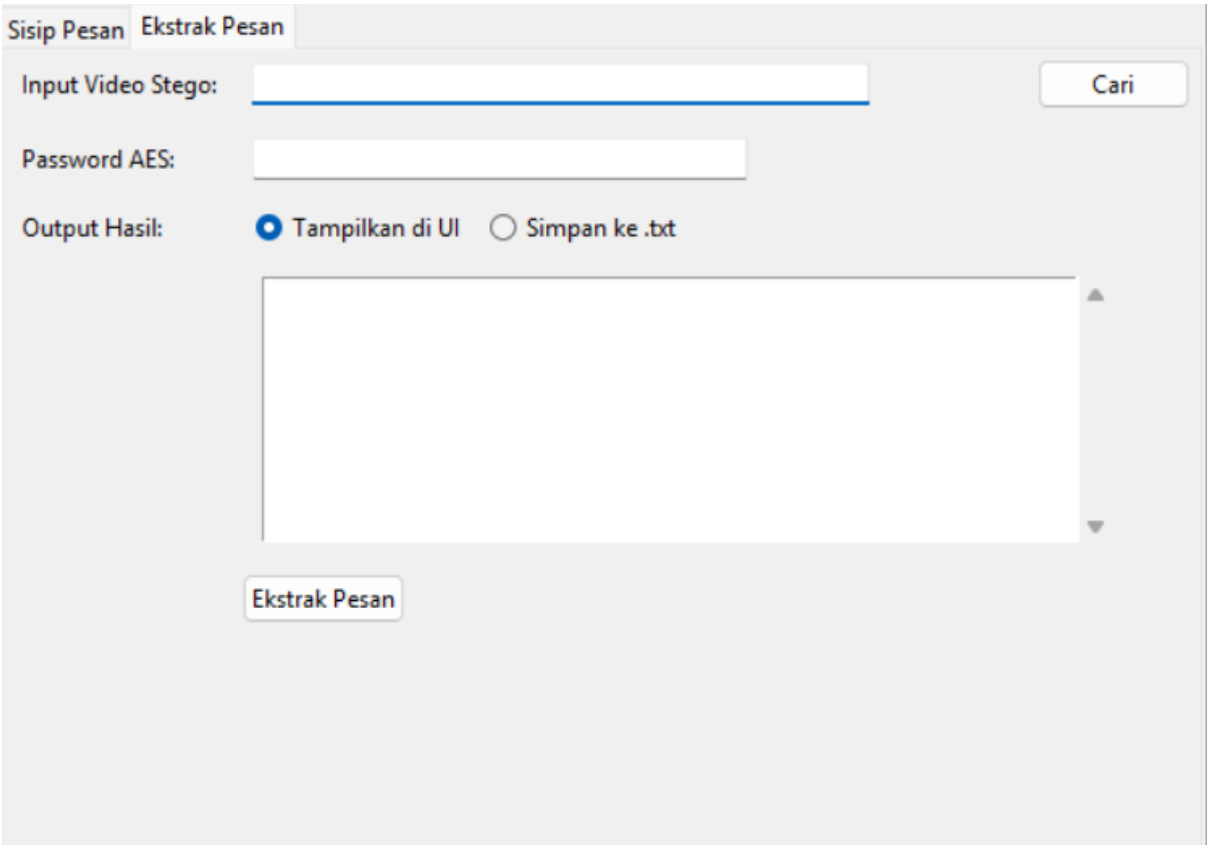
a. Pemilihan Berkas Video (*Video Selection*): Tombol pencarian berkas (*browse*) memungkinkan pengguna menelusuri direktori penyimpanan lokal untuk memilih video sumber (format .mp4 atau .avi). Sistem akan menampilkan path atau lokasi berkas yang dipilih pada kolom indikator.
b. Area Input Pesan (*Message Input*): Sebuah kolom teks area (*text field*) disediakan bagi pengguna untuk mengetikkan pesan rahasia yang akan disisipkan.
c. Konfigurasi Kunci (*Security Key*): Kolom input password mewajibkan pengguna memasukkan frasa sandi (*passphrase*). Input ini bersifat krusial karena akan digunakan oleh modul AES untuk membangkitkan kunci enkripsi yang unik.
d. Tombol Eksekusi: Tombol proses berfungsi untuk memicu validasi input. Jika seluruh data telah terisi, sistem akan menjalankan proses enkripsi dan penyisipan di latar belakang.



Gambar 4.1 Antarmuka Menu Sisip Pesan.

2. Halaman Menu Ekstraksi Pesan Halaman ekstraksi dirancang dengan tampilan yang lebih minimalis karena hanya membutuhkan parameter kunci untuk memulihkan pesan. Tampilan halaman ini dapat dilihat pada Gambar 4.2. Komponen utamanya meliputi:

a. Input Video Stego: Pengguna memilih berkas video yang diduga mengandung pesan rahasia.
b. Input Kunci Dekripsi: Kolom ini digunakan untuk memasukkan password. Sistem akan menggunakan password ini untuk merekonstruksi kunci AES. Jika password tidak cocok dengan yang digunakan saat penyisipan, proses dekripsi akan gagal.
c. Area Hasil Pesan (*Output Display*): Berbeda dengan halaman penyisipan, area teks pada halaman ini bersifat *read-only* (hanya baca) dan berfungsi untuk menampilkan hasil pesan teks yang berhasil diekstraksi dan didekripsi dari video.



Gambar 4.2 Antarmuka Menu Ekstrak Pesan.

4.1.2 Implementasi Modul Enkripsi dan Penyisipan

Implementasi modul ini bertanggung jawab atas dua fungsi keamanan utama: mengamankan kerahasiaan pesan melalui kriptografi *Advanced Encryption Standard* (AES) dan menyembunyikan keberadaan pesan tersebut menggunakan teknik steganografi *Least Significant Bit* (LSB).

A. Implementasi Algoritma Enkripsi AES

Realisasi keamanan data diawali dengan pembentukan kunci enkripsi yang dinamis pada kelas AESCipher. Sistem tidak menggunakan *passphrase* pengguna secara mentah sebagai kunci, melainkan memprosesnya terlebih dahulu melalui fungsi derivasi kunci PBKDF2 (*Password-Based Key Derivation Function 2*). Fungsi ini melakukan iterasi *hashing* sebanyak 1.000.000 kali dengan tambahan *Salt* acak, yang bertujuan untuk memperlambat serangan komputasi (*brute-force attack*) terhadap kata sandi. Selanjutnya, proses enkripsi dilakukan menggunakan mode operasi CBC (*Cipher Block Chaining*) yang membutuhkan *Initialization Vector* (IV) unik setiap kali proses berjalan. Pesan teks yang masuk akan melalui proses

padding agar panjang bitnya sesuai dengan kelipatan blok AES sebelum diubah menjadi *ciphertext*.

```
def encrypt(self, data: bytes) -> bytes:
        salt = get_random_bytes(16)
        iv = get_random_bytes(16)
        key = self._derive_key(salt)
        cipher = AES.new(key, AES.MODE_CBC, iv)
        ct = cipher.encrypt(pad(data, AES.block_size))
        return salt + iv + ct
```

Gambar 4.3 Kode Program Algoritma Enkripsi AES.

Berdasarkan kode di atas, hasil akhir dari fungsi enkripsi bukanlah sekadar teks tersandi, melainkan paket data biner yang terdiri dari penggabungan tiga komponen vital:

3. *Salt* (16 byte): Komponen pengacak kunci.
4. IV (16 byte): Komponen pengacak blok awal.
5. *Ciphertext*: Pesan yang telah terenkripsi.

Ketiga komponen ini digabungkan secara sekuensial (*salt* + iv + ct) untuk memastikan bahwa pada saat dekripsi, sistem penerima memiliki seluruh parameter yang dibutuhkan untuk merekonstruksi kunci. Sebagai langkah penyesuaian teknis agar data biner ini dapat disisipkan ke dalam format teks steganografi tanpa error, hasil penggabungan tersebut kemudian dikonversi ke format Base64 sebelum diteruskan ke modul penyisipan.

B. Implementasi Metode Penyisipan LSB pada Video

Setelah pesan diamankan melalui enkripsi, proses beralih ke manipulasi media video yang diatur oleh fungsi encode. Mengingat video merupakan kumpulan gambar bergerak, sistem memanfaatkan pustaka OpenCV untuk mengekstraksi fail video menjadi deretan frame gambar statis berformat PNG. Format PNG dipilih karena sifat kompresinya yang *lossless*, sehingga bit pesan yang disisipkan tidak akan rusak akibat kompresi gambar.

Sebelum pesan disisipkan, sistem memastikan data telah dibungkus dengan penanda khusus (*delimiter*) berupa <<<ENDMSG>>> di akhir. Penanda ini berfungsi sebagai rambu bagi sistem ekstraksi nanti untuk mengetahui batas valid data. Agar modifikasi piksel tersebar merata dan tidak menumpuk pada satu titik yang mencurigakan, paket pesan lengkap ini dipecah (*split*) menjadi potongan-potongan kecil sesuai dengan jumlah total frame yang tersedia.

```
Def embedded(self, video_path: str, message: str, output_video="Embedded_Video.avi"):
        cap = cv2.VideoCapture(video_path)
        fps = cap.get(cv2.CAP_PROP_FPS)
        cap.release()

        total_frames = self.extract_frames(video_path)
        split_msg = self._split_message(message, total_frames)

        for i, chunk in enumerate(split_msg[:total_frames]):
            frame_path = os.path.join(self.temp_dir, f"{i}.png")
            secret = lsb.hide(frame_path, chunk)
            secret.save(frame_path)

        # Tambahkan penanda akhir
        last_frame_path = os.path.join(self.temp_dir, f"{min(len(split_msg), total_frames)-
1}.png")
        secret = lsb.hide(last_frame_path, split_msg[-1] + "<ENDMSG>")
        secret.save(last_frame_path)

        # Audio & Rebuild
        call(["ffmpeg",     "-i",     video_path,     "-q:a",     "0",     "-map",     "a",
f"{self.temp_dir}/audio.mp3", "-y"], stdout=open(os.devnull, "w"), stderr=STDOUT)
        call(["ffmpeg",  "-framerate",  str(fps),  "-i",  f"{self.temp_dir}/%d.png",  "-c:v",
"ffv1",    f"{self.temp_dir}/temp_video.avi",    "-y"],    stdout=open(os.devnull,    "w"),
stderr=STDOUT)
        call(["ffmpeg",         "-i",         f"{self.temp_dir}/temp_video.avi",         "-i",
f"{self.temp_dir}/audio.mp3", "-codec", "copy", output_video, "-y"], stdout=open(os.devnull,
"w"), stderr=STDOUT)

        self.cleanup()
```

Gambar 4.4 Kode Program Algoritma Penyisipan LSB.

Inti dari proses steganografi terjadi saat fungsi melakukan iterasi pada setiap frame. Kode program memanggil fungsi lsb.hide untuk menyisipkan potongan pesan ke dalam bit paling tidak signifikan (*Least Significant Bit*) pada kanal warna piksel. Setiap frame yang telah dimodifikasi kemudian disimpan ulang ke dalam direktori sementara. Setelah seluruh potongan pesan berhasil disisipkan, kumpulan frame tersebut disusun kembali (*rebuild*) menjadi satu fail video utuh berformat .avi menggunakan utilitas FFmpeg. Penggunaan FFmpeg ini krusial untuk menyatukan kembali visual dan audio tanpa merusak struktur bit data yang baru saja disisipkan.

4.1.3 Implementasi Modul Ekstraksi dan Dekripsi

Kebalikan dari proses penyisipan, modul ini bertujuan untuk memulihkan kembali pesan asli dari video steganografi. Proses ini melibatkan dua tahapan kritis: ekstraksi data

tersembunyi dari frame video dan dekripsi *ciphertext* menjadi teks yang dapat dibaca (*plaintext*).

A. Implementasi Ekstraksi Pesan dari Video

Proses pemulihan data dimulai dengan memecah video steganografi menjadi kumpulan frame gambar. Fungsi ekstraksi (*decode*) bekerja dengan menelusuri setiap frame secara berurutan untuk membaca bit LSB yang tersembunyi. Mengingat proses pembacaan ribuan frame membutuhkan waktu komputasi yang signifikan, sistem menerapkan mekanisme validasi awal *(Early Exit*). Sistem akan memeriksa frame pertama terlebih dahulu, jika tidak ditemukan data atau pola yang valid, proses akan langsung dihentikan. Hal ini mencegah sistem membuang waktu memproses video kosong atau video yang bukan merupakan hasil steganografi.

```
def extract(self, video_path: str) -> str:
        total_frames = self.extract_frames(video_path)
        if total_frames == 0:
            self.cleanup()
            return ""

        # --- Early Exit jika frame pertama kosong ---
        first_frame = os.path.join(self.temp_dir, "0.png")
        try:
            first_check = lsb.reveal(first_frame)
            if not first_check:
                self.cleanup()
                return ""
        except:
            self.cleanup()
            return ""

        secret_msg = []
        for i in range(total_frames):
            frame_path = os.path.join(self.temp_dir, f"{i}.png")
            try:
                msg = lsb.reveal(frame_path)
                if msg:
                    secret_msg.append(msg)
                    if "<ENDMSG>" in msg: break
                else: break
            except: continue
```

```
        message = ''.join(secret_msg).replace("<ENDMSG>", "")
        self.cleanup()
        return message
```

Gambar 4.5 Kode Program Algoritma Ekstraksi.

Setelah data dari setiap frame berhasil dikumpulkan dan digabungkan kembali menjadi satu untaian *string*, sistem melakukan validasi integritas menggunakan penanda (*marker*) yang telah disisipkan sebelumnya. Sistem akan mencari penanda akhir <<<ENDMSG>>>. Data yang diambil hanyalah karakter yang berada di sebelum penanda tersebut. Mekanisme ini memastikan bahwa karakter sampah (*noise*) yang mungkin terbaca dari frame kosong setelah pesan berakhir akan dibuang secara otomatis, sehingga yang tersisa hanyalah *payload* data terenkripsi yang bersih.

B. Implementasi Algoritma Dekripsi dan Verifikasi Kunci

Data bersih hasil ekstraksi masih berada dalam format Base64. Sebelum didekripsi, data tersebut dikembalikan (*decode*) ke format byte array asli. Fungsi dekripsi kemudian mengambil peran vital untuk memisahkan komponen-komponen kriptografi yang tergabung di dalamnya melalui teknik pemotongan data (*slicing*).

```
def decrypt(self, enc_data: bytes) -> bytes:
        salt = enc_data[:16]
        iv = enc_data[16:32]
        key = self._derive_key(salt)
        cipher = AES.new(key, AES.MODE_CBC, iv)
        ct = enc_data[32:]
        pt = unpad(cipher.decrypt(ct), AES.block_size)
        return pt
```

Gambar 4.6 Kode Program Algoritma Dekripsi.

Berdasarkan logika kode di atas, 16 byte pertama dari data diambil sebagai *Salt*, dan 16 byte berikutnya diambil sebagai Initialization Vector (IV). Sisa data setelah byte ke-32 dianggap sebagai *Ciphertext*. *Salt* yang diekstrak ini digunakan bersama dengan password masukan pengguna untuk merekonstruksi kunci enkripsi yang sama persis dengan kunci saat penyisipan. Proses dekripsi dilakukan menggunakan algoritma AES mode CBC. Validasi keberhasilan dekripsi bergantung pada fungsi unpad. Jika *padding* data sesuai dengan standar PKCS#7, pesan asli akan dikembalikan. Sebaliknya, jika kunci salah, struktur data akan kacau

dan unpad akan menghasilkan error, yang oleh sistem diterjemahkan sebagai kegagalan autentikasi (kunci salah).

4.2 Pengujian Fungsional

Pengujian fungsional dilaksanakan dengan mengacu secara ketat pada skenario pengujian Black Box yang telah dirumuskan pada Bab 3. Tujuan utama dari rangkaian pengujian ini adalah memverifikasi kesesuaian antara masukan (input) yang diberikan pengguna dengan keluaran (output) yang dihasilkan oleh sistem, serta memastikan bahwa mekanisme penanganan kesalahan (*error handling*) berjalan efektif.

Pengujian dibagi menjadi tiga kategori skenario utama, yaitu uji validasi input, uji proses enkripsi-penyisipan, dan uji proses ekstraksi-dekripsi. Rangkuman hasil pelaksanaan pengujian tersebut disajikan secara rinci dalam Tabel 4.1.

Tabel 4.1 Hasil Pengujian Fungsional.

No	Kategori Skenario	Detail Langkah Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Uji Validasi Input	Mengklik tombol "Mulai" dengan membiarkan kolom *path video* atau password kosong.	Sistem menolak memproses data dan menampilkan pesan peringatan (error).	Muncul notifikasi *pop-up*: "Harap isi semua kolom". Proses berhenti.	Valid
2	Uji Validasi Input	Mencoba mengunggah berkas dengan format non-video (contoh: .docx, .jpg) pada menu input.	Sistem membatasi seleksi berkas hanya pada format video yang didukung (.mp4, .avi).	File explorer hanya menampilkan berkas video, berkas lain tidak dapat dipilih.	Valid
3	Uji Proses Enkripsi-Penyisipan	Memasukkan video MP4, pesan teks, dan password, lalu	Terbentuk berkas video baru (*stego video*) dengan	Berkas Embedded_Video.avi berhasil dibuat.	Valid

		menjalankan proses penyisipan.	format AVI di direktori tujuan.	Ukuran file sedikit bertambah dari asli.	
4	Uji Proses Enkripsi-Penyisipan	Memutar video hasil steganografi (stego video) pada pemutar video standar (VLC/Media Player).	Video dapat diputar dengan lancar, durasi sesuai asli, dan tidak terjadi crash.	Video berjalan normal. Visual dan audio sinkron selayaknya video biasa.	Valid
5	Uji Proses Ekstraksi-Dekripsi	Melakukan ekstraksi pada video stego menggunakan password yang benar.	Sistem berhasil mendekripsi pesan dan menampilkan teks asli secara utuh.	Pesan rahasia muncul pada kolom output (Isi pesan identik dengan input).	Valid
6	Uji Proses Ekstraksi-Dekripsi	Melakukan ekstraksi pada video stego menggunakan password yang salah.	Sistem menolak mendekripsi pesan dan memberikan indikasi kegagalan.	Muncul pesan kesalahan (error) atau hasil kosong. Pesan asli tetap tersembunyi.	Valid

Berdasarkan data yang tercantum pada Tabel 4.1, dapat dilakukan analisis terhadap kinerja sistem pada setiap skenario pengujian yang telah dilakukan. Pada skenario pengujian nomor 1 dan 2, sistem menunjukkan respons yang tepat dalam menangani input yang tidak valid. Pembatasan format berkas pada dialog pemilihan file berfungsi sebagai langkah preventif untuk mencegah pengguna memasukkan format data yang tidak kompatibel dengan pemrosesan OpenCV. Selain itu, mekanisme validasi terhadap kolom input yang kosong memastikan bahwa proses kriptografi tidak dijalankan dengan parameter bernilai *null* yang berpotensi menyebabkan kesalahan eksekusi (*runtime error*). Hal ini menunjukkan bahwa sistem telah dilengkapi dengan mekanisme pra-pemrosesan yang andal sebelum memasuki tahap komputasi utama.

Selanjutnya, hasil pengujian pada skenario nomor 3 dan 4 membuktikan keberhasilan integrasi antara modul kriptografi AES dan steganografi berbasis *Least Significant Bit* (LSB). Video steganografi yang dihasilkan dapat diputar secara normal menggunakan pemutar media standar seperti VLC Player, yang menandakan bahwa proses penyisipan pesan pada bit LSB tidak merusak struktur *header* maupun frame video secara signifikan. Meskipun terjadi perubahan pada data piksel secara teknis, video hasil penyisipan tetap mempertahankan fungsinya sebagai media multimedia. Temuan ini sejalan dengan prinsip utama steganografi, yaitu *imperceptibility*, di mana keberadaan pesan tersembunyi tidak menimbulkan gangguan terhadap fungsi maupun kualitas visual media penampung.

Pada skenario pengujian nomor 5 dan 6, aspek keamanan dan autentikasi pesan diuji secara menyeluruh. Ketika kunci yang dimasukkan benar, sistem mampu menjalankan seluruh rangkaian proses secara berurutan, mulai dari ekstraksi bit LSB, pendeteksian penanda, konversi data dari format Base64, hingga proses dekripsi menggunakan algoritma AES, sehingga pesan asli dapat dikembalikan tanpa mengalami kerusakan karakter. Sebaliknya, ketika kunci yang dimasukkan tidak sesuai, sistem gagal menampilkan pesan yang diekstraksi. Kondisi ini bukan merupakan kesalahan sistem, melainkan mekanisme keamanan yang disengaja. Hal tersebut disebabkan oleh sifat sensitivitas tinggi algoritma AES (*avalanche effect*), di mana perbedaan satu karakter pada kunci akan menghasilkan kunci turunan yang sepenuhnya berbeda, sehingga proses dekripsi dan *unpadding* data tidak dapat dilakukan. Dengan demikian, sistem telah memenuhi aspek kerahasiaan (*confidentiality*), di mana hanya pengguna dengan kunci yang valid yang dapat mengakses pesan rahasia yang disisipkan dalam video.

4.2.1 Pengujian Enkripsi-Penyisipan Pesan

Pada tahap pengujian enkripsi, sistem diuji dengan memasukkan pesan teks menggunakan dua metode, yaitu input manual melalui antarmuka aplikasi dan input melalui berkas .txt. Pengguna juga memasukkan kunci AES sebagai parameter wajib untuk proses enkripsi. Hasil pengujian menunjukkan bahwa pesan berhasil dikonversi menjadi *ciphertext* secara konsisten, dan *ciphertext* yang dihasilkan bersifat acak tanpa pola yang dapat dikenali. Setiap percobaan enkripsi memberikan output yang berbeda ketika kunci berubah, sehingga ini menunjukkan bahwa implementasi AES berjalan sesuai prinsip keamanan yang diharapkan.

Pengujian penyisipan dilakukan dengan memasukkan *ciphertext* yang telah dihasilkan ke dalam video menggunakan metode LSB. Sistem diuji pada tiga resolusi video, yaitu 144p, 360p, dan 720p, untuk memastikan penyisipan dapat berjalan pada berbagai tingkat kepadatan

piksel. Hasil pengujian menunjukkan bahwa proses penyisipan berhasil dilakukan pada seluruh video tanpa menyebabkan kerusakan file maupun error saat penyimpanan. Stego-video yang dihasilkan dapat diputar dengan normal, dan tidak ditemukan perubahan visual yang signifikan dibandingkan video asli. Hal ini membuktikan bahwa proses enkripsi dan penyisipan dapat berjalan dengan baik dan konsisten.

4.3 Pengujian Keamanan dan Ketahanan

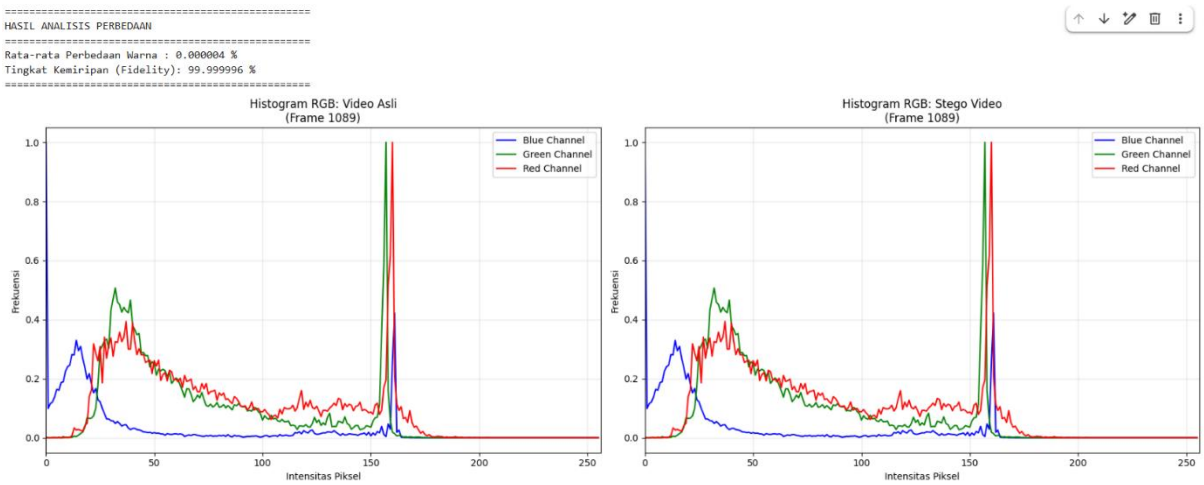
Pengujian keamanan dan ketahanan dirancang untuk mengevaluasi kemampuan sistem dalam melindungi kerahasiaan data terhadap serangan pihak ketiga serta mendeteksi anomali yang mungkin timbul akibat proses steganografi. Berdasarkan skenario yang telah ditetapkan pada Bab 3, pengujian ini dibagi menjadi empat kategori utama yautu Pengujian Kerahasiaan, Analisis Statistik Histogram, Analisis Ukuran Berkas, dan Uji Ketahanan Terhadap Frame Drop Attack.

4.3.1 Pengujian Kerahasiaan

Skenario pengujian ini mensimulasikan upaya akses ilegal (*unauthorized access*), di mana pihak ketiga mencoba mengekstrak pesan tersembunyi menggunakan kunci (*passphrase*) yang tidak valid. Pengujian dilakukan untuk memverifikasi integritas mekanisme *padding* PKCS#7 serta tingkat sensitivitas algoritma *Advanced Encryption Standard* (AES) terhadap perubahan kunci. Dalam pengujian ini, pesan asli yang disisipkan adalah “Data Rahasia Negara” dengan password yang benar “Admin123”, sementara pihak penyusup mencoba melakukan ekstraksi menggunakan password “Admin124” yang hanya berbeda satu karakter. Hasil pengujian menunjukkan bahwa sistem menolak menampilkan pesan apa pun kepada pengguna. Pada sisi backend, sistem mendeteksi terjadinya pengecualian berupa kesalahan *padding* (*padding error*), sehingga proses dekripsi dihentikan. Tidak adanya teks acak atau informasi tidak bermakna yang ditampilkan ke antarmuka pengguna menunjukkan bahwa sistem telah menangani kegagalan dekripsi dengan benar. Kondisi ini membuktikan penerapan prinsip Avalanche Effect pada algoritma AES, di mana perubahan kecil pada input kunci menghasilkan perbedaan kunci derivasi yang signifikan. Akibatnya, blok data hasil dekripsi tidak memiliki format *padding* yang valid, sehingga proses *unpadding* gagal. Mekanisme ini berperan penting dalam menjaga keamanan sistem, karena mencegah penyerang memperoleh umpan balik berupa pola teks yang dapat dimanfaatkan dalam upaya kriptanalisis.

4.3.2 Analisis Statistik Histogram

Skenario pengujian ini mensimulasikan serangan visual *steganalysis* dengan tujuan membuktikan terpenuhinya aspek *imperceptibility* (ketidaktampakan), yaitu kondisi di mana proses penyisipan pesan tidak menyebabkan perubahan signifikan pada distribusi warna piksel yang dapat dideteksi oleh pengamatan visual manusia maupun analisis statistik sederhana. Pada pengujian ini, diambil satu frame secara acak dari video asli (*cover*) dan frame yang sama dari video hasil penyisipan (*stego*). Kedua frame tersebut kemudian dianalisis distribusi warna Merah, Hijau, dan Biru (RGB) menggunakan grafik histogram. Hasil pengujian, sebagaimana ditunjukkan pada Gambar 4.3, memperlihatkan bahwa pola histogram antara citra *cover* dan *stego* memiliki kemiripan yang sangat tinggi dan tampak identik secara kasat mata, sehingga mengindikasikan bahwa penyisipan pesan tidak menimbulkan distorsi visual yang signifikan pada video.



Gambar 4.7 Hasil Pengujian Histogram.

Kemiripan tinggi ini terjadi karena metode *Least Significant Bit* (LSB) hanya mengubah nilai bit terakhir dari sebuah byte warna. Dalam rentang nilai piksel 0–255, perubahan +1 atau -1 pada satu bit LSB tidak memberikan dampak visual yang nyata terhadap intensitas warna. Hal ini mengindikasikan bahwa pesan sulit dideteksi melalui inspeksi visual maupun analisis histogram standar, sehingga sistem dinilai aman dari serangan deteksi visual dasar.

4.3.3 Analisis Ukuran Berkas

Skenario pengujian ini bertujuan untuk menganalisis dampak proses penyisipan data terhadap ukuran berkas keluaran dengan membandingkan ukuran video asli (*cover*) berformat

MP4 dan video hasil steganografi (*stego*) berformat AVI. Hasil pengujian menunjukkan adanya peningkatan ukuran berkas yang cukup signifikan setelah proses penyisipan pesan, sebagaimana ditampilkan pada Tabel 4.2, yang mengindikasikan bahwa perubahan format serta proses steganografi berkontribusi terhadap bertambahnya ukuran file video.

Tabel 4.2 Hasil Pengujian Ukuran Berkas

Sampel Video	Ukuran Awal (KB)	Ukuran Akhir (KB)	Persentase Kenaikan
Video_144p	2351	48100	1946,7%
Video_360p	5230	250000	4679,2%
Video_720p	21857	862000	3844,5%

Peningkatan ukuran berkas pada video hasil steganografi merupakan konsekuensi teknis yang tidak dapat dihindari dalam penerapan metode *Least Significant Bit* (LSB) pada media video. Perubahan ini bukan merupakan kesalahan sistem, melainkan dampak langsung dari strategi yang digunakan untuk menjaga keutuhan pesan rahasia yang disisipkan pada bit paling tidak signifikan dari setiap piksel.

Faktor pertama yang menyebabkan peningkatan ukuran berkas adalah perubahan format kontainer video. Video masukan umumnya menggunakan format MP4 dengan tingkat kompresi tinggi, seperti kodek H.264. Namun, untuk memastikan bahwa data yang disisipkan tidak mengalami perubahan, video keluaran disimpan dalam format yang mendukung penyimpanan data piksel secara presisi, yaitu AVI. Format ini memungkinkan penyimpanan data secara lossless atau near-lossless, sehingga integritas bit-bit LSB dapat dipertahankan.

Faktor kedua berkaitan dengan penghindaran penggunaan kompresi *lossy*. Metode LSB sangat sensitif terhadap proses kompresi, karena algoritma kompresi cenderung menghapus perubahan kecil pada bit terakhir yang dianggap sebagai *noise*. Jika video hasil steganografi dikompresi kembali untuk memperkecil ukuran berkas, maka pesan tersembunyi berpotensi hilang secara permanen dan tidak dapat diekstraksi kembali.

Oleh karena itu, sistem dirancang untuk mempertahankan kualitas piksel asli tanpa melakukan kompresi agresif, meskipun konsekuensinya adalah peningkatan ukuran berkas yang cukup signifikan atau terjadinya file *size overhead*. Meskipun kondisi ini berpotensi menarik perhatian dalam analisis forensik berbasis ukuran file, peningkatan ukuran tersebut merupakan bentuk *trade-off* yang diperlukan untuk menjamin ketersediaan data (*availability*) serta keberhasilan proses ekstraksi pesan rahasia dari video steganografi.

4.3.4 Uji Ketahanan Terhadap Frame Drop Attack

Skenario pengujian ini bertujuan untuk menguji batas toleransi sistem (*robustness*) terhadap kerusakan media video. Pengujian dilakukan dengan mempertimbangkan karakteristik metode *Least Significant Bit* (LSB) yang bersifat rapuh (*fragile*), sehingga bertujuan untuk mengonfirmasi perilaku sistem ketika sebagian data pada media penampung mengalami kehilangan atau kerusakan.

Pada prosedur pengujian, video steganografi dipotong sebanyak satu frame saja. Setelah proses pemotongan tersebut, dilakukan upaya ekstraksi pesan tersembunyi dari video yang telah mengalami perubahan struktur data.

Hasil pengujian menunjukkan bahwa sistem gagal mengembalikan pesan asli. Proses ekstraksi berhenti atau menghasilkan kesalahan (error), yang disebabkan oleh terputusnya struktur data pesan dalam format Base64 di tengah proses pembacaan data.

Kondisi ini mengonfirmasi hipotesis bahwa metode steganografi LSB bersifat rapuh (*fragile*). Dalam implementasi sistem ini, pesan dienkripsi kemudian dipecah dan disebarkan ke seluruh frame video untuk pemerataan modifikasi bit. Kehilangan atau kerusakan satu frame saja sudah cukup untuk memutus rantai data terenkripsi, sehingga proses ekstraksi tidak dapat diselesaikan. Meskipun terlihat sebagai kelemahan, sifat ini justru menunjukkan bahwa sistem lebih sesuai digunakan pada skenario komunikasi tertutup yang menuntut integritas data tinggi, seperti penyimpanan bukti digital, di mana setiap kerusakan file harus terdeteksi sebagai kegagalan total, dan bukan untuk transmisi data pada saluran yang bising atau tidak stabil.

4.4 Pengujian Kualitas Video (PSNR)

Pengujian tahap akhir difokuskan pada aspek *imperceptibility* (ketidaktampakan) untuk memastikan bahwa penyisipan pesan rahasia tidak mendegradasi kualitas visual video secara kasat mata. Pengukuran kualitas citra dilakukan secara objektif menggunakan parameter *Peak Signal-to-Noise Ratio* (PSNR).

Pengukuran dilakukan dengan membandingkan nilai piksel frame demi frame antara video asli (*cover video*) dengan video hasil steganografi (*stego video*). Proses perhitungan dilakukan secara komputasi menggunakan skrip pemrograman Python untuk mendapatkan nilai rata-rata MSE dan PSNR dari seluruh frame video.

Nilai MSE merepresentasikan tingkat kesalahan atau perbedaan data antar citra, sedangkan nilai PSNR merepresentasikan perbandingan sinyal terhadap *noise*. Sesuai standar

pengolahan citra digital, kualitas steganografi dikatakan baik jika nilai MSE mendekati nol dan nilai PSNR berada di atas 30 dB.

Berdasarkan pengujian terhadap sampel video dengan resolusi yang berbeda, diperoleh hasil perhitungan kuantitatif seperti yang disajikan pada Tabel 4.3.

Tabel 4.3 Hasil Nilai PSNR

No.	Resolusi Video	Ukuran Pesan (KB)	*Frame Rate* (FPS)	Nilai PSNR (dB)	Keterangan Kualitas
1	144p	0,471	30	92,69	Sangat Baik
2	144p	1,45	30	87,97	Sangat Baik
3	144p	1900	30	61,67	Sangat Baik
4	360p	0,471	30	100,7	Sangat Baik
5	360p	1,45	30	95,98	Sangat Baik
6	360p	4290	30	66,14	Sangat Baik
7	720p	0,471	30	106,6	Sangat Baik
8	720p	1,45	30	102,03	Sangat Baik
9	720p	16600	30	66,28	Sangat Baik

Berdasarkan data yang disajikan pada Tabel 4.3, dapat dilakukan analisis terhadap kualitas visual video hasil steganografi menggunakan parameter *Peak Signal-to-Noise Ratio* (PSNR). Hasil pengujian menunjukkan adanya variasi nilai PSNR pada setiap sampel video yang diuji, namun tetap berada pada rentang yang sangat tinggi.

Rentang nilai PSNR yang diperoleh berada antara 61,67 dB hingga 106,60 dB. Nilai terendah yang dihasilkan, yaitu 61,67 dB, masih jauh melampaui standar kelayakan umum pada sistem steganografi, yang umumnya berada di atas 40 dB. Hal ini menunjukkan bahwa algoritma yang diterapkan mampu menjaga konsistensi kualitas visual video meskipun dilakukan proses penyisipan pesan pada berbagai jenis dan karakteristik sampel video.

Selain itu, terdapat beberapa sampel yang menghasilkan nilai PSNR di atas 100 dB, yaitu pada Sampel 4 dan Sampel 7. Secara matematis, nilai PSNR yang sangat tinggi ini mengindikasikan bahwa nilai *Mean Square Error* (MSE) mendekati nol. Kondisi tersebut terjadi karena kapasitas piksel pada video cover jauh lebih besar dibandingkan jumlah bit pesan

yang disisipkan, sehingga sebagian besar frame video tidak mengalami perubahan sama sekali. Akibatnya, keaslian data piksel tetap terjaga hampir sepenuhnya.

Dengan nilai rata-rata PSNR keseluruhan sampel yang berada pada kisaran ±87 dB, dapat disimpulkan bahwa distorsi visual yang ditimbulkan oleh sistem steganografi ini bersifat sangat minimal hingga tidak terdeteksi. Mata manusia tidak memiliki sensitivitas yang cukup untuk membedakan perubahan intensitas warna pada level bit yang sangat kecil tersebut. Oleh karena itu, video hasil steganografi yang dihasilkan telah memenuhi aspek keamanan *imperceptibility* (ketidaktampakan) dengan kategori Sangat Baik.

4.5 Pembahasan Hasil Pengujian

Berdasarkan seluruh rangkaian pengujian yang telah dilakukan, mulai dari uji fungsional, keamanan, hingga pengukuran kualitas citra, bagian ini akan menguraikan analisis mendalam mengenai kinerja sistem secara keseluruhan serta membandingkannya dengan penelitian terkait.

4.5.1 Efektivitas Keamanan dan Imperceptibility

Penggabungan algoritma kriptografi AES dan metode steganografi LSB terbukti memberikan perlindungan ganda yang efektif bagi keamanan pesan rahasia. Dari sisi kerahasiaan data (*confidentiality*), implementasi AES memastikan bahwa pesan diubah menjadi *ciphertext* acak yang tidak dapat dipahami maknanya tanpa kunci yang benar sebelum proses penyisipan dilakukan. Mekanisme ini menjamin keamanan konten meskipun lapisan steganografi berhasil ditembus.

Sementara itu, dari sisi penyembunyian (*imperceptibility*), metode LSB mampu menjaga aspek penyamaran dengan sangat baik. Video steganografi yang dihasilkan tidak menunjukkan perubahan visual yang mencolok dan tetap dapat diputar secara normal sebagaimana video aslinya. Hal ini diperkuat oleh hasil analisis histogram yang menunjukkan perbedaan distribusi warna antara video asli dan video stego berada pada tingkat yang sangat kecil, sehingga keberadaan informasi rahasia di dalamnya sulit dideteksi melalui analisis statistik visual (*steganalysis*).

4.5.2 Korelasi Resolusi Video terhadap Kualitas (PSNR)

Analisis terhadap kualitas visual menggunakan parameter *Peak Signal-to-Noise Ratio* (PSNR) menunjukkan adanya korelasi positif yang signifikan antara resolusi video dengan kapasitas penyisipan pesan.

Pada Resolusi menengah dan tinggi yakni 360p dan 720p didapatkan bahwa Sistem mampu mempertahankan nilai PSNR yang sangat tinggi, bahkan mencapai lebih dari 100 dB untuk kategori pesan pendek. Hal ini mengindikasikan bahwa ketersediaan jumlah piksel yang lebih besar memungkinkan distribusi bit pesan dilakukan secara lebih menyebar (*spread*), sehingga meminimalkan modifikasi pada area yang berdekatan dan menjaga kualitas visual tetap prima.

Kemudian pada resolusi rendah (144p) didapati penyisipan pesan sebesar 1900 KB menyebabkan penurunan nilai PSNR ke angka 61,67 dB. Meskipun nilai ini masih masuk dalam kategori sangat baik (>40 dB), penurunan ini mencerminkan adanya degradasi kualitas yang lebih terasa akibat keterbatasan kapasitas piksel pada resolusi rendah.

4.5.3 Komparasi dengan Penelitian Terdahulu

Untuk memvalidasi posisi dan kontribusi penelitian ini, dilakukan perbandingan karakteristik metode yang digunakan dengan beberapa penelitian sebelumnya yang relevan. Perbandingan difokuskan pada aspek kualitas visual dan ketahanan data untuk menunjukkan keunggulan serta keterbatasan pendekatan yang diusulkan.

Pada aspek kualitas visual, penelitian ini dibandingkan dengan penelitian steganografi video yang menggunakan algoritma *End of File* (EOF) dan Rijndael. Metode berbasis EOF menghasilkan nilai *Peak Signal-to-Noise Ratio* (PSNR) yang tak hingga, karena proses penyisipan data dilakukan di luar struktur piksel video sehingga tidak menyebabkan perubahan matematis pada citra (Riadi et al., 2021). Sebaliknya, pada penelitian ini, proses penyisipan dilakukan langsung pada bit *Least Significant Bit* (LSB) piksel, sehingga menghasilkan nilai PSNR yang bersifat terukur (*finite*). Meskipun demikian, nilai PSNR yang diperoleh tetap sangat tinggi, dengan rata-rata di atas 85 dB, yang menunjukkan bahwa metode LSB mampu menjaga kualitas visual video secara optimal dan tetap memenuhi aspek *imperceptibility* tanpa menimbulkan kecurigaan visual.

Sementara itu, dari aspek ketahanan data, hasil pengujian Frame Dropping Attack menunjukkan bahwa penghapusan satu frame saja sudah menyebabkan kegagalan dalam proses ekstraksi pesan. Temuan ini sejalan dengan penelitian yang mengombinasikan Vigenere Cipher dan EOF, di mana manipulasi fisik terhadap file video dapat merusak urutan data pesan yang disisipkan. Hal ini menegaskan bahwa kedua pendekatan tersebut sama-sama memiliki sifat sensitif atau rapuh (*fragile*) terhadap kerusakan media (Minarni et al., 2023).

Meskipun demikian, penelitian ini memiliki keunggulan yang signifikan pada aspek keamanan kriptografi. Penggunaan algoritma *Advanced Encryption Standard* (AES) sebagai

mekanisme enkripsi memberikan tingkat keamanan yang lebih tinggi dibandingkan penggunaan Vigenere Cipher yang merupakan kriptografi klasik dan relatif mudah dianalisis serta diretas. Dengan demikian, meskipun sama-sama memiliki keterbatasan pada ketahanan terhadap kerusakan file, penelitian ini menawarkan peningkatan pada aspek kerahasiaan dan keamanan data yang lebih relevan dengan kebutuhan sistem modern.

4.5.4 Keterbatasan Sistem

Meskipun sistem steganografi video yang dikembangkan menunjukkan kinerja yang baik dari sisi keamanan data dan kualitas visual, hasil pengujian juga mengungkapkan adanya beberapa keterbatasan teknis yang perlu diperhatikan. Keterbatasan ini berkaitan dengan aspek kompatibilitas format serta efisiensi waktu pemrosesan sistem.

Keterbatasan pertama terletak pada kompatibilitas format video. Sistem saat ini hanya menerima masukan video berformat MP4, namun menghasilkan keluaran dalam format AVI. Kondisi ini terjadi karena metode *Least Significant Bit* (LSB) menuntut penggunaan format video yang menghindari kompresi lossy, yang berpotensi merusak bit-bit pesan yang disisipkan. Oleh karena itu, format AVI dipilih untuk menjaga integritas data piksel, meskipun konsekuensinya adalah peningkatan ukuran berkas dan keterbatasan fleksibilitas format keluaran.

Keterbatasan berikutnya berkaitan dengan kompleksitas waktu pemrosesan. Waktu yang dibutuhkan sistem untuk menjalankan proses enkripsi, penyisipan pesan pada setiap frame, serta rekonstruksi ulang video sangat dipengaruhi oleh durasi video masukan. Semakin panjang durasi video, yang berarti semakin banyak jumlah frame yang harus diproses, maka semakin lama pula waktu pemrosesan yang dibutuhkan. Hal ini menunjukkan bahwa kompleksitas komputasi sistem berbanding lurus dengan jumlah frame video, sehingga menjadi catatan penting untuk pengembangan dan optimasi kinerja sistem pada penelitian selanjutnya.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian sistem steganografi video menggunakan kombinasi algoritma kriptografi *Advanced Encryption Standard* (AES) dan metode *Least Significant Bit* (LSB), dapat ditarik beberapa kesimpulan sebagai berikut:

1. Keberhasilan Integrasi Sistem Aplikasi steganografi video berbasis desktop telah berhasil dikembangkan menggunakan bahasa pemrograman Python. Sistem mampu menyisipkan pesan teks yang telah diamankan dengan enkripsi AES ke dalam frame video berformat MP4 dan menghasilkan keluaran video berformat AVI. Proses ekstraksi pesan juga berhasil dilakukan dengan syarat kunci yang dimasukkan tepat.
2. Kualitas Visual dan *Imperceptibility* Penerapan metode LSB terbukti sangat efektif dalam menjaga kualitas visual video. Berdasarkan pengujian *Peak Signal-to-Noise Ratio* (PSNR), sistem menghasilkan nilai rata-rata kualitas citra yang sangat tinggi, berkisar antara 61,67 dB hingga 106,60 dB tergantung pada resolusi video. Nilai ini jauh melampaui ambang batas standar kelayakan visual (40 dB), yang berarti perubahan pada video tidak dapat dideteksi oleh indra penglihatan manusia maupun analisis histogram sederhana.
3. Keamanan Data (*Security*) Penggunaan algoritma AES dengan mode *Cipher Block Chaining* (CBC) memberikan lapisan keamanan yang kuat sebelum proses penyisipan. Hasil pengujian menunjukkan bahwa kesalahan satu karakter saja pada input kunci dekripsi menyebabkan kegagalan total dalam pemulihan pesan (*Avalanche Effect*), sehingga kerahasiaan pesan tetap terjamin meskipun keberadaan steganografi diketahui pihak ketiga.
4. Karakteristik Ketahanan (*Robustness*) dan Efisiensi Sistem yang dibangun memiliki karakteristik rapuh (*fragile*) terhadap manipulasi media. Pengujian menunjukkan bahwa penghapusan frame (*frame dropping*) atau kompresi ulang video akan merusak struktur data LSB dan menyebabkan pesan tidak dapat diekstrak. Selain itu, demi menjaga integritas bit pesan, sistem menghasilkan ukuran berkas video keluaran yang lebih besar dibandingkan video asli karena penggunaan format kontainer tanpa kompresi *lossy*.

5.2 Saran

Berdasarkan keterbatasan yang ditemukan selama proses penelitian dan pengujian, terdapat beberapa saran untuk pengembangan penelitian selanjutnya agar sistem dapat menjadi lebih optimal:

1. Pengembangan Metode yang Lebih Tangguh (*Robust*) Untuk mengatasi kelemahan metode LSB yang rentan terhadap kompresi dan manipulasi frame, penelitian selanjutnya disarankan untuk menerapkan metode steganografi pada domain frekuensi, seperti *Discrete Cosine Transform* (DCT) atau *Discrete Wavelet Transform* (DWT). Metode ini memungkinkan pesan disisipkan pada koefisien frekuensi sehingga lebih tahan terhadap serangan kompresi dan *scaling*.
2. Optimasi Ukuran Berkas Sistem saat ini menghasilkan video format AVI dengan ukuran yang cukup besar. Pengembangan selanjutnya dapat mengeksplorasi penggunaan format kontainer video modern (seperti MKV atau implementasi codec H.264 *lossless*) yang mampu menampung data piksel mentah dengan efisiensi penyimpanan yang lebih baik tanpa merusak bit pesan.
3. Peningkatan Efisiensi Waktu Komputasi Mengingat proses enkripsi dan penyisipan dilakukan secara sekuensial pada setiap frame, waktu pemrosesan menjadi lambat pada video berdurasi panjang. Disarankan untuk mengimplementasikan teknik pemrosesan paralel (*multithreading* atau *multiprocessing*) atau memanfaatkan akselerasi GPU (*GPU Acceleration*) untuk mempercepat proses manipulasi piksel.
4. Ekspansi Platform Aplikasi Aplikasi saat ini masih berbasis desktop. Penelitian selanjutnya dapat mengembangkan sistem ini ke dalam platform berbasis web atau aplikasi seluler (*mobile*) untuk meningkatkan fleksibilitas dan kemudahan akses bagi pengguna dalam mengamankan pertukaran data multimedia.

DAFTAR PUSTAKA

Aslam, M. A., Rashid, M., Azam, F., Abbas, M., & ... (2022). Image steganography using least significant bit (lsb)-a systematic literature review. *... on Computing and ...*. https://ieeexplore.ieee.org/abstract/document/9711628/

Belyaev, E. (2023). An Efficient Compressive Sensed Video Codec with Inter-Frame Decoding and Low-Complexity Intra-Frame Encoding. *Sensors*, *23*(3). https://doi.org/10.3390/s23031368

Fitriani, L. A. (2020). Analisa Keamanan Data Teks Dengan Menerapkan Kriptografi RSA Dan Steganografi LSB. In *J. Comput. Syst. Informatics*. download.garuda.kemdikbud.go.id. http://download.garuda.kemdikbud.go.id/article.php?article=1604485&val=18036&title=Analisa Keamanan Data Teks Dengan Menerapkan Kriptografi RSA Dan Steganografi LSB

Fuad, M., & Ernawan, F. (2020). Video steganography based on DCT psychovisual and object motion. *Bulletin of Electrical Engineering and Informatics*. https://beei.org/index.php/EEI/article/view/1859

Hacimurtazaoglu, M., & Tutuncu, K. (2022). LSB-based pre-embedding video steganography with rotating &shifting poly-pattern block matrix. In *PeerJ Computer Science*. peerj.com. https://peerj.com/articles/cs-843/

Kunhoth, J., Subramanian, N., Al-Maadeed, S., & ... (2023). Video steganography: recent advances and challenges. In *Multimedia Tools and ...*. Springer. https://doi.org/10.1007/s11042-023-14844-w

Laksono, A. W., Suhada, S., & Zakaria, A. (2024). Implementasi Metode Least Significant Bit (Lsb) Dalam Teknik Steganografi Pada Citra Digital Menggunakan Matlab. *Diffusion: Journal of Systems ...*. https://ejurnal.ung.ac.id/index.php/diffusion/article/view/24194

Malvi, A., & Painem, P. (2020). Pengamanan File Gambar pada Media Video dengan Kriptografi Algoritma RSA dan Steganografi Algoritma End of File (EOF). *Informatik: Jurnal Ilmu Komputer*. https://ejournal.upnvj.ac.id/informatik/article/view/1860

Minarni, M., Ikram, A., Warman, I., & Swara, G. Y. (2023). Implementasi Algoritma Vigenere Cipher Dan End Of File Pada Steganografi Video. *Jurnal Minfo Polgan*. https://www.jurnal.polgan.ac.id/index.php/jmp/article/view/12418

Nirmala, E. (2020). Penerapan Steganografi File Gambar Menggunakan Metode Least Significant Bit (LSB) Dan Algoritma Kriptografi *Advanced Encryption Standard* (AES).

In *Jurnal Informatika Universitas Pamulang*. academia.edu. https://www.academia.edu/download/72551649/pdf.pdf

Oktavani, S., Rizky, F., & Gunawan, I. (2023). *Analisis Keamanan Data Dengan Menggunakan Kriptografi Modern Algoritma Advance Encryption Standar (AES) JURNAL MEDIA INFORMATIKA [JUMIN]*. *4*, 97–101.

Riadi, I., Sunardi, S., & Aryanto, D. (2020). Steganografi Video Digital dengan Algoritma LSB (Least Significant Bit) dan Rijndael. In *J. Innov. Inf* .... download.garuda.kemdikbud.go.id. http://download.garuda.kemdikbud.go.id/article.php?article=2070624&val=18481&title=Steganografi Video Digital dengan Algoritma LSB Least Significant Bit dan Rijndael

Riadi, I., Sunardi, S., & Aryanto, D. (2021). Algoritma End of File dan Rijndael pada Steganografi Video. *JRST (Jurnal Riset Sains* https://jurnalnasional.ump.ac.id/index.php/JRST/article/view/9187

Şahin, F., Çevik, T., & Takaoğlu, M. (2021). Review of the Literature on the Steganography Concept. In *International Journal of Computer* .... researchgate.net. https://www.researchgate.net/profile/Mustafa-Takaoglu/publication/351713920_Review_of_the_Literature_on_the_Steganography_Concept/links/60a6499e299bf10d2eb7c581/Review-of-the-Literature-on-the-Steganography-Concept.pdf

Set, F., Bana, C. M. N., Anunut, M. A., & ... (2025). Penerapan Steganografi LSB dan Enkripsi AES untuk Keamanan Data Rahasia pada Gambar Digital. *Blantika* https://blantika.publikasiku.id/index.php/bl/article/view/382

Simbolon, B. J., & Nusantara, S. P. (2021). Steganografi Penyisipan Pesan Pada File Citra Menggunakan Metode LSB (Least Significant Bit). In *Jurnal* .... download.garuda.kemdikbud.go.id. http://download.garuda.kemdikbud.go.id/article.php?article=1929454&val=13467&title=Steganografi Penyisipan Pesan Pada File Citra Dengan Menggunakan Metode LSB Least Significant Bit

Unggul Budi Astowo. (2024). *Desain Komunikasi Dan Keamanan Data Arsitektur Aplikasi Multimasjid*. https://dspace.uii.ac.id/bitstream/handle/123456789/51606/20523026.pdf?sequence=1&isAllowed=y