PENGEMBANGAN APLIKASI WEB STEGANOGRAFI UNTUK KEAMANAN DATA: IMPLEMENTASI DAN ANALISIS KEAMANAN

Shelinna Puspita Ali[1*] [1*], Wahyu Noviani Purwanti[2]
[1,2]Sistem Informasi, Universitas Terbuka, Tangerang Selatan, Banten Indonesia

*****Penulis korespondensi: selinapuspitaa@gmail.com*

ABSTRAK

Di tengah meningkatnya ancaman siber, keamanan data menjadi hal penting dalam pertukaran informasi digital. Oleh karena itu, dibutuhkan aplikasi yang dapat melindungi data saat dikirim. Penelitian ini mengembangkan aplikasi web steganografi bernama InvisiByte untuk menyisipkan pesan rahasia ke dalam gambar secara aman, menggunakan kombinasi algoritma *Least Significant Bit* (LSB) dan *Advanced Encryption Standard* (AES). Pengembangan aplikasi menggunakan metode *Rapid Application Development* (RAD), yang mencakup pemodelan bisnis, data, proses, dan implementasi dengan Laravel (PHP) dan Python. Fitur utamanya berupa proses *encode* untuk menyisipkan pesan dengan perlindungan kata sandi dan *decode* untuk mengambil pesan menggunakan kata sandi yang sama. Pengujian dilakukan untuk menilai fungsi, keamanan terhadap serangan steganalisis, serta efektivitas pada berbagai ukuran gambar. Hasilnya menunjukkan bahwa InvisiByte dapat menyembunyikan data tanpa menurunkan kualitas gambar secara signifikan, meski masih rentan terhadap teknik deteksi canggih. Integrasi AES terbukti memperkuat keamanan pesan, sementara LSB tetap efektif untuk penyisipan teks. Aplikasi ini berpotensi digunakan dalam intelijen, keamanan siber, dan komunikasi rahasia, dengan kemungkinan pengembangan lebih lanjut melalui kecerdasan buatan dan teknik steganografi lanjutan.

Kata kunci: Advanced Encryption Standard, Kriptografi, Least Significant Bit, Steganografi

1. PENDAHULUAN

Di era digital yang semakin maju, keamanan data menjadi salah satu aspek paling krusial dalam kehidupan saat ini. Pesatnya perkembangan teknologi informasi telah mempermudah akses dan pertukaran data, namun di sisi lain juga membuka peluang bagi ancaman seperti pencurian informasi, peretasan, dan penyadapan. Sunardi et al (2020) mengungkapkan "Sepanjang tahun 2018, Badan Siber Sandi Negara (BSSN) mencatat terdapat 225,9 kejahatan dengan menggunakan teknologi komputer sehingga merugikan ekonomi Indonesia hingga 400 Triliun Rupiah." (p. 1). Data sensitif, baik milik individu, perusahaan, maupun pemerintahan, kini menjadi target utama pihak-pihak yang tidak bertanggung jawab. Dalam konteks ini, kebutuhan akan metode perlindungan data yang efektif dan tidak mudah terdeteksi menjadi semakin mendesak. Salah satu solusi yang muncul adalah steganografi, "Steganografi merupakan metode menyembunyikan pesan dalam media digital sedemikian rupa sehingga orang lain tidak menyadari ada sesuatu pesan di dalam media tersebut" (Wijaya, et al., 2025, p. 337). Berbeda dengan kriptografi yang mengacak data menjadi bentuk yang tidak terbaca, sehingga pesan yang disampaikan dalam kriptografi menjadi mencurigakan (Adhimah, et al., 2023).

Peran steganografi dalam melindungi komunikasi rahasia menjadikannya alat penting di tengah tantangan keamanan digital saat ini, terutama untuk menjaga kerahasiaan informasi yang bersifat sensitif. Melihat pentingnya pendekatan ini, penelitian ini berangkat dari dua pertanyaan utama yang perlu dijawab. Pertama, bagaimana merancang aplikasi web steganografi yang praktis dan efisien untuk menyembunyikan data? Pertanyaan ini mencerminkan perlunya panduan teknis dalam membangun sistem yang mudah diakses

melalui platform berbasis web. Kedua, sejauh mana keamanan metode steganografi terhadap serangan steganalisis? Steganalisis sendiri merupakan teknik untuk mendeteksi dan mengungkap keberadaan pesan tersembunyi dalam media digital (Sunardi et al., 2020, p. 1). Tingkat ketahanan terhadap serangan ini menjadi indikator penting dalam menilai efektivitas steganografi sebagai alat perlindungan data, terutama di tengah berkembangnya metode forensik digital. Kedua pertanyaan tersebut menjadi dasar dalam merancang dan mengevaluasi aplikasi web steganografi yang aman dan efisien di era digital saat ini.

Merujuk pada latar belakang dan rumusan masalah yang telah dibahas, penelitian ini memiliki beberapa tujuan yang ingin dicapai. Salah satunya adalah mengembangkan aplikasi web steganografi menggunakan *framework* Laravel dan bahasa pemrograman Python, dengan memanfaatkan algoritma *Least Significant Bit* (LSB) sebagai metode utama dalam menyisipkan pesan ke dalam gambar. Menurut Permana et al. (2023), LSB merupakan salah satu teknik yang paling umum digunakan dalam steganografi gambar. Teknik ini memiliki keunggulan karena tidak secara nyata mengubah kualitas visual gambar, sehingga pesan tersembunyi menjadi sulit terdeteksi (Wijaya et al., 2025).

Selain pengembangan aplikasi, penelitian ini juga difokuskan pada analisis aspek keamanannya, terutama dalam menghadapi serangan steganalisis. Evaluasi ini penting untuk mengukur seberapa tahan metode yang digunakan terhadap upaya pendeteksian dan pembongkaran pesan oleh pihak yang tidak berwenang. Dengan pendekatan ini, penelitian diharapkan tidak hanya menghasilkan solusi teknologi berbasis web, tetapi juga memberikan kontribusi terhadap pengembangan sistem perlindungan data yang adaptif terhadap ancaman digital yang terus berkembang.

2. METODE

Penelitian ini menggunakan pendekatan rekayasa perangkat lunak yang bertujuan untuk membangun dan mengevaluasi sebuah aplikasi web steganografi bernama InvisiByte. Pendekatan ini menekankan pada proses iteratif dan sistematis, dimulai dari identifikasi masalah hingga pengujian sistem. Setiap tahapan dilaksanakan secara terstruktur untuk memastikan bahwa pengembangan aplikasi berjalan sesuai dengan tujuan dan menghasilkan solusi yang layak serta relevan dalam konteks keamanan data digital.

2.1 Skema Alur Penelitian

Adapun alur penelitian ini terdiri dari enam tahapan utama sebagai berikut:

a. Identifikasi Masalah
   Pada tahap ini, penulis mengamati meningkatnya kebutuhan akan perlindungan data dalam pertukaran informasi digital, terutama dalam menghadapi ancaman siber yang semakin kompleks. Isu ini mendorong lahirnya ide untuk membangun sebuah aplikasi steganografi berbasis web yang mampu menyisipkan pesan rahasia secara aman di dalam gambar digital.
b. Pengumpulan Data
   Penulis melakukan studi literatur dari berbagai sumber seperti jurnal ilmiah, artikel teknologi, dan dokumentasi teknis untuk memahami lebih dalam tentang algoritma steganografi, metode enkripsi data, dan teknologi pengembangan aplikasi web. Informasi yang dikumpulkan digunakan sebagai dasar dalam menentukan pendekatan dan teknologi yang akan diterapkan dalam pengembangan sistem.
c. Analisis Sistem
   Pada tahap ini, penulis menganalisis kebutuhan sistem baik dari sisi fungsional maupun non-fungsional. Kebutuhan tersebut meliputi proses *encode* dan *decode*, pengamanan dengan enkripsi, *user interface* yang sederhana, serta performa sistem saat menangani berbagai ukuran gambar dan panjang pesan. Analisis ini digunakan untuk menentukan spesifikasi sistem yang akan dibangun.

d. Perancangan dan Implementasi Sistem
   Penulis merancang antarmuka pengguna serta arsitektur sistem dengan mempertimbangkan kemudahan penggunaan dan efisiensi proses. Setelah itu, dilakukan implementasi sistem menggunakan *framework* Laravel untuk pengelolaan antarmuka dan logika *backend*, serta Python untuk proses steganografi menggunakan *library* Pillow dan Stepic dan enkripsi AES menggunakan OpenSSL.
e. Pengujian
   Aplikasi diuji dari segi fungsionalitas, keamanan, dan efektivitas. Pengujian dilakukan dengan menyisipkan dan mengekstraksi pesan menggunakan gambar dari berbagai ukuran, serta melakukan simulasi serangan steganalisis seperti *cropping* untuk mengevaluasi ketahanan sistem. Hasil pengujian ini menjadi dasar untuk mengetahui keandalan aplikasi dalam skenario nyata.
f. Hasil dan Kesimpulan
   Berdasarkan hasil pengujian, penulis menarik kesimpulan mengenai performa dan efektivitas aplikasi InvisiByte dalam menyembunyikan pesan secara aman. Selain itu, diberikan pula saran pengembangan lanjutan agar sistem dapat menghadapi tantangan teknologi dan ancaman keamanan yang lebih kompleks di masa depan.

2.2 Metode Pengembangan Sistem

Pengembangan aplikasi InvisiByte dilakukan dengan pendekatan *Rapid Application Development* (RAD), *Rapid Application Development* (RAD) merupakan metode pengembangan perangkat lunak yang dirancang untuk mempercepat proses pembangunan sistem melalui siklus yang lebih singkat. Pendekatan ini mengadaptasi model *waterfall* namun dengan alur yang lebih cepat, serta mengombinasikan teknik *prototyping*, pengembangan terstruktur, dan keterlibatan pengguna secara aktif. Dengan demikian, pengembangan aplikasi menggunakan metode RAD dapat dilakukan dalam waktu yang relatif lebih singkat dibandingkan metode konvensional (Putri & Effendi, 2018).

Berikut adalah tahapan metode RAD yang digunakan dalam pengembangan aplikasi ini:

a. Pemodelan Bisnis
   Pada tahap ini, penulis memetakan kebutuhan utama dari aplikasi steganografi, yaitu menyisipkan dan mengekstraksi pesan dengan aman melalui media gambar. Fokus diarahkan pada bagaimana aplikasi akan digunakan oleh pengguna dan bagaimana proses perlindungan data berjalan secara praktis di lingkungan web.
b. Pemodelan Data
   Tahap ini mencakup perancangan struktur data yang dibutuhkan, seperti format pesan, format gambar, hasil enkripsi, dan input *password*. Selain itu, dilakukan perancangan bagaimana data ditangani selama proses *encode* dan *decode*, termasuk hubungan antara antarmuka pengguna, sistem *backend*, dan proses steganografi.
c. Pemodelan Proses
   Penulis mendeskripsikan alur kerja aplikasi secara logis, dari pengguna mengunggah gambar dan memasukkan pesan, hingga sistem melakukan enkripsi dan penyisipan dengan algoritma LSB. Demikian pula untuk proses *decode*, dijelaskan alur bagaimana sistem memverifikasi *password* dan menampilkan pesan rahasia jika berhasil didekripsi.
d. Pembuatan Aplikasi
   Implementasi dilakukan menggunakan Laravel (PHP) untuk pengembangan antarmuka dan pengelolaan logika aplikasi, serta Python dengan *library* Pillow dan Stepic untuk proses steganografi. Enkripsi dilakukan menggunakan algoritma AES yang diintegrasikan melalui OpenSSL. Proses ini juga melibatkan penggunaan JavaScript dan AJAX untuk meningkatkan pengalaman pengguna.
e. Pengujian dan Pergantian

Setelah aplikasi dibangun, dilakukan pengujian awal untuk memastikan seluruh fitur berjalan sesuai rencana. Jika ditemukan kekurangan, penyesuaian segera dilakukan tanpa harus mengulang dari awal. Prototipe aplikasi terus disempurnakan hingga mencapai versi yang stabil.

2.3 Arsitektur Sistem

Aplikasi steganografi web yang dikembangkan dalam penelitian ini menggunakan arsitektur berbasis *client-server*. Dalam arsitektur ini, proses-proses penting seperti enkripsi, penyisipan pesan ke dalam gambar, serta ekstraksi pesan dilakukan di sisi server, sementara sisi *client* berfungsi sebagai antarmuka pengguna yang mengirimkan permintaan dan menerima hasilnya melalui browser.

Secara umum, sistem ini dibagi menjadi beberapa komponen utama, yaitu:

- *Frontend*
  Bagian *frontend* bertanggung jawab dalam menyediakan antarmuka interaktif bagi pengguna. Komponen ini dibangun menggunakan HTML, CSS, dan JavaScript untuk memastikan tampilan responsif dan proses interaksi yang dinamis, seperti pengunggahan gambar dan pengisian form pesan serta *password*.
- *Backend*
  Backend berfungsi mengatur logika aplikasi dan pemrosesan data. *Framework* Laravel berbasis PHP digunakan sebagai kerangka kerja utama untuk menangani permintaan dari *frontend*, melakukan validasi data, mengelola file sementara, dan menghubungkan ke modul Python untuk proses steganografi dan enkripsi.
- Modul Steganografi
  Proses penyisipan dan ekstraksi pesan rahasia ke atau dari gambar dilakukan dengan bahasa pemrograman Python menggunakan *library* Pillow dan Stepic. Algoritma yang digunakan adalah *Least Significant Bit* (LSB), yang berfungsi dengan memodifikasi bit terkecil pada citra digital untuk menyisipkan informasi tanpa mengubah tampilan visual secara signifikan.
- Modul Kriptografi
  Sebelum disisipkan ke dalam gambar, pesan rahasia terlebih dahulu dienkripsi menggunakan metode *Advanced Encryption Standard* (AES). Proses enkripsi dan dekripsi dilakukan dengan bantuan pustaka OpenSSL, untuk memastikan pesan tetap aman meskipun gambar berhasil diakses oleh pihak yang tidak berwenang.

2.4 Fitur Aplikasi

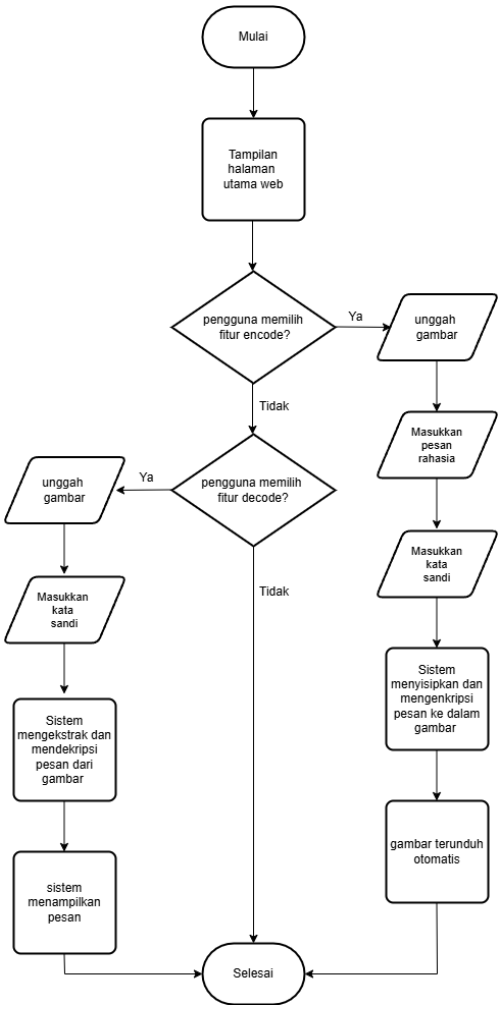
Aplikasi web steganografi yang dikembangkan memiliki dua fitur utama, yaitu proses penyisipan pesan rahasia (*encode*) dan proses pengambilan pesan kembali (*decode*). Seluruh proses ini dilakukan melalui antarmuka web yang responsif dan mudah diakses oleh pengguna.

a. *Encode*
   Fitur ini berfungsi bagi pengguna untuk menyisipkan pesan rahasia ke dalam sebuah gambar dengan pengamanan berupa kata sandi. Prosesnya terdiri atas beberapa langkah:
   1. Pengguna mengunggah gambar yang akan digunakan sebagai media penyisipan.
   2. Pengguna memasukkan pesan rahasia dan kata sandi.
   3. Sistem mengenkripsi pesan menggunakan algoritma AES (*Advanced Encryption Standard*).
   4. Pesan yang telah terenkripsi kemudian disisipkan ke dalam gambar menggunakan metode *Least Significant Bit* (LSB).
   5. Gambar hasil penyisipan (*stego image*) dapat diunduh oleh pengguna.

b. *Decode*

Fitur ini digunakan untuk mengekstrak dan menampilkan kembali pesan rahasia dari gambar yang telah disisipkan sebelumnya, dengan langkah-langkah berikut:

1. Pengguna mengunggah gambar stego yang berisi pesan.
2. Pengguna memasukkan kata sandi yang sesuai.
3. Sistem mengekstrak pesan tersembunyi, lalu mendekripsinya dengan metode AES.
4. Jika kata sandi sesuai, pesan rahasia ditampilkan kepada pengguna.



Gambar 1. Flowchart Sistem InvisiByte

2.5 Metode Pengujian

Pengujian dalam penelitian ini dilakukan untuk memastikan bahwa sistem steganografi yang dibangun dapat berfungsi dengan baik, aman terhadap deteksi pihak ketiga, serta efektif dalam berbagai kondisi penggunaan. Terdapat empat aspek utama yang diuji dalam pengembangan aplikasi InvisiByte, yaitu fungsionalitas, keamanan, efektivitas, dan ketahanan.

a. Aspek Fungsionalitas

Aspek ini bertujuan untuk memastikan bahwa seluruh fitur utama dalam aplikasi berjalan sesuai dengan perancangan. Fokusnya adalah pada akurasi proses penyisipan (*encode*) dan pengambilan (*decode*) pesan. Pengujian dilakukan dengan cara menginput pesan rahasia dan *password* pada fitur *encode*, lalu memastikan bahwa pesan tersebut dapat diambil kembali secara utuh melalui fitur *decode* dengan *password* yang sesuai.

b. Aspek Keamanan

Aspek keamanan diuji dengan melakukan simulasi serangan steganalisis untuk menilai apakah pesan yang disisipkan dapat terdeteksi oleh pihak yang tidak berwenang. Dalam

penelitian ini, aplikasi pihak ketiga bernama StegSpy digunakan untuk mendeteksi keberadaan pesan tersembunyi pada gambar stego. Jika gambar lolos dari deteksi aplikasi tersebut, maka metode penyisipan dapat dikatakan cukup aman dari analisis statis berbasis deteksi pola.

c. Aspek Efektivitas
Pengujian pada aspek ini dilakukan untuk mengevaluasi performa aplikasi saat digunakan dengan berbagai ukuran gambar dan panjang pesan yang berbeda. Tujuannya adalah untuk melihat batas kemampuan sistem dalam mempertahankan kualitas visual gambar serta kestabilan performa saat data yang disisipkan bertambah besar.

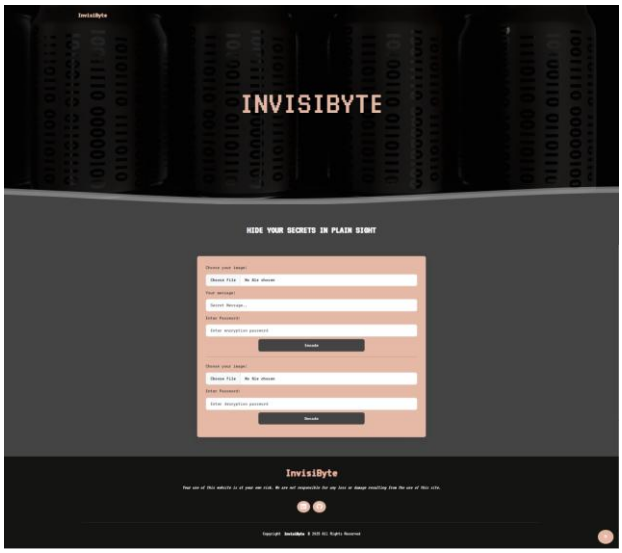
d. Aspek Ketahanan
Untuk menguji seberapa tangguh data tersembunyi terhadap manipulasi sederhana, dilakukan percobaan pemotongan sebagian gambar (*cropping*) pada gambar stego. Setelah itu, fitur *decode* digunakan untuk memeriksa apakah pesan masih dapat diambil secara utuh atau mengalami kerusakan. Hasil dari pengujian ini memberikan gambaran sejauh mana integritas data dapat dipertahankan apabila media gambar mengalami perubahan.

Melalui pemaparan metode penelitian ini, mulai dari alur penelitian, pendekatan pengembangan sistem, arsitektur, fitur aplikasi, hingga metode pengujian, diharapkan seluruh tahapan yang telah dirancang dapat mendukung proses pengembangan aplikasi steganografi InvisiByte secara optimal. Dengan pendekatan yang terstruktur dan pengujian yang komprehensif, penelitian ini diharapkan mampu menghasilkan solusi yang tidak hanya fungsional, tetapi juga aman dan relevan dengan kebutuhan perlindungan data di era digital

3. HASIL DAN PEMBAHASAN

3.1 Antarmuka Aplikasi

Pengembangan aplikasi web InvisiByte menghasilkan antarmuka yang mendukung proses penyisipan dan pengambilan pesan rahasia. Antarmuka aplikasi web InvisiByte dirancang secara intuitif dengan tema gelap yang mendukung tujuan aplikasi untuk menjaga keamanan data. Pada bagian atas halaman utama, terdapat logo InvisiByte dengan latar belakang berpola biner, diikuti oleh *tagline* "*Hide your secrets in plain sight*" yang mencerminkan fungsi utama aplikasi.

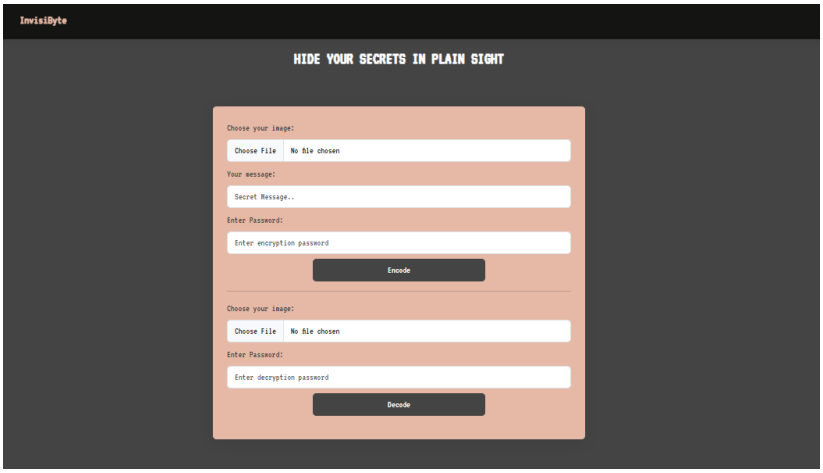


Gambar 2. Antarmuka aplikasi web InvisiByte

Antarmuka utama terbagi menjadi dua bagian utama: fitur *encode* dan *decode*. Pada fitur *encode*, pengguna dapat mengunggah gambar melalui tombol *Choose your image*,

memasukkan pesan rahasia pada kolom *Secret Message*, dan menambahkan kata sandi pada kolom *Enter Password* untuk mengenkripsi pesan sebelum disisipkan. Setelah semua input diisi, tombol *Encode* yang berwarna abu-abu tua akan memulai proses penyisipan dan gambar stego akan otomatis terunduh.

Sementara itu, fitur *decode* berfungsi ketika pengguna mengunggah gambar stego melalui tombol *Choose your image*, memasukkan kata sandi dekripsi pada kolom *Enter Password*, dan menekan tombol *Decode* untuk mengekstrak pesan.



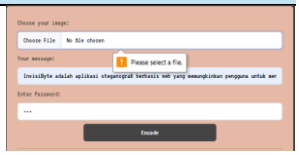
Gambar 3. Fitur Encode dan Decode pada InvisiByte

3.2 Pengujian Aplikasi

Pengujian dengan *white-box* dilakukan untuk memastikan bahwa aspek fungsionalitas dan alur logika dalam kode program berjalan sesuai dengan skenario yang telah ditentukan. Hasil dari pengujian tersebut disajikan pada tabel berikut:

Tabel 1. Fitur *Encode*

Skenario pengujian	Kode yang diuji	Hasil yang diharapkan	Hasil aktual

Pengguna tidak mengunggah gambar	HTML
	`<div class="mb-2">`
	`<label for="encodeImage" class="form-label">Choose your image:</label>`
	`<input type="file" id="encodeImage" name="image" class="form-control" required>`
	`</div>`
	LARAVEL
	`$request->validate([`
	`'image' => 'required|image|mimes:png,jpg,jpeg',`
	`'message' => 'required|string',`
	`'password' => 'required|string|min:4',`
	`]);`
		Muncul pesan "*please select a file*"	
			Sesuai

Pengguna memasukkan file yang bukan png, jpg dan jpeg	HTML
	`<div class="mb-2">`
	`<label for="encodeImage" class="form-label">Choose your image:</label>`
	`<input type="file" id="encodeImage" name="image" class="form-control" required accept=".jpg,.jpeg,.png">`
	`</div>`
	LARAVEL
	`$request->validate([`
	`'image' => 'required|image|mimes:png,jpg,jpeg',`
	`'message' => 'required|string',`
	`'password' => 'required|string|min:4',`
	`]);`
		User tidak bisa menginput file yang bukan png, jpg dan jpeg	Sesuai: Pengguna tidak bisa memilih file yang bukan png, jpg dan jpeg

Skenario pengujian	Kode yang diuji	Hasil yang diharapkan	Hasil aktual

Pengguna tidak memasukkan pesan	HTML
	`<div class="mb-2">`
	`  <label for="message" class="form-label">Your message:</label>`
	`  <input type="text" id="message" name="message" class="form-control"`
	`  placeholder="Secret Message.." required>`
	`</div>`
	LARAVEL
	`$request->validate([`
	`  'image' => 'required|image|mimes:png,jpg,jpeg',`
	`  'message' => 'required|string',`
	`  'password' => 'required|string|min:4',`
	`]);`
		Muncul pesan "*please fill out this field*"	
			Sesuai

Pengguna tidak memasukkan password	HTML
	`<div class="mb-2">`
	`  <label for="encodePassword" class="form-label">Enter Password:</label>`
	`  <input type="password" id="encodePassword" name="password"`
	`  class="form-control" placeholder="Enter encryption password" required>`
	`</div>`
	JAVASCRIPT
	`let password = document.getElementById("encodePassword").value.trim();`
	`if (!password) {`
	`  alert("Password harus diisi!");`
	`  return;`
	`}`
	LARAVEL
	`$request->validate([`
	`  'image' => 'required|image|mimes:png,jpg,jpeg',`
	`  'message' => 'required|string',`
	`  'password' => 'required|string|min:4',`
	`]);`
		Muncul pesan "*please fill out this field*"	
			Sesuai

Pengguna menekan tombol *encode*, kemudian gambar terunduh otomatis	JAVASCRIPT
	`.then(response => response.json())`
	`.then(data => {`
	`  if (data.success) {`
	`  let a = document.createElement('a');`
	`  a.href = data.downloadUrl;`
	`  a.download = '';`
	`  document.body.appendChild(a);`
	`  a.click();`
	`  document.body.removeChild(a);`
		Gambar stego terunduh otomatis	
			Sesuai

Pengguna menekan tombol *encode*, sistem melakukan *reload* dan menampilkan pesan sukses	JAVASCRIPT
	`  setTimeout(() => {`
	`  window.location.href = '/';`
	`  }, 1000);`
	`  } else {`
	`  alert(data.error || "Terjadi kesalahan saat encoding.");`
	`  }`
	LARAVEL
	`session()->flash('success', 'The message has been successfully encoded and downloaded.');`
	HTML
	`@if (session('success'))`
	`  <div class="alert alert-success">`
	`  {{ session('success') }}`
	`  </div>`
	`@endif`
		Sistem reload dan menampilkan pesan "*The message has been successfully encoded and downloaded.*"	
			Sesuai

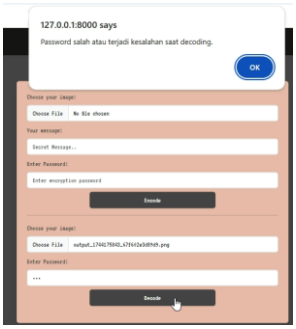
Tabel 2. Fitur Decode

Skenario pengujian	Kode yang diuji	Hasil yang diharapkan	Hasil aktual

Pengguna tidak mengunggah gambar	HTML
	`<div class="mb-2">`
	`<label for="decodeImage" class="form-label">Choose your image:</label>`
	`<input type="file" id="decodeImage" name="image" class="form-control" required>`
	`</div>`
	LARAVEL
	`$request->validate([`
	`'image' => 'required|image|mimes:png,jpg,jpeg',`
	`'password' => 'required|string|min:4',`
	`]);`
		Muncul pesan "*please select a file*"	
			Sesuai

Pengguna tidak memasukkan password	HTML
	`<div class="mb-2">`
	`<label for="decodePassword" class="form-label">Enter Password:</label>`
	`<input type="password" id="decodePassword" name="password" class="form-control" placeholder="Enter decryption password" required>`
	`</div>`
	JAVASCRIPT
	`let password = document.getElementById("decodePassword").value.trim();`
	`if (!password) {`
	`alert("Password harus diisi!");`
	`return;`
	`}`
	LARAVEL
	`$request->validate([`
	`'image' => 'required|image|mimes:png,jpg,jpeg',`
	`'password' => 'required|string|min:4',`
	`]);`
		Muncul pesan "*please fill out this field*"	
			Sesuai

Pengguna memasukkan file yang bukan png, jpg dan jpeg	HTML
	`<div class="mb-2">`
	`<label for="decodeImage" class="form-label">Choose your image:</label>`
	`<input type="file" id="decodeImage" name="image" class="form-control" required accept=".jpg,.jpeg,.png">`
	`</div>`
	LARAVEL
	`$request->validate([`
	`'image' => 'required|image|mimes:png,jpg,jpeg',`
	`'password' => 'required|string|min:4',`
	`]);`
		User tidak bisa menginput file yang bukan png, jpg dan jpeg	Sesuai: Pengguna tidak bisa memilih file yang bukan png, jpg dan jpeg

Pengguna memasukkan password yang salah	JAVASCRIPT
	`.then(response => response.json())`
	`.then(data => {`
	`if (data.success) {`
	`document.getElementById('decodedMessage').innerHTML = "Pesan Rahasia: " + data.message;`
	`} else {`
	`alert(data.error || "Password salah atau terjadi kesalahan saat decoding.");`
	`}`
	`})`
	`.catch(error => {`
	`console.error('Error:', error);`
	`alert("Password salah atau terjadi kesalahan saat decoding.");`
	`});`
		Muncul *alert* "Password salah atau terjadi kesalahan saat decoding."	
			Sesuai

Pengguna menekan tombol *decode*, kemudian sistem menampilkan pesan rahasia	HTML
	`</form>`
	`<div id="decodedMessage" class="mt-3 text-center" style="font-weight: bold;"></div>`
	`</div>`
	JAVASCRIPT
	`.then(response => response.json())`
	`.then(data => {`
	`if (data.success) {`
	`document.getElementById('decodedMessage').innerHTML = "Pesan Rahasia: " + data.message;`
	`} else {`
		Sistem menampilkan pesan rahasia	
			Sesuai

3.3 Pengujian Gambar

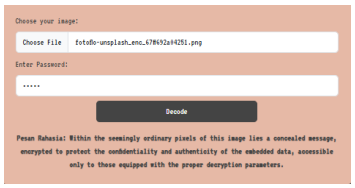
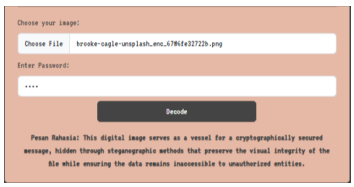
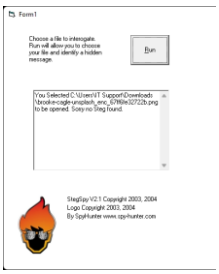
Pada tahap ini dilakukan pengujian dan perbandingan antara gambar asli dan gambar stego untuk mengevaluasi sejauh mana proses penyisipan pesan memengaruhi kualitas visual serta karakteristik teknis dari citra. Gambar stego merupakan hasil dari proses encoding, di mana pesan rahasia disisipkan ke dalam gambar asli menggunakan metode steganografi berbasis Least Significant Bit (LSB).

Pengujian mencakup beberapa aspek utama. Dari segi keamanan, dilakukan uji deteksi menggunakan aplikasi pihak ketiga, StegSpy, untuk menilai apakah pesan yang disisipkan dapat terdeteksi oleh pihak yang tidak berwenang. Jika gambar stego tidak terdeteksi mengandung pesan oleh aplikasi tersebut, maka teknik penyisipan dianggap cukup aman dari serangan steganalisis berbasis deteksi pola.

Selanjutnya, pengujian efektivitas dilakukan melalui analisis terhadap performa aplikasi ketika digunakan dengan gambar berukuran berbeda dan pesan dengan panjang bervariasi. Evaluasi mencakup sejauh mana sistem mampu mempertahankan kualitas visual serta kestabilan fungsi saat menangani data tersembunyi dalam jumlah yang lebih besar. Pengujian juga mencakup perbandingan ukuran file dan resolusi gambar sebelum dan sesudah penyisipan pesan untuk menilai efisiensi proses encoding secara teknis.

Sementara itu, aspek ketahanan diuji melalui simulasi manipulasi terhadap gambar stego, yaitu dengan melakukan *cropping* (pemotongan sebagian area gambar). Setelah manipulasi dilakukan, fitur decode digunakan untuk memeriksa apakah pesan rahasia masih dapat diambil secara utuh atau mengalami kerusakan. Hasil dari pengujian ini memberikan gambaran mengenai seberapa baik integritas pesan dapat dipertahankan apabila media gambar mengalami gangguan atau perubahan.

Tabel 3. Perbandingan gambar asli dan stego serta hasil deteksi pesan oleh aplikasi StegSpy.

Gambar Asli	Gambar Stego	Pesan Rahasia	Steganalisis
 tyler-lastovich-unsplash.png	 tyler-lastovich-unsplash_enc_67f7446611a89.png	 Panjang karakter 106	 Steganografi tidak terdeteksi oleh aplikasi StegSpy
 fotoflo-unsplash.png	 fotoflo-unsplash_enc_67ff692a04251.png	 Panjang karakter 231	 Steganografi tidak terdeteksi oleh aplikasi StegSpy
 v2osk-unsplash.png	 v2osk-unsplash_enc_67ff6cc01bc52.png	 Panjang karakter 10	 Steganografi tidak terdeteksi oleh aplikasi StegSpy
 brooke-cagle-unsplash.png	 brooke-cagle-unsplash_enc_67ff6fe32722b.png	 Panjang karakter 238	 Steganografi tidak terdeteksi oleh aplikasi StegSpy

Tabel 4. Tabel perbandingan ukuran gambar asli dan gambar stego

Nama Gambar	Resolusi		Ukuran File		Perubahan Ukuran
	Gambar Asli	Gambar Stego	Gambar Asli	Gambar Stego	
tyler-lastovich-unsplash.png	640x400	640x400	22.3kb	84.2kb	3.78x
fotoflo-unsplash.png	640x424	640x424	25.3kb	147kb	5.8x
v2osk-unsplash.png	640x381	640x381	60.1kb	349kb	5.8x
brooke-cagle-unsplash.png	640x427	640x427	61.2kb	345kb	5.6x

Berdasarkan Tabel 3, dapat dilihat bahwa seluruh gambar stego yang telah disisipi pesan rahasia dengan panjang karakter bervariasi tetap tidak terdeteksi sebagai steganografi oleh aplikasi StegSpy. Hal ini menunjukkan bahwa metode penyisipan pesan yang digunakan pada aplikasi ini mampu menyembunyikan informasi dengan cukup baik tanpa mengindikasikan adanya anomali visual atau *metadata* yang mudah dikenali oleh *tool* deteksi steganografi.

Sementara itu, pada Tabel 4, terlihat adanya peningkatan ukuran file yang signifikan setelah penyisipan pesan, meskipun resolusi gambar tidak mengalami perubahan, peningkatan ukuran ini kemungkinan besar disebabkan oleh proses *encoding* pesan ke dalam bit-bit piksel yang mengubah struktur kompresi gambar, terutama pada format PNG yang bersifat *lossless*. Namun demikian, meskipun terjadi peningkatan ukuran file, hal tersebut tidak menyebabkan terdeteksinya steganografi oleh aplikasi StegSpy.

Tabel 5. Pengujian ketahanan gambar terhadap *cropping*

Gambar Asli	Gambar Stego Hasil Cropping	Keterangan
	 Crop 30%	Pesan gagal diekstrak
	 Crop 10%	Pesan gagal diekstrak
	 Crop 5%	Pesan gagal diekstrak

Pesan gagal diekstrak

Crop 2%

Pengujian ketahanan gambar terhadap manipulasi *cropping* dilakukan dengan memotong gambar menggunakan berbagai proporsi. Setiap gambar telah disisipkan pesan dengan panjang karakter yang sama, yaitu 100 karakter, menggunakan kalimat: *"This embedded message is a test to evaluate the resilience of steganography against cropping"*

Berdasarkan hasil pengujian yang telah dilakukan, ditemukan bahwa proses pemotongan (*cropping*) terhadap gambar stego berdampak langsung pada kemampuan sistem untuk mengekstraksi pesan yang telah disisipkan. Pada seluruh variasi proporsi pemotongan yang diuji, tidak satu pun dari gambar hasil *cropping* yang berhasil menghasilkan pesan tersembunyi saat dilakukan proses *decoding*. Hal ini menunjukkan bahwa metode steganografi yang digunakan dalam aplikasi belum memiliki ketahanan yang baik terhadap manipulasi jenis ini, sehingga kehilangan sebagian informasi piksel akibat pemotongan dapat menyebabkan rusaknya struktur data tersembunyi.

4. KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan hasil penelitian dan pengembangan yang telah dilakukan, dapat disimpulkan bahwa aplikasi web steganografi yang dikembangkan berhasil menyisipkan dan mengekstrak pesan rahasia ke dalam sebuah citra digital. Aplikasi ini menggabungkan metode steganografi berbasis *Least Significant Bit* (LSB) dengan kriptografi *Advanced Encryption Standard* (AES) sebagai lapisan keamanan tambahan, yang memberikan perlindungan terhadap isi pesan apabila gambar berhasil dianalisis oleh pihak yang tidak berwenang.

Pengujian yang dilakukan menunjukkan bahwa pesan dapat disisipkan tanpa menimbulkan perubahan visual yang signifikan pada gambar stego, sehingga tidak mudah dikenali secara kasat mata. Namun, proses penyisipan tetap menghasilkan peningkatan ukuran file yang cukup terlihat, khususnya pada gambar dengan warna kompleks. Tingkat keamanan metode yang digunakan cukup baik karena pesan rahasia dienkripsi terlebih dahulu sebelum disisipkan, sehingga meskipun berhasil diambil oleh pihak ketiga, isi pesan tetap tidak dapat dibaca tanpa kunci enkripsi yang sesuai.

4.2 Saran

Untuk pengembangan lebih lanjut, disarankan agar aplikasi ini dapat dilengkapi dengan teknologi berbasis kecerdasan buatan (AI) untuk mendeteksi kualitas gambar yang ideal sebelum penyisipan pesan, serta membantu menganalisis kemungkinan deteksi oleh alat steganalisis. Selain itu, penggunaan metode steganografi lain seperti *Discrete Cosine Transform* (DCT), *Discrete Wavelet Transform* (DWT), atau *spread spectrum* dapat dipertimbangkan untuk meningkatkan ketahanan terhadap manipulasi seperti *cropping*, kompresi, atau pengubahan resolusi. Di sisi kriptografi, penggabungan metode hybrid seperti AES-RSA dapat meningkatkan keamanan sistem secara menyeluruh.

Terakhir, aplikasi ini memiliki potensi besar untuk diterapkan dalam dunia nyata, terutama dalam bidang komunikasi rahasia, perlindungan dokumen penting, dan sistem keamanan data digital. Oleh karena itu, pengembangan berkelanjutan dan pengujian pada skala lebih besar sangat disarankan untuk memastikan aplikasi ini dapat diandalkan dalam kondisi nyata.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada Ibu Wahyu Noviani Purwanti, S.Si., M.Si., selaku dosen pembimbing, atas bimbingan, arahan, serta dukungan yang telah diberikan selama proses penyusunan dan pelaksanaan penelitian ini. Bantuan dan masukan yang diberikan sangat berarti dalam menyempurnakan hasil penelitian yang penulis lakukan.

DAFTAR PUSTAKA

Adhimah, L. F., Nurhafiyah, I., Muntahar, A. A., Kristiaji, F., & Mustofa, D. (2023). Implementasi aplikasi steganografi berbasis web menggunakan algoritma LSB dan BPCS. *Komputa*: *Jurnal Ilmiah Komputer dan Informatika*, *12*(2), 100-108. Doi: https://doi.org/10.34010/komputa.v12i2.10319

Anwar, S. (2017). Implementasi pengamanan data dan informasi dengan metode steganografi lsb dan algoritma kriptografi aes. *Jurnal Format, 6*(1), 65-74. http://neliti.com/id/publications/224678/implementasi-pengamanan-data-daninformasi-dengan-metode-steganografi-lsb-dan-al

Hidayatulloh, N. W., Tahir, M., Amalia, H., Basyar, N. A., Prianggara, A. F., & Yasin, M. (2023). Mengenal advance encryption standard (aes) sebagai algoritma kriptografi dalam mengamankan data. *Digital Transformation Technology, 3*(1), 1–10. https://www.jurnal.itscience.org/index.php/digitech/article/view/2293

Junianto, M. B. S., & Ardiansyah, H. (2020). Analisa Dan Perancangan Sistem Informasi Pengaman Dokumen Dengan Metode Algoritma Xor Dan Aes Berbasis Web (Studi Kasus: Bimbingan Belajar Matriks Pamulang). *Joaiia J*, *1*(2), 61-66.

Permana, F. R., Setiyanto, R. F., Fauzi, A. R. (2023). Image steganography dengan menggunakan metode lsb pada python. Triple a: *Jurnal Pendidikan Teknologi Informasi*, *2*(1), 1-7. https://jurnal.umj.ac.id/index.php/TripleA/article/view/17825

Putra, I. G. N. B. P., Suhartana, I. K. G., Mogi, I. K. A., Pramartha, C. R. A., Suputra, I. P. G. H., & Wibawa, I. G. A. (2022). Penerapan steganography untuk perlindungan hak cipta menggunakan metode least significant bit (lsb). *Jurnal Elektronik Ilmu Komputer Udayana, 10*(4), 329-340. https://ojs.unud.ac.id/index.php/jlk/article/download/86055/44836

Putri, M. P., Effendi, H. (2018). Implementasi metode rapid application development pada website service guide “waterfall tour south Sumatera”. *Jurnal Sisfokom, 7*(2), 130-136. https://media.neliti.com/media/publications/265635-implementasi-metode-rad-pada-website-ser-fa285f1d.pdf

Rahmatillah, S. R., Tahir, M., Detina, H. I. L., Darmasaputra, A., Laili, I. I., Aliy, Z., & Soleha, R. (2024). Steganografi: keamanan data dengan metode least significant bit menggunakan python. *Jurnal Riset Sistem Informasi Dan Teknologi Informasi* (*JURSISTEKNI*), *6*(2), 439-447. https://jursistekni.nusaputra.ac.id/article/view/327

Sabrina, F. N. (2021). Aplikasi steganografi pada media gambar menggunakan algoritma least significant bit. *Jurnal Tera*, *1*(2), 185–201. https://jurnal.undira.ac.id/jurnaltera/article/view/55

Sinlae, F., Irwanda, E., Maulana, Z., Syahputra, V. E. (2024). Penggunaan framework Laravel dalam membangun aplikasi website berbasis php. *Jurnal Siber Multi Disiplin, 2*(2), 119-132. Doi: https://doi.org/10.38035/jsmd.v2i2.186

Sunardi., Riadi, I., Akbar, M. H. (2020). Steganalisis bukti digital pada media penyimpanan menggunakan metode static forensics. *Jurnal Nasional Teknologi dan Sistem Informasi, 6*(1), 1-8. Doi: https://doi.org/10.25077/TEKNOSI.v6i1.2020.1-8

Syawal, M. F., Fikriansyah, D. C., Agani, N. (2016). Implementasi teknik steganografi menggunakan algoritma vigenere cipher. Jurnal TICOM, 4(3), 91-99.

https://media.neliti.com/media/publications/93707-ID-implementasi-tekniksteganografi-menggun.pdf

Wijaya, G. E. P., Kadnyanan, I. G. A. G. A. (2025). Penerapan steganografi untuk pengamanan konten gambar dalam media sosial. *Jurnal Nasional Teknologi Informasi dan Aplikasinya, 3*(2), 337-342. https://ojs.unud.ac.id/index.php/jnatia/article/download/116064/58099