Jurnal TICOM: Technology of Information and Communication
Volume 11, Nomor 2, Januari 2023 Hal: 113-118 E-ISSN: 2962-7982

Implementasi Keamanan File Menggunakan Metode Kriptografi Base-64 dan Steganografi Least Significant Bit (LSB) Random 2-Bit Berbasis Web

Agnes Aryasanti[1], Ratna Ujiandari[2*], Mardi Hardjianto[3], Ratna Kusumawardani[4]

[1,2,4]Fakultas Teknologi Informasi, Sistem Informasi, Universitas Budi Luhur, Jakarta, Indonesia
[3]Fakultas Teknologi Informasi ,Teknik Informatika, Universitas Budi Luhur, Jakarta, Indonesia
Jl. Raya Ciledug, Petukangan Utara, Jakarta Selatan 12260
[1]agnes.aryasanti@budiluhur.ac.id, [2*]ratna.ujiandari@budiluhur.ac.id,
[3]mardi.hardjianto@budiluhur.ac.id, [4]ratna.kusumawardani@budiluhur.ac.id
(*: corresponding author)

Abstrak— **Keamanan dan kerahasiaan adalah dua aspek penting dari transmisi data dalam saluran komunikasi. Dalam pengiriman yang bersifat data rahasia, pengirim menginginkan data tersebut terkirim ke tujuan tanpa ada orang lain yang mengetahuinya. Salah satu cara agar data rahasia agar tidak dapat dibaca oleh orang yang tidak berwenang adalah menggunakan steganografi. Teknik steganografi menyembunyikan data rahasia ke dalam suatu obyek (*cover object*). Media yang digunakan untuk *cover object* adalah citra digital dengan kedalaman warna 24-bit. Metode steganografi yang digunakan untuk mengamankan data adalah *Least Significant Bit* (LSB) *Random* yang dikombinasikan dengan m-Bit. Selain menggunakan steganografi, pengamanan data ini juga menggunakan algoritme enkripsi Base-64 yang sudah dimodifikasi. Penggunaan kombinasi metode kriptografi dan steganografi dapat meningkatkan keamanan data. *Stego object* yang dihasilkan jika dibandingkan dengan *cover object* hampir tidak ada bedanya, karena nilai PNSR lebih dari 30.**

Kata Kunci— **Kriptografi; steganografi; Base-64; LSB Random; LSB m-Bit.**

Abstract— ***Security and confidentiality are two important aspects of data transmission in communication channels. In sending confidential data, the sender wants the data to be sent to the destination without anyone knowing about it. One way to make confidential data unreadable by unauthorized persons is to use steganography. Steganography techniques hide secret data into an object (cover object). The media used for the cover object is a digital image with a color depth of 24 bits. The steganographic method used to secure data is Least Significant Bit (LSB) Random combined with m-Bit. Apart from using steganography, this data security also uses a modified BASE-64 encryption algorithm. The combined use of cryptographic and steganographic methods can increase data security. The stego object produced when compared to the cover object is almost no different, because the PNSR value is more than 30.***

Keywords— ***cryptography; steganography; Base-64; LSB Random; LSB m-Bit***

I. PENDAHULUAN

Dalam teknologi komunikasi saat ini, keamanan dan kerahasiaan informasi merupakan hal yang sangat penting. Sepenggal informasi bila menyangkut aspek keamanan, pribadi, kepentingan publik atau keputusan bisnis, akan dinilai berharga dan lebih tinggi.

Saat ini, lalu lintas pengiriman pesan semakin pesat sesuai dengan perkembangan dunia digital. Pertukaran data atau informasi sudah semakin mudah dilakukan dengan adanya teknologi internet yang bisa dilakukan tanpa adanya media fisik. Hal ini mengakibatkan faktor keamanan dan privasi menjadi hal yang penting dalam berkomunikasi melalui jaringan internet.

Perkembangan teknologi informasi ini juga memicu perkembangan teknik kejahatan. Berbagai cara dapat dilakukan untuk mengakses informasi yang bukan milikinya secara ilegal. Hal ini menyebabkan perlunya upaya-upaya untuk mengamankan dan merahasiakan data dan informasi yang dikirim. Berbagai macam cara digunakan untuk mengamankan data penting, diantaranya adalah kriptografi. Walaupun penggunaan kriptografi sudah cukup aman, namun pesan yang disandikan masih tetap terlihat.

Selain kriptografi, ada metode lain yang sering digunakan untuk mengamankan data, yaitu steganografi. Metode steganografi dapat digunakan untuk menyembunyikan data rahasia ke dalam suatu wadah sehingga keberadaan data rahasia sulit dideteksi. Wadah yang sering digunakan untuk menyembunyikan data rahasia adalah file *image*, *audio* atau video. Penggunaan metode steganografi sering kali digabung dengan kriptografi untuk meningkatkan keamanan.

Banyak metode yang digunakan untuk steganografi, salah satu yang sering digunakan adalah *Least Significant Bit* (LSB). Metode LSB sendiri mempunyai banyak varian, antara lain LSB *Sequential*, *Random* dan M-Bit. Metode LSB *Sequential* menyembunyikan data rahasia secara *sequential*, artinya satu piksel untuk menyimpan satu bit data rahasia. Metode LSB *Random* menyembunyikan data rahasia pada lokasi piksel secara *random*. Metode M-Bit memanfaatkan beberapa bit untuk penyimpanan data rahasia.

Penelitian sebelumnya yang dilakukan oleh Gilbijatno, berhasil membuat sistem keamanan pesan menggunakan teknik Steganografi dengan metode *End of File* [1]. Aryasanti berhasil membuat aplikasi mengamankan file soal ujian akhir, metode yang digunakan Steganografi LSB dan Kompresi dengan

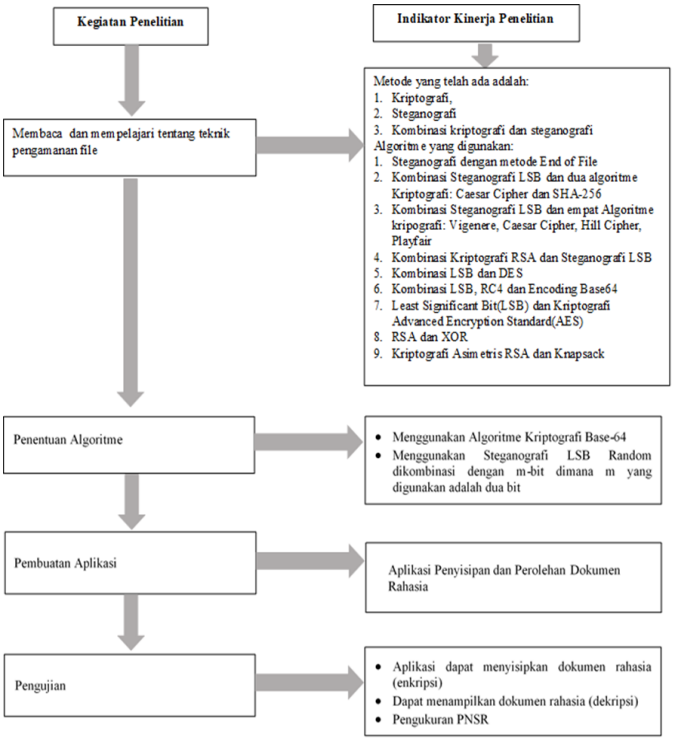
menambahkan teknik *error detection* menggunakan *hash function* untuk menjaga keabsahan file *stego image*[2]. Aplikasi pengamanan pesan rahasia menggunakan kombinasi Steganografi LSB, Kriptografi Caesar Cipher dan SHA-26 berhasil menyembunyikan pesan dengan keamanan berlapis [3]. Penelitian yang dilakukan oleh Frobenius, berhasil mengamankan informasi dengan menggabungkan Steganografi LSB dan 4 algoritme Kripografi yaitu Vigenere, Caesar Cipher, Hill Cipher, Playfair[4]. Penyembunyian pesan rahasia menggunakan algoritme Vigenere Cipher dengan Hill Cipher berhasil dilakukan dengan hasil nilai rata -rata *Avalanche Effect* 52,46%[5]. Penggabungan algoritme Kriptografi Rivest Shamir Adleman (RSA) dan algoritme Steganografi LSB untuk mengamankan pesan rahasia berhasil direkontruksi dengan baik[6] dan mampu mengamankan data [7]. Kombinasi Steganografi LSB dan Kriptografi DES pada penyisipan pesan menghasilkan nilai rata-rata PNSR lebih dari standar yang ditentukan yaitu 76.01882 dB[8]. Penggabungan teknik steganografi LSB, Kriptografi RC4 dan base 64 pada penyisipan pesan rahasia di *cover image*, terbukti aman dan tidak terlihat perbedaan secara kasat mata. Proses pengembalian pesan rahasia berhasil dijalankan pada proses *decode*. Terdapat beberapa kegagalan dalam proses pengujian setelah *stego image* dikirimkan melalui e-mail, media sosial dan aplikasi pengiriman pesan [9]. Kombinasi teknik kriptografi *Advanced Encrytion Standard*(AES) dan Stegano LSB pada aplikasi android dapat berjalan dalam mengenkripsi dan dekripsi pesan yang disembunyikan ke dalam file gambar [10]. Penerapan algoritme RSA dan algoritme XOR pada proses enkripsi dan dekripsi menghasilkan Rata-rata MSE 0,8768, dan PSNR sekitar 50,1588 [11]. Penerapan algoritme Kriptografi asimetris RSA dan Knapsack pada sistem keamanan dan kerahasiaan data terbukti mampu mengatasi masalah penyalahgunaan, pencurian maupun korupsi data[12]. Base64 yang menggunakan kunci akan mempersulit proses dekripsinya bila dibandingkan dengan proses standar pada algoritme Base64 itu sendiri[13]. Pengenkripsian alamat URL pada sebuah website dapat mengunakan kombinasi metode Base64 dan ROT13[14] .

Pada penelitian ini, kami menggunakan metode kriptografi dan steganografi untuk mengamankan data rahasia. Algoritme kriptografi yang kami pilih adalah Base-64 yang telah dimodifikasi. Pada penelitian sebelumnya penggunakan Base-64 masih bersifat standar. Hal ini dapat dengan mudah untuk dilakukan dekripsi. Modifikasi yang kami lakukan adalah merubah tabel Base-64 dan menambah penggunaan kata kunci untuk menentukan *ciphertext*-nya. Steganografi yang digunakan adalah LSB *Random* yang dikombinasi dengan m-bit, dimana m yang digunakan adalah dua bit. LSB *Random* berarti penyisipan data rahasia dilakukan ke lokasi secara acak/ *random*, tidak secara berurutan. Lokasi acak ini dibangkitkan dengan Pseudo *Random Number Generator* (PRNG). Penelitian ini juga mencari nilai *Peak Signal to Noise Ratio* (PSNR) yang dihasilkan untuk mengetahui apakah *stego image* yang dihasilkan memiliki kualitas yang baik atau tidak. Sistem yang dibangun untuk mengamankan data rahasia ini menggunakan berbasis web.

II. METODE PENELITIAN

A. *Langkah-langkah Penelitian*

Pada langkah awal, kami mempelajari teknik pengamanan data pada penelitian-penelitian terdahulu. Dari hasil mempelajari teknik pengamanan data, diperoleh bahwa peneliti-peneliti terdahulu menggunakan kriptografi atau steganografi. Ada pula yang menggabungkan kriptografi dan steganografi agar lebih kuat dalam pengamanannya. Langkah selanjutnya kami menentukan metode pengamanan, yaitu menggunakan steganografi yang dikombinasi dengan kriptografi. Dokumen rahasia yang mau disembunyikan, dienkripsi terlebih dahulu dengan metode Base-64 yang telah dimodifikasi. Metode steganografi yang digunakan adalah *Least Significant Bit Random* yang dikombinasi dengan m-Bit dimana m menggunakan dua bit. Setelah aplikasi selesai dibuat, maka perlu dilakukan pengujian. Pengujian dilakukan dengan melakukan percobaan menyembunyikan dokumen rahasia ke dalam media *cover object*, lalu diperoleh kembali dokumen rahasia tersebut. Pengujian juga mengukur nilai *Peak Signal to Noise Ratio* (PSNR) untuk mengetahui apakah kualitas file *stego object* masih baik atau tidak. Langkah-langkah penelitian dalam penelitian ini ditunjukkan pada Gambar 1.

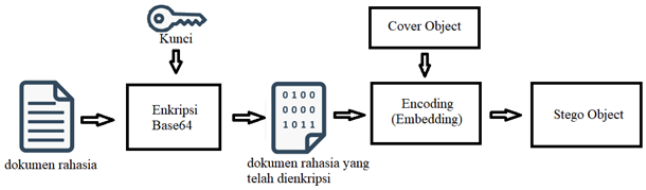


Gambar 1. Langkah-langkah Penelitian

B. *Proses Penyisipan Dokumen Rahasia*

Masukan pada proses penyisipan dokumen rahasia terdiri dari dua file, yaitu file dokumen rahasia yang akan disembunyikan dan file *cover object* sebagai media penampungan. Proses pengamanan ini juga membutuhkan kunci rahasia. Dokumen rahasia yang akan disembunyikan dapat dalam format apa saja, sedangkan *cover object* yang digunakan adalah file gambar dengan format .bmp dan .png.

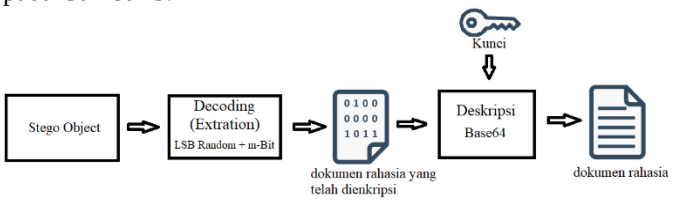
Langkah awal yang dilakukan adalah *user* memilih file dokumen rahasia yang akan disembunyikan, file gambar sebagai *cover object* serta memasukan kunci rahasia. Proses berikutnya adalah membaca file dokumen rahasia dan melakukan enkripsi dengan metode base-64 yang sudah dimodifikasi. Modifikasi dilakukan dengan mengubah isi tabel Base64 dan menambahkan kunci rahasia dalam menentukan karakter di tabel Base64. Proses penyisipan dokumen rahasia ditunjukkan pada Gambar 2.



Gambar 2. Penyisipan Dokumen Rahasia

C. *Proses Pengambilan Dokumen*

Proses pengambilan dokumen membutuhkan dua masukan yaitu file gambar yang berisi pesan rahasia (*stego object*) dan kata kunci keamanan. *User* memasukkan file gambar yang berisi pesan rahasia, lalu memasukkan kata kunci keamanan yang sama pada proses penyisipan dokumen. Selanjutnya program akan mengambil pesan rahasia dari file *stego object* dengan metode *Least Significant Bit Random* dikombinasi dengan m-bit dimana m yang digunakan adalah dua bit. Pesan rahasia yang diperoleh masih dalam bentuk terenkripsi dan harus dilakukan dekripsi dengan metode Base-64. Setelah dilakukan dekripsi, diperoleh pesan yang nantinya menjadi file dokumen rahasia. Proses pengambil dokumen ditunjukkan pada Gambar 3.



Gambar 3. Pengambilan Dokumen Rahasia

D. *Algoritme Base64*

Base64 adalah sistem penyandian dengan cara mentransformasi data yang berukuran 8-bit menjadi 6-bit. Karakter yag dihasilkan pada transforasi Base64 ini terdiri dari A s/d Z, a s/d z, 0 s/d 9 dan ditambah dengan karakter +, / serta = yang digunakan untuk mengenapi data *binary* hasil *encoding*. Pada penelitian ini, kami mengubah algoritme Base64 dengan menambahkan kunci rahasia dan mengganti tabel Base-64. Karakter yang digunakan untuk menggenap data *binary* hasil *encoding* adalah +. Adapun tabel Base-64 yang digunakan ditunjukkan pada Tabel I.
Berikut adalah algoritme Base64 *encoding* dan *decoding* hasil modifikasi.

```
Fungsi Encode64(plaintext, password)
Base64←“AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXx
YyZz9876543210=/+”
i ← 0
nkey ← -1
hasil ← ””
lenkey ← length(password)
while (i < length(plaintext))
   in[0] ← 0, in[1] ← 0, in{2] ← 0
   out[0] ← 0, out[1] ← 0, out[2] ← 0, out[3] ← 0
   len ← 0

   For j ← 0 to 2
      If i < length(plaintext) Then
         in[j] ← ord(substr(plaintext, i, 1))
         len ← len + 1
         i ← i + 1
      Endif
   Next

   nkey ← (nkey + 1) mod lenkey
   key ← ord(substr(password, nkey, 1))
   out[0] ← ((in[0] >> 2) + key) mod 64

   nkey ← (nkey + 1) mod lenkey
   key ← ord(substr(password, nkey, 1))
   out[1] ← ((((in[0] & 3) << 4) + (in[1] >> 4)) + key) mod 64

   If (len > 1) Then
      nkey ← (nkey + 1) mod lenkey
      key ← ord(substr(password, nkey, 1))
      out[2] ← ((((in[1] & 15) << 2) + (in[2] >> 6)) + key) mod 64
   Else
      out[2] ← 64
   Endif

   If (len > 2) Then
      nkey ← (nkey + 1) mod lenkey
      key ← ord(substr(password, nkey, 1))
      out[3] ← ((in[2] & 63) + key) mod 64
   Else
      out[3] ← 64
   Endif
   hasil ← hasil + substr($base, out[0], 1) + substr($base, out[1], 1) +
substr($base, out[2], 1) + substr($base, out[3], 1)
EndWhile
Return hasil


Fungsi decode64(ciphertext, password)
Base64←“AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXx
YyZz9876543210=/+”
i ← 0
nkey ← -1
hasil ← “”
lenkey ← length(password)
while (i < length(ciphertext))
   len ← 0
   in[0] ← 0, in[1] ← 0, in{2] ← 0, in[3] ← 0
   out[0] ← 0, out[1] ← 0, out[2] ← 0, out[3] ← 0

   For j ← 0 to 3
      If (i < strlen(ciphertext)) Then
         in[j] ← strpos(base, substr(ciphertext, i, 1))
         len ← len + 1
         i ← i + 1
      Endif
    Next

   nkey ← (nkey + 1) mod lenkey;
   xkey ← ord(substr(password, nkey, 1))
   out[0] ← (in[0] – xkey + 192) mod 64

   nkey ← (nkey + 1) mod lenkey
   xkey ← ord(substr(password, nkey, 1))
   out[1] ← (in[1] - xkey + 192) mod 64

   Iif (in[2] <> 64) Then
```

```
    nkey ← (nkey + 1) mod lenkey
    xkey ← ord(substr(password, nkey, 1))
    out[2] ← (int[2] - xkey + 192) mod 64
  Endif

  If (in[3] <> 64) Then
    nkey ← (nkey + 1) mod lenkey;
    xkey ← ord(substr(password, nkey, 1))
    out[3] ← (in[3] - xkey + 192) mod 64
  Endif

  c[0] ← (out[0] << 2) or (out[1] >> 4)
  c[1] ← (out[1] << 4) or (out[2] >> 2)
  c[2] ← (out[2] << 6) or out[3]

  hasil ← hasil + chr(c[0]) + chr(c[1]) + chr(c[2])
EndWhile
Return hasil
```

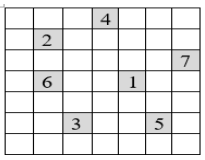
TABEL I
INDEX BASE64

Tabel Pengkodean Base64							
Nilai	Char	Nilai	Char	Nilai	Char	Nilai	Char
0	A	16	I	32	Q	48	Y
1	a	17	i	33	q	49	y
2	B	18	J	34	R	50	Z
3	b	19	j	35	r	51	z
4	C	20	K	36	S	52	9
5	c	21	k	37	s	53	8
6	D	22	L	38	T	54	7
7	d	23	l	39	t	55	6
8	E	24	M	40	U	56	5
9	e	25	m	41	u	57	4
10	F	26	N	42	V	58	3
11	f	27	n	43	v	59	2
12	G	28	O	44	W	60	1
13	g	29	o	45	w	61	0
14	H	30	P	46	X	62	=
15	h	31	p	47	x	63	/

E. *Algoritme LSB Random m-Bit*

LSB (*Least Significant Bit*) adalah bit yang berada pada posisi paling kanan dalam bilangan biner dan memiliki nilai paling rendah. Proses penyisipan LSB dengan cara memodifikasi bit paling kanan dalam satu byte data. Bila *cover object* berupa citra berwarna 24-bit, dimana setiap pixel mempunyai tiga warna utama (R, G, B) masing-masing berukuran 8-bit, penyisipan dapat dilakukan pada ketiga warna utama tersebut.

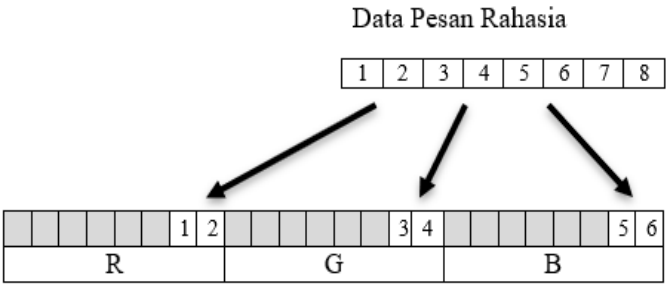
Ada beberapa varian dari LSB, diantaranya adalah LSB *Random* dan m-Bit. LSB *Random* menyembunyikan pesan pada lokasi pixel yang dipilih secara acak. Penentuan lokasi pixel menggunakan *Pseudo Randon Number Generator* (PRNG) yang harus mengikuti pendekatan tidak berulang, artinya tidak boleh ada pengulangan pixel yang sama dalam urutan pemilihan. Gambar 4 menunjukkan contoh tujuh pixel yang dipilih secara acak dari gambar yang berukuran 7x7 pixel.



Gambar 4. Pixel Yang Dipilih Secara Acak

Metode LSB m-Bit menggunakan m bit untuk menyimpan dokumen rahasia. Semakin besar nilai m yang digunakan, maka semakin banyak data rahasia yang dapat disimpan, namun kualitas gambar semakin tidak bagus.

Dalam penelitian ini, kami menggunakan metode LSB *Random* yang dikombinasi dengan m-Bit dimana m yang digunakan adalah dua. Proses penyisipan 2-bit data rahasia ditunjukkan pada Gambar 5.



Gambar 5. Proses Penyisipan 2-bit Data Rahasia

F. *Peak Signal to Noise Ratio (PSNR)*

Setelah proses penyisipan pada citra, kualitas citra perlu diketahui. Kualitas citra dapat diketahui melalui perhitungan *Peak Signal to Noise Ratio* (PSNR). Proses perhitungan PSNR dengan cara membandingkan citra sebelum dan sesudah disisipkan data rahasia apakah terdapat perubahan (*noise*) yang signifikan. Sebelum menentukan PSNR, terlebih dahulu menghitung nilai *Mean Square Error* (MSE). Nilai MSE adalah rata-rata kuadrat *error* dari citra asli (*cover object*) dengan citra hasil penyisipan (*stego object*). Perhitungan MSE menggunakan rumus:

$$MSE = \sqrt{\frac{1}{MN}\sum_{i=0}^{M}\sum_{j=0}^{N}(\mathrm{I}ij - \hat{\mathrm{I}}ij)^2}$$

Di mana
I = *Cover Object*
Î = *Stego Object*
ij = koordinat pixel
M = lebar citra dalam pixel
N = panjang citra dalam pixel

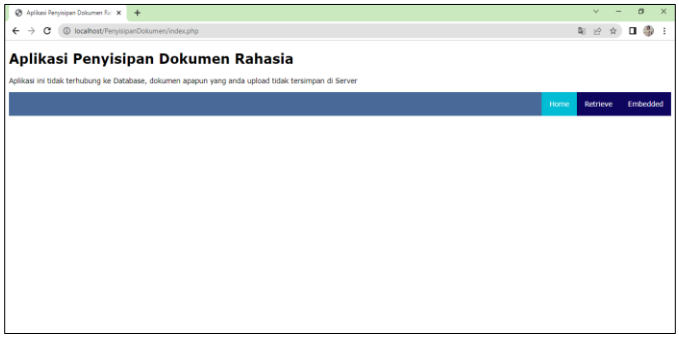
Rumus perhitungan PSNR sebagai berikut

$$PSNR = 20\,log10\left(\frac{255}{MSE}\right)$$

III. HASIL DAN PEMBAHASAN

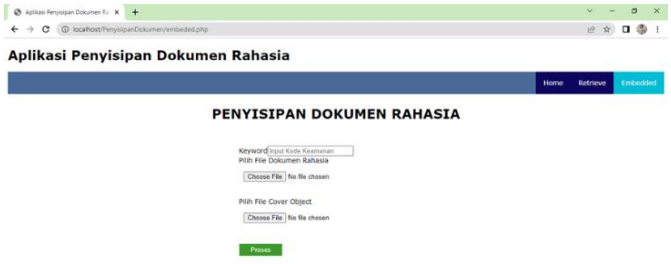
G. *Implementasi Aplikasi*

Implementasi aplikasi diawali dengan memunculkan halaman utama, seperti yang ditunjukkan pada Gambar 6.



Gambar 6. Tampilan Awal Aplikasi Penyisipan Dokumen Rahasia

Pada tampilan awal, terdapat dua pilihan submenu, yaitu submenu *Retrieve* dan *Embedded*. Submenu *Retrieve* digunakan untuk mengambil kembali dokumen rahasia dari file *stego object*, submenu *Embedded* digunakan untuk menyembunyikan dokumen rahasia ke file *cover object*. Bila submenu *Embedded* diklik, akan muncul halaman seperti ditunjukkan pada Gambar 7.



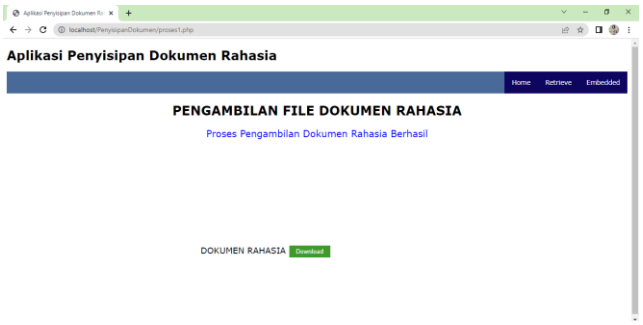
Gambar 7. Halaman Penyisipan Dokumen Rahasia

Pada halaman ini *user* diminta untuk memasukan kunci rahasia, file dokumen rahasia dan file *cover object*. Kunci rahasia harus diingat oleh *user* karena dibutuhkan pada saat pengambilan kembali dokumen rahasia. Bila kunci rahasia lupa, maka file dokumen rahasia tidak dapat diambil. File *cover object* yang dapat digunakan ada file citra yang berekstensi .PNG dan .BMP. Setelah file dokumen rahasia selesai disembunyikan ke file *cover object*, maka *user* dapat mendownload file *stego image*-nya. Gambar 8 menunjukkan Halaman *download* file *stego image*.

Gambar 8. Halaman Download File Stego Image

Pengembalian dokumen rahasia melalui submenu *Retrieve*. Halaman pengembalian dokumen rahasia ditunjukkan pada Gambar 9. Pada halaman ini, *user* diminta untuk memasukan kunci rahasia yang digunakan waktu penyisipan file dokumen rahasia dan file *stego image*. Aplikasi akan mengambil dokumen rahasia dari file *stego image*. Setelah dokumen rahasia berhasil diambil, *user* dapat men-*download* file tersebut. Halaman *download* file dokumen rahasia ditunjukkan pada Gambar 10.

Gambar 8. Halaman Pengembalian File Dokumen Rahasia



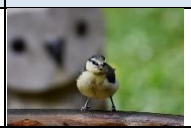
Gambar 9. Halaman Download File Dokumen Rahasia

H. *Pengujian*

Pengujian perlu dilakukan untuk memastikan bahwa aplikasi dalam menyembunyikan dokumen rahasia dan pengambilan kembali dokumen rahasia. Dokumen yang disembunyikan harus dapat diambil kembali. Pengujian ini menggunakan beberapa file *cover object* dan dokumen rahasia. File *cover object* yang digunakan untuk pengujian berjumlah lima file dan bertipe citra PNG dan BMP. Informasi file *cover object* ditunjukkan pada Tabel II.

Dokumen rahasia yang digunakan untuk pengujian berjumlah tiga file dan bertipe txt, jpg, xlsx dan jpg. Informasi dokumen rahasia yang digunakan ditunjukkan pada Tabel III. Hasil pengukuran waktu penyisipan, pengambilan pesan dan nilai PSNR ditunjukkan pada Tabel IV.

TABELII
FILE *COVER OBJECT* YANG DIGUNAKAN

No.	Nama Gambar	Gambar	Bit-Depth	Dimensi Gambar	Ukuran Gambar
1.	burung.bmp		24-Bit	901 x 601 pixel	1.624.503 Byte

2.	burung-1.png		24-Bit	648 x 432 pixel	839.808 Byte
3.	harimau.png		24-Bit	525 x 349 pixel	549.675 Byte
4.	pir.png		24-Bit	647 x 372 pixel	722.052 Byte
5.	lembah.bmp		24-Bit	748 x 498 pixel	1,117.512 Byte

TABEL III
DOKUMEN RAHASIA UNTUK PENGUJIAN 443.481

No.	Nama File	Ukuran File
1.	Pesan-1.txt	115.227 Byte
2.	Pesan-2.jpg	209.940 Byte
3.	Pesan-3.xlsx	97,117 Byte

TABEL IV.
HASIL PENGUKURAN WAKTU DAN PSNR

No.	Cover Object	Dokumen Rahasia	Waktu Penyisipan (Detik)	Waktu Pengambilan (Detik)	PSNR (dB)
1.	Burung.bmp	Pesan-1.txt	130,09	117,06	49,20
2.	Burung.bmp	Pesan-2.jpg	103.25	91,01	49,65
3.	Burung.bmp	Pesan-3.xlsx	112,12	104,84	49,40
4.	Burung-1.png	Pesan-1.txt	179,56	182,85	46,34
5.	Burung-1.png	Pesan-2.jpg	103,93	105,45	46,79
6.	Burung-1.png	Pesan-3.xlsx	123,90	126,01	46,55
7.	Harimau.png	Pesan-1.txt	524,20	411.73	44,46
8.	Harimau.png	Pesan-2.jpg	163,44	164,40	44,93
9.	Harimau.png	Pesan-3.xlsx	239,72	250,32	44,69
10.	Pir.png	Pesan-1.txt	161,55	162,97	45,64
11.	Pir.png	Pesan-2.jpg	113,15	114,26	46,09
12.	Pir.png	Pesan-3.xlsx	137,52	137,26	45,85
13.	Lembah.bmp	Pesan-1.txt	139.42	135,29	47,58
14.	Lembah.bmp	Pesan-2.jpg	108,92	102,85	48,05
15.	Lembah.bmp	Pesan-3.xlsx	123,93	113,83	47,79

Berdasarkan hasil pengujian diketahui bahwa 15 proses penyisipan dan mengambilan kembali dokumen rahasia berjalan dengan baik. Dokumen rahasia yang disembunyikan berhasil diperoleh dan dapat dibuka kembali. Kualitas *stego image* yang dihasilkan masih dinyatakan baik dan kualitas citra tidak mengalami perubahan yang signifikan. Hal ini dinyatakan dengan nilai PSNR-nya rata-rata sekitar 40-an dB. Kualitas *stego image* yang baik bila nilai PSNR-nya di atas 30 dB[15].

IV. PENUTUP

Berdasarkan hasil pengujian dari penelitian steganografi menggunakan metode LSB *Random* yang dikombinasi dengan m-Bit dimana m yang digunakan adalah 2-bit, diperoleh proses berjalan dengan baik. Dokumen rahasia yang disembunyikan, dapat diambil kembali. File *stego image* masih memiliki kualitas citra yang masih baik karena nilai PSNR yang dihasilkan rata-rata di atas 40 dB. Penggunaan 2-bit untuk penyisipan dokumen rahasia dapat meningkatkan ukuran dokumen yang disimpan. Dikarenakan nilai PSNR masih cukup tinggi, maka disarankan untuk menaikkan jumlah bit yang digunakan agar ukuran dokumen rahasia yang disisipkan menjadi bertambah banyak.

REFERENSI

[1] A. A. Gilbijatno and P. Kasih, "Sistem Keamanan Data Teks dengan Steganografi Citra Gambar Menggunakan Algoritma End Of File," in *Seminar Nasional Inovasi Teknologi*, 2020, pp. 197–204.
[2] A. Aryasanti and M. Hardjianto, "Model Pengamanan Berkas Bank Soal dengan Metode Steganografi LSB dan Kompresi," *Journal TICOM (Technology of Information and Communication)*, vol. 2, no. 2, pp. 127–135, 2014.
[3] Y. P. Dewi, "Pengembangan Teknik Steganografi dengan Kriptografi Modifikasi dari Caesar Cipher dan SHA-256 untuk Merahasiakan Pesan," *Jurnal Ilmu Komputer dan Desain Komunikasi Visual*, vol. 5, no. 1, pp. 10–21, 2020.
[4] A. C. Frobenius and E. R. Hidayat, "Steganografi LSB dengan Modifikasi Kriptografi: Caesar, Vigenere, Hill Cipher dan Playfair pada Image," *Melek IT Information Technology Journal*, vol. 6, no. 1, pp. 33–40, 2020.
[5] L. B. Handoko and Abdussalam, "Sekuriti Teks Menggunakan Vigenere Cipher dan Hill Cipher," *Bit (Fakultas Teknologi Informasi Universitas Budi Luhur)*, vol. 19, no. 1, pp. 37–47, 2022.
[6] A. R. Mido and E. I. H. Ujianto, "Analis Pengaruh Citra Terhadap Kombinasi Kriptografi RSA dan Steganografi LSB," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, vol. 9, no. 2, pp. 279–286, 2022, doi: 10.25126/jtiik.202294852.
[7] M. R. Rambe, E. V. Haryanto, and A. Setiawan, "Aplikasi Pengamanan Data dan Disisipkan pada Gambar dengan Algoritma RSA dan Modified LSB Berbasis Android," *IT Journal*, vol. 7, no. 01, pp. 51–62, 2019.
[8] I. U. WM, W. S. Sari, C. A. Sari, and D. R. I. M. Setiadi, "Kombinasi Steganografi-Kriptografi Citra Menggunakan LSB Dan DES," in *Prosiding SNATIF*, 2018, pp. 31–36.
[9] P. Aplikasi Steganografi Dengan Teknik *et al.*, "Perancangan Aplikasi Steganografi Dengan Teknik LSB dan AlgoritmaRC4 & Base64 Encoding," 2018.
[10] E. Nirmala, "Penerapan Steganografi File Gambar Menggunakan Metode Least Significant Bit (LSB) dan Algoritma Kriptografi Advanced Encryption Standard (AES) Berbasis Android," *Jurnal Informatika Universitas Pamulang*, vol. 5, no. 1, pp. 36–47, 2020, [Online]. Available: http://openjournal.unpam.ac.id/index.php/informatika36
[11] S. Agustini and M. Kurniawan, "Peningkatan Keamanan Teks Menggunakan Kriptografi dan Steganografi," *SCAN Jurnal Teknologi Informasi dan Komunikasi*, vol. XIV, no. 3, pp. 33–38, 2019.
[12] R. S. Y. Simbolon and P. Silitonga, "Pengamanan Data dengan Menggunakan Teknik Kriptografi Public RSA dan Knapsack," *KAKIFIKOM (Kumpulan Artikel Karya Ilmiah Fakultas Ilmu Komputer)*, vol. 03, no. 01, pp. 9–12, 2021.
[13] R. Sudiyarno, "Modifikasi Metode Base64 Menggunakan Caesar Cipher dan Kunci Rahasia," *JURTI*, vol. 5, no. 1, pp. 1–8, 2021.
[14] M. F. Arif and M. Misdram, "Implementasi Enkripsi URL pada Website Menggunakan Metode Base64 dan Rotation 13," *Jurnal SPIRIT*, vol. 12, no. 1, pp. 20–25, 2020.
[15] G. Badshah, S. C. Liew, J. M. Zain, and M. Ali, "Watermark Compression in Medical Image Watermarking Using Lempel-Ziv-Welch (LZW) Lossless Compression Technique," *Journal of Digital Imaging*, vol. 29, no. 2, pp. 216–225, 2016, doi: 10.1007/s10278-015-9822-4.