

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/338356974>

Implementasi Steganografi LSB dengan Enkripsi Base64 Pada Citra dengan Ruang Warna CMYK

Article · January 2020

CITATIONS

4

READS

295

2 authors:



Faisal Al Isfahani

Siliwangi University

6 PUBLICATIONS 17 CITATIONS

SEE PROFILE



Fuji Nugraha

Siliwangi University

6 PUBLICATIONS 16 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Sistem Terdistribusi [View project](#)

Implementasi Steganografi LSB dengan Enkripsi Base64 Pada Citra dengan Ruang Warna CMYK

Faisal Al Isfahani
Informatic
Siliwangi University
Tasikmalaya
177006047@student.unsil.ac.id

Fuji Nugraha
Informatic
Siliwangi University
Tasikmalaya
177006052@student.unsil.ac.id

Abstract—Kemajuan teknologi yang sangat bermanfaat pada kehidupan manusia sekarang adalah kecepatan dalam menyampaikan informasi dari tempat yang jauh yaitu melalui Internet. Dalam pengiriman informasi tersebut terdapat masalah yang mengganggu keamanan yang dilakukan oleh pihak-pihak yang tidak bertanggung jawab. Terlepas dari itu file gambar merupakan file yang banyak dicari dan dikirim dan banyak juga mengandung informasi-informasi penting di dalamnya. Keamanan file gambar tentu menjadi sangat penting agar tidak adanya pihak-pihak yang tidak berwenang meretas atau memanipulasi ataupun mengeksploitasi informasi dari gambar tersebut. Terdapat cara untuk mengamankan suatu informasi agar informasi itu tidak bocor kepada pihak yang tidak berwenang, yaitu dengan menggunakan kriptografi dan steganografi. Kriptografi digunakan untuk mengubah pesan rahasia yang dapat dimengerti menjadi pesan yang tak dapat dimengerti, sedangkan steganografi digunakan untuk menyisipkan sebuah pesan rahasia dalam media penampung sehingga tidak akan ada yang menyadari letak pesan rahasia tersebut. Penggabungan kedua metode ini merupakan upaya untuk menjaga kerahasiaan dan keamanan terhadap sebuah file terutama file gambar karena mengimplementasikan prinsip multi layer secure. Dalam penelitian ini algoritma yang digunakan adalah algoritma kriptografi BASE64 dan metode steganografi LSB dengan media citra gambar dengan ruang warna CMYK(32 bit).

Keywords—Steganografi, Kriptografi, BASE64, LSB, CMYK, PSNR

I. PENDAHULUAN

Pesatnya perkembangan teknologi informasi dan komunikasi tentunya memberikan pengaruh besar bagi seluruh kehidupan manusia. Sebagai contoh perkembangan jaringan internet yang memungkinkan setiap orang untuk saling bertukar data atau informasi melalui jaringan internet tersebut. Seiring dengan perkembangan jaringan internet, maka kejahatan atas teknologi komunikasi dan informasi juga turut berkembang dan maju, seperti yang sering kita dengar adalah *hacker*, *cracker*, *carder*, *phreaker* dan sebagainya [1], [2]. Seiring dengan pesatnya kebutuhan informasi pada manusia seperti yang terjadi pada saat ini tentu diperlukan keamanan terhadap pesan yang dikirim maupun diterima. Keamanan tersebut diperlukan untuk menghindari adanya penyadapan atau pembajakan terhadap gambar yang mengandung informasi penting bagi penggunaannya. Keamanan diperlukan untuk menjaga integritas gambar tersebut agar tetap aman [3], [4].

Terdapat beberapa cara untuk mengamankan pesan gambar tersebut, yaitu dengan kriptografi dan steganografi [3]. Kriptografi dan steganografi saling berelasi satu sama lain. Perbedaan utama antara kriptografi dan steganografi adalah kriptografi mengacak-acak pesan hingga pesan tersebut menjadi sulit untuk dimengerti, namun steganografi lebih ke arah menyembunyikan pesan [5]. kerahasiaan pesan

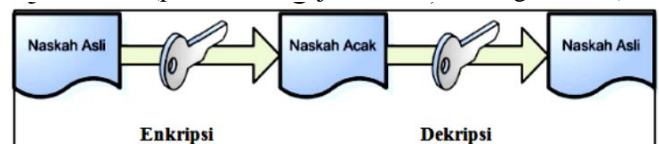
yang ingin disampaikan merupakan faktor utama sehingga digunakan metode steganografi. Dengan metode steganografi, pesan yang ingin disampaikan disembunyikan dalam suatu media umum sehingga diharapkan tidak akan menimbulkan kecurigaan dari pihak lain yang tidak diinginkan untuk mengetahui pesan rahasia tersebut. Oleh sebab itu metode steganografi terus digunakan dan dikembangkan sampai saat ini [4].

Pada penelitian ini bertujuan untuk membuat sebuah program steganografi yang mampu menyisipkan data atau informasi berupa teks dengan menggunakan teknik LSB dan enkripsi BASE64 pada media citra dengan ruang warna CMYK dan melakukan pengujian dengan metode PSNR.

II. TINJAUAN PUSTAKA

A. Kriptografi

Kriptografi adalah ilmu mengenai teknik pengacakan data menggunakan suatu kunci yang disebut teknik enkripsi dari data plain menjadi data yang sulit dibaca oleh seseorang yang tidak memiliki kunci untuk melakukan proses dekripsi. Dekripsi menggunakan kunci dekripsi mendapatkan kembali data asli. Proses enkripsi dilakukan menggunakan suatu algoritma dengan beberapa parameter. Biasanya algoritma tidak dirahasiakan, bahkan enkripsi yang mengandalkan kerahasiaan algoritma dianggap sesuatu yang tidak baik. Kerahasiaan terletak di beberapa parameter yang digunakan. Maka kunci adalah sebuah parameter penentu. Parameter yang menentukan kunci dekripsi itulah yang harus dirahasiakan (parameter menjadi ekuivalen dengan kunci).



Gambar 1. Proses Enkripsi Dekripsi

Gambar 1 menunjukkan efek dari proses enkripsi dan proses dekripsi. Secara garis besar, proses enkripsi adalah proses pengacakan plain text menjadi ciphertext yang “sulit untuk dibaca” oleh seseorang yang tidak mempunyai kunci dekripsi. Yang dimaksud dengan “sulit untuk dibaca” disini adalah probabilitas mendapat kembali naskah asli oleh seseorang yang tidak mempunyai kunci dekripsi dalam waktu yang tidak terlalu lama adalah sangat kecil. Jadi suatu proses enkripsi yang baik menghasilkan naskah acak yang memerlukan waktu yang lama (contohnya satu juta tahun) untuk didekripsi oleh seseorang yang tidak mempunyai kunci dekripsi [6].

B. Algoritma BASE64

Transformasi base64 merupakan salah satu algoritma untuk encoding dan decoding suatu data ke dalam format

ASCII, yang didasarkan pada bilangan dasar 64 atau bisa dikatakan sebagai salah satu metoda yang digunakan untuk melakukan encoding (penyandian) terhadap data binary. Karakter yang dihasilkan pada transformasi base64 ini terdiri dari A..Z, a..z dan 0..9, serta ditambah simbol “+” dan “/” serta satu buah karakter sama dengan (=) di dua karakter terakhir yang dipakai untuk pengisian pad atau dengan kata lain penyesuaian dan menggenapkan data binary. Karakter simbol yang akan dihasilkan akan tergantung dari proses algoritma yang berjalan.

Kriptografi transformasi base64 banyak digunakan di dunia Internet sebagai media data format untuk mengirimkan data, penggunaan tersebut dikarenakan hasil dari encode base64 berupa plaintext, maka data ini akan jauh lebih mudah dikirim, dibandingkan dengan format data yang berupa binary. Algoritma base64 menggunakan kode ASCII dan kode index base64 dalam melakukan proses enkripsi ataupun dekripsinya. Dalam melakukan enkripsi pada URL website, kode index base64 perlu dimodifikasi. Simbol “+” dimodifikasi menjadi “-” dan simbol simbol “/” menjadi “_” [7], [8]. Tabel index base64 dapat dilihat pada tabel 1.

Tabel 1. Kode index Base64

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

Menurut Ariyus (2008) yang dikutip oleh [6], enkripsi base64 ternyata sederhana, jika ada sebuah string dan ingin di enkrip menjadi base64 maka tahapanya adalah sebagai bserikut:

1. Pecah string bytes tersebut ke per-3 bytes.
2. Gabungkan 3 bytes menjadi 24 bit. dengan catatan 1 bytes = 8 bit, sehingga 3 x 8 = 24 bit.
3. Lalu 24 bit yang disimpan di-buffer (disatukan) dipecah-pecah menjadi 6 bit, maka akan menghasilkan 4 pecahan.
4. Masing masing pecahan diubah ke dalam nilai desimal, dimana maksimal nilai 6 bit dalah 63.
5. Terakhir, jadikan nilai-nilai desimal tersebut menjadi index untuk memilih maksimal index ke 64 atau karakter ke 63 dari penyusun base64.

C. Steganografi

Steganografi (steganography) berasal dari bahasa Yunani yaitu “steganos” yang berarti “tersembunyi” atau “terselubung”, dan “graphein” yang artinya “menulis”. Steganografi dapat diartikan “tulisan tersembunyi” (covered writing). Steganografi adalah ilmu dan seni menyembunyikan pesan rahasia di dalam pesan lain sehingga keberadaan pesan rahasia tersebut tidak dapat diketahui. Steganografi membutuhkan dua properti, yaitu media penampung dan pesan rahasia. Media penampung yang umum digunakan adalah gambar, suara, video, atau teks. Pesan yang disembunyikan dapat berupa sebuah artikel, gambar, kode program, atau pesan lain. Proses penyisipan pesan ke dalam media covertext dinamakan encoding, sedangkan ekstraksi pesan dari stegotext dinamakan decoding. Kedua proses ini mungkin memerlukan kunci

rahasia (yang dinamakan stegokey) agar hanya pihak yang berhak saja yang dapat melakukan penyisipan pesan dan ekstraksi [2], [4], [9].

D. Metode LSB

Teknik Steganografi dengan menggunakan metode modifikasi Least Significant Bit (LSB) adalah teknik yang paaling sederhana, pendekatan yang sederhana untuk menyisipkan informasi di dalam suatu citra digital. Mengkonversi suatu gambar dari format GIF atau BMP, yang merekonstruksi pesan yang sama dengan aslinya (lossless compression) ke JPEG yang lossy compression, dan ketika dilakukan kembali akan menghancurkan informasi yang tersembunyi dalam LSB.

Untuk menyembunyikan suatu gambar dalam LSB pada setiap byte dari gambar 24-bit, dapat disimpan 3 byte dalam setiap pixel. Gambar 1,024 x 768 mempunyai potensi untuk disembunyikan seluruhnya dari 2,359,296 bit (294,912 byte) pada informasi. Jika pesan tersebut dikompres untuk disembunyikan sebelum ditempelkan, dapat menyembunyikan sejumlah besar dari informasi. Pada pandangan mata manusia, hasil stego-image akan terlihat sama dengan gambar cover [9]–[11].

E. PSNR

Kualitas media penampung setelah ditambahkan pesan rahasia tidak jauh berbeda dengan kualitas media penampung sebelum ditambahkan pesan. Setelah penambahan pesan rahasia, kualitas citra penampung tidak jauh berubah, masih terlihat dengan baik. Untuk mengukur kualitas citra steganografi diperlukan suatu pengujian secara obyektif. Pengujian secara obyektif adalah dilakukan dengan menghitung nilai PSNR. Peak Signal to Noise Ratio (PSNR) adalah perbandingan antara nilai maksimum dari sinyal yang diukur dengan besarnya derau yang berpengaruh pada sinyal tersebut. PSNR diukur dalam satuan desibel. Pada penelitian ini, PSNR digunakan untuk mengetahui perbandingan kualitas gambar sebelum dan sesudah disisipkan pesan. Untuk menentukan PSNR, terlebih dahulu harus ditentukan MSE (Mean Square Error). MSE secara matematis dapat dirumuskan sebagai berikut:

$$MSE = \frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2$$

Dimana :

MSE = Nilai Mean Square Error citra steganografi

m = Panjang citra stego (dalam pixel)

I(i,j) = nilai piksel dari citra cover

n = Lebar citra stego (dalam pixel)

K(i,j) = nilai piksel pada citra stego

Setelah diperoleh nilai MSE maka nilai PSNR dapat dihitung dari kuadrat nilai maksimum dibagi dengan MSE. Secara matematis, nilai PSNR dirumuskan sebagai berikut :

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right)$$

Dimana: MSE = nilai MSE, MAX_i = nilai maksimum dari pixel citra yang digunakan. Semakin rendah Nilai MSE maka akan semakin baik, dan semakin besar nilai PSNR maka semakin baik kualitas citra steganografi [12].

III. METODE PENELITIAN

Metode yang digunakan dalam penelitian ini terdiri dari 4 tahap, sebagai berikut :

A. Analisis dan Pengumpulan data

Pada tahap ini dilakukan studi pustaka dari beberapa sumber terkait steganografi LSB, kriptografi BASE64, dan image dengan ruang warna CMYK. Informasi didapat dari jurnal maupun artikel media online dan juga terdapat pada dokumentasi resmi dari penyedia teknologi terkait

B. Perancangan Sistem

Para tahap ini dilakukan perancangan arsitektur sistem sesuai dengan hasil informasi dan pengetahuan yang didapat dari tahap sebelumnya. Rancangan arsitektur yang dibuat meliputi permodelan aplikasi implementasi kriptografi BASE64 dan steganografi LSB untuk image dengan ruang warna CMYK.

C. Implementasi

Rancangan arsitektur aplikasi yang telah dibuat sebelumnya, pada tahap ini akan diimplementasikan ke dalam aplikasi dengan memanfaatkan teknologi informasi. Implementasi bermula pada. Persiapan environment kemudian merepresentasikan algoritma kriptografi BASE64 dan steganografi LSB kedalam program dengan bahasa pemrograman PHP dan berbasis web.

D. Pengujian dan Analisis

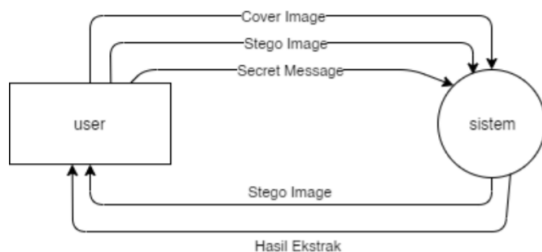
Pada tahap ini dilakukan pengujian terhadap hasil implementasi dari aplikasi yang telah dibangun. Pengujian dilakukan dengan metode PSNR, sebelum didapatkan nilai PSNR (Peak Signal to Noise Ratio) terlebih dahulu harus dicari nilai MSE seperti yang telah di jelaskan sebelumnya. Setelah didapatkan nilai MSE selanjutnya mencari nilai PSNR.

IV. HASIL DAN PEMBAHASAN

A. Perancangan sistem

1) Model fungsional

Model fungsional perangkat lunak memberikan gambaran umum mengenai proses-proses yang terjadi dalam perangkat lunak tanpa memberikan detail mengenai bagaimana proses tersebut berjalan. Model fungsional aplikasi dapat dilihat pada gambar 2.



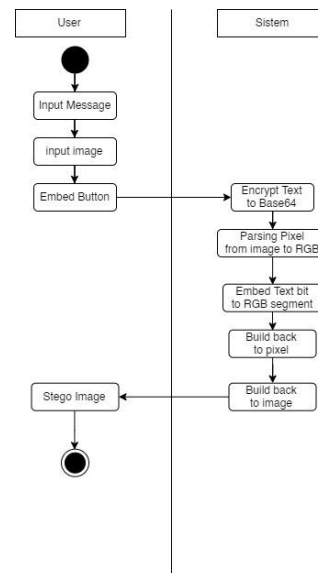
Gambar 2. Model fungsional aplikasi

Gambar 2 merepresentasikan alur kerja sistem yang akan dibangun dimana user memasukkan cover image dan secret message kemudian sistem akan mengembalikan stego image yang telah melalui proses embed. Fungsi selanjutnya yaitu fungsi ekstrak dimana user akan memasukkan stego image kemudian sistem akan mengembalikan hasil ekstrak

kepada user dalam bentuk plaintext atau secret message yang di inputkan diawal.

2) Diagram aktivitas

Diagram aktivitas merepresentasikan alur kerja sistem mulai dari state awal hingga state akhir suatu aktivitas yang dikerjakan oleh sistem. Perancangan diagram aktivitas terbagi menjadi 2 yakni proses embed dan ekstrak. Proses embed terbagi menjadi 2 karena akan dilakukan dua kali percobaan yakni percobaan dengan menggunakan algoritma parsing tiga segment warna yakni RGB dan algoritma parsing empat segment warna yakni CMYK dikarenakan bahasa PHP tidak dapat secara langsung mengambil atau mendapatkan segment warna CMYK pada suatu gambar. Diagram aktivitas embed image dengan algoritma parsing tiga segment warna RGB seperti yang dapat dilihat di gambar 3.

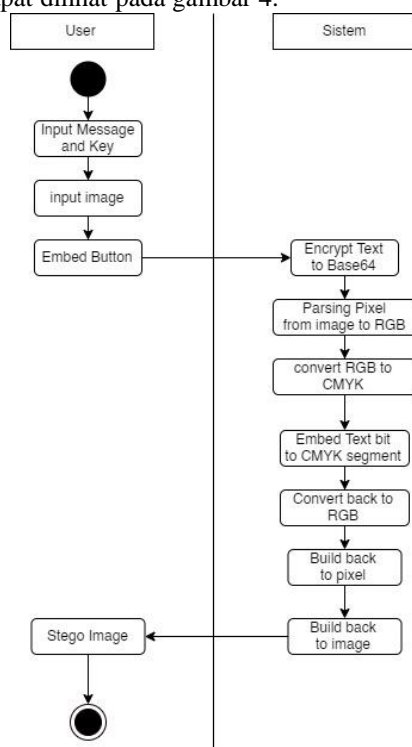


Gambar 3. Diagram aktivitas embed RGB

Gambar 3 merupakan diagram aktivitas dari proses embedding secret message dengan algoritma parsing segment warna RGB. Dimulai dari user menginputkan text atau secret message, dilanjutkan user menginputkan cover image dengan format jpg atau jpeg, selanjutnya user akan mengklik tombol embed, dari sana akan dilakukan proses oleh sistem yang diawali oleh sistem akan membaca text input kemudian akan dilakukan enkripsi terhadap text atau secret message yang diinputkan oleh user kedalam bentuk Base64. Jadi yang di-embedkan kedalam stego image bukanlah secret message yang asli namun hasil enkripsi ke Base64 terlebih dahulu. Kemudian dilakukan parsing pixel dan pengambilan segment warna tiap pixel yaitu warna red, green, dan blue. Masing masing dari segment warna ini akan dimuatkan 1 bit dari secret message. Setelah di sisipkan bit dari secret message segment warna baru di build ulang kedalam pixel dengan koordinat tertentu, dan ketika semua bit dari secret image sudah dimuat seluruhnya proses berikutnya adalah membentuk ulang kedalam sebuah kesatuan gambar. Gambar ini lah yang disebut stego image dan diterima oleh user sebagai output dari proses yang dikerjakan sistem.

Diagram aktivitas selanjutnya adalah proses embedding dengan menggunakan algoritma parsing segment warna CMYK. Perbedaan dengan algoritma sebelumnya adalah, jika

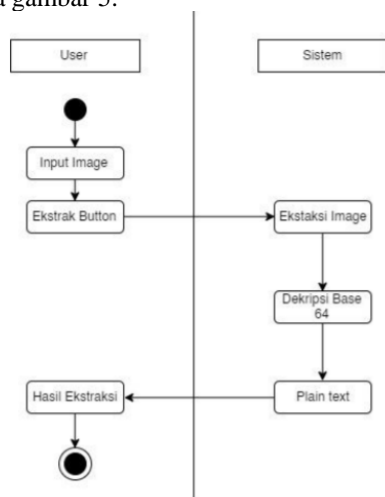
parsing pixel kedalam segment warna RGB itu berarti satu pixel hanya dapat memuat 3 bit dari bit secret message. Namun dengan menerapkan algoritma parsing pixel kedalam segment warna CMYK, maka satu pixel dapat menerima 4 bit dari bit secret message (dengan syarat image harus bertipe 32 bit atau memiliki color format CMYK). Diagram aktivitas embed image dengan algoritma CMYK dapat dilihat pada gambar 4.



Gambar 4. Diagram aktivitas embed CMYK

Gambar 4 menunjukan alur dari aktivitas embed CMYK. Yang berbeda dari alur sebelumnya adalah setelah pendefinisian segment warna RGB pada pixel gambar terjadi penkonversian dari RGB menjadi CMYK kemudian proses embed dilakukan pada segment CMYK. Setelah proses tersebut selesai maka segment CMYK dikembalikan menjadi RGB untuk nantinya di rebuild menjadi pixel dan image serta dikebalikan ke user dalam bentuk stego image.

Selanjutnya diagram aktivitas ekstraksi data seperti yang terdapat pada gambar 5.



Gambar 5. Diagram aktivitas ekstraksi

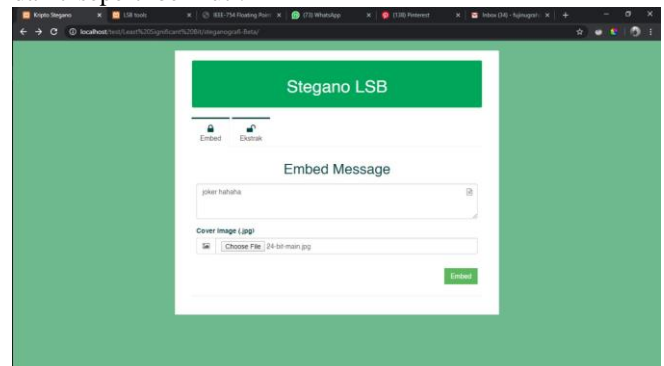
Gambar 5 merupakan representasi dari sebuah proses ekstraksi yang dirancang untuk nantinya di implementasi ke dalam aplikasi. Dalam gambar proses dimulai dari user yang menginputkan gambar (stego image) untuk kemudian ketika user menekan tombol ekstrak sistem akan membaca gambar tersebut dan memarsing atau mengekstrak informasi didalam image tersebut. Hasil dari ekstraksi masih dalam bentuk Base64 maka dari itu harus dilakukan dekripsi Base64 untuk mendapatkan plaintext. Plaintext di kembalikan ke user sebagai output dari sistem.

B. Implementasi

Setelah melewati tahap perancangan, pada tahap implementasi adalah dimana rancangan di implementasikan kedalam bentuk aplikasi. Aplikasi pada percobaan kali ini adalah berbasis web dengan bahasa pemrograman PHP. Implementasi dibagi menjadi dua iterasi, iterasi pertama atau bisa disebut alpha menggunakan atau menerapkan algoritma parsing segment RGB dan iterasi kedua atau bisa disebut beta menggunakan atau menerapkan algoritma parsing segment warna CMYK.

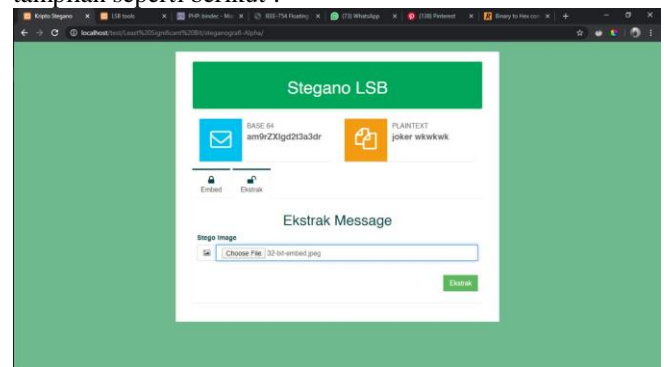
1) Implementasi tahap alpha

Hasil implementasi tahap alpha bisa dilihat pada gambar 6 dan 7 seperti berikut :



Gambar 6. Hasil implementasi Embed

Gambar 6 merupakan tampilan dari fitur embed dimana terdapat form untuk memasukan secret message dan input cover image. Jika pergi ke menu ekstrak akan menampilkan tampilan seperti berikut :



Gambar 7. Hasil implementasi Ekstrak

Gambar 7 merupakan tampilan dari fitur ekstrak dimana terdapat form input file untuk stego image dan ketika telah menekan tombol ekstrak akan muncul hasilnya berupa plaintext dan secret message yang masih dalam bentuk base64.

2) Implementasi tahap beta

Hasil implementasi tahap beta tidak ada perbedaan dari segi tampilan dengan tahap alpha yang membedakan adalah source code embed stegano LSBnya saja. Perbedaan algoritma antara tahap alpha dan tahap beta seperti berikut : Source code steghide tahap alpha:

```
// Membaca RGB dari pixel gambar
$rgb = ImageColorAt($pic, $x, $y);
$cols = array();
$cols[] = ($rgb >> 16) & 0xFF;
$cols[] = ($rgb >> 8) & 0xFF;
$cols[] = $rgb & 0xFF;

// Menyisipkan binary dari masing-masing karakter
// pesan pada setiap segment warna
for ($j = 0; $j < sizeof($cols); $j++) {
    if ($make_odd[$i + $j] === true && is_even($cols[$j])) {
        $cols[$j]++;
    } else if ($make_odd[$i + $j] === false && !is_even($cols[$j])) {
        $cols[$j]--;
    }
}

// Memasukan kembali pixel-pixel baru hasil dari penyisipan pesan
$temp_col = ImageColorAllocate($outpic, $cols[0], $cols[1], $cols[2]);
ImageSetPixel($outpic, $x, $y, $temp_col);
```

Source code steghide tahap beta:

```
// Membaca RGB dari pixel gambar
$rgb = ImageColorAt($pic, $x, $y);
$cols = array();
$cols[] = ($rgb >> 16) & 0xFF;
$cols[] = ($rgb >> 8) & 0xFF;
$cols[] = $rgb & 0xFF;

// Konversi RGB to CMYK
$Rc = $cols[0] / 255;
$Gc = $cols[1] / 255;
$Bc = $cols[2] / 255;

$K = 1 - max($Rc, $Gc, $Bc);
$C = (1 - $Rc - $K) / (1 - $K);
$M = (1 - $Gc - $K) / (1 - $K);
$Y = (1 - $Bc - $K) / (1 - $K);

$cols[0] = $C;
$cols[1] = $M;
$cols[2] = $Y;
$cols[3] = $K;

$cols[0] = round($cols[0] * 255);
$cols[1] = round($cols[1] * 255);
$cols[2] = round($cols[2] * 255);
$cols[3] = round($cols[3] * 255);

// Menyisipkan binary dari masing-masing karakter
// pesan pada setiap segment warna
for ($j = 0; $j < sizeof($cols); $j++) {
    if (is_even($cols[$j])) {
        $cols[$j]++;
    }
}
```

```
} else if (is_odd($cols[$j])) {
    $cols[$j]--;
}
}

// Konversi CMYK to RGB
$cols[0] = $cols[0] / 255;
$cols[1] = $cols[1] / 255;
$cols[2] = $cols[2] / 255;
$cols[3] = $cols[3] / 255;

$red = round(255 * (1 -
$cols[0] / 100) * (1 - $cols[3] / 100));
$green = round(255 * (1 -
$cols[1] / 100) * (1 - $cols[3] / 100));
$blue = round(255 * (1 -
$cols[2] / 100) * (1 - $cols[3] / 100));

$temp_col = ImageColorAllocate($outpic, $red,
$green, $blue);
ImageSetPixel($outpic, $x, $y, $temp_col);
```

Dari kedua Source code implementasi terdapat perbedaan yaitu jika pada tahap alpha setelah dilakukan parsing tiap pixel menjadi RGB langsung disisipkan oleh bit bit dari pesan, namun untuk tahap beta setelah di parsing menjadi RGB dilakukan tahap konversi menjadi CMYK untuk selanjutnya hasil konversi inilah yang disisipi bit bit dari secret image. Alasan pengkonversian ini adalah karena dalam bahasa pemrograman PHP tidak memungkinkan untuk pembacaan segment warna CMYK secara langsung.

C. Pengujian dan analisis hasil

Pengujian dilakukan dengan beberapa case, dan image yang digunakan yaitu dua image dengan format warna RGB dan CMYK atau 24 bit dan 32 bit.

1) Pengujian Encode dengan jumlah karakter yang sama

Pada pengujian ini, image yang diuji adalah 2 tipe image dengan tipe ruang warna berbeda yaitu RGB dan CMYK dengan ukuran dan resolusi yang berbeda beda dan akan dimasukan atau disisipi teks dengan jumlah karakter yang sama. File akan didisipkan teks 'joker wkwkwk' dengan ukuran file awal 24bit = 189kb dan 32bit = 2.43Mb. pengujian ini dilakukan untuk mengetahui berapa ukuran file citra setelah disisipkan pesan yang sama. Hasil ditunjukan pada tabel 2.

Tabel 2. Hasil pengujian encode karakter

No	Nama File	Resolusi Cover	Ukuran Cover	Resolusi Stego	Ukuran Stego
1.	24-bit.jpg	714x1000	189Kb	714x1000	1,36Mb

2.	32-bit.jpg	720x1129	2.43Mb	720x1129	1.21Mb
----	------------	----------	--------	----------	--------

2) Pengujian Ketahanan Citra

Dalam pengujian ketahanan ini, pengujian dilakukan menggunakan aplikasi alpha dengan mengujikan dua format gambar yang berbeda dan telah sisipi pesan(stego image) yaitu yang memiliki segment warna RGB dan CMYK. Jadi pengujian ini lebih berfokus kepada ketahanan stego image ,apakah masih dapat diekstrak ketika sudah dilakukan berbagai manipulasi

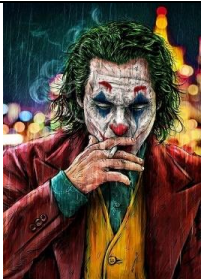


Uji ketahanan stego image dengan segment RGB (24 bit)

Citra akan diujikan meiliki ukuran resolusi citra 714x1000 pixel dan ukuran file 1,36Mb. Citra yang diujikan seperti pada gambar 8. Dan hasil dari pengujian seperti yang ditunjukkan pada tabel 3, 4, dan 5.






Gambar 8. Stego image dengan segment warna RGB



Tabel 3. Hasil pengujian Scaling pada Stego Image

No.	Scaling menjadi	Ukuran Sesudah	Stego-image setelah discaling	Hasil Decode	
				Sukses	Gagal
1.	357x500 Pixel	119,8 KB		-	Gagal
2.	179x250 Pixel	33.1 KB		-	Gagal
3.	90x125	9.9 KB		-	Gagal

Tabel 4. Hasil pengujian Rotasi pada Stego Image

No.	Rotation Sebesar	Ukuran Sesudah	Stego-image setelah di rotaion	Hasil Decode	
				Sukses	Gagal
1.	90°	332.2 KB		-	Gagal
2.	180°	356.9 KB		-	Gagal
3.	270°	352.7 KB		-	Gagal




Tabel 5. Hasil pengujian Cropping pada Stego Image

No.	Ukuran Sesudah	Stego-image setelah di Crop	Hasil Decode	
			Sukses	Gagal
1.	357.1 KB		-	Gagal
2.	329.1 KB		-	Gagal


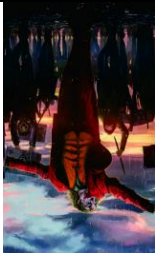

Uji Ketahanan stego image dengna segment CMYK(32 bit)
Citra akan dujikan memiliki resolusi 720x1129 pixel dan ukuran file 1.21 Mb. Citra yang diujika adalah pada gambar

9. Dan hasil dari pengujian seperti yang ditunjukkan pada tabel 6,7 dan 8.



Tabel 6. Hasil pengujian Scaling pada Stego Image

No.	Scaling menjadi	Ukuran Sesudah	Stego-image setelah discaling	Hasil Decode	
				Sukses	Gagal
1.	360x595 Pixel	452.6 KB		-	Gagal
2.	180x298 Pixel	125.0 KB		-	Gagal
3.	90x149	34.7 KB		-	Gagal

Tabel 7. Hasil pengujian rotasi pada Stego Image

No.	Rotasi Sebesar	Ukuran Sesudah	Stego-image setelah di rotaion	Hasil Decode	
				Sukses	Gagal
1.	90°	1.6 MB		-	Gagal
2.	180°	1.5 MB		-	Gagal
3.	270°	1.6 MB		-	Gagal

Tabel 8. Hasil Pengujian Cropping pada Stego Image

No.	Ukuran Sesudah	Stego-image setelah di rotaion	Hasil Decode	
			Sukses	Gagal
1.	1.2 MB		-	Gagal
2.	1.0 MB		-	Gagal

Dari hasil pengujian yang dilakukan terhadap 2 tipe image yang berbeda yaitu RGB dan CMYK dari hasil yang terdapat pada tabel dapat disimpulkan bahwa stegano LSB tidak tahan akan perubahan stego image baik Scaling, Rotating, ataupun Cropping. Ketika stego image telah melewati salah satu dari kegiatan perubahan maka stego image tersebut akan kembali menjadi cover image atau tidak dapat di extrak secret message yang terkandung didalamnya.

3) Pengujian Nilai Error pada Citra

Selain pengujian untuk mengetahui ketahanan dari steggo image ketika dihadapkan ke beberapa kondisi. penting adanya untuk melakukan pengujian terhadap kualitas gambar stego. Metode yang digunakan untuk mengetahui kualitas suatu gambar stgo adalah metode PSNR. PSNR merupakan sebuah metode penguan secara Objektif dan matematis. Hasil pengujian PSNR terdapat pada tabel 9.

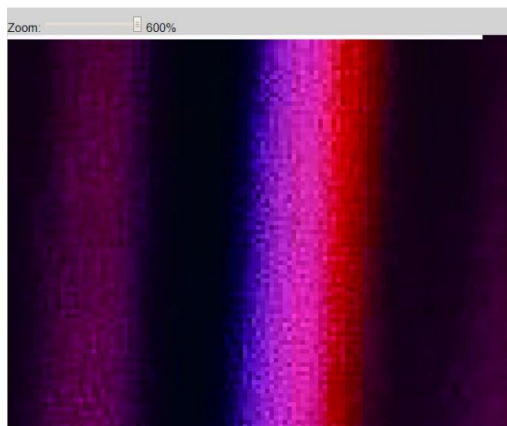
Tabel 9. Hasil pengujian PSNR

Gambar	Resolusi	Cover size	Stego Size	MSE	PSNR
24-bit-embed	714x1000	189Kb	1,36Mb	8.788	53.08
32-bit-embed	720x1129	2.43Mb	1.21Mb	10.25	49,74

Tabel 9 merupakan tabel pengujiain kualitas gambar stego dengan nama file 24-bit-embed dan 32-bit-embed. Pada tabel diatas menjelaskan baahwa kualitas penyisipan sudah baik dilihat dari nilai MSE dan nilai PSNR, semakin kecil nilai MSE berarti semakin kecil nilai error dan semakin besar nilai PSNR berarti proses penyisipan semakn baik.

4) Pengujian menggunakan aplikasi Beta

Hasil stego iimage yang dikeluarkan oleh aplikasi beta adalah sebagai berikut :



Gambar 8. Hasil embed aplikasi beta

Gambar 8 menunjukkan alasan kenapa aplikasi beta tidak dilanjutkan pengembangannya ataupun tidak menjadi aplikasi untuk tahap pengujian sebelumnya. Dikarenakan proses embedding sudah merusak pixel menjadikan pixel yang di sisipi oleh pesan rusak dan tanpa dilakukan steganalisis pun sudah terlihat kerusakannya.

V. KESIMPULAN

- 1) Penerapan LSB dengan ddampingi oleh Kriptografi Base64 merupakan sebuah upaya menambah penjagaan, pengamanan, dan kerahasiaan yang dilakukan pada spesimen Gambar yang menerapkan prinsip multi layer security sehingga jika stego image dapat di ekstrakpun maka hasil ekstrak tidak langsung dalam bentuk plainteks melainkan dalam bentuk enkripsi Base64.
- 2) Dari hasil pengujian dapat disimpulkan bahwa LSB merupakan metode steganografi yang baik karena tidak merubah ukuran file secara signifikan namun untuk ketahanan stego image metode LSB dirasa kurang baik dikarenakan jika sedikit saja dilakukan perubahan pada stego image, maka tidak akan bisa dilakukan ekstraksi pesan rahasia.
- 3) Dilihat dari hasil PSNR hasil MSE dan PSNR dapat dikatakan baik karena nilai MSE kecil dan nilai PSNR terbilang besar dengan selisih 40 – 50 dB dari MSE. Maka aplikasi ini dapat diimplementasikan untuk mengamankan dan menjaga kerahasiaan file gambar namun untuk ketahanan stego image masih dinilai rendah.

VI. DAFTAR PUSTAKA

- [1] A. Rahmatulloh and R. Munir, "Pencegahan Ancaman Reverse Engineering Source Code PHP dengan Teknik Obfuscation Code pada Extension PHP," *Univ. Siliwangi*, no. October 2015, 2015.
- [2] J. V. Purba, M. Situmorang, and D. Arisandi, "Implementasi Steganografi Pesan Text Ke Dalam File Sound (.Wav) Dengan Modifikasi Jarak Byte Pada Algoritma Least Significant Bit (LSB)," *J. Dunia Teknoogi Inf. Vol. 1, No. 1, 50-55*, vol. 1, no. 1, pp. 50–55, 2012.
- [3] G. W. Bhaudhayana and I. M. Widiartha,

"Implementasi algoritma kriptografi aes 256 dan metode steganografi lsb pada gambar bitmap," *J. Ilmu Komput. Univ. Udayana*, vol. 8, no. 2, pp. 15–25, 2015.

- [4] N. Laila and A. S. R. Sinaga, "Implementasi Steganografi LSB Dengan Enkripsi Vigenere Cipher Pada Citra," *Sci. Comput. Sci. Informatics J.*, vol. 1, no. 2, p. 47, 2019.
- [5] M. Juneja and P. S. Sandhu, "An Improved LSB Based Steganography Technique for RGB Color Images," *Int. J. Comput. Commun. Eng.*, vol. 2, no. 4, pp. 513–517, 2013.
- [6] S. Kromodimoeljo, *Teori & Aplikasi Kriptografi*. SPK IT consulting.
- [7] A. P. Nugraha and E. Gunadhi, "Penerapan Kriptografi Base64 Untuk Keamanan URL (Uniform Resource Locator) Website Dari Serangan SQL Injection," *J. Algoritm. Sekol. Tinggi Teknol. Garut*, pp. 491–498, 2016.
- [8] R. Aulia, A. Zakir, and D. A. Purwanto, "Penerapan Kombinasi Algoritma Base64 Dan Rot47 Untuk Enkripsi Database Pasien Rumah Sakit Jiwa Prof. Dr. Muhammad Ildrem," *InfoTekJar (Jurnal Nas. Inform. dan Teknol. Jaringan)*, vol. 2, no. 2, pp. 146–151, 2018.
- [9] M. Hariri, R. Karimi, and M. Nosrati, "An introduction to steganography methods," *World Appl. Program.*, no. 13, pp. 191–195, 2011.
- [10] S. Gupta, G. Gujral, and N. Aggarwal, "Enhanced Least Significant Bit algorithm For Image Steganography," *IJCEM Int. J. Comput. Eng. Manag. ISSN*, vol. 15, no. 4, p. 22307893, 2012.
- [11] W. Laksito, "Modifikasi Least Significant Bit dalam Steganografi," *J. Ilm. SINUS*, vol. 6, pp. 1–8, 2008.
- [12] G. M. Male, Wirawan, and E. Setijadi, "Analisa Kualitas Citra Pada Steganografi untuk Aplikasi e-Government," in *Prosiding Seminar Nasional Manajemen Teknologi XV*, 2012, pp. 1–9.