Original Article

e-ISSN: 2774-2016 - https://journal.itera.ac.id/index.php/indojam/
p-ISSN: 2774-2067

Received 9th February 2025
Accepted 23rd April 2025
Published 25th April 2025

Open Access

DOI:
https://doi.org/10.35472/indojam.v5i1.1887

Implementasi Kombinasi *Secret Sharing* dan Steganografi Citra *Least Significant Bit* dengan *QR Code*

Dwi Putri Pebriani[a]*, Rini Marwati[a], Dewi Rachmatin[a]

[a] *Program Studi Matematika, Universitas Pendidikan Indonesia*

* *Koresponden E-mail: rinimarwati@upi.edu*

Abstract: Information security becomes very important as long as information technology continues to develop. This research combines Shamir Secret Sharing (Scheme $(\boldsymbol{t}, \boldsymbol{w})$) cryptography and Least Significant Bit (LSB) steganography to improve information security in the aspect of confidentiality. The need to collect a minimum number of shares, Scheme $(\boldsymbol{t}, \boldsymbol{w})$ method makes it more difficult for hackers to reconstruct the message. The implementation of the combination of cryptography and steganography produces an application program created with Python programming languange version 3.11.2, where the Scheme $(\boldsymbol{t},\boldsymbol{w})$ used is $\mathbf{2} \leq \boldsymbol{t} \leq \boldsymbol{w} \leq \mathbf{10}$ and RGB image as the cover object of LSB method. In determining the location of image pixels that will be inserted into the message, random numbers generated by the Linear Congruential Generator (LCG) algorithm is used. The program can create a share of text messages with a maximum of 8 characters contained in ASCII characters 32 to 126. The result obtained from the share construction program is $\boldsymbol{w}$ QR codes that refers to the stego image that has been inserted by the share, so that the existence of information in the QR code is difficult to know. The program can also reconstruct the message back from $\boldsymbol{t}$ QR codes.

Keywords: *Cryptography, Shamir Secret Sharing, Steganography, RGB Image, Least Significant Bit, LCG, QR Code.*

Abstrak: Keamanan informasi menjadi sangat penting selama teknologi informasi terus berkembang. Penelitian ini mengombinasikan kriptografi *Shamir Secret Sharing* (Skema $(\boldsymbol{t},\boldsymbol{w})$) dan steganografi *Least Significant Bit* (LSB) untuk meningkatkan keamanan informasi pada aspek kerahasiaan. Perlunya mengumpulkan jumlah minimal *share*, metode Skema $(\boldsymbol{t},\boldsymbol{w})$ membuat peretas lebih sulit untuk merekonstruksi pesan. Implementasi dari kombinasi kriptografi dan steganografi tersebut menghasilkan program aplikasi yang dibuat dengan bahasa pemograman Python versi 3.11.2, di mana Skema $(\boldsymbol{t},\boldsymbol{w})$ yang digunakan adalah $\mathbf{2} \leq \boldsymbol{t} \leq \boldsymbol{w} \leq \mathbf{10}$ dan citra RGB sebagai *cover object* metode LSB. Dalam menentukan letak *pixel* citra yang akan disisipkan pesan, digunakan bilangan acak yang dibangkitkan oleh algoritma *Linear Congruential Generator* (LCG). Program dapat membuat *share* dari pesan teks dengan ketentuan maksimal 8 karakter yang termuat dalam ASCII karakter ke 32 sampai 126. Hasil yang diperoleh dari program konstruksi *share* adalah $\boldsymbol{w}$ buah *QR Code* yang merujuk kepada *stego image* yang telah disisipkan *share*, sehingga keberadaan informasi dalam *QR Code* tersebut sulit diketahui. Program juga dapat merekonstruksi pesan kembali dari $\boldsymbol{t}$ buah *QR Code*.

Kata Kunci: *Kriptografi, Shamir Secret Sharing, Steganografi, Citra RGB, Least Significant Bit, LCG, QR Code.*

Pendahuluan

Di era pesatnya kemajuan teknologi informasi, keamanan informasi menjadi sangat penting. Kemajuan ini dapat meningkatkan efisiensi dan konektivitas, tetapi juga berisiko adanya ancaman keamanan informasi yang bersifat rahasia [1][2]. Kepercayaan yang diberikan kepada pihak lain untuk mengakses informasi rahasia harus diawasi secara ketat karena adanya kemungkinan penyimpangan kesepakatan dari pihak tersebut. Untuk meningkatkan keamanan berbagi informasi rahasia, perlu memodifikasi informasi tersebut sehingga penerima tidak mengetahui informasi yang dimaksud secara langsung . Pengamanan informasi dapat dilakukan

Original Article

dengan teknik kriptografi atau steganografi [1]. Kriptografi adalah seni dan ilmu untuk menyandikan pesan sehingga tidak dapat dipahami maknanya [1][2]. Sedangkan, steganografi adalah seni dan ilmu untuk menyembunyikan pesan ke dalam media penampung (*cover object*) sehingga keberadaan pesan tidak dapat diketahui [1][2].

Dalam kriptografi terdapat metode *secret sharing*, yaitu membagi pesan rahasia (*secret*) menjadi potongan-potongan informasi (*share*) yang diberikan kepada sekolompok orang (partisipan), sehingga mereka harus mengumpulkan potongan informasi tersebut jika ingin mendapatkan kembali pesan [3]. Konsep *secret sharing* ditemukan oleh Adi Shamir pada tahun 1979 dikenal dengan *Shamir Secret Sharing* atau Skema (t, w), di mana w merupakan jumlah partisipan dan t merupakan jumlah minimal *share* yang harus dikumpulkan untuk dapat mengembalikan pesan [4].

Meskipun kriptografi *Shamir Secret Sharing* dapat menyamarkan pesan, peretas masih dapat menemukan keberadaan *share* [1]. Oleh karena itu, untuk menyembunyikan *share* diperlukan teknik steganografi. Metode steganografi yang digunakan pada penelitian ini adalah *Least Significant Bit* (LSB) dengan *cover* berupa citra RGB dengan format png. Keunggulan metode LSB yaitu tidak ada perbedaan secara kasat mata antara citra asli dan citra yang sudah melalui proses peyisipan (*embedding*) meskipun prosesnya mudah dan sederhana [6].

Chuang *et al.* (2010) membuat program *Shamir Secret Sharing* dengan masukkan pesan berupa angka dan *share* yang dihasilkan berupa *QR Code* Humaira *et al.* (2023) membuat program penggabungan kriptografi Skema (3,4) dan steganografi audio LSB, dengan masukkan pesan berupa PIN 6 digit di mana digit pertama tidak sama dengan nol.

Pada penelitian ini, pesan yang dapat dikonstruksi *share*-nya diperluas menjadi pesan teks dengan batasan maksimal 8 karakter yang terdiri dari angka, huruf, simbol, maupun kombinasi ketiganya. Karakter pesan termuat dalam karakter ASCII 32 sampai 126, karena ordinal karakter pada rentang tersebut *printable* dalam Python. Skema (t, w) yang digunakan yaitu $2 \leq t \leq w \leq 10$. Setiap *share* hasil Skema (t, w) disembunyikan ke dalam *cover* citra RGB dengan format png oleh metode LSB. Partisipan akan menerima *share* dalam bentuk *QR Code* yang merujuk kepada *stego image*.

QR Code merupakan tipe kode batang yang dapat memuat lebih dari 4000 karakter, dapat menyimpan informasi lebih banyak dari *barcode*[7][8]. *QR Code* dapat berisi tautan yang merujuk kepada pesan teks, gambar, video, audio, dan lainnya [4][7]. Di era digital, penggunaan *QR Code* mempermudah cara berbagi informasi.

Kombinasi kriptografi dan steganografi dengan metode yang telah dipaparkan tersebut dapat meningkatkan keamanan informasi rahasia pada aspek kerahasiaan. Untuk meretas informasi rahasia, peretas harus mengumpulkan t *QR Code* dan menemukan keberadaan *share* dalam *stego image*.

Metode

Skema (t, w)

Misalkan t dan w adalah bilangan bulat positif dengan $t \leq w$. Skema (t, w) adalah metode pembagian pesan kepada w partisipan sedemikian sehingga sembarang himpunan bagian yang terdiri dari t partisipan dapat merekonstruksi pesan. Dalam Skema (t, w), pesan diinterpretasikan sebagai bilangan bulat bulat positif, dinotasikan dengan M. Berikut langkah membuat *share* dari Skema (t, w) [1]:

1. Pilih bilangan prima p, dengan $p > M$ dan $p > w$. Bilangan p tidak perlu dirahasiakan.
2. Pilih $t-1$ bilangan bulat dalam modulo p secara acak, misalkan $a_1, \dots, a_{t-1}$, kemudian nyatakan dalam polinom berikut:

$$s(x) \equiv M + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1} (mod\ p) \quad (1)$$

   Rahasiakan polinom $s(x)$.
3. Pilih w bilangan bulat berbeda, misalkan $x_1, \dots, x_w$. Substitusikan x_i, $1 \leq i \leq w$, ke persamaan (1) sehingga diperoleh $y_i = s(x_i)$. Setiap partisipan ke-i menerima *share* (x_i, y_i).

Least Significant Bit (LSB)

Pada *byte* terdapat bit yang paling kurang berarti (*least significant bit*), yaitu bit digit ke-8 [5]. Pada citra RGB, setiap *pixel* berukuran 3 (tiga) *byte* (24 bit) di mana setiap *byte* mewakili nilai intensitas warna *Red*, *Green*, dan *Blue* [9]. Misalkan terdapat 3 (tiga) *pixel* dengan nilai intensitas warna pada setiap *pixel* yang telah dikonversi ke dalam biner diperoleh bilangan *biner* sebagai berikut:

(11110111 00010010 10101010
11000100 11101001 00000001
00001001 11010001 00011101)

Misalkan pesan yang akan disisipkan (*embedded*) adalah karakter "R", berdasarkan ASCII bilangan biner karakter tersebut adalah 01010010. *Embedding* pesan dilakukan dengan mengganti bit LSB dari setiap *byte* oleh bit dari pesan [5]. Hasil *embedding* memberikan nilai *pixel* baru sebagai berikut:

(11110110 00010011 10101010
11000101 11101000 00000000
00001001 11010000 00011101)

Peretas akan kesulitan mencari keberadaan pesan pada *stego image* karena sangat mirip dengan *cover image*, tidak bisa dibedakan jika hanya menggunakan indera manusia. LSB menyebabkan nilai intensitas warna pada *pixel* citra hanya berubah satu lebih besar atau lebih kecil setelah *embedding*[9].

Algoritma *Linear Congruential Generator* (LCG)

Jika menggunakan hanya menggunakan metode LSB, pesan yang disisipkan dalam *cove image* akan mudah diekstraksi, yaitu dengan mengambil bit terakhir pada setiap *byte*. Oleh karena itu diperlukan *stego key* untuk menentukan letak penyisipan bit pesan. Pada penelitian ini, digunakan algoritma LCG yang merupakan salah satu metode *Pseudo Random Number Generator*. LCG digunakan untuk menghasilkan bilangan acak sebagai penentu letak *byte pixel* yang akan disisipkan bit pesan. Bilangan acak tersebut dibangkitkan dari persamaan berikut [10]:

$$X_{n+1} = aX_n + b\ (mod\ m) \qquad (2)$$

di mana:
X_{n+1} : bilangan acak ke-$(n+1)$
X_n : bilangan acak ke-n
a : konstanta pengali
b : konstanta penambah
m : konstanta modulus

Persamaan (2) memerlukan nilai awal X_0 sebagai pembangkit (*seed*), di mana $0 < X_0 \leq m$. Berikut syarat-syarat agar LCG mempunyai periode tidak lebih besar dari m dan mempunyai periode penuh [9]:

a. $a > 0, b > 0$
b. b relatif prima terhadap m
c. $(a-1)$ dapat dibagi oleh semua faktor prima dari m.
d. Jika m kelipatan 4 maka $(a-1)$ kelipatan 4
e. $m > maks(a, b, X_0)$

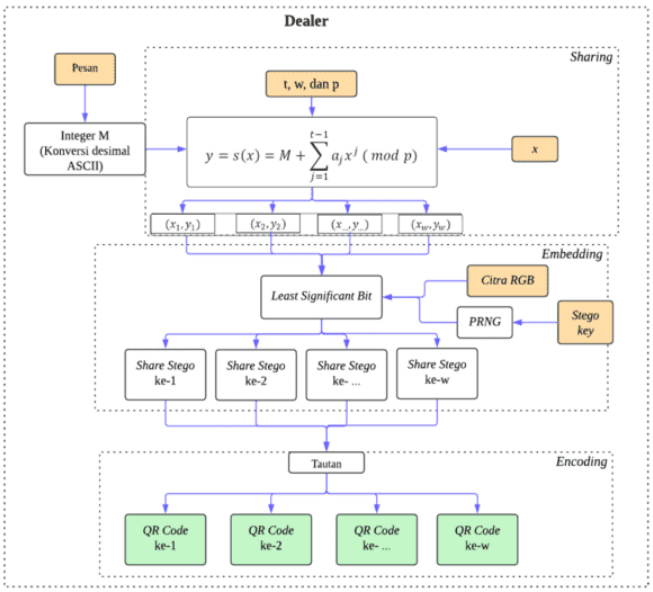
Konstanta modulus m adalah batas nilai maksimum dan batas jumlah maksimum bilangan acak yang dihasilkan. Pada penelitian ini, nilai m merupakan maksimal panjang string "(x_i, y_i)", yaitu *share* yang dihasilkan dari Skema (t, w).

Model Dasar

Program aplikasi dikonstruksi dari skema konstruksi *share* dan skema rekonstruksi *secret*. Skema konstruksi *share* memperlihatkan tahapan mengonstruksi *share QR Code* dari pesan seperti pada Gambar 1. Konstruksi *share* dilakukan oleh *dealer*, dimulai dengan mengubah pesan teks menjadi bilangan bulat M dengan menggabungkan nilai desimal dari setiap karakter pesan. Setelah menentukan nilai t, w, dan p, proses *sharing* terhadap M menggunakan Skema (t, w) sehingga menghasilkan w buah titik *share*. Kemudian *embedding* setiap titik *share* pada citra RGB yang sama dengan metode LSB sehingga menghasilkan w *stego image*. Kemudian setiap *stego image* di-*hosting* pada suatu *web* untuk mendapatkan tautan yang merujuk pada *stego image* tersebut. Layanan *web hosting* yang digunakan dalam penelitian ini adalah *https://imgbb.com*. Selanjutnya, setiap tautan tersebut disimpan dalam *QR Code* yang kemudian dibagikan kepada partisipan.

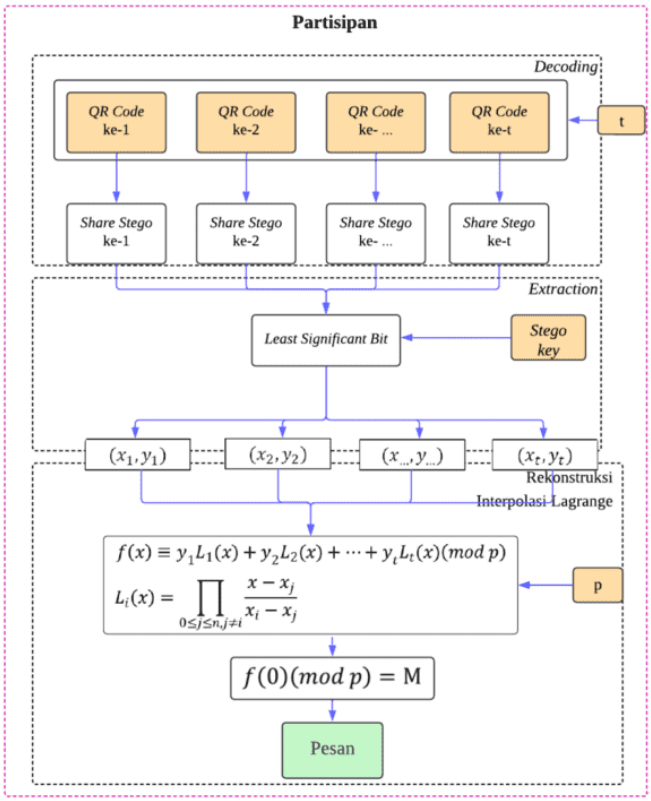
Published by: Lembaga Penelitian dan Pengabdian Masyarakat (LPPM)
Institut Teknologi Sumatera, Lampung Selatan, Indonesia

Original Article



Gambar 1. Skema konstruksi *share*

Skema rekonstruksi *secret* merupakan tahapan mengembalikan pesan dari *share QR Code* seperti pada Gambar 2. Rekonstruksi *secret* dilakukan oleh partisipan setelah mengumpulkan t buah *QR Code* dan mempunyai *stego key*. *Decoding* setiap *QR Code* sehingga *stego image* dapat diakses. Kemudian ekstraksi setiap *stego image* sehingga diperoleh informasi yang tersembunyi yaitu titik *share* (x, y). Substitusikan t buah titik tersebut ke dalam polinom Lagrange, nilai M dapat diperoleh dengan cara menghitung nilai $f(0) \bmod p$. Pesan teks dapat dikembalikan dengan cara mengonversi nilai desimal menjadi karakter berdasarkan ASCII.



Gambar 2. Skema rekonstruksi *secret*

Hasil dan Diskusi

Program aplikasi dibuat untuk memudahkan *user* membuat *share QR Code* dan mengembalikan pesan rahasia dengan kriptografi Skema (t, w) dan steganografi citra LSB. Program tidak hanya menerima *input* pesan berupa angka (bilangan bulat), melainkan dapat menerima *input* pesan berupa karakter yang termuat dalam ASCII 32-126 dengan maksimal 8 karakter. Program memudahkan *user* menentukan bilangan prima p yang memenuhi $p \geq M$ dengan memberikan daftar pilihan bilangan prima setelah M dihitung. Program juga memudahkan *user* menentukan *stego key* dengan memberikan pilihan konstanta pengali (a) dan konstanta penambah (b) untuk membangkitkan bilangan acak dari algoritma LCG. Nilaia a dan b yang diberikan sudah memenuhi syarat untuk menghasilkan periode penuh terhadap m.

Tampilan Aplikasi dan Penggunaannya

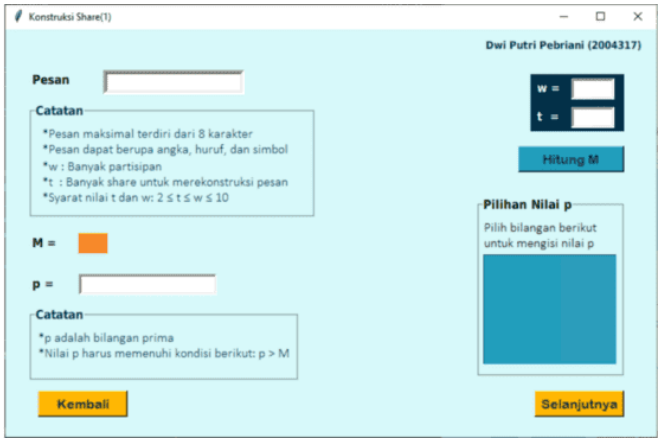
Program aplikasi dibuat menggunakan Python versi 3.11.2 dengan *software* Visual Studio Code.

Original Article

Tampilan menu utama pada program dapat dilihat pada Gambar 3.

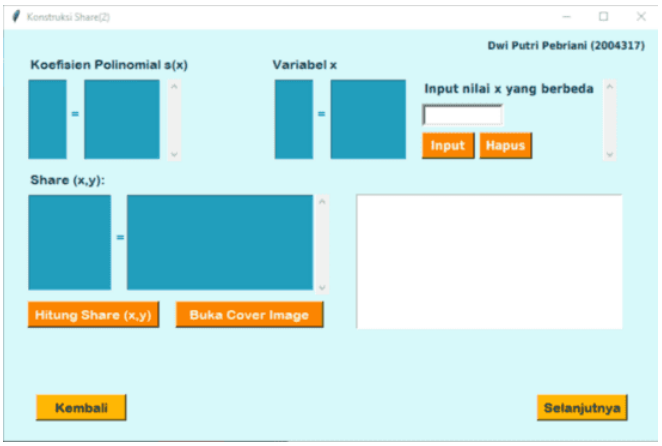
Gambar 3. Menu utama program

Jika *user* menekan *button* "Konstruksi *Share*", maka Gambar 4 ditampilkan. *User* memasukkan pesan, t, dan w sesuai ketentuan yang tertera dalam "Catatan". Kemudian *user* menekan *button* "Hitung M", nilai M dan pilihan nilai p ditampilkan program. *User* memilih salah satu bilangan prima dari pilihan nilai p untuk memasukkan nilai p. Kemudian *user* menekan "Selanjutnya", program akan menampilkan Gambar 5.



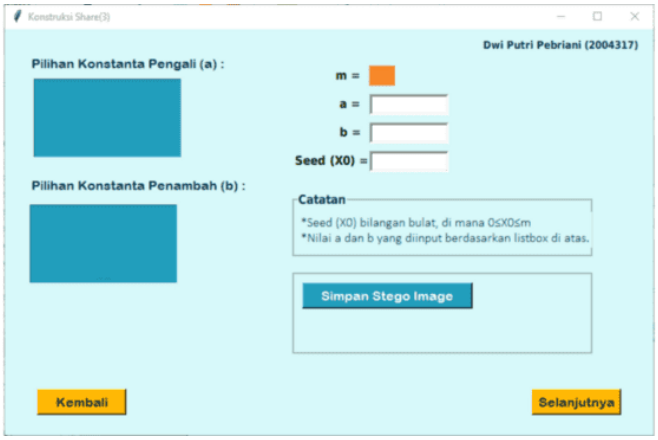
Gambar 4. Konstruksi *share* pada program (1)

Pada tampilan berikut program memberikan $t-1$ koefisien polinomial $s(x)$. *User* memasukkan w bilangan bulat acak, kemudian menekan *button* "Hitung Share (x,y)" untuk memperoleh *share* hasil Skema (t,w). Kemudian *user* menekan *button* "Buka Cover Image", *user* diarahkan ke *file explorer* untuk memilih *cover image*. Kemudian *user* menekan "Selanjutnya", Gambar 6 ditampilkan.



Gambar 5. Konstruksi *share* pada program (2)

Pada tampilan Gambar 6 berikut, program memberikan nilai m dan pilihan konstanta a dan b. *User* memasukkan nilai a, b, dan *seed* sesuai ketentuan yang tertera pada "Catatan". Program melakukan *embedding* ketika *user* menekan "Simpan Stego Image", setelah *embedding* selesai Gambar 7 ditampilkan.

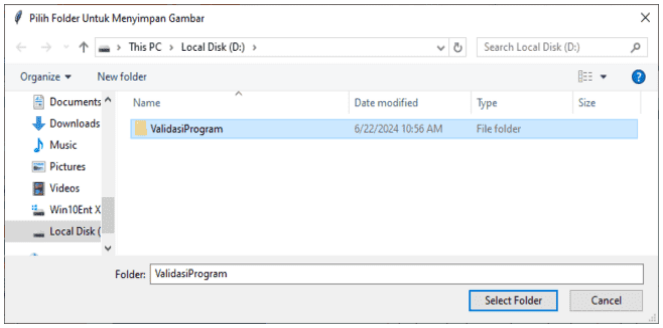


Gambar 6. Konstruksi *share* pada program (3)

User memilih *folder* atau membuat *folder* baru untuk menyimpan *stego image*. *Stego image* harus disimpan terlebih dahulu agar program dapat mengunggah ke *website* ImgBB sehingga diperoleh tautan yang merujuk kepada *stego image* untuk disimpan dalam *QR Code*.

Published by: Lembaga Penelitian dan Pengabdian Masyarakat (LPPM)
Institut Teknologi Sumatera, Lampung Selatan, Indonesia

Original Article



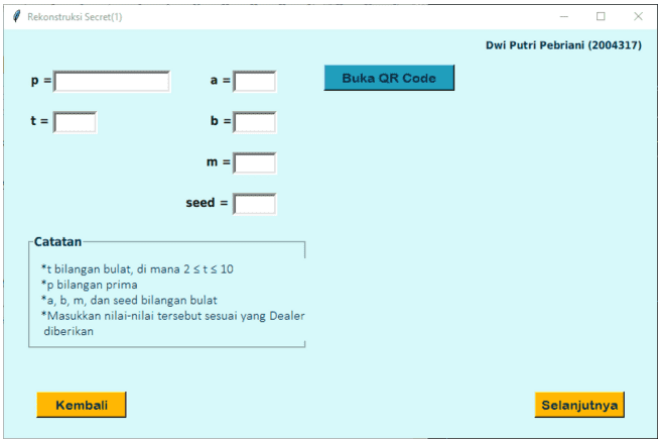
Gambar 7. Tampilan *file explorer*

Kemudian *user* menekan "Selanjutnya", Gambar 8 ditampilkan. Pada tampilan tersebut, program memberikan share QR Code sebagai hasil rangkaian proses kontruksi Share. Share *QR Code* dapat disimpan dengan menekan "Simpan QR Code".

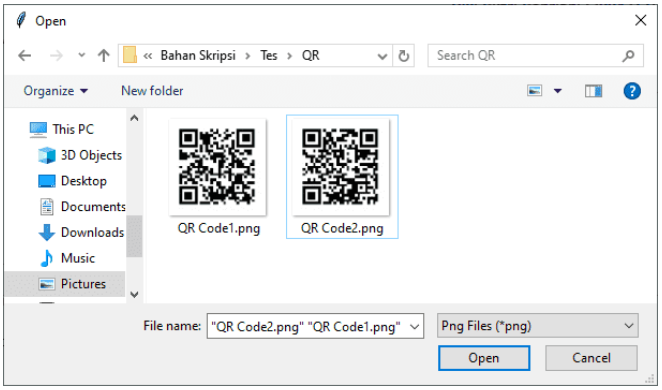
Gambar 8. Konstruksi *share* pada program (4)

Selain memberikan *share QR Code* pada partisipan, *Dealer* juga memberikan informasi yang berisi t, p, dan *stego key*. Informasi tersebut sudah disimpan program dalam *file* "Informasi.txt" pada *folder* yang sama saat menyimpan *stego image*.

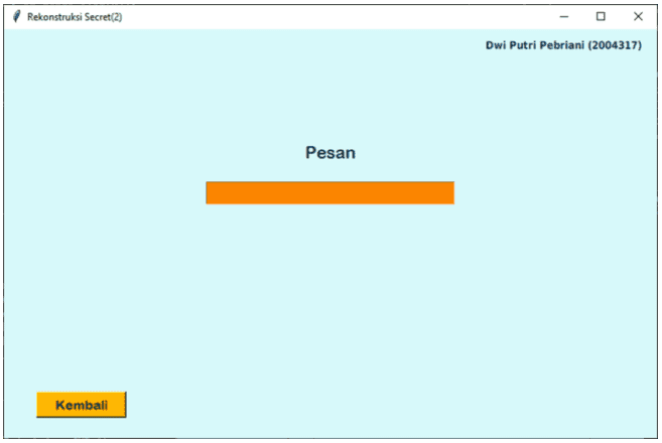
Jika *user* menekan "Rekonstruksi *Secret*" Gambar 9 ditampilkan. Pada tampilan berikut *user* memasukkan informasi yang diberikan *user* yaitu , p, a, b, dan *seed*. Kemudian *user* menekan "Buka QR Code", diarahkan untuk memilih t *share QR Code* secara bersamaan seperti pada Gambar 10. Setelah *QR Code* ditampilkan pada Gambar 9, *user* menekan "Selanjutnya" dan tunggu beberapa saat untuk memperoleh pesan asli yang akan ditampilkan pada Gambar 11. Pada masa transisi ini program melakukan tiga proses yaitu mengunduh *stego image* dari ImgBB, mengekstraksi *stego image*, dan merekonstruksi nilai M menggunakan interpolasi Lagrange.



Gambar 9. Rekonstruksi *secret* pada program (1)



Gambar 10. *Folder* berisi kumpulan *share QR Code*



Gambar 11. Rekonstruksi *secret* pada program (2)

Original Article

Validasi Program Aplikasi

Validasi program aplikasi dilakukan dengan tiga cara. Validasi pertama yaitu membandingkan hasil share Skema (t, w) pada program dengan hasil perhitungan Microsoft Excel. Validasi kedua yaitu membandingkan pesan yang menjadi masukkan pada konstruksi *share* sama dengan pesan yang dihasilkan pada rekonstruksi *secret*. Validasi terakhir yaitu menguji kualitas setiap *stego image* berdasarkan parameter PSNR (*Peak Signal to Noise Ratio*).

PSNR merupakan perbandingan antara nilai maksimum dari sinyal dan besarnya *noise* yang berpengaruh pada sinyal tersebut. PSNR diukur dalam satuan desibel (dB). Semakin besar nilai parameter PSNR, maka *stego image* dan *cover image* semakin mirip. Berikut persamaan untuk menghitung PSNR [11]:

$$\text{PSNR} = 10.log_{10}\left(\frac{MAX_I^2}{MSE}\right) \qquad (3)$$

Nilai MSE dihitung menggunakan persamaan berikut:

$$\text{MSE} = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}|I(i,j) - K(i,j)|^2 \qquad (4)$$

dengan
MAX_I : maksimum nilai *pixel*,
m : panjang citra (dalam *pixel*),
n : lebar citra (dalam *pixel*),
(i, j) : koordinat,
I : *cover image*,
K : *stego image*.

Berikut kualifikasi kualitas citra berdasarkan nilai PSNR:

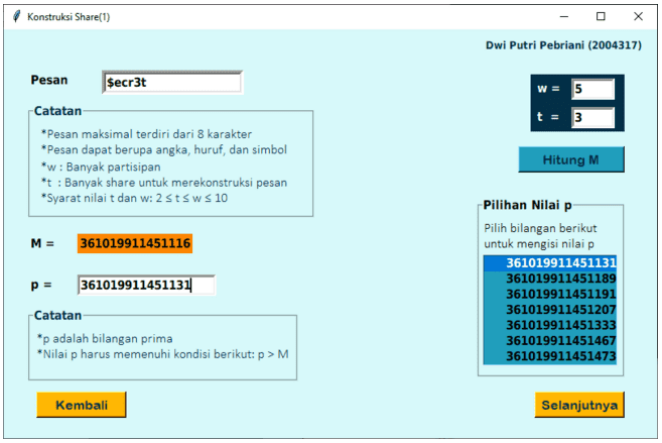
TABEL 1. Kualitas citra berdasarkan PSNR

PSNR	Kualitas
60 dB	*Excellent*
50 dB	*Good*
40 dB	*Reasonable*
30 dB	*Poor*
20 dB	*Unusable*

Validasi diterapkan pada contoh berikut. Pesan yang dimasukkan adalah "$ecr3t" dan *share* dikonstruksi dengan Skema (3,5). Berdasarkan ASCII, nilai desimal dari karakter pesan adalah sebagai berikut:

a. "$" = 36
b. "e" = 101
c. "c" = 99
d. "r" = 114
e. "3" = 51
f. "t" = 116

Dengan menggabungkan nilai desimal tersebut diperoleh M = 361019911451116. Kemudian dipilih p = 361019911451131.



Gambar 12. Konstruksi *share* Skema (3,5) (1)

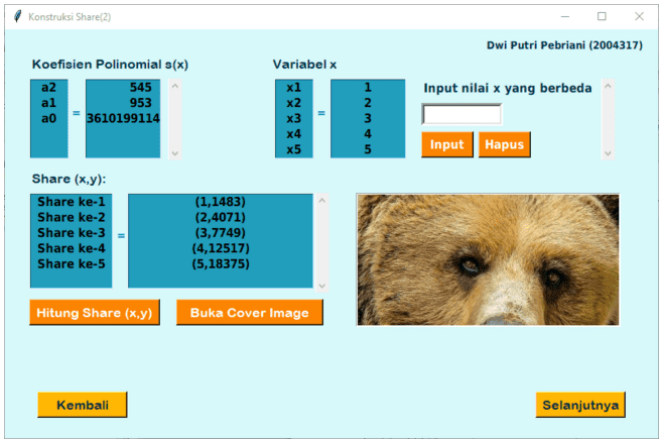
Karena $t = 3$, maka program memberikan dua bilangan bulat acak yaitu $a_1 = 953$ dan $a_2 = 545$ sehingga diperoleh polinom:

$$s(x) = y = 361019911451116 + 953x + 545x^2 \ (mod\ 361019911451131) \qquad (5)$$

Karena $w = 5$, dipilih 5 bilangan bulat acak, misalkan
$x_1 = 1, x_2 = 2, x_3 = 3, x_4 = 4, x_5 = 5$
Substitusikan $x_i, 1 \le i \le 5$ pada persamaan (5) untuk memperoleh *share* ke-i. Dapat dilihat bahwa *share* yang ditampilkan program Gambar 13 sama dengan *share* yang dihitung menggunakan Microsoft Excel (Gambar 14). Hal ini berarti program telah memenuhi validasi pertama.

Published by: Lembaga Penelitian dan Pengabdian Masyarakat (LPPM)
Institut Teknologi Sumatera, Lampung Selatan, Indonesia

Original Article



Gambar 13. Konstruksi *share* Skema (3,5) (2)

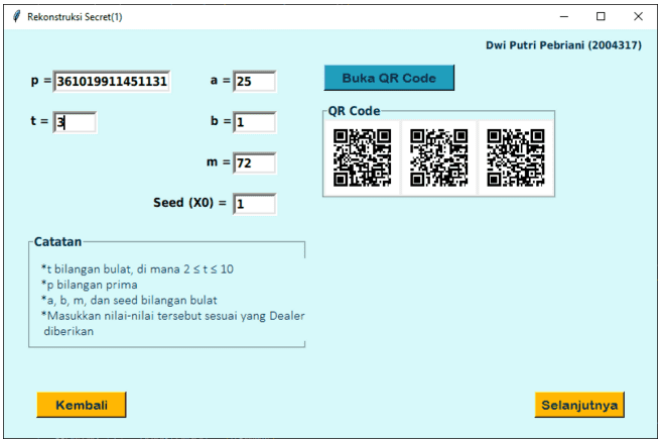
	A	B	C	D	E	F	G	H
4								
5		Koefisien			i	xi	yi (mod p)	
6		a0 = M	361019911451116		1	1	1483	
7		a1	953		2	2	4071	
8		a2	545		3	3	7749	
9					4	4	12517	
10		p			5	5	18375	
11		361019911451131						
12								

Gambar 14. Perhitungan Skema (3,5) pada Microsoft Excel

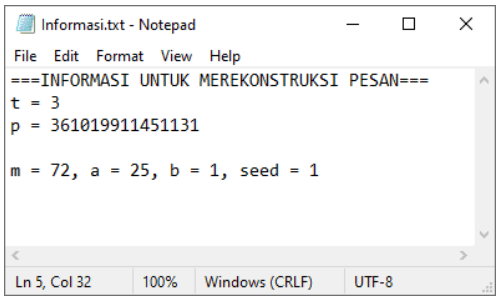
Nilai $y_i \ (mod\ p)$ pada Gambar 14 diperoleh dari formula berikut:

a. $y_1 = MOD(\$C\$6 + (\$C\$7 * F6) + (\$C\$8 * F6\text{^}2); \$B\$11)$
b. $y_2 = MOD(\$C\$6 + (\$C\$7 * F7) + (\$C\$8 * F7\text{^}2); \$B\$11)$
c. $y_3 = MOD(\$C\$6 + (\$C\$7 * F8) + (\$C\$8 * F8\text{^}2); \$B\$11)$
d. $y_4 = MOD(\$C\$6 + (\$C\$7 * F9) + (\$C\$8 * F9\text{^}2); \$B\$11)$
e. $y_5 = MOD(\$C\$6 + (\$C\$7 * F10) + (\$C\$8 * F10\text{^}2); \$B\$11)$

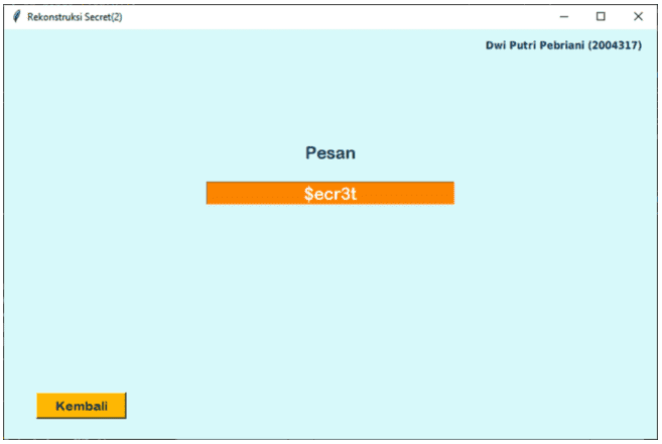
Selanjutnya, tiga *share QR Code* dari hasil konstruksi *share* tersebut menjadi masukkan rekonstruksi *secret*. Pada Gambar 15 nilai-nilai yang dimasukkan diperoleh dari *file* Informasi.txt pada Gambar 16. Dapat dilihat pada Gambar 17, pesan yang dihasilkan sama dengan pesan yang dimasukkan saat konstruksi *share* yaitu "$ecr3t". Hal ini berarti program telah memenuhi validasi kedua.



Gambar 15. Rekonstruksi *secret* Skema (3,5) (1)



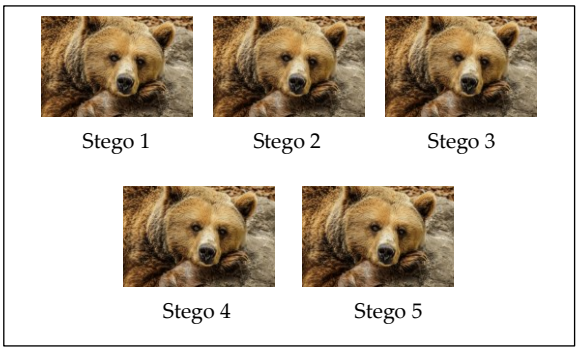
Gambar 16. *File* Informasi.txt Skema (3,5)



Gambar 17. Rekonstruksi *secret* Skema (3,5) (2)

Gambar 18 merupakan 5 *stego image* yang tersimpan dari konstruksi *share* Skema (3,5). Nilai PSNR dari kelima *stego image* dapat dilihat pada Tabel 2, di mana memiliki nilai PSNR lebih dari 60 dB. Berdasarkan Tabel 1 nilai tersebut menunjukkan bahwa kualitas stego image termasuk sangat baik, artinya *stego image* yang dihasilkan sangat mirip dengan *cover image*

sehingga dapat menghindari kecurigaan adanya informasi yang disisipkan dalam citra.



Gambar 18. 5 (lima) buah *stego image* Skema (3,5)

TABEL 2. Hasil PSNR *Stego Image* dari Skema (3,5)

Gambar	PSNR
Stego 1	67, 99 dB
Stego 2	67,71 dB
Stego 3	67,71 dB
Stego 4	67,74 dB
Stego 5	68,06 dB

Karena ketiga validasi telah terpenuhi, yaitu hasil *share* dari perhitungan program sama dengan perhitungan Microsoft Excel, *QR Code* dapat dikembalikan menjadi pesan semula, serta kualitas setiap *stego image* sangat mirip dengan *cover image*, maka program aplikasi dinyatakan dapat berjalan sebagaimana mestinya.

Kesimpulan

Kombinasi kriptografi *Shamir Secret Sharing* dan steganografi citra *Least Significant Bit* pada penelitian dapat meminimalisir adanya kriptanalisis, karena membutuhkan kumpulan *QR Code* dan perlu mencari keberadaan *share* dalam *QR Code* yang memuat *stego image*, serta perlu mengetahui *stego key* untuk dapat merekonstruksi pesan rahasia. Program aplikasi Kriptografi *Shamir Secret Sharing* dan Steganografi Citra RGB *Least Significant Bit* dibuat menggunakan Python sebagai implementasi kombinasi tersebut. Program dapat mengonstruksi *share* dalam bentuk *QR Code* sebanyak w buah, di mana $w \leq 10$, dari pesan maksimal 8 karakter yang terdiri dari angka, huruf, atau simbol. Program juga dapat merekonstruksi pesan semula dari t buah *QR* Code, di mana $2 \leq t \leq w$.

Konflik Kepentingan

Tidak ada konflik kepentingan yang dinyatakan.

Referensi

[1] Humaira, A. F., Marwati, R., & Yulianti, K., " Implementasi Kriptografi *Secret Sharing Scheme* dan Steganografi *Audio Least Significant Bit* (LSB)," JMT (Jurnal Matematika Terapan), vol. 5, no. 1, pp 1-11, February 2023 .

[2] Yusup, I. M., Carudin, C., & Purnamasari, I., "Implementasi Algoritma *Caesar Cipher* dan Steganografi *Least Significant Bit* Untuk *File* Dokumen," Jurnal Teknik Informatika dan Sistem Informasi, vol. 6, no. 3, pp. 434-441, December 2020.

[3] Sinaga, R., Purba, S., & Siburian, R. M., "APLIKASI PEMAHAMAN DAN PENERAPAN *FAST* (k,n) *THRESHOLD SECRET SHARING SCHEME*," Jurnal Teknologi, Informasi dan Industri, vol. 3, no. 1, pp. 76-83, 2023.

[4] Chuang, J. C., Hu, Y. C., & Ko, H. J., "*A Novel Secret Sharing Technique Using QR Code*," International Journal Of Image Processing, vol. 4, no. 5, pp. 468-475, 2010.

[5] Djuwitaningrum, E. R., & Apriyani, M., "Teknik Steganografi Pesan Teks menggunakan Metode *Least Significant Bit* dan Algoritma *Linear Congruential Generator*," JUITA: Jurnal Informatika, vol. 4, no. 2, pp. 79-85, 2017.

[6] Singh, P., "A Comparative Study of Audio Steganography Techniques," International Research Journal of Engineering and Technology (IRJET), vol. 3, no. 4, pp.580-585, 2016.

[7] Az-Zahra, F., Marwati, R., & Sispiyati, R., "Implementasi *QR Code* dengan Algoritma SHA-256 dan RSA yang Ditingkatkan untuk Autentikasi Dokumen Digital," Jurnal EurekaMatika, vol. 12, no. 1, pp. 11-22, 2023.

[8] Ispandi, I., Fauzi, A., & Sugiono, S., "Stegnografi Menggunakan Metode *Least Significant Bit* dan *Quick Response Code* (QR-Code)," JURIKOM (Jurnal Riset Komputer), vol. 6, no. 5, October 2019.

[9] Kurniasih, F., Marwati, R., & Sispiyati, R., "Penyisipan Pesan Rahasia pada Gambar dengan Menggunakan *Affine Cipher* dan *Least Significant Bit-2* (LSB-2)," JEM (Jurnal Eureka Matika, vol. 11, no. 2, pp. 79-88, 2023.

[10] Fahrizal, M., & Solichin, A., "Pengamanan *M-Commerce* Menggunakan *One Time Password* Metode *Pseudo Random Number Generator* (PRNG)," Rabit: Jurnal Teknologi dan Sistem Informasi Univrab, vol. 5, no. 2, pp. 108-116, July 2020.

[11] Nurfitri, K., & Suyanto, M., "Penilaian Kualitas Pemampatan Citra pada Aplikasi-Aplikasi *Instant Messenger*," MULTITEK INDONESIA, vol. 10, no. 2, February 2016.

Published by: Lembaga Penelitian dan Pengabdian Masyarakat (LPPM)
Institut Teknologi Sumatera, Lampung Selatan, Indonesia