Analisis Algoritma RSA Dan LSB pada *One Time Password* untuk *Financial Technology*

Galuh Sitoresmi[1], Wijanarto[2]
Jurusan Teknik Informatika, Fakultas Ilmu Komputer UDINUS, Semarang
Universitas Dian Nuswantoro, Jl. Imam Bonjol 207, telp: (+6224) 3517261
e-mail: [1]galuhsitoresmi2501@gmail.com, [2]wijanarto@dsn.dinus.ac.id

Abstrak

Financial Technology (fintech) merupakan inovasi dibidang pelayanan keuangan yang memanfaatkan teknologi untuk meningkatkan pelayanan dan mempermudah transaksi. Karena berkaitan dengan keuangan, maka keamanan dalam fintech harus kuat dan terjamin. Penyedia layanan fintech seperti GoPay, Paypro, dan T-cash menggunakan One Time Password (OTP) dengan format 4 dan 6 digit bilangan acak numerik yang dikirimkan melalui SMS sebagai solusi autentikasi. Namun kode tersebut mudah terbaca sehingga rawan disalahgunakan oleh orang tidak bertanggung jawab dan dapat merugikan pengguna. Kriptografi merupakan teknik menyandikan informasi ke dalam karakter tertentu dan disusun acak sehingga sulit dimengerti. RSA merupakan algoritma kriptografi yang andal karena menggunakan kunci berbeda untuk menyandikan informasi. Metode LSB merupakan teknik steganografi sederhana untuk menyembunyikan informasi ke dalam citra sehingga mempersulit informasi dapat dibaca dan disalahgunakan. Dengan menerapkan algoritma RSA dan teknik LSB maka dapat meningkatkan keamanan kode OTP, karena kode OTP yang tersembunyi di dalam citra tidak dapat diketahui oleh mata manusia. Hasil dari penelitian ini, kedua algoritma tersebut melakukan keseluruhan proses sangat cepat, durasi rata-rata untuk memproses kode OTP format 4 digit yaitu 0,6738 detik, sedangkan format 6 digit sebesar 0,7005 detik. Hasil kualitas citra stego dengan pengujian MSE dan PSNR memiliki nilai rata-rata yang tinggi yaitu MSE sebesar 0,000531 dan PSNR sebesar 83,110348 dB.

Kata kunci: *Algoritma, Kriptografi, Steganografi, Financial Technology.*

Abstract

Financial Technology (fintech) is an innovation in the field of financial services that utilizes technology to improve services and facilitate transactions. Because it is related to finance, security in fintech must be strong and guaranteed. Fintech service providers such as GoPay, Paypro, and T-cash use One Time Password (OTP) with a 4 and 6 digit numeric random number format sent via SMS as an authentication solution. However, the code is easy to read so that it is prone to being misused by irresponsible people and can harm users. Cryptography is the technique of encoding information into certain characters and arranged randomly so that it is difficult to understand. RSA is a reliable cryptographic algorithm because it uses different keys to encode information. The LSB method is a simple steganography technique to hide information into an image, making it difficult for information to be read and misused. By applying the RSA algorithm and LSB technique, it can improve the security of the OTP code, because the OTP code hidden in the image cannot be known by the human eye. The results of this study, the two algorithms do the whole process very quickly, the average duration to process the 4-digit format OTP code is 0.6738 seconds, while the 6-digit format is 0.7005 seconds. The results of stego image quality with MSE and PSNR testing have a high average value, namely MSE of 0.000531 and PSNR of 83.110348 dB.

Keywords*: Algorithm, Cryptography, Steganography, Financial Technology.*

1. Pendahuluan

Perusahaan *financial technology* (*fintech*) merupakan perusahaan yang menerapkan inovasi baru dibidang pelayanan keuangan dengan memanfaatkan teknologi untuk meningkatkan pelayanan dan mempermudah pengguna dalam bertransaksi, manajemen aset, asuransi, dan menggunakan layanan keuangan [1]. Contoh perusahaan yang menerapkan *fintech* adalah Gojek dengan GoPay, Telkomsel dengan T-cash, Indosat dengan Paypro, dan sebagainya. Dalam sepuluh tahun, tercatat pertumbuhan pengguna

fintech di Indonesia naik tajam menjadi 78% dari sebelumnya hanya 7% pada tahun 2007, sebanyak 43% dari total jumlah 135-140 perusahaan *fintech* yang tercatat merupakan perusahaan di bidang pembayaran [2].

Karena berkaitan dengan keuangan, maka keamanan sistem pada *fintech* harus kuat dan terjamin. Penyedia layanan *fintech* menerapkan *two factor authentication* menggunakan *One Time Password* (OTP) sebagai solusi metode autentikasi. OTP merupakan *password* yang hanya berlaku satu kali pemakaian untuk masuk ke dalam akun pengguna atau untuk persetujuan transaksi *online* [3].

Urutan angka yang muncul pada OTP dibangkitkan berdasarkan waktu dengan perhitungan rumus tertentu yang dikirimkan oleh server penyedia layanan melalui *Short Message Service* (SMS) ke nomor *handphone* pengguna yang didaftarkan pada layanan *fintech* [3]. Dalam SMS tersebut, ditampilkan kode OTP sebanyak 4 atau 6 digit yang mudah dibaca. Walaupun kode OTP hanya berlaku untuk satu kali pemakaian, namun kode OTP ini merupakan hal yang bersifat rahasia yang hanya boleh diketahui oleh pengguna dan penyedia layanan. Oleh karena itu, kode verifikasi ini tidak boleh diketahui oleh orang lain agar tidak disalahgunakan. Contoh penyalahgunaan kode OTP, seperti pada kasus penipuan akun *fintech*, yaitu penipu meminta kode verifikasi kepada pemilik akun dengan alasan kesalahan memasukkan nomor *handphone* karena nomor hampir mirip. Setelah kode diberikan, saldo uang dalam akun pengguna lenyap [4] [5].

Kasus tersebut sangat merugikan pengguna. Oleh karena itu, diperlukan analisis terhadap kode OTP dengan mengimplementasikan teknik kriptografi. Kriptografi adalah teknik menyandikan informasi ke dalam bentuk karakter tertentu dan disusun secara acak sehingga tidak mudah dimengerti. RSA (*Rivest Shamir Adleman*) merupakan algoritma kriptografi yang andal karena menggunakan kunci yang berbeda pada proses enkripsi dan dekripsi. Algoritma RSA memiliki tingkat keamanan yang tinggi dibanding metode lain pada teknik kunci asimetris karena RSA menggunakan pemfaktoran bilangan prima untuk membangkitkan kunci [6].

Karena teknik kriptografi hanya menyamarkan informasi namun tidak disembunyikan, jadi masih dapat dibaca pihak lain sehingga dapat menimbulkan kecurigaan. Oleh karena itu, maka digunakan teknik steganografi untuk memaksimalkan keamanan[13, 14]. Metode *Least Significant Bit* (LSB) adalah teknik steganografi yang sederhana dan efektif untuk menyembunyikan informasi ke dalam media lain, sehingga orang awam tidak tahu jika terdapat informasi rahasia yang terdapat di dalam media tersebut [7,15]. LSB bekerja menyisipkan informasi dengan mengganti bit terakhir dari citra *cover* dengan bit informasi. Teknik LSB memiliki keunggulan yaitu citra hanya mengalami sedikit penurunan kualitas setelah pesan disisipkan ke dalam citra *cover*, jadi untuk penyisipan informasi pada citra lebih baik menggunakan metode LSB [8].

LSB bekerja dengan baik pada citra *lossless* dan kurang bagus jika diaplikasikan pada citra lossy karena kompresi citra *lossy* dapat menghilangkan informasi yang terdapat dalam citra [9,12]. Kompresi *lossless* digunakan pada citra yang memiliki informasi penting yang tidak boleh hilang atau rusak. BITMAP (*.bmp) merupakan salah satu format citra yang termasuk ke dalam kompresi *lossless*. Format citra *bitmap* memiliki hasil terbaik dengan nilai MSE kecil dan PSNR yang tinggi serta waktu rata-rata pemrosesan yang tidak terlalu lambat [10, 11].

Penelitian ini bertujuan untuk melakukan analisis terhadap penerapan algoritma kriptografi RSA dan teknik steganografi LSB untuk kode *One Time Password* pada akun pengguna layanan *financial technology* yang melakukan transaksi.

2. Metode Penelitian

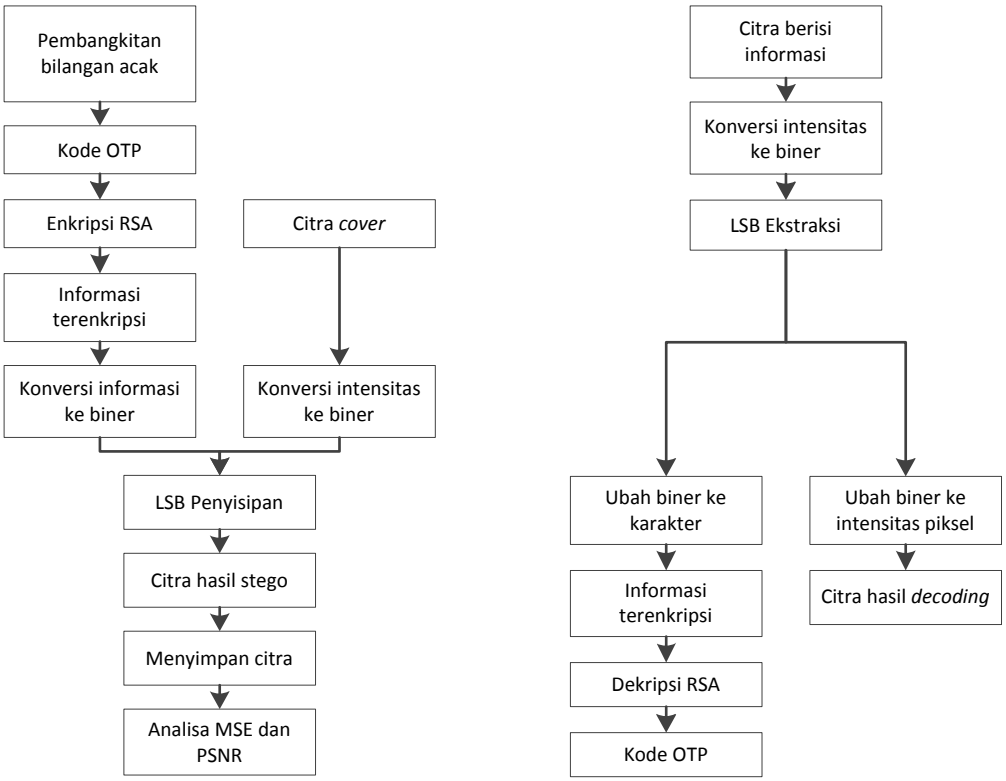
Dalam penelitian ini, disajikan gambaran umum yang berupa langkah-langkah dalam menyelesaikan masalah. Hal ini terkait dengan analisis yang ingin dihasilkan oleh metode seperti gambar 1 di bawah ini. Proses enkripsi dan penyisipan informasi rahasia yang diusulkan pada gambar 1.a adalah:

1. Menentukan jumlah digit bilangan acak 4 atau 6 digit yang dihasilkan menggunakan fungsi *rand()* dari Matlab sebagai kode OTP.
2. Didapat bilangan acak dengan format 4 atau 6 digit sebagai kode OTP dan mempersiapkan citra *cover* yang akan digunakan untuk menyisipkan informasi rahasia ke dalam citra.
3. Mengenkripsi kode OTP menggunakan algoritma kriptografi RSA sehingga dihasilkan kode OTP yang telah terenkripsi.
4. Mengubah kode OTP yang telah terenkripsi dan intensitas tiap piksel citra *cover* ke dalam biner.
5. Melakukan proses penyisipan kode OTP terenkripsi ke dalam citra *cover* menggunakan teknik steganografi LSB sehingga dihasilkan citra hasil stego.
6. Kemudian citra hasil stego disimpan.

7. Melakukan evaluasi dengan menggunakan nilai MSE dan PSNR pada citra hasil stego terhadap citra asli.

Proses pengambilan kembali dan dekripsi informasi rahasia yang diusulkan dari gambar 1.b adalah:

1. Mengambil citra hasil stego yang berisi informasi rahasia terenkripsi.
2. Mengubah intensitas tiap piksel citra ke dalam biner.
3. Melakukan pengambilan informasi rahasia yang terdapat dalam citra hasil stego menggunakan teknik steganografi LSB.
4. Dihasilkan dua macam data yaitu data biner informasi rahasia dan data biner intensitas piksel citra. Ubah data biner informasi rahasia ke dalam karakter informasi rahasia terenkripsi dan ubah data biner intensitas piksel ke dalam intensitas citra.
5. Dihasilkan informasi rahasia terenkripsi dan citra hasil *decoding*.
6. Melakukan dekripsi terhadap informasi rahasia menggunakan algoritma kriptografi RSA.
7. Dihasilkan informasi rahasia kode OTP dengan format 4 atau 6 digit.



Gambar 1. Metode yang diusulkan (a) Enkripsi, (b) Dekripsi.

3. Hasil dan Pembahasan

Berikut adalah paparan hasil yang dapat disajikan dalam penelitian ini sesuai dengan metode atau langkah yang diusulkan pada bagian sebelumnya.

3.1. Proses Pembangkitan Kunci RSA

Proses pada algoritma kriptografi RSA diawali dengan pembangkitan pasangan kunci. Langkah untuk membangkitkan pasangan kunci RSA adalah dengan persamaan dalam [6,7,11]:

1. Memilih sembarang dua bilangan bulat positif prima yang disimpan dalam *variable p* dan *q*. Dipilih nilai $p = 11$ dan $q = 31$.
2. Menghitung nilai n menggunakan persamaan:
   $n = p * q = 11 * 31 = 341$
3. Menghitung $\phi(n)$ dengan menggunakan persamaan:
   $\phi(n) = (p - 1) * (q - 1) = (11 - 1) * (31 - 1) = 300$

4. Memilih kunci publik *e* yang relatif prima terhadap nilai ϕ(*n*). Nilai *e* yang dipilih yaitu *e* = 17, karena 17 relatif prima dengan 300.
5. Membangkitkan kunci privat *d* menggunakan persamaan:
$$d = \frac{1 + k\phi(n)}{e} = \frac{1 + (3x300)}{17} = 53$$

Dengan mencoba nilai *k* = 1, 2,3, ..., didapat nilai *d* yang bulat yaitu *d* = 53. Sehingga dari perhitungan pembangkitan kunci didapatkan pasangan kunci yaitu:

Kunci publik: (*e* = 17, *n* = 341) untuk proses enkripsi, dan

Kunci privat: (*d* = 53, *n* = 341) untuk proses dekripsi

3.2. Proses Enkripsi RSA dan Penyisipan LSB

3.2.1. Proses Enkripsi RSA

Plainteks yang digunakan untuk contoh perhitungan enkripsi menggunakan kode OTP dengan format 6 digit yang diperoleh dari pembangkitan bilangan secara *random* menggunakan fungsi *rand()* dari Matlab yaitu: *m* = **6 4 8 7 9 5**

Kemudian *plainteks* m dipecah menjadi beberapa blok $m_1, m_2, ..., m_n$.

$m_1 = 6$ $m_3 = 8$ $m_5 = 9$
$m_2 = 4$ $m_4 = 7$ $m_6 = 5$

Selanjutnya setiap blok *plainteks m*, di enkripsi menggunakan kunci publik yaitu *e*=17 dan *n*=341. Proses perhitungan enkripsi menggunakan persamaan 1 sebagai berikut:

$$c_i = m_i^e \bmod n \quad (1)$$

Proses perhitungan untuk setiap blok *plainteks* yaitu:

$c_1 = 6^{17} \bmod 341$
$= 6^{(16+1)} \bmod 341$
$= (6^{16} \times 6^1) \bmod 341$
$= ((6^{16} \bmod 341) \times (6^1 \bmod 341)) \bmod 341$
$= (335 \times 6) \bmod 341 =$ **305**
.....
$c_6 = 5^{17} \bmod 341$
$= 5^{(16+1)} \bmod 341$
$= (5^{16} \times 5^1) \bmod 341$
$= ((5^{16} \bmod 341) \times (5^1 \bmod 341) \bmod 341$
$= (5 \times 5) \bmod 341 =$ **25**

Dari proses enkripsi *plainteks* menggunakan algoritma kriptografi RSA di atas, maka diperoleh *cipherteks* *c* = **305 16 2 204 81 25**

3.2.2. Proses Penyisipan LSB

Pada bagian ini akan mengimplementasikan teknik LSB dengan langkah-langkah sebagai berikut [8,9]:

1. Mempersiapkan informasi rahasia
   Misalkan *cipherteks* kode OTP yang akan disisipkan ke dalam citra *cover* diperoleh dari enkripsi *plainteks* pada proses sebelumnya yaitu *c* = 305 16 2 204 81 25. Pada proses LSB ini, *cipherteks* yang masih dalam bentuk integer, masing-masing diubah menjadi biner 16 bit:
   0000 0001 0011 0001 0000 0000 0001 0000 0000 0000 0000 0010
   0000 0000 1100 1100 0000 0000 0101 0001 0000 0000 0001 1001
2. Mempersiapkan citra *cover* sebagai media penyisipan informasi rahasia.
   Proses penyisipan menggunakan teknik steganografi LSB, tiap bit informasi rahasia akan menggantikan bit terakhir tiap piksel citra *cover* [16], di mana jumlah piksel yang digunakan untuk penyisipan adalah sejumlah panjang informasi rahasia yang akan disisipkan. Berikut citra *cover* yang akan digunakan:

Gambar 2. Citra *cover*.

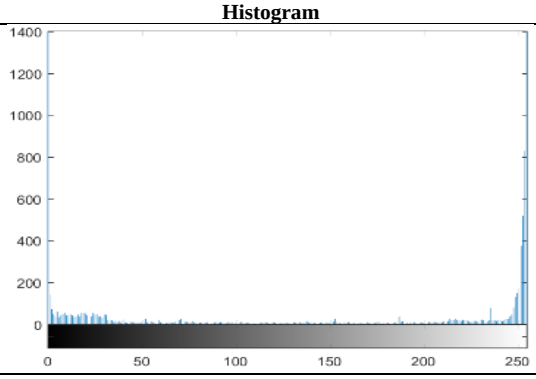
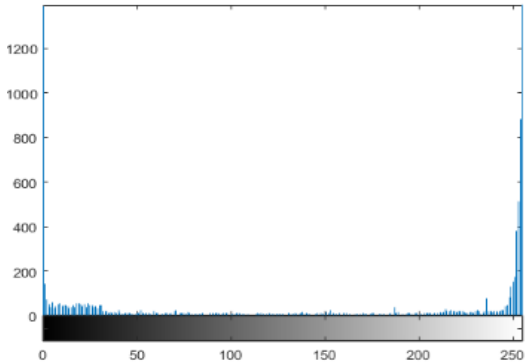
3. Melakukan proses steganografi menggunakan teknik LSB.
Berikut contoh proses penyisipan serta perubahan nilai intensitas piksel sebelum dan sesudah disisipi oleh bit *cipherteks* yang disajikan dalam tabel 1 di bawah ini:

Tabel 1. Proses penyisipan.

Nilai piksel sebelum disisipi		Bit cipherteks		Nilai piksel setelah disisipi	
Desimal	Biner			Biner	Desimal
255	1111 1111	0	menjadi	1111 1110	254
255	1111 1111	0		1111 1110	254
255	1111 1111	1		1111 1111	255
255	1111 1111	0		1111 1110	254

4. Menyimpan citra hasil penyisipan
Citra hasil penyisipan kemudian disimpan. Perbandingan citra *cover* dan citra stego dapat juga dilihat dari histogramnya, yang dapat dilihat pada tabel 2 berikut:

Tabel 2. Perbandingan citra *cover* dan citra stego.

Jenis citra	Citra	Histogram
Cover	 cover_1.bmp	
Stego	 stego_91.bmp	

Berikut hasil perbandingan histogram citra *cover* dan citra stego pada layer Red pada Tabel 3:

Tabel 3. Perbandingan histogram citra pada *layer red.*

Range	**Citra *Cover***	**Citra Stego**
Gradien Warna	0 - 255	0 -255

Mean	194,17	194,17
Median	255	255
Standar Deviasi	**99,70**	**99,70**
Pixels	16384	16384
Percent	100	100

3.3. Proses Ekstraksi LSB dan Dekripsi RSA

3.3.1. Proses Ekstraksi LSB

1. Mempersiapkan citra yang mengandung informasi.
   Pada proses ekstraksi dibutuhkan citra yang mengandung informasi rahasia yang akan dikeluarkan.
2. Membaca intensitas piksel yang berisi informasi
   Dibutuhkan informasi panjang informasi rahasia yang disisipkan 4 atau 6 digit kode OTP, serta panjang biner informasi rahasia pada setiap blok, di mana pada penelitian ini setiap blok informasi rahasia memiliki panjang biner 16 bit. Hal ini diperlukan untuk mempermudah pada proses ekstraksi.
3. Mengambil bit terakhir nilai piksel
   Nilai piksel diubah ke biner, kemudian bit terakhir atau bit paling akhir dari nilai piksel citra diambil dan disusun kembali menjadi bit-bit biner informasi *cipherteks* kode OTP. Contoh prosesnya dapat dilihat pada Tabel 4 berikut:

Tabel 4. Proses Ekstraksi LSB.

Nilai piksel citra stego		Bit cipherteks terekstrak		Nilai piksel setelah diekstraksi	
Desimal	**Biner**			**Biner**	**Desimal**
254	1111 1110	→ 0	menjadi	1111 1111	255
254	1111 1110	→ 0		1111 1111	255
255	1111 1111	→ 1		1111 1111	255
254	1111 1110	→ 0		1111 1111	255

4. Hasil Ekstraksi
   Dari proses ekstraksi, dihasilkan dua data yaitu biner citra hasil ekstraksi dan biner *cipherteks*. Biner citra hasil ekstraksi kemudian dijadikan desimal sehingga didapat nilai piksel citra. Hasil ekstraksi *cipherteks* dalam biner yang dihasilkan yaitu:
   C = **0000 0001 0011 0001 0000 0000 0001 0000 0000 0000 0000 0010**
   0000 0000 1100 1100 0000 0000 0101 0001 0000 0000 0001 1001
   Hasil di atas masih berupa satu blok *cipherteks*, kemudian ubah setiap 16 bit biner menjadi desimal, sehingga diperoleh C = 305 16 2 204 81 25

3.3.2. Proses Dekripsi RSA

Cipherteks yang digunakan untuk contoh perhitungan dekripsi menggunakan *cipherteks* yang diperoleh dari proses ekstraksi sebelumnya, yaitu *c* = 305 16 2 204 81 25. Kemudian *cipherteks c* dipecah kembali menjadi beberapa blok c_1, c_2, ..., c_n, menjadi:

$c_1 = 305$ $c_3 = 2$ $c_5 = 81$
$c_2 = 16$ $c_4 = 204$ $c_6 = 25$

Selanjutnya setiap blok *cipherteks c*, didekripsi menggunakan kunci privatnya yang didapat dari proses pembangkitan kunci sebelumnya yaitu d = 53 dan n = 341. Proses perhitungan dekripsi menggunakan persamaan (1), Proses perhitungan untuk setiap blok *plainteks* yaitu:

$m_1 = 305^{53} \bmod 341$
$= 305^{(32+16+4+1)} \bmod 341$
$= (305^{32} \times 305^{16} \times 305^{4} \times 305^{1}) \bmod 341$
$= ((305^{32} \bmod 341) \times (305^{16} \bmod 341) \times (305^{4} \bmod 341) \times (305^{1} \bmod 341) \bmod 341$
$= (273 \times 36 \times 191 \times 305) \bmod 341$
$= (((273 \times 36) \bmod 341) \times ((191 \times 305) \bmod 341)) \bmod 341$
$= (280 \times 285) \bmod 341 =$ **6**

……

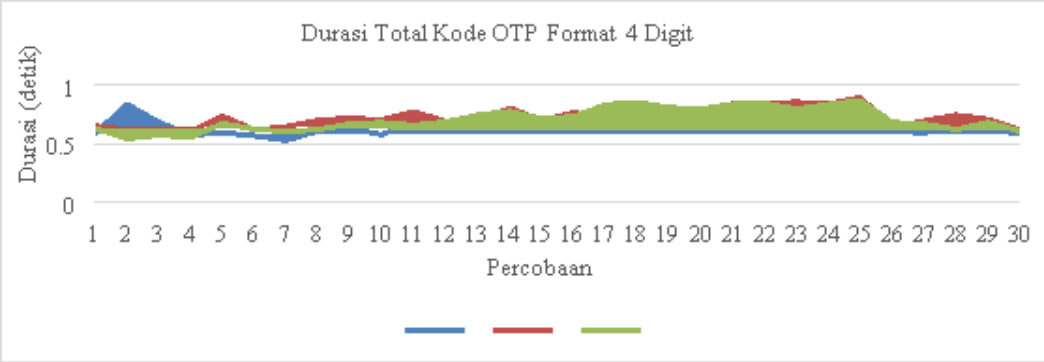
$m_6 = 25^{53} \bmod 341$
$= 25^{(32+16+4+1)} \bmod 341$
$= (25^{32} \times 25^{16} \times 25^{4} \times 25^{1}) \bmod 341$
$= ((25^{32} \bmod 341) \times (25^{16} \bmod 341) \times (25^{4} \bmod 341) \times (25^{1} \bmod 341)) \bmod 341$

= (284 x 25 x 180 x 25) mod 341
= (((284 x 25) mod 341) x ((180 x 25) mod 341)) mod 341
= (280 x 67) mod 341 = **5**

Sehingga diperoleh *plainteks* kembali dari dekripsi *cipherteks*, yaitu *m* = **6 4 8 7 9 5**

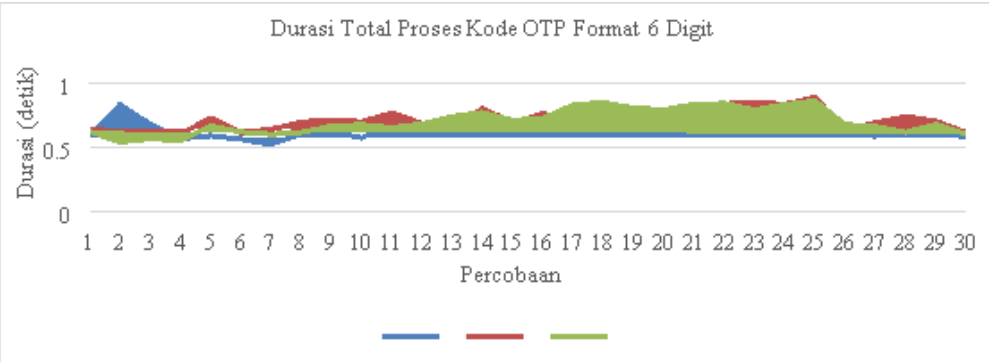
3.4. Hasil Durasi Waktu Pemrosesan

Berikut diberikan grafik durasi waktu yang diperlukan untuk melakukan keseluruhan proses menggunakan metode yang diusulkan. Dari grafik pada Gambar 3 di atas, dapat dilihat melalui garis yang menunjukkan durasi total proses enkripsi, penyisipan, ekstraksi, dan dekripsi pada kode OTP dengan format 4 digit yang dilakukan sebanyak 30 kali. Durasi tertinggi pada percobaan ini yaitu 0,8847 detik yang merupakan durasi dari citra "cover_2.bmp" pada percobaan ke-30. Untuk waktu terendah yaitu 0,5308 detik, merupakan durasi dari citra "cover_1.bmp" pada percobaan ke-16. Durasi waktu mengalami kenaikan dan penurunan pada rentang 0,5300 – 08850 detik. Waktu terlama yang dibutuhkan untuk melakukan total proses enkripsi, penyisipan, ekstraksi, dan dekripsi pada kode OTP dengan format 4 digit adalah 0,8847 detik. Sedangkan waktu tercepat adalah 0,5308 detik, serta rata-rata durasi total yang dibutuhkan adalah 0,6738 detik.



Gambar 3. Grafik durasi total proses kode OTP format 4 digit.

Dari grafik pada Gambar 4 di bawah dapat dilihat melalui garis yang menunjukkan durasi total waktu proses enkripsi, penyisipan, ekstraksi, dan dekripsi pada kode OTP dengan format 6 digit yang dilakukan sebanyak 30 kali. Durasi tertinggi pada percobaan ini yaitu 0,8933 detik yang merupakan durasi dari citra "cover_2.bmp" pada percobaan ke-25. Untuk waktu terendah yaitu 0,5282 detik, merupakan durasi dari citra "cover_1.bmp" pada percobaan ke-7. Berdasarkan grafik pada gambar 4.17, durasi waktu mengalami kenaikan dan penurunan pada rentang 0,5250 – 08950 detik. Waktu terlama yang dibutuhkan untuk melakukan total proses enkripsi, penyisipan, ekstraksi, dan dekripsi pada kode OTP dengan format 6 digit adalah 0,8933 detik. Untuk waktu tercepat adalah 0,5282 detik, serta rata-rata durasi total yang dibutuhkan adalah 0,7005 detik.



Gambar 4. Grafik durasi total proses kode OTP format 4 digit.

3.5. Hasil Evaluasi

Bagian ini melakukan evaluasi untuk mengetahui kualitas citra hasil penyisipan terhadap citra *cover* menggunakan nilai MSE dan PSNR. MSE *(Mean Square Error*) digunakan untuk mengetahui tingkat kesalahan atau *error* yang terdapat dalam citra hasil terhadap citra *cover*. Semakin kecil nilai MSE maka semakin bagus kualitas citra hasil. Berikut persamaan 2 perhitungan MSE:

$$MSE = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}\left(S_{xy} - C_{xy}\right)^2 \quad (2)$$

dengan *x* dan *y* adalah koordinat citra, S_{xy} adalah citra stego dan C_{xy} adalah *cover* citra, yang menghasilkan nilai MSE sebesar **0,001363.**

Peak Signal Noise Ratio (PSNR) digunakan untuk membandingkan kualitas citra sebelum dan sesudah disisipi *cipherteks* informasi rahasia kode OTP. Semakin tinggi nilai PSNR sebuah citra hasil maka semakin bagus kualitas dan semakin mirip pula citra itu dengan citra *cover*. Berikut persamaan 3 untuk menghitung nilai PSNR:

$$PSNR = 10\log_{10}\left(\frac{C_{max}^2}{MSE}\right) \quad (3)$$

dalam penelitian ini nilai yang di hasilkan adalah sebesar 76,785468 dan secara keseluruhan distribusi rata-rata nilai MSE dan PSNR disajikan dalam tabel 5 berikut di bawah ini.

Tabel 5. Nilai MSE dan PSNR Terbaik Keseluruhan**.**

Ukuran citra	Format kode OTP	Rata-rata MSE	Rata-rata PSNR
128 x 128	4 digit	0,000946	78,379441
256 x 256		0,000276	83,728935
512 x 512		0,000689	89,749537
128 x 128	6 digit	0,001394	76,692703
256 x 256		0,000397	82,151221
512 x 512		0,000104	87,960252
∑ Rata-rata		**0,000531**	**83,110348**

Pada tabel 5 di atas dapat dilihat rata-rata nilai MSE dan PSNR untuk tiap ukuran citra beserta format kode OTP yang disisipkan dalam citra. Dari tabel 4.34 nilai rata-rata MSE terendah yaitu 0,000104 dari citra berukuran 512 x 512 piksel yang disisipi kode OTP format 6 digit, dan nilai MSE tertinggi yaitu 0,001394 dari citra berukuran 128 x 128 piksel yang disisipi kode OTP format 6 digit. Dari keseluruhan percobaan yang dilakukan maka diperoleh rata-rata nilai MSE sebesar 0,000531. Untuk nilai rata-rata PSNR terendah yaitu 76,692703 dB dari citra berukuran 128 x 128 piksel yang disisipi kode OTP format 6 digit, dan nilai PSNR tertinggi yaitu 89,749537 dB dari citra berukuran 512 x 512 piksel yang disisipi kode OTP format 4 digit. Dari keseluruhan percobaan yang dilakukan maka diperoleh rata-rata nilai PSNR sebesar 83,110348 dB.

Berdasarkan rata-rata nilai MSE dan PSNR pada masing-masing ukuran citra, citra dengan ukuran 512 x 512 piksel yang disisipi kode OTP dengan format 6 digit memiliki kualitas terbaik pada percobaan ini karena memiliki nilai rata-rata MSE yang rendah dan nilai rata-rata PSNR yang tinggi.

4. Kesimpulan

Berdasarkan hasil penelitian yang sudah dilakukan, berikut hasil dari penelitian yang telah dilakukan oleh penulis:

4.1. Hasil Durasi Waktu Pemrosesan

Hasil rata-rata durasi proses enkripsi dengan algoritma RSA untuk 30 kali percobaan pada kode OTP format 4 digit sebesar 0,001953 detik, sedangkan dengan format 6 digit sebesar 0,001870 detik. Hasil rata-rata durasi proses penyisipan dengan LSB pada kode OTP format 4 digit sebesar 0,335004 detik, sedangkan dengan format 6 digit sebesar 0,35076 detik. Hasil rata-rata durasi pada proses ekstraksi dengan LSB pada kode OTP format 4 digit sebesar 0,3346 detik, sedangkan dengan format 6 digit sebesar 0,3467 detik. Hasil rata-rata durasi pada proses dekripsi dengan algoritma RSA pada kode OTP format 4 digit sebesar 0,0014 detik, sedangkan dengan format 6 digit sebesar 0,0016 detik. Hasil rata-rata untuk

melakukan keseluruhan proses enkripsi, penyisipan, ekstraksi dan dekripsi pada kode OTP format 4 digit sebesar 0,6738 detik, sedangkan pada kode OTP format 6 digit sebesar 0,7005 detik.

4.2. Hasil evaluasi MSE dan PSNR

Algoritma kriptografi RSA dan teknik steganografi LSB, sebelumnya digunakan oleh Richard Apau dan Clement Adomako [11]. Hasilnya nilai rata-rata MSE adalah 0,0336 sedangkan nilai rata-rata PSNR adalah 63,6032 dB. Berdasarkan nilai MSE dan PSNR tersebut, penelitian yang dilakukan penulis hasilnya lebih tinggi yaitu MSE sebesar 0,000531 dan nilai PSNR sebesar 83,110348 dB dibanding hasil pada penelitian [11], sehingga penelitian yang dilakukan penulis lebih baik. Serta diperoleh citra dengan ukuran 512 x 512 piksel yang disisipi kode OTP format 6 digit memiliki kualitas terbaik pada percobaan ini, dengan nilai rata-rata MSE 0,000104 serta nilai rata-rata PSNR 87,960252 dB. Panjang dan variasi angka pada kode OTP yang disisipkan serta ukuran citra sangat mempengaruhi nilai MSE dan PSNR.

Dari hasil durasi waktu untuk memproses kode OTP dengan format 4 dan 6 digit menggunakan metode yang diusulkan dapat dikatakan sangat cepat. Karena pada praktiknya dibutuhkan waktu yang cepat untuk mengeksekusi proses ini, sehingga dengan durasi rata-rata yang diperoleh tidak sampai 1 detik menandakan algoritma kriptografi RSA dan teknik steganografi LSB yang diajukan untuk kode OTP berhasil. Dari evaluasi nilai MSE dan PSNR diperoleh nilai yang tinggi, menandakan kualitas citra stego sangat bagus dan hampir mirip citra aslinya. Selain itu, kode OTP yang terdapat dalam citra stego dapat diambil kembali dan didekripsi menjadi kode OTP semula. Jadi, citra stego yang mengandung kode OTP yang dihasilkan dari penelitian ini dapat menjadi solusi untuk menghindarkan pengguna layanan *fintech* dari kejahatan penyalahgunaan kode OTP oleh orang tidak bertanggung jawab. Sehingga penerapan algoritma kriptografi RSA dan teknik steganografi LSB untuk kode OTP yang diajukan penulis berhasil dilakukan.

5. Saran

Dari hasil penelitian yang telah dilakukan oleh penulis, maka diharapkan penelitian selanjutnya dapat mengembangkan beberapa hal berikut:

1. Dilakukan pemilihan piksel yang digunakan untuk penyisipan pada citra *cover*, sehingga lokasi penyisipan lebih acak dan tidak mudah diketahui serta bertujuan untuk mengetahui pengaruh pada hasil MSE dan PSNR
2. Kemungkinan pemilihan parameter nilai p dan q yang lebih besar pada pembangkitan kunci algoritma RSA, sehingga kunci untuk enkripsi dan dekripsi yang dihasilkan memiliki nilai yang lebih besar.

Daftar Pustaka

[1] International Monetary Fund, "Fintech and Financial Services : Initial Considerations," pp. 1–49, 2017.

[2] Cekindo, "Perkembangan Teknologi Finansial (Fintech) di Indonesia," 2017. [Online]. Available: http://www.cekindo.com/id/perkembangan-teknologi-finansial-fintech-di-indonesia.html%0A. [Accessed: 20-Feb-2018].

[3] Shally and G. S. Aujla, "A review of one time password mobile verification," *Int. J. Comput. Sci. Eng. Inf. Technol. Res.*, vol. 4, no. 3, pp. 113–118, 2014.

[4] D. Anastasia, "Waspada Modus Penipuan Baru! Jika Diminta Kode Verifikasi oleh Olshop, Jangan Langsung Percaya - Halaman all - Tribun Jabar," 2017. [Online]. Available: http://jabar.tribunnews.com/2017/07/07/waspada-modus-penipuan-baru-jika-diminta-kode-verifikasi-oleh-olshop-jangan-langsung-percaya. [Accessed: 20-Feb-2018].

[5] S. PUSPITA, "Cerita Pengguna Go-Jek soal Modus Pencurian Saldo Go-Pay - Kompas," 2017. [Online].Available:http://megapolitan.kompas.com/read/2017/06/18/14175361/cerita.pengguna.go-jek.soal.modus.pencurian.saldo.go-pay.%0A. [Accessed: 20-Feb-2018].

[6] R. Tripathi and S. Agrawal, "Comparative Study of Symmetric and Asymmetric Cryptography Techniquesin 2," *Int. J. Adv. Found. Res. Comput.*, vol. 1, no. 6, pp. 2348–4853, 2014.

[7] S. Mittal, S. Arora, and R. Jain, "Data security using RSA encryption combined with image steganography," *India Int. Conf. Inf. Process. IICIP 2016 - Proc.*, 2017.

[8] T. Sahata Pandapotan and T. Zebua, "Analisa Perbandingan Least Significant Bit dan End Of File Untuk Steganografi Citra Digital Menggunakan Matlab," *Semin. Nas. Inov. dan Teknol.*, no. 3, pp. 604–608, 2016.

[9] R. Roy and S. Changder, "Quality Evaluation of Image Steganography Techniques : A Heuristics based Approach," *Int. J. Secur. Its Apl.*, vol. 10, no. 4, pp. 179–196, 2016.

[10] H. Kaur and A. Kakkar, "Comparison of different image formats using LSB Steganography," *4th IEEE Int. Conf. Signal Process. Comput. Control*, pp. 97–101, 2017.
[11] R. Apau and C. Adomako, "Design of Image Steganography based on RSA Algorithm and LSB Insertion for Android Smartphones," *Int. J. Comput. Appl.*, vol. 164, no. 1, pp. 975–8887, 2017.
[12] M. F. Alamsyah, "Implementasi Metode Steganografi Least Significant," no. 5, 2015.
[13] S. K. Kulkarni, "A Survey of Password Attacks, Countermeasures and Comparative Analysis of Secure Authentication Methods," *Int. J. Adv. Res. Comput. Sci. Manag. Stud.*, vol. 3, no. 11, pp. 319–331, 2015.
[14] M. L. Das and N. Samdaria, "On the security of SSL / TLS-enabled applications," *Appl. Comput. Informatics*, vol. 10, no. 1–2, pp. 68–81, 2014.
[15] K. U. Singh, "A Survey on Image Steganography Techniques," *Int. J. Comput. Appl.*, vol. 97, no. 18, pp. 975–8887, 2014.
[16] "Logo - Fakultas Ilmu Komputer - Udinus Gallery." [Online]. Available: http://dinus.ac.id/gallery2/displayimage.php?pid=2895. [Accessed: 09-Mar-2018].