APLIKASI PENGAMANAN DATA DAN DISISIPKAN PADA GAMBAR DENGAN ALGORITMA RSA DAN MODIFIED LSB BERBASIS ANDROID

DATA SECURITY APPLICATIONS AND INSERTED ON PICTURE WITH RSA AND MODIFIED LSB ALGORITHM BASED ON ANDROID

Muhammad Ridwan Rambe[1], Edy Victor Haryanto[2], Adil Setiawan[3]
[1]Jurusan Teknik Informatika Universitas Potensi Utama
[2,3]Dosen Jurusan Teknik Informatika Universitas Potensi Utama
Universitas Potensi Utama, Jl.K.L. Yos Sudarso Km. 6,5 No 3A Tanjung Mulia Medan
Email : ridwan.eno@gmail.com

Abstrak

Menyimpan username dan password beberapa akun yang kita miliki didalam smartphone tentu bukan pilihan tepat mengingat hal tersebut dapat jatuh ketangan orang lain untuk disalahgunakan. Dibutuhkan sebuah aplikasi yang dapat mengamankan teks dalam perangkat android agar keamanan username dan password yang kita simpan didalam smartphone dapat terjaga. Metode kriptografi dapat digunakan untuk merubah teks menjadi bentuk yang tidak bermakna dengan melakukan perhitungan matematika. Namun perubahan bentuk tersebut menimbulkan kecurigaan oleh pihak lain. Metode steganografi hadir untuk menyembunyikan teks kedalam sebuah media agar teks tersebut tidak diketahui oleh orang lain. Dengan melakukan dua kombinasi antara kriptografi dan steganografi diharapkan dapat menjamin keamanan data username dan password setiap akun yang disimpan didalam smartphone berbasis android.

***Kata Kunci** : Kriptografi, Steganografi, Keamanan, Android*

Abstract

Saving the username and password of some accounts that we have in the smartphone is not the right choice considering it can fall into the hands of others to be abused. It takes an application that can secure the text in the android device for the security of username and password that we store in the smartphone can be maintained. Cryptographic methods can be used to transform text into meaningless forms by performing mathematical calculations. But the change in form raises suspicion by others. Steganography method is present to hide the text into a media so that text is not known by others. By doing two combinations between cryptography and steganography is expected to ensure the security of username and password data of each account stored in the smartphone based on android.

***Keywords** : Cryptography, Steganography, Security, Android*

1. PENDAHULUAN

Keamanan untuk menyimpan teks yang berupa username dan password dalam perangkat android tentu harus menjadi perhatian khusus saat ini. Disamping penggunaan smartphone yang sudah meluas, penyalahgunaan teknologi juga ikut meluas mengikuti perkembangan para penggunanya. Diperlukan sebuah aplikasi yang dapat mengamankan teks berisi username dan password akun penggunanya yang disimpan dalam smartphone berbasis android mereka.

Kriptografi adalah ilmu sekaligus seni untuk menjaga keamanan pesan. Kriptografi mengubah informasi asli (*plaintext*) melalui proses enkripsi menjadi informasi acak (*ciphertext*) menggunakan algoritma dan kunci tertentu, lalu setelah diterima oleh penerima informasi, *ciphertext* akan diubah kembali menjadi plaintext melalui proses dekripsi menggunakan algoritma dan kunci yang sama dengan proses enkripsi [1]. Hal ini membuat pesan yang sudah di enkripsi hanya dapat diketahui oleh sipenerima pesan yang memiliki kunci untuk melakukan proses dekripsi. Namun kriptografi dinilai dapat mengundak kecurigaan oleh pihak lain karena perubahan bentuk menjadi tak bermakna sangat mengundang kecurigaan, oleh karena itu dibutuhkan cara agar dapat menyembunyikan hasil enkripsi kriptografi menjadi tidak diketahui oleh orang lain.

Steganografi lahir dari perkembangan kriptografi dimana ia berperan untuk menyembunyikan pesan kedalam sebuah media agar tidak dapat diketahui keberadaan nya oleh orang lain. Steganografi menggunakan teknik substitusi dan mengganti nilai bit – bit terendah pada tiap byte dalam *cover-object* dengan pesan yang ingin disembunyikan adalah metode LSB (*Least Significant Bit*). Metode LSB mengganti nilai bit terkecil yang perubahannya tidak signifikan sehingga menghasilkan *stego-image* yang secara kasat mata terlihat sama dengan *cover-object* [2].

2. METODE PENELITIAN

2.1. Teknik Pengumpulan Data

Pada tahap ini dilakukan dengan mempelajari teori dasar yang mendukung penelitian, pencarian dan pengumpulan data-data yang dibutuhkan. Untuk mengumpulkan data yang dibutuhkan, maka penulis memakai teknik :

1. Penelitian kelapangan

   Pada metode ini penulis terjun langsung kelapangan untuk mengumpulkan data yang berkaitan dengan pelaksanaan penelitian yang dikutip dari pengamatan langsung, wawancara, dan *sampling.*

   a. Pengamatan langsung (Observation)
      Melakukan pengamatan secara langsung ke tempat objek pembahasan yang ingin diperoleh yaitu bagian – bagian terpenting dalam pengambilan data yang diperlukan berkaitan tentang kriptografi dan steganografi.
   b. Wawancara (Interview)
      Teknik ini secara langsung bertatap muka dengan seseorang yang mengerti mengenai ilmu kriptografi untuk mendapatkan penjelasan dari masalah – masalah yang sebelumnya kurang jelas.
   c. Sampling
      Meneliti dan memilih data – data yang tersedia sesuai dengan bidang ilmu yang dipilih sebagai berkas lampiran.

2. Penelitian perpustakaan

   Pada metode ini penulis menguti dari beberapa bacaan yang berkaitan dengan pelaksanaan penelitian yang berupa teori – teori yang sudah ada.

2.2. Algoritma RSA

Algoritma RSA mengambil nama dari Ron Rivest, Adi Shamir dan Len Adleman yang menciptakan metode tersebut pada tahun 1977. Teknik dasarnya ditemukan pertama kali pada tahun 1973 oleh *Clifford Cock* dari CESG (bagian dari *British* GCHQ) tetapi dirahasiakan sampai tahun 1977. Paten dimiliki oleh RSA *Labs* dan telah *expired*. Algoritma RSA adalah enkripsi yang paling umum digunakan dan algoritma otentikasi. Algoritma *RSA* melibatkan mengalikan dua bilangan prima besar, setelah kunci telah dibuat, bilangan prima asli tidak lagi penting dan dapat dibuang. Baik kunci publik dan kunci privat dibutuhkan untuk enkripsi/dekripsi. Pada algoritma RSA, kunci privat tidak pernah perlu dikirim. Kunci privat digunakan untuk mendekripsi teks yang telah dienkripsi dengan kunci publik. Algoritma kriptografi RSA merupakan algoritma yang termasuk dalam kategori algoritma asimetri bisa disebut juga algoritma kunci publik, disebut algoritma

asimetri karena algoritma yang digunakan pada proses enkripsi dan proses dekripsi berbeda dan disebut algoritma kunci publik karena kunci yang digunakan untuk proses enkripsi bisa dipublikasikan dan diketahui oleh banyak orang [3].

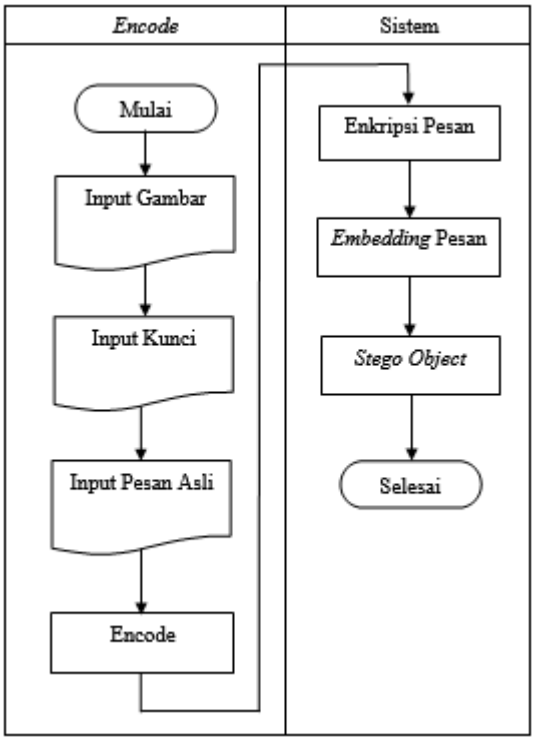
2.3. Metode Steganografi LSB

Metode LSB merupakan metode steganografi yang paling sederhana dan mudah diimplementasikan. Metode ini menggunakan citra digital sebagai *covertext*. Pada susunan bit di dalam sebuah byte (1 byte = 8 bit), ada bit yang paling depan (*most significant bit* atau MSB) dan bit yang paling akhir (*least significant bit* atau LSB). Sebagai contoh byte 11010010, angka bit 1 (pertama, digarisbawahi) adalah bit MSB, dan angka bit 0 (terakhir, digaris-bawahi) adalah bit LSB. Bit yang cocok untuk diganti adalah bit LSB, sebab perubahan tersebut hanya mengubah nilai *byte* satu lebih tinggi atau satu lebih rendah dari nilai sebelumnya. Misalkan *byte* tersebut menyatakan warna merah, maka perubahan satu bit LSB tidak mengubah warna merah tersebut secara berarti. Mata manusia tidak dapat membedakan perubahan kecil tersebut [4].

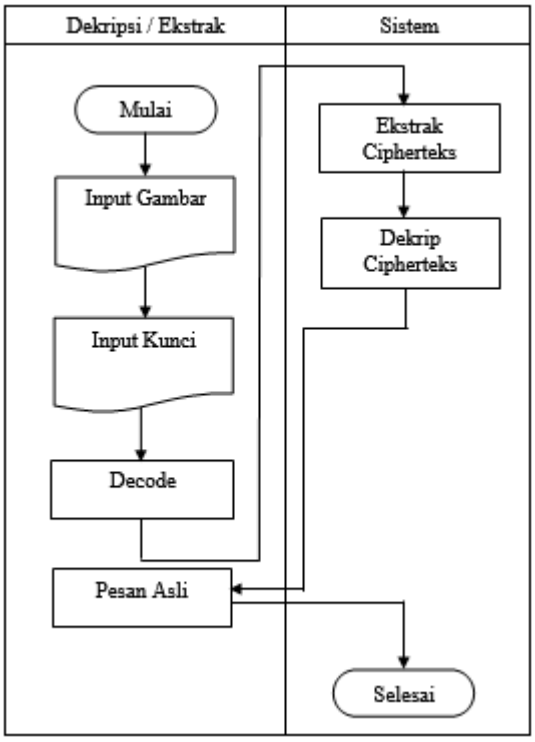
3. HASIL DAN PEMBAHASAN

3.1. Penerapan Algoritma RSA dan Metode *Modified* LSB

Pada perancangan aplikasi ini dibutuhkan beberapa perhitungan yang sesuai dengan algoritma dan metode yang digunakan. Adapun bentuk flowchart sistem encode dan decode yang digunakan sebagai berikut :



Gambar 1. Flowchart sistem Encode



Gambar 2. Flowchart sistem Decode

Pada algoritma RSA,dibutuhkan pasangan kunci yang disebut *public key* untuk proses enkripsi dan *private key* untuk proses dekripsi. Berikut adalah rumus pembangkitan kunci :

a. Menentukan 2 buah bilangan prima.
   Kedua bilangan tidak boleh bernilai sama (P1 ≠ P2).

b. Mencari nilai n
   n = P1 x P2

c. Mencari nilai Θn
   Θn = (P1 -1) x (P2 – 1)

d. Mencari nilai e
   e = 2
   While Θn mod e ≠ 0
   e = e + 1
   End While

e. Mencari nilai d
   $U_1 = 1$
   $U_2 = 0$
   $U_3 = \Theta n$
   $V_1 = 0$
   $V_2 = 1$
   $V_3 = e$
   While $V_3 = 0$
   $Q = Int(U_3 / V_3)$
   $N_1 = U_1 - (Q \times V_1)$
   $N_2 = U_2 - (Q \times V_2)$
   $N_3 = U_3 - (Q \times V_3)$
   $U_1 = V_1$
   $U_2 = V_2$
   $U_3 = V_3$

$V_1 = N_1$
$V_2 = N_2$
$V_3 = N_3$
End While

Setelah pasangan kunci didapatkan, maka dilakukan proses enkripsi dengan algoritma RSA dan juga proses *encode* dengan metode LSB. Berikut adalah penjelasannya :

Untuk melakukan proses enkripsi maka dilakukan perhitungan dengan menggunakan pasangan kunci yang sudah dibangkitkan dan rumus enkripsi algoritma RSA berikut.

$C_i = P_i^e \bmod n$

Keterangan :
C_i = *Ciphertext* hasil enkrip
P_i = *Plaintext* yang akan dienkrip
e = Fungsi eksponensial
mod = Sisa Bagi/Modulus
n = Hasil perkalian dua buah bilangan prima

Setelah dilakukan perhitungan, maka didapatkan cihperteks dengan teks asli RSA adalah *Ciphertext* : chr(917), chr(941), chr(761). Kemudian cipherteks tersebut diubah menjadi bentuk desimal dengan panduan tabel ASCII yang kemudian akan disisipkan pada setiap bit terakhir pixel *cover object* berwarna merah.

Untuk mengembalikan data yang berupa teks yang sebelumnya sudah disisipkan kedalam *cover object*. Maka dilakukan proses ekstraksi dengan metode LSB yang kemudian hasil ekstraksi tersebut akan di dekripsi menggunakan algoritma RSA sehingga mendapatkan data teks yang sama dengan saat proses enkripsi sebelumnya. Dari hasil ekstraksi *stego object* pada pixel warna merah, didapatkan bit terakhir yang membentuk *Ciphertext* : chr(917), chr(941), chr(761). Maka selanjutnya dilakukan perhitungan dekripsi algoritma RSA dengan pasangan kunci yang sama pada saat proses dekripsi dan rumus berikut :

$P_i = C_i^d \bmod n$

Keterangan :
P_i = *Plaintext* hasil dekrip
C_i = *Ciphertext* yang akan didekrip
d = Fungsi eksponensial kunci *public*
mod = Sisa Bagi/Modulus
n = Fungsi perkalian dua bilangan prima

Dari hasil perhitungan diatas, maka didapatkan pesan asli dari cipherteks tersebut adalah RSA.

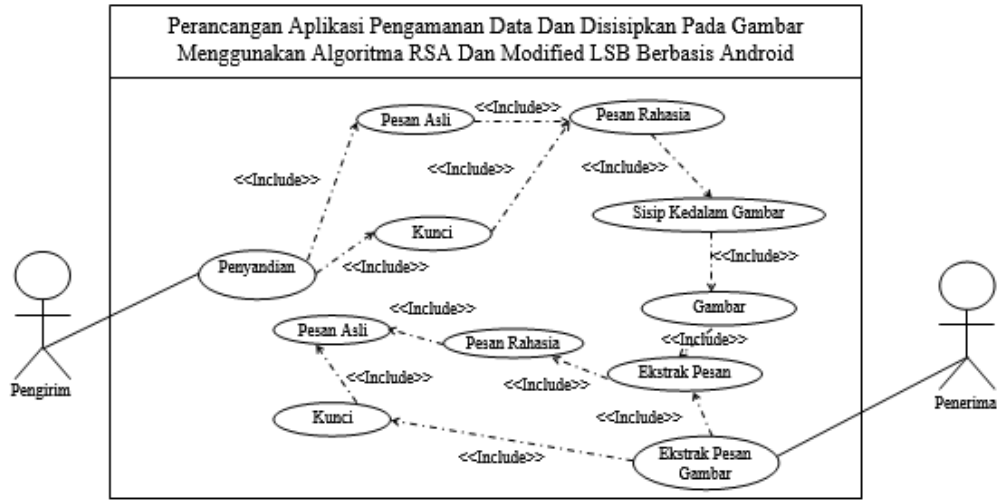
3.2. Desain Sistem

Desain sistem atau perancangan sistem adalah proses pengembangan spesifikasi baru berdasarkan hasil rekomendasi analisis sistem. Dalam tahap perancangan, diharuskan merancang spesifikasi yang dibutuhkan.

Bentuk rancangan sistem yang penulis gunakan adalah beberapa bentuk diagram dari UML (*Unified Modeling Language*) yaitu *Use Case* Diagram, dan *Activity* Diagram.

1. *Use Case* Diagram

   Adapun rancangan *use case* diagram yang penulis rancang adalah sebagai berikut :



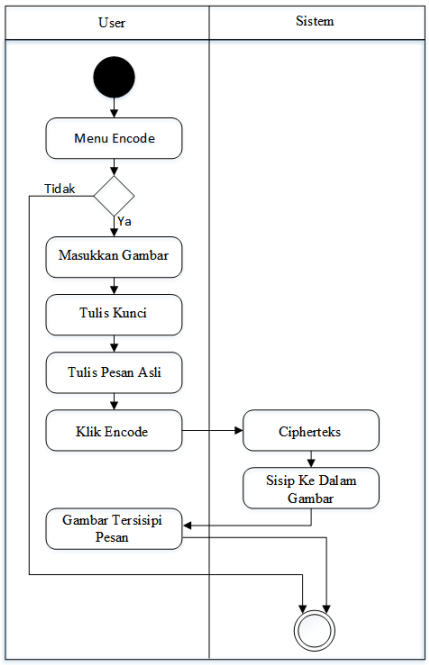
Gambar 3. *Use Case* Diagram Sistem

2. *Activity* Diagram

Pada proses ini kita akan membuat alur dari sistem yang dirancang yaitu *activity* diagram. Berikut adalah *activity* diagram sistem yang dirancang.

a. *Activity* Diagram *Encode*

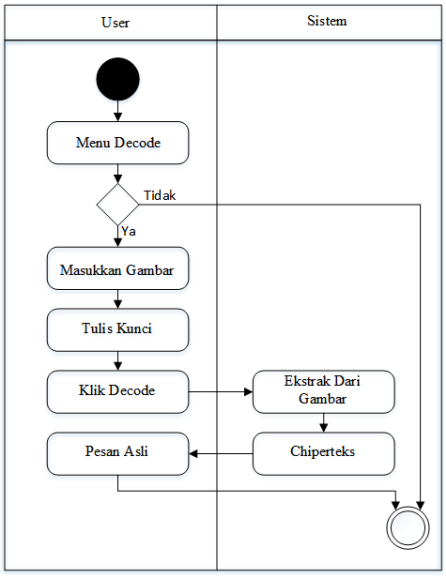
Aktivitas yang dilakukan pada saat proses *Encode* dapat dilihat pada gambar 4 :



Gambar 4. *Activity* Diagram *Encode*

b. *Activity* Diagram *Decode*

Aktivitas yang dilakukan pada saat proses *Decode* dapat dilihat pada gambar 5 :



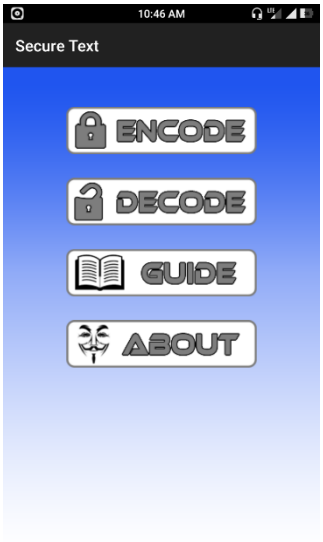
Gambar 5. *Activity* Diagram *Decode*

3.3. Tampilan Hasil

Berikut ini adalah tampilan hasil dari program Perancangan Aplikasi Keamanan Data dan Disisipkan Pada Gambar Dengan Algoritma RSA dan *Modified* LSB Berbasis Android.

1. Tampilan Menu Utama

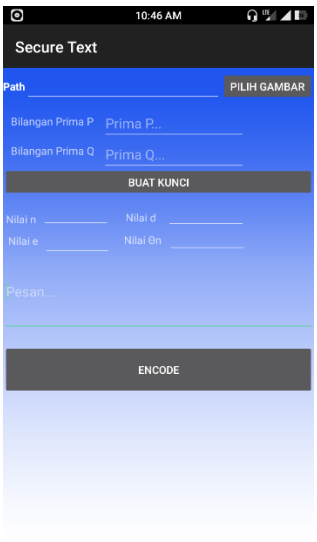
   Tampilan yang diberikan sistem untuk menampilkan Menu Utama dapat dilihat pada gambar 6 berikut :



Gambar 6. Tampilan Menu Utama

2. Tampilan Menu Encode

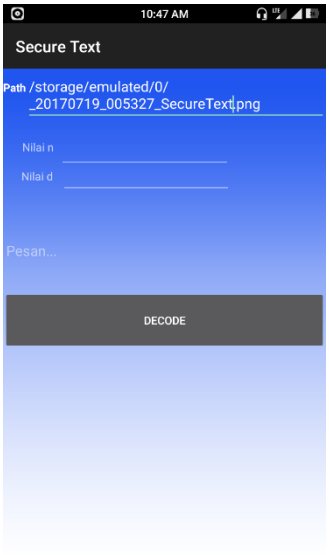
   Tampilan yang diberikan sistem untuk menampilkan Menu Encode dapat dilihat pada gambar 7 berikut :



Gambar 7. Tampilan Menu Encode

3. Tampilan Menu Decode

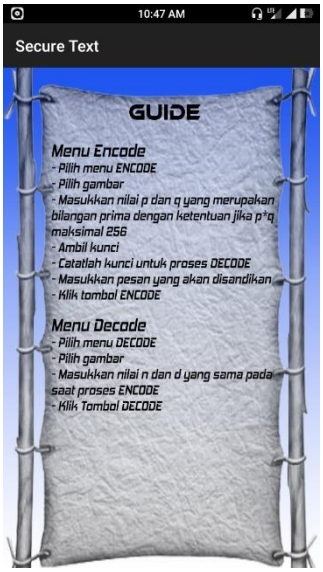
Tampilan yang diberikan sistem untuk menampilkan Menu Decode dapat dilihat pada gambar 8 berikut :



Gambar 8. Tampilan Menu Decode

4. Tampilan Menu Guide

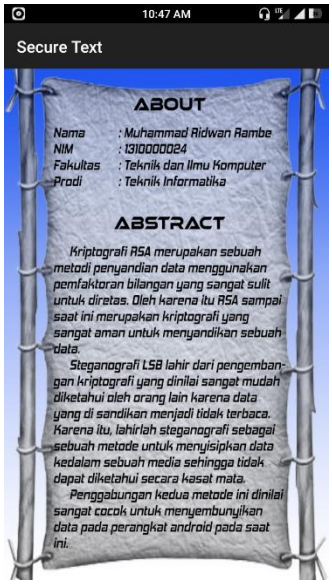
Tampilan yang diberikan sistem untuk menampilkan Menu Guide dapat dilihat pada gambar 9 berikut :



Gambar 9. Tampilan Menu Guide

5. Tampilan Menu About

Tampilan yang diberikan sistem untuk menampilkan Menu About dapat dilihat pada gambar 10 berikut :



Gambar 10. Tampilan Menu About

3.4. Uji Coba

Uji coba aplikasi dilakukan dengan menggunakan aspek *recovery*, dimana data yang disisipkan pada saat dilakukan proses *encode*, akan sama dengan data yang dihasilkan pada proses *decode*. Hal ini bertujuan untuk memastikan bahwa aplikasi yang dibangun sudah berada pada kondisi siap pakai. Berikut adalah hasil pengujian sistem :

Tabel 1. Pengujian sistem dengan aspek *recovery*

No.	*Stego Object*	Cipherteks Encode	Cipherteks Decode	Kunci	Plainteks Decode
1.	 Name : Butterfly.png	>?>}@>C}AB@}?>B}A@A}E=}CAE}AB@}>?=}CE>}CAE}C<B}DE@}>D=}>?=}A@A}@>C}CAE}AB@}C<B}>?>}>?=}CE>}>E<}CE>}C<B}A<<}>D=}D<=}C<B}?>E}A@A}DB}CE>}D<C}AB@}><C}=<?}=<?}	>?>}@>C}AB@}?>B}A@A}E=}CAE}AB@}>?=}CE>}CAE}C<B}DE@}>D=}>?=}A@A}@>C}CAE}AB@}C<B}>?>}>?=}CE>}>E<}CE>}C<B}A<<}>D=}D<=}C<B}?>E}A@A}DB}CE>}D<C}AB@}><C}=<?}=<?}	d=587 n=943	Universitas Potensi Utama Top Sekali.!!
2.	 Name : Cloud.png	=@@<}=ABC}=@>}>@BE}=<=>}>@BE}==<>}=DCE}@D@}D>@}>@BE}=CB@}=>=?}=DCE}=>BD}>EC>A<@}=<=A}>@BE}>@A=}=?>E}	=@@<}=ABC}=@>}>@BE}=<=>}>@BE}==<>}=DCE}@D@}D>@}>@BE}=CB@}=>=?}=DCE}=>BD}>EC>A<@}=<=A}>@BE}>@A=}=?>E}	d=343 n=2501	TIF A Pagi Stambuk 2013 is the best (y) :D XD

		><E?}>=<E} >@BE}D>@ }>@AD}>@ BE}=>=?}= @<A}=@>@ }>@BE}>EC }=@>@}>@ AD}=>=?}> @BE}><@} @DC}D><} >}BA?}>@ BE}=D??} BA?}	><E?}>=<E} >@BE}D>@ }>@AD}>@ BE}=>=?}= @<A}=@>@ }>@BE}>EC }=@>@}>@ AD}=>=?}> @BE}><@} @DC}D><} >}BA?}>@ BE}=D??} BA?}		

4. KESIMPULAN

Berdasarkan pembahasan dari penelitian yang telah dilakukan, maka dapat diambil beberapa kesimpulan sebagai berikut :

1. Proses *encoding* menggunakan metode *modified* LSB dilakukan dengan menyisipkan bit – bit cipherteks ke dalam byte – byte *cover object*, dimana bit terakhir pada tiap byte *cover object* yang mengalami perubahan. Selanjutnya proses *decoding* dilakukan dengan mengekstrak bit terakhir dari tiap byte *cover object*.
2. Pembangkitan kunci menggunakan bilangan prima yang acak membuat algoritma RSA sulit untuk diketahui karena harus memfaktorkan nilai n yang merupakan perkalian dari bilangan prima p dan q. Untuk menyembunyikan cipherteks hasil enkripsi algoritma RSA, maka dilakukan penyembunyian pada objek gambar menggunakan metode LSB yang sudah dimodifikasi yang penyisipan nya dilakukan pada bit – bit terakhir setiap byte.
3. Kombinasi metode keamanan menggunakan metode kriptografi RSA dan steganografi *modified* LSB pada perangkat android yang dibuat dengan pemrograman Eclipse berhasil mencapai tujuan untuk mengamankan data pada perangkat android.

5. SARAN

Untuk pengembangan lebih lanjut pada aplikasi pengamanan data dan disipkan pada gambar dengan algoritma RSA dan *Modified* LSB ini, maka dapat diberikan saran sebagai berikut :

1. Pembangkitan bilangan prima p dan q sebaiknya menghasilkan nilai n yang besar, sehingga menghasilkan nilai n yang sulit untuk difaktorkan agar keamanan kunci *private* pada algoritma RSA tetap dapat terjaga.
2. Penelitian selanjutnya dapat mengembangkan metode dengan menggabungkan beberapa algoritma kriptografi sehingga cipherteks sulit untuk diketahui maknanya. Pengembangan metode *modified* LSB juga dapat dilakukan dengan menambah byte warna biru dan hijau, atau menyisipkan bit – bit cipherteks secara diagonal kedalam byte – byte *cover object*.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada keluarga penulis dan staff / pegawai Universitas Potensi Utama yang telah memberi dukungan terhadap penelitian ini.

DAFTAR PUSTAKA

[1] Haryanto, E.V., 2015. Penerapan Metode Adaptif Dalam Penyembunyian Pesan Pada Citra. *Proceedings Konferensi Nasional Sistem dan Informatika (KNS&I)*.

[2] Akbar, M.B. and Haryanto, E.V., 2016. Aplikasi Steganografi dengan Menggunakan Metode F5. *JUSITI: Jurnal Sistem Informasi dan Teknologi Informasi*, *4*(2), pp.165-175

[3] Joko Dewanto, dkk, 2013. Pembuatan Aplikasi RSA Dengan Android. *Forum Ilmiah*, Vol. 10, No. 2.

[4] Gede Wisnu Bhaudhayana, dkk, 2015. Implementasi Algoritma Kriptografi AES 256 Dengan Metode Steganografi LSB Pada Gambar Bitmap. *Jurnal Ilmiah Komputer Universitas Udayana*. Vol. 8, No. 2.