Kriptografi Citra Digital Menggunakan Algoritma *Hill Cipher* Dan *Affine Cipher* Berbasis Android

Erla Rizky Febrianto[*1)], Eko Adi Sarwoko[*2)]

[*]Departemen Ilmu Komputer/Informatika, Fakultas Sains dan Matematika,
Universitas Diponegoro
[1)]erlarizky.f@gmail.com, [2)]eko.adi.sarwoko@gmail.com

Abstrak

Perkembangan teknologi saat ini mengakibatkan semakin mudah pertukaran informasi. Citra merupakan data yang bersifat informatif, bahkan mengandung informasi penting bagi sebagian pihak. Keamanan terhadap kerahasiaan informasi atau data dari citra harus terjaga dan terjamin keasliannya sehingga tidak terjadi penyalahgunaan informasi oleh pihak yang tidak bertanggung jawab. Pada penelitian tugas akhir ini membahas tentang kriptografi citra digital menggunakan algoritma Hill Cipher dan Affine Cipher berbasis Android. Algoritma Hill Cipher dipilih karena salah satu dari algoritma kriptografi klasik yang cukup aman dan tidak mudah dipecahkan. Algoritma Affine Cipher dipilih karena sederhana dan mudah dalam pengimplementasiannya. Aplikasi ini dapat melakukan proses enkripsi dan dekripsi pada perangkat smartphone yang menggunakan sistem operasi android. Hasil dari aplikasi ini adalah citra rekayasa yang telah diolah berdasarkan kunci yang dimasukkan. Durasi lama waktu proses enkripsi dan dekripsi tergantung pada kombinasi kunci serta dimensi ukuran citra. Semakin besar dimensi ukuran citra yang dimasukkan maka semakin lama waktu pemrosesan. Aplikasi kriptografi citra digital ini telah diuji dengan membandingkan dengan aplikasi serupa yang hanya menggunakan algoritma Hill Cipher atau Affine Cipher saja. Hasil pengujian nilai Peak Signal to Noise Ratio (PSNR) dengan menggabungkan kedua algoritma menghasilkan nilai rata- rata PSNR yang lebih rendah jika dibandingkan dengan aplikasi yang hanya menggunakan algoritma Hill Cipher atau algoritma Affine Cipher saja.

Kata kunci : *Citra, Android, Enkripsi, Dekripsi, Hill, Affine, PSNR*

Abstract

Current technological developments have resulted in easier exchange of information. Image is data that is informative, even contains important information for some parties. The security of the confidentiality of information or data from the image must be maintained and guaranteed authenticity so that there is no misuse of information by irresponsible parties. In this final project research discusses digital image cryptography using the Hill Cipher and Affine Cipher algorithms based on Android. The Hill Cipher algorithm is chosen because one of the classic cryptographic algorithms is safe and not easily solved. The Affine Cipher algorithm is chosen because it is simple and easy to implement. This application can process encryption and decryption on smartphone devices that use the Android operating system. The results of this application are engineered imagery that has been processed based on the key entered. The duration of the encryption and decryption process depends on the combination of the key and the dimensions of the image size. The greater the dimensions of the image size entered, the longer the processing time. This digital image cryptographic application has been tested by comparing with similar applications that only use the Hill Cipher algorithm or Affine Cipher only. The results of testing the Peak Signal to Noise Ratio (PSNR) value by combining the two

algorithms results in lower PSNR average values compared to applications that only use the Hill Cipher algorithm or Affine Cipher algorithm only.

Keywords : *Image, Android, Encryption, Decryption, Hill, Affine, PSNR*

1 PENDAHULUAN

Smartphone saat ini sudah menjadi kebutuhan primer bagi sebagian masyarakat di dunia. Melalui media ini, kita dapat saling bertukar informasi tanpa perlu mengkhawatirkan jarak dan waktu, termasuk informasi yang bersifat rahasia dan pribadi. Melalui *smartphone* kita bisa berbagi informasi melalui pesan teks, gambar maupun suara. Pada *smartphone* terdapat berbagai macam sistem operasi yang digunakan yang salah satunya adalah Android. Android merupakan salah satu sistem informasi yang sedang berkembang dan bersifat *open source* sehingga tidak heran jika Android menjadi salah satu sistem operasi yang paling banyak digunakan pada saat ini.

Seiring dengan perkembangan teknologi informasi tuntutan akan keamanan terhadap kerahasiaan informasi yang saling dipertukarkan melalui *smartphone* semakin meningkat, salah satunya adalah data atau informasi berupa citra digital [9]. Citra digital telah digunakan secara luas dalam berbagai macam kegiatan sehingga perlindungan pada citra digital menjadi sangat penting. Instansi pemerintah, rumah sakit, militer, badan keuangan serta perusahaan swasta telah menggunakan citra digital untuk menyimpan informasi penting, misalnya data pemeriksaan pasien pada rumah sakit, data militer, data nasabah, data karyawan dan sebagainya. Hampir semua informasi ini akan mengalami proses pertukaran informasi melalui internet. Pengguna perlu untuk mengirim dan menerima informasi yang bersifat *private* hingga tidak boleh seorangpun yang mengetahui, cara yang paling baik untuk berkomunikasi aman adalah dengan mengubah data menjadi bentuk lain, sehingga data hasil perubahan hanya dapat dimengerti oleh penerima yang mengetahui cara mengembalikanya ke bentuk semula [1].

Salah satu metode untuk mengamankan data tersebut adalah dengan kriptografi. Kriptografi adalah suatu studi teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, otentikasi entitas dan otentikasi keaslian data. Kriptografi tidak hanya berarti penyediaan keamanan informasi, melainkan sebuah himpunan teknik-teknik [5].

Dengan semakin majunya perkembangan pada perangkat *mobile*, implementasi kriptografi menjadi mungkin untuk diterapkan. Dalam kriptografi, teknik penyandian data dibagi menjadi dua, yaitu kriptografi klasik dan kriptografi modern. Dalam kriptografi klasik terdapat dua teknik dasar yang digunakan, yaitu teknik substitusi dan teknik transposisi. Teknik substitusi dilakukan dengan mengganti karakter asli dengan karakter lain, sedangkan transposisi dilakukan dengan permutasi karakter. Salah satu algoritma kriptografi klasik adalah *Hill Cipher*. *Hill Cipher* termasuk algoritma kriptografi klasik yang sulit dipecahkan apabila hanya mengetahui berkas *ciphertext* saja, karena *Hill Cipher* tidak mengganti setiap abjad yang sama pada *plaintext* dengan

abjad lainnya yang sama pada *ciphertext* [7]. Metode lain dalam kriptografi klasik yang dapat digunakan adalah *Affine Cipher*. *Affine Cipher* adalah jenis monoalfabetik *cipher* substitusi yang setiap huruf dalam alfabet dipetakan ke numerik, dienkripsi menggunakan fungsi matematika sederhana, dan diubah kembali ke teks. Rumus yang digunakan berarti bahwa setiap huruf mengenkripsi dengan satu huruf lain, dan kembali lagi, berarti *cipher* ini pada dasarnya adalah *cipher* substitusi standar. Dengan demikian, ia memiliki kelemahan semua *cipher* substitusi [9].

Penerapan Algoritma *Affine Cipher* pada proses enkripsi citra dapat menghasilkan citra rekayasa yang rusak. Akan tetapi seperti *cipher* subtitusi yang lainnya algoritma *Affine Cipher* memiliki kelemahan dalam hal keamanan [9]. Oleh karena itu pada penelitian ini digunakan algoritma *Hill Cipher* untuk menutupi kekurangan yang dimiliki oleh algoritma *Affine Cipher*. Penggunaan algoritma *Hill Cipher* pada penelitian ini diharapkan dapat menambah keamanan pada proses enkripsi dan dekripsi serta menghasilkan citra rekayasa yang lebih rusak jika dibandingkan hanya menggunakan algoritma *Affine Cipher* atau *Hill Cipher* saja.

Penelitian ini membangun sebuah aplikasi dengan menggabungkan algoritma *Hill Cipher* dan *Affine Cipher* untuk proses enkripsi dan dekripsi pesan citra pada perangkat *smartphone* berbasis Android yang bertujuan untuk melindungi data atau informasi dari sebuah citra digital.

1.1 RUMUSAN MASALAH

Rumusan masalah yang diangkat pada penelitian ini adalah bagaimana membangun sebuah aplikasi pada perangkat *smartphone* berbasis Android yang mampu melindungi informasi atau data dari sebuah citra digital menggunakan penggabungan algoritma *Hill Cipher* dan *Affine Cipher*.

1.2 TUJUAN PENULISAN

Tujuan dilaksanakannya penelitian ini adalah untuk membangun sebuah aplikasi pada perangkat smartphone berbasis Android yang mampu melindungi informasi atau data dari sebuah citra digital menggunakan algoritma Hill Cipher dan Affine Cipher.

1.3 RUANG LINGKUP

Penelitian ini dilakukan beberapa pembatasan ruang lingkup, diantaranya adalah sebagai berikut:

1. *Input* data berupa *file* citra dengan format **.jpg*, **jpeg*, **.png* atau **.bmp*.
2. *Output* berupa citra rekayasa dengan format **.png*.
3. Kunci yang digunakan pada proses enkripsi dan dekripsi menggunakan algoritma *Hill Cipher* adalah matriks dengan ukuran 3x3 atau panjang sembilan karakter.
4. Sistem operasi *mobile* yang digunakan dalam pengembangan aplikasi ini adalah Google® Android versi 6.0 dengan sistem operasi minimum yang dapat dites adalah Android versi 5.0.
5. Metode perancangan aplikasi yang digunakan adalah metode perancangan *Unified Process.*

2 TINJAUAN PUSTAKA

2.1 CITRA DIGITAL

Citra digital (citra) adalah suatu gambar yang disimpan dalam bentuk rangkaian bilangan biner dan disimpan pada media penyimpanan digital, seperti *Random Access Memory* (RAM), *harddisk*, dan lain sebagainya. Citra dapat ditampilkan pada layar monitor ataupun dicetak pada kertas.

Suatu citra terdiri dari susunan elemen-elemen citra (piksel) sebanyak lebar kali tinggi citra. Setiap piksel memiliki nilai yang menunjukan tingkat kecerahan dan warna pada titik tersebut. Untuk citra berwarna, nilai piksel dapat dibagi menjadi sejumlah kanal warna yang digunakan. Sebagai contoh, untuk citra berwarna RGB, setiap piksel menyimpan nilai untuk intensitas warna merah, hijau, dan biru secara berurutan [8].

2.2 KRIPTOGRAFI

Kriptografi adalah suatu studi teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, otentikasi entitas dan otentikasi keaslian data [5]. Terdapat empat prinsip atau tujuan yang mendasari kriptografi yaitu kerahasiaan (*confidelity*), integritas data (*integrity*), otentikasi (authentication), dan nonrepudiasi. Kerahasiaan berfungsi agar isi pesan yang dikirimkan oleh pihak pengirim ke pihak penerima tetap terjaga dan tidak diketahui oleh pihak lain. Integritas data berhubungan dengan penjagaan dari perubahan data secara tidak sah. Otentikasi berhubungan dengan identifikasi atau pengenalan, baik secara kesatuan sistem maupun informasi itu sendiri. Nonrepudiasi adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman atau terciptanya suatu informasi oleh yang mengirimkan.

Kriptografi dibagi menjadi dua, kriptografi klasik dan kriptografi modern. Kriptografi klasik menggunakan metode substitusi, transposisi, atau keduanya dalam menyembunyikan pesan. Teknik substitusi adalah menggantikan karakter dalam *plaintext* menjadi karakter lain yang hasilnya adalah *ciphertext* sedangkan teknik transposisi adalah teknik mengubah *plaintext* menjadi *ciphertext* dengan cara permutasi karakter [9]. Pada kriptografi modern umumnya beroperasi dalam mode *bit* karena berjalan mengikuti operasi komputer digital sehingga membutuhkan operasi matematika untuk pengoperasiannya. Berdasarkan kunci yang dipakai, algoritma kriptografi dapat dibedakan atas dua golongan, yaitu:

2.2.1 SYMMETRIC ALGORITHMS

Algoritma kriptografi simeteris atau disebut juga algoritma kriptografi konvensional adalah algoritma yang menggunakan kunci untuk proses enkripsi sama dengan kunci untuk proses dekripsi. Algoritma kriptografi simeteris dibagi menjadi dua kategori yaitu algoritma aliran (*Stream Ciphers*) dan algoritma blok (*Block Ciphers*). Pada algoritma aliran, proses penyandiannya berorientasi pada satu bit atau satu byte data. Sedang pada algoritma blok, proses penyandiannya berorientasi pada sekumpulan bit atau *byte* data (per blok). Contoh algoritma kunci simetris yang terkenal adalah DES (*Data Encryption Standard*).

2.2.2 ASYMMETRIC ALGORITHMS

Algoritma kriptografi asimetris adalah algoritma yang menggunakan kunci yang berbeda untuk proses enkripsi dan dekripsinya. Algoritma ini disebut juga algoritma kunci umum (*public key algorithm*) karena kunci untuk enkripsi dibuat umum (*public key*) atau dapat diketahui oleh setiap orang, tapi kunci untuk dekripsi hanya diketahui oleh orang yang berwenang mengetahui data yang disandikan atau sering disebut kunci pribadi (*private key*). Contoh algoritma terkenal yang menggunakan kunci asimetris adalah RSA dan ECC.

2.3 ALGORITMA *HILL CIPHER*

Algoritma *Hill cipher* merupakan algoritma klasik yang ditemukan oleh Leister

S. Hill pada tahun 1929 yang menggunakan prinsip perkalian matriks. Di dalam algoritma *Hill cipher*, *plaintext*, *ciphertext*, dan kunci direpresentasikan ke dalam bentuk matriks. Kunci pada algoritma ini adalah matriks *n x n* di mana *n* merupakan ukuran blok. Pada penelitian ini menggunakan kunci matriks berukuran 3x3. Alasan penggunaan kunci matriks berukuran 3x3 adalah dirasa cukup aman dan jika menggunakan ukuran yang lebih besar akan menambah waktu proses saat enkripsi dan dekripsi. *Plaintext* pada penelitian ini merupakan nilai RGB dari setiap piksel citra yang diinputkan. Pada proses enkripsi, *plaintext* dibagi menjadi blok-blok berurutan yang sesuai dengan ukuran matriks kunci yang digunakan [1]. Matriks plainteks *P* pada algoritma *Hill cipher* ditunjukkan pada persamaan 1 dan matriks kunci *K* pada algoritma *Hill cipher* ditunjukkan pada persamaan 2.

$$P = \begin{bmatrix} p_{11} & p_{12} & \dots & p_{1n} \\ p_{21} & p_{22} & \dots & p_{2n} \\ \dots & \dots & \dots & \dots \\ p_{n1} & p_{n2} & \dots & p_{nn} \end{bmatrix} \quad (1)$$

$$K = \begin{bmatrix} k_{11} & k_{12} & \dots & k_{1n} \\ k_{21} & k_{22} & \dots & k_{2n} \\ \dots & \dots & \dots & \dots \\ k_{n1} & k_{n2} & \dots & k_{nn} \end{bmatrix} \quad (2)$$

Matriks kunci *K* adalah matriks yang *invertible*, yaitu matriks yang memiliki *multiplicative invers K*$^{-1}$ sehingga $K * K^{-1} = I$. Dalam proses enkripsi, *plaintext* yang ada diubah ke dalam bentuk blok-blok secara berurutan sesuai dengan ukuran matriks [1]. Jika terdapat sebuah *plaintext P* yang memiliki matriks kunci *K* akan dienkripsi menjadi *ciphertext C*, maka secara matematis proses enkripsi dapat dirumuskan pada persamaan 3 dan proses dekripsi pada persamaan 4.

Enkripsi : $C = K * P$ (3)

Dekripsi : $P = K^{-1} * C$ (4)

Dengan:
C = *Ciphertext*,
K = Kunci,
P = *Plaintext*.

Misalkan matriks kunci *K* yang digunakan berordo 3x3, maka proses enkripsi dapat dinyatakan dengan persamaan 5.

$$C_1 = (k_{11}P_{11} + k_{12}P_{12} + k_{13}P_{13}) \bmod 256$$
$$C_2 = (k_{21}P_{11} + k_{22}P_{12} + k_{23}P_{13}) \bmod 256 \quad (5)$$
$$C_3 = (k_{31}P_{11} + k_{32}P_{12} + k_{33}P_{13}) \bmod 256$$

Persamaan 5 juga dapat dinyatakan dalam bentuk vektor kolom dan matriks yang ditunjukkan pada persamaan 6.

$$\begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} \begin{bmatrix} P_{11} \\ P_{12} \\ P_{13} \end{bmatrix} \quad (6)$$

2.4 ALGORITMA *AFFINE CIPHER*

Affine Cipher adalah teknik *cipher* yang merupakan perluasan dari *Caesar cipher* yang mengalikan plainteks dengan sebuah nilai dan menambahkannya dengan sebuah pergeseran [7]. *Affine cipher* tergolong dalam algoritma klasik yang merupakan algoritma penyandian yang sudah ada sebelum era digital sekarang ini. Algoritma klasik pada dasarnya hanya terdiri dari *cipher* subtitusi dan *cipher* tranposisi. *Cipher* subtitusi yaitu proses mensubtitusi karakter-karakter yang ada pada *plaintext*. Sedangkan *cipher* tranposisi yaitu proses pertukaran huruf-huruf yang terdapat dalam suatu *string*. *Affine cipher* merupakan metode kriptografi yang menggunakan kunci simetris, yang mana kunci yang digunakan untuk melakukan enkripsi sama dengan kunci yang digunakan untuk dekripsi. Secara matematis enkripsi *plainteks* menghasilkan *cipherteks* dinyatakan dengan fungsi yang ditunjukkan pada persamaan 7.

$$C(P) = (mP + b) \bmod n \quad (7)$$

Sementara dekripsi *cipherteks* menjadi *plainteks* dinyatakan oleh persamaan 8.

$$P(C) = (m - 1\,(C - b)) \bmod n \quad (8)$$

Dengan nilai *n* adalah ukuran maksimal nilai RGB sebesar 256, P adalah *plainteks* yang dikonversi menjadi bilangan bulat dari 0 sampai *n* – 1 sesuai dengan nilai RGB, C adalah *cipherteks* yang dikonversi menjadi bilangan bulat dari 0 sampai *n* – 1 sesuai dengan nilai RGB, *m* adalah kunci berupa bilangan bulat yang harus relatif prima dengan *n* (jika tidak relatif prima, maka dekripsi tidak dapat dilakukan) dan *b* adalah kunci yang digunakan untuk menentukan jumlah pergeseran.

2.5 *PEAK SIGNAL TO NOISE RATIO*

PSNR digunakan untuk menentukan kualitas citra. Nilai PSNR diperoleh dengan membandingkan citra asli dan citra rekayasa. Untuk menentukan nilai PSNR digunakan rumus pada persamaan 9 [10].

$$PSNR(dB) = 10 * log \frac{255^2}{MSE} \quad (9)$$

Dimana nilai 255 merupakan nilai tertinggi intensitas suatu piksel dan MSE (*Mean Square Error*) merupakan nilai rata-rata dari jumlah kuadrat *Absolute Error* antara *plainfile* dengan *cipherfile.* Nilai ini dapat diperoleh dengan menggunakan rumusan persamaan 10.

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} I(i,j) - K(i,j) \quad (10)$$

Dimana:
m : Jumlah baris/ lebar citra dalam piksel
n : Jumlah kolom/ tinggi citra dalam piksel
$I_{(i, j)}$: Nilai piksel dari citra asli
$K_{(i, j)}$: Nilai piksel dari citra rekayasa

Nilai PSNR menunjukkan ukuran kualitas citra yang dibandingkan. Semakin rendah nilai PSNR yang didapatkan maka kualitas citra yang dibandingkan semakin berbeda atau tidak identik. Sebaliknya jika nilai PSNR yang didapatkan semakin besar maka citra yang dibandingkan semakin identik. Ukuran kualitas citra disajikan dalam Tabel 1.

Tabel 1 Nilai Kualitas Citra PSNR [11]

PSNR (dB)	Kualitas Citra
>60	Istimewa (*Excellent*)
50	Bagus (*Good*)
40	Layak (*Reasonable*)
30	Cukup (*Poor Picture*)
20	Tidak Dapat Dipakai (*Unusable*)

3 METODE PENELITIAN

3.1 PROSES ENKRIPSI CITRA

Alur proses enkripsi citra adalah sebagai berikut:

1. Mengubah bentuk citra dan kunci menjadi matriks. Citra yang dimasukkan akan diubah ke dalam bentuk matriks angka (nilai RGB) untuk digunakan dalam proses enkripsi. Kunci yang dimasukkan juga diubah ke bentuk matriks.
2. Melakukan pengecekan kunci. Pengecekan terhadap kunci dilakukan dengan memeriksa ukuran kunci apakah sesuai dengan ketentuan. Aplikasi menggunakan kunci matriks 3x3 sehingga masukan kunci harus memiliki sembilan karakter. Selain itu juga menghitung nilai determinan dari matriks. Matriks kunci adalah matriks yang *invertible* sehingga determinan kunci tidak boleh sama dengan nol. Jika kunci tidak memenuhi ketentuan tersebut, aplikasi akan memberikan peringatan.
3. Melakukan proses enkripsi algoritma *Hill Cipher*. Citra dan kunci yang telah diubah ke dalam bentuk matriks kemudian dilakukan proses ekripsi algoritma *Hill Cipher*. Proses dilakukan dengan mengalikan matriks kunci dengan matriks citra masukan. Setelah itu dilakukan proses modulo 256 terhadap nilai hasil perkalian matriks.
4. Melakukan enkripsi algoritma *Affine Cipher*. Hasil proses enkripsi algoritma *Hill Cipher* kemudian dilakukan proses enkripsi kembali menggunakan algoritma

Affine Cipher dengan kunci yang telah dimasukkan.

5. Memperoleh hasil enkripsi dalam bentuk matriks. Hasil dari perhitungan proses enkripsi adalah *cipherfile* dalam bentuk matriks. Matriks ini akan selanjutnya akan dibuah menjadi *array*.
6. Mengubah matriks kembali menjadi bentuk citra. *Cipherfile* yang masih dalam bentuk matriks akan diubah menjadi array. Matriks citra diubah menjadi *array* dua dimensi sesuai dengan ukuran panjang dan lebarnya.
7. Hasil keluaran proses adalah *cipherfile* citra. Keluaran akhir proses enkripsi adalah satu citra acak (*cipherfile*) yang kemudian dapat digunakan sebagai masukan untuk proses dekripsi.

3.2 Proses Dekripsi Citra

Alur proses dekripsi citra adalah sebagai berikut:

1. Mengubah *cipherfile* dan kunci menjadi matriks. *File* citra dan kunci yang digunakan sebagai masukan diubah ke dalam bentuk matriks terlebih dahulu.
2. Menghitung nilai invers dari matriks kunci untuk selanjutnya digunakan dalam proses dekripsi.
3. Melakukan proses dekripsi *cipherfile* dengan algoritma *Affine Cipher*. Setelah diubah ke bentuk matriks, proses selanjutnya adalah mendekripsi *cipherfile* menggunakan algoritma *Affine Cipher*.
4. Melakukan proses dekripsi dengan algoritma *Hill Cipher*. Hasil dari proses dekripsi algoritma *Affine Cipher* kemudian dilakukan proses dekripsi algoritma *Hill Cipher* dengan cara mengalikan invers matriks kunci dengan matriks *cipherfile*.
5. Memperoleh hasil dekripsi dalam bentuk matriks. Hasil dari perkalian matriks kunci dengan matriks *cipherfile* dilanjutkan dengan proses modulo 256.
6. Mengubah matriks kembali menjadi citra. Matriks citra hasil perkalian diubah ke dalam array dua dimensi. Kemudian array angka *file* citra diubah menjadi derajat keabuan citra.
7. Keluaran akhir proses berupa *file* citra. Hasil akhir dari proses dekripsi adalah *file* citra yang menyerupai citra awal.

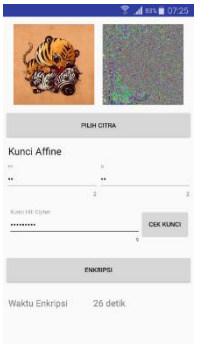
4 Hasil dan Pembahasan

4.1 Implementasi Antarmuka

Implementasi antarmuka halaman utama untuk menampilkan menu-menu yang terdapat dalam aplikasi dapat dilihat pada Gambar 1.

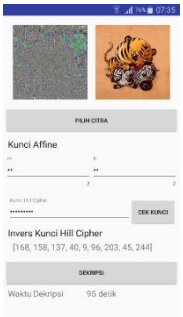
Gambar 1 Implementasi Halaman Utama

Implementasi antarmuka halaman enkripsi untuk melakukan proses enkripsi citra dapat dilihat pada Gambar 2



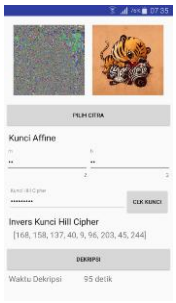
Gambar 2 Implementasi Halaman Enkripsi

Implementasi antarmuka halaman dekripsi untuk melakukan proses dekripsi citra dapat dilihat pada Gambar 3



Gambar 3 Implementasi Halaman Dekripsi

Implementasi antarmuka hitung PSNR untuk menghitung perbandingan kualitas dua citra dapat dilihat pada Gambar 4



Gambar 4 Implementasi Halaman Hitung PSNR

4.2 PENGUJIAN

4.2.1 PENGUJIAN ENKRIPSI DAN DEKRIPSI

Pengujian dilakukan dengan melakukan enkripsi dan dekripsi terhadap sejumlah citra dengan ukuran yang berbeda-beda dengan menggunakan dua buah kunci. Waktu proses enkripsi dan dekripsi citra masing-masing akan dibandingkan. Hasil pengujian dapat dilihat pada Tabel 2.

Tabel 2 Hasil Pengujian Enkripsi dan Dekripsi

No	Nama File	Dimensi Citra	Kunci	Waktu Proses Enkripsi (detik)	Waktu Proses Dekripsi (detik)
1.	lena.jpeg	256 x 256	9901, 17, erlarizky	1	9
			17, 11, bbabcbbdd	1	9
2.	lenagrey.jpeg	256 x 256	9901, 17, erlarizky	1	9
			17, 11, bbabcbbdd	1	10
3.	camera.bmp	256 x 256	9901, 17, erlarizky	1	11
			17, 11, bbabcbbdd	1	11
4.	goldhill.bmp	256 x 256	9901, 17, erlarizky	1	10
			17, 11, bbabcbbdd	1	10
5.	san.bmp	256 x 256	9901, 17, erlarizky	1	9
			17, 11, bbabcbbdd	1	11
6.	baboon.png	512 x 512	9901, 17, erlarizky	5	80
			17, 11, bbabcbbdd	5	89
7.	boat.png	512 x 512	9901, 17, erlarizky	5	95
			17, 11, bbabcbbdd	5	90
8.	barbara.bmp	512 x 512	9901, 17, erlarizky	5	99
			17, 11, bbabcbbdd	5	90
9.	hiu.jpeg	540 x 540	9901, 17, erlarizky	5	102
			17, 11, bbabcbbdd	5	106
10.	owltikus.jpeg	540 x 540	9901, 17, erlarizky	5	102
			17, 11, bbabcbbdd	6	94
11.	kucing.jpeg	800 x 800	9901, 17, erlarizky	12	230
			17, 11, bbabcbbdd	13	216
12.	pinguin.png	800 x 800	9901, 17, erlarizky	12	226
			17, 11, bbabcbbdd	13	219
13.	tank.jpeg	800 x 800	9901, 17, erlarizky	12	242
			17, 11, bbabcbbdd	13	172
14.	tasya.png	1000 x 1000	9901, 17, erlarizky	21	358
			17, 11, bbabcbbdd	21	359
15.	sg.jpeg	1000 x 1000	9901, 17, erlarizky	19	361
			17, 11, bbabcbbdd	21	330

Pengujian juga dilakukan dengan membandingkan aplikasi serupa yang hanya menggunakan algoritma *Hill Cipher* atau *Affine Cipher* saja. Hasil pengujian enkripsi dan dekripsi menggunakan algoritma *Hill Cipher* dapat dilihat pada Tabel 3 dan hasil pengujian enkripsi dan dekripsi menggunakan algoritma *Affine Cipher* dapat dilihat pada Tabel 4.

Tabel 3 Hasil Pengujian Enkripsi dan Dekripsi menggunakan Algoritma *Hill Cipher*

No	Nama File	Dimensi Citra	Kunci	Waktu Proses Enkripsi (detik)	Waktu Proses Dekripsi (detik)
1.	lena.jpeg	256 x 256	erlarizky	1	1
			babcbbdd	1	1
2.	lenagrey.jpeg	256 x 256	erlarizky	1	1
			babcbbdd	1	1
3.	camera.bmp	256 x 256	erlarizky	1	1
			babcbbdd	1	1
4.	goldhill.bmp	256 x 256	erlarizky	1	1
			babcbbdd	1	1
5.	san.bmp	256 x 256	erlarizky	1	1
			babcbbdd	1	1
6.	baboon.png	512 x 512	erlarizky	1	1
			babcbbdd	1	1
7.	boat.png	512 x 512	erlarizky	1	1
			babcbbdd	1	1
8.	barbara.bmp	512 x 512	erlarizky	1	1
			babcbbdd	1	1
9.	hiu.jpeg	540 x 540	erlarizky	1	1
			babcbbdd	1	1
10.	owltikus.jpeg	540 x 540	erlarizky	1	1
			babcbbdd	1	1
11.	kucing.jpeg	800 x 800	erlarizky	3	3
			babcbbdd	3	3
12.	pinguin.png	800 x 800	erlarizky	3	3
			babcbbdd	3	3
13.	tank.jpeg	800 x 800	erlarizky	3	3
			babcbbdd	3	3
14.	tasya.png	1000 x 1000	erlarizky	5	6
			babcbbdd	5	6
15.	sg.jpeg	1000 x 1000	erlarizky	5	6
			babcbbdd	5	6

Tabel 4 Hasil Pengujian Enkripsi dan Dekripsi menggunakan Algoritma *Hill Cipher*

No	Nama File	Dimensi Citra	Kunci	Waktu Proses Enkripsi (detik)	Waktu Proses Dekripsi (detik)
1.	lena.jpeg	256 x 256	9901 dan 17	1	11
			17 dan 11	1	14
2.	lenagrey.jpeg	256 x 256	9901 dan 17	1	11
			17 dan 11	1	12
3.	camera.bmp	256 x 256	9901 dan 17	1	9
			17 dan 11	1	14
4.	goldhill.bmp	256 x 256	9901 dan 17	1	12
			17 dan 11	1	13
5.	san.bmp	256 x 256	9901 dan 17	1	11
			17 dan 11	1	12
6.	baboon.png	512 x 512	9901 dan 17	3	59
			17 dan 11	4	56
7.	boat.png	512 x 512	9901 dan 17	3	60
			17 dan 11	4	58
8.	barbara.bmp	512 x 512	9901 dan 17	3	58
			17 dan 11	4	64
9.	hiu.jpeg	540 x 540	9901 dan 17	3	75
			17 dan 11	5	87
10.	owltikus.jpeg	540 x 540	9901 dan 17	3	72
			17 dan 11	5	86
11.	kucing.jpeg	800 x 800	9901 dan 17	7	196
			17 dan 11	14	185
12.	pinguin.png	800 x 800	9901 dan 17	7	210
			17 dan 11	13	186
13.	tank.jpeg	800 x 800	9901 dan 17	7	183
			17 dan 11	12	188
14.	tasya.png	1000 x 1000	9901 dan 17	12	332
			17 dan 11	19	306
15.	sg.jpeg	1000 x 1000	9901 dan 17	12	310
			17 dan 11	19	317

4.2.2 PENGUJIAN BERDASARKAN KUALITAS CITRA

Pengujian dilakukan dengan cara mendapatkan nilai PSNR antara citra asli dan citra hasil rekayasa pada aplikasi kriptografi citra digital yang menggunakan algoritma *Hill Cipher* dan *Affine Cipher*. Masukkan berupa citra dengan dimensi ukuran dan kata kunci yang sama. Hasil pengujian dapat dilihat pada Tabel 5.

Tabel 5 Hasil Pengujian Berdasarkan Kualitas Citra

No	Nama File	Dimensi Citra	Kunci	Nilai PSNR Enkripsi (dB)	Nilai PSNR Dekripsi (dB)
1.	lena.jpeg	256 x 256	9901, 17, erlarizky	3.84881	100
			17, 11, bbabcbbdd	3.83344	100
2.	lenagrey.jpeg	256 x 256	9901, 17, erlarizky	3.75996	100
			17, 11, bbabcbbdd	3.76743	100
3.	camera.bmp	256 x 256	9901, 17, erlarizky	3.68786	100
			17, 11, bbabcbbdd	3.54972	100
4.	goldhill.bmp	256 x 256	9901, 17, erlarizky	4.27313	100
			17, 11, bbabcbbdd	4.28781	100
5.	san.bmp	256 x 256	9901, 17, erlarizky	4.13926	100
			17, 11, bbabcbbdd	4.19505	100
No	**Nama File**	**Dimensi Citra**	**Kunci**	**Nilai PSNR Enkripsi (dB)**	**Nilai PSNR Dekripsi (dB)**
6.	baboon.png	512 x 512	9901, 17, erlarizky	100	100
			17, 11, bbabcbbdd	100	100
7.	boat.png	512 x 512	9901, 17, erlarizky	9.96773	100
			17, 11, bbabcbbdd	9.92616	100
8.	barbara.bmp	512 x 512	9901, 17, erlarizky	9.44260	100
			17, 11, bbabcbbdd	9.58896	100
9.	hiu.jpeg	540 x 540	9901, 17, erlarizky	15.44502	100
			17, 11, bbabcbbdd	13.08066	100
10.	owltikus.jpeg	540 x 540	9901, 17, erlarizky	100	100
			17, 11, bbabcbbdd	100	100
11.	kucing.jpeg	800 x 800	9901, 17, erlarizky	100	100
			17, 11, bbabcbbdd	100	100
12.	pinguin.png	800 x 800	9901, 17, erlarizky	18.81114	100
			17, 11, bbabcbbdd	100	100
13.	tank.jpeg	800 x 800	9901, 17, erlarizky	100	100

			17, 11, bbabcbbdd	36.08960	100
14.	tasya.png	1000 x 1000	9901, 17, erlarizky	17.72083	100
			17, 11, bbabcbbdd	100	100
15.	sg.jpeg	1000 x 1000	9901, 17, erlarizky	20.36379	100
			17, 11, bbabcbbdd	21.68642	100

Pengujian juga dilakukan dengan membandingkan aplikasi serupa yang hanya menggunakan algoritma *Hill Cipher* atau *Affine Cipher* saja. Hasil pengujian berdasarkan kualitas citra menggunakan algoritma *Hill Cipher* dapat dilihat pada Tabel 6 dan hasil pengujian berdasarkan kualitas citra menggunakan algoritma *Affine Cipher* dapat dilihat pada Tabel 7.

Tabel 6 Hasil Pengujian Berdasarkan Kualitas Citra menggunakan Algoritma *Hill Cipher*

No	Nama File	Dimensi Citra	Kunci	Nilai PSNR Enkripsi (dB)	Nilai PSNR Dekripsi (dB)
1.	lena.jpeg	256 x 256	erlarizky	3.85903	100
			babcbbdd	4.12422	100
2.	lenagrey.jpeg	256 x 256	erlarizky	3.76075	100
			babcbbdd	4.01174	100
3.	camera.bmp	256 x 256	erlarizky	3.71668	100
			babcbbdd	5.24093	100
4.	goldhill.bmp	256 x 256	erlarizky	4.29534	100
			babcbbdd	4.04857	100
5.	san.bmp	256 x 256	erlarizky	4.17893	100
			babcbbdd	4.18068	100
6.	baboon.png	512 x 512	erlarizky	100	100
			babcbbdd	100	100
No	**Nama File**	**Dimensi Citra**	**Kunci**	**Nilai PSNR Enkripsi (dB)**	**Nilai PSNR Dekripsi (dB)**
7.	boat.png	512 x 512	erlarizky	10.05681	100
			babcbbdd	100	100
8.	barbara.bmp	512 x 512	erlarizky	9.39353	100
			babcbbdd	100	100
9.	hiu.jpeg	540 x 540	erlarizky	14.96900	100
			babcbbdd	100	100
10.	owltikus.jpeg	540 x 540	erlarizky	100	100
			babcbbdd	100	100
11.	kucing.jpeg	800 x 800	erlarizky	100	100
			babcbbdd	100	100
12.	pinguin.png	800 x 800	erlarizky	100	100
			babcbbdd	100	100
13.	tank.jpeg	800 x 800	erlarizky	13.5793	100
			babcbbdd	14.04840	100
14.	tasya.png	1000 x 1000	erlarizky	16.40186	100
			babcbbdd	100	100
15.	sg.jpeg	1000 x 1000	erlarizky	100	100
			babcbbdd	15.96387	100

Tabel 7 Hasil Pengujian Berdasarkan Kualitas Citra menggunakan Algoritma *Affine Cipher*

No	Nama File	Dimensi Citra	Kunci	Nilai PSNR	Nilai PSNR
				Enkripsi (dB)	Dekripsi (dB)
1.	lena.jpeg	256 x 256	9901 dan 17	3.92943	100
			17 dan 11	3.78974	100
2.	lenagrey.jpeg	256 x 256	9901 dan 17	4.03401	100
			17 dan 11	3.69927	100
3.	camera.bmp	256 x 256	9901 dan 17	3.40119	100
			17 dan 11	3.02214	100
4.	goldhill.bmp	256 x 256	9901 dan 17	4.39157	100
			17 dan 11	4.18769	100
5.	san.bmp	256 x 256	9901 dan 17	4.28190	100
			17 dan 11	4.13043	100
6.	baboon.png	512 x 512	9901 dan 17	100	100
			17 dan 11	100	100
7.	boat.png	512 x 512	9901 dan 17	10.5591	100
			17 dan 11	9.22436	100
8.	barbara.bmp	512 x 512	9901 dan 17	9.60966	100
			17 dan 11	9.29703	100
9.	hiu.jpeg	540 x 540	9901 dan 17	100	100
			17 dan 11	12.4676	100
10.	owltikus.jpeg	540 x 540	9901 dan 17	100	100
			17 dan 11	100	100
11.	kucing.jpeg	800 x 800	9901 dan 17	100	100
			17 dan 11	100	100
12.	pinguin.png	800 x 800	9901 dan 17	100	100
			17 dan 11	100	100
13.	tank.jpeg	800 x 800	9901 dan 17	100	100
			17 dan 11	100	100
14.	tasya.png	1000 x 1000	9901 dan 17	100	100
			17 dan 11	100	100
15.	sg.jpeg	1000 x 1000	9901 dan 17	18.2720	100
			17 dan 11	21.1237	100

5 KESIMPULAN

Kesimpulan yang dapat diambil dari penelitian mengenai aplikasi kriptografi citra digital menggunakan algoritma *Hill Cipher* dan *Affine Cipher* pada perangkat Android adalah aplikasi berhasil dibangun dan dapat melakukan proses enkripsi, dekripsi dan menghitung nilai PSNR pada perangkat android sesuai dengan *requirements* yang dibutuhkan.

Aplikasi ini juga telah diuji dan dibandingkan dengan aplikasi yang hanya menggunakan algoritma *Hill Cipher* atau *Affine Cipher* saja. Hasil yang diperoleh adalah penerapan kombinasi dua algoritma antara *Hill Cipher* dan *Affine Cipher* berhasil menghasilkan suatu citra yang lebih acak dan hampir tidak dikenali lagi jika dibandingkan dengan aplikasi serupa yang hanya menggunakan algoritma *Hill Cipher* atau *Affine Cipher* saja. Waktu yang dibutuhkan untuk proses enkripsi dan dekripsi pada kombinasi dua algoritma antara *Hill Cipher* dan *Affine Cipher* memiliki waktu yang lebih

lama jika dibandingkan dengan aplikasi kriptografi citra digital yang hanya menggunakan algoritma *Hill Cipher* atau *Affine Cipher* saja. Nilai PSNR yang dihasilkan dari membandingkan citra asli dengan citra hasil proses enkripsi pada aplikasi kriptografi citra digital menggunakan algoritma *Hill Cipher* dan *Affine Cipher* memiliki nilai rata-rata yang paling rendah jika dibandingkan dengan aplikasi serupa yang menggunakan algoritma *Hill Cipher* atau *Affine Cipher* saja.

Waktu yang dibutuhkan untuk proses enkripsi dan dekripsi dipengaruhi oleh ukuran citra dan kunci yang dimasukkan. Semakin besar ukuran citra dan kombinasi kunci yang dimasukkan membutuhkan waktu yang lebih lama untuk aplikasi melakukan proses enkripsi dan dekripsi.

DAFTAR PUSTAKA

[1] Acharya, B., Kumar, S. P., & Panda, G. 2010. *Image Encryption Using Advanced Hill Cipher Algorithm*. *ACEEE International Journal on Signal and Image Processing Vol. 1, No 1, Jan 2010.*

[2] Ariyus, Dony. 2008. *Pengantar Ilmu Kriptografi Teori Analisis dan Implementasi*. Yogyakarta: Penerbit Andi.

[3] Cox, C., & Kratzke, C. (2012). *Research and Development of Mobile Application for Android Platform*. *International Electronic Journal of Health Education, 15*(1), 72-82.

[4] Kadir, A. (2013). *From Zero to A Pro: Pemrograman Aplikasi Android* (1st ed.). Yogyakarta: Penerbit Andi.

[5] Menezes, A. dkk. 1996. *Handbook of Applied Cryptography*. CRC Press, Inc.

[6] Miles, R., & Hamilton, K. (2006). *Learning UML 2.0* (1st ed.). California: O'Reilly Media Inc.

[7] Munir, R. 2006. *Kriptografi*. Bandung: Institut Teknologi Bandung.

[8] Shapiro, Linda G. dan George C. Stockman. 2001. *Computer Vision.* New Jersey: Prentice Hall.

[9] Sadikin, Rifki. 2012. *Kriptografi untuk Keamanan Jaringan dan Implementasinya dalam Bahasa Java*. Yogyakarta: Penerbit Andi.

[10] Cole E. 2003. *Hiding in Plain Sight Steganography and the Art of Covert Communication.* Indiana: Wiley.

[11] Anwar, K., Sugiharto, A., & Sasongko, P. S. (2008). *Kompresi Citra Medis Menggunakan Discrete Wavelet Transform (DWT) dan Embedded Zerotree Wavelet (EZW).* Jurnal Matematika 11:73-77.

[12] Booch, G., Rumbaugh, J., & Jacobson, I. (2005). *The Unified Modelling language User Guide SECOND EDITION*. Addison Wesley.

[13] Arlow, J., & Neustadt, I. (2002). *UML and The Unified Process Practical Object-Oriented Analysist & Design.* Pearson Education Limited, Great Britain.

[14] Jacobson, I., Booch, G., & Rumbaugh, J. (1999). *The Unified Software Development Process.* Addison Wesley.