PENGGABUNGAN ALGORITMA *REVERSED VIGENERE ENCRYPTION* DENGAN MODIFIKASI ALGORITMA *SKIPJACK* PADA PENYANDIAN CITRA RGB

SKRIPSI

Oleh

Ahmad Rico Santoso
NIM 141810101007

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2018

PENGGABUNGAN ALGORITMA *REVERSED VIGENERE ENCRYPTION* DENGAN MODIFIKASI ALGORITMA *SKIPJACK* PADA PENYANDIAN CITRA RGB

SKRIPSI

diajukan guna memenuhi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Matematika (S1)
dan mencapai gelar Sarjana Sains

Oleh

Ahmad Rico Santoso
NIM 141810101007

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2018

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Kedua Orang tuaku tercinta, Ayahanda Samsuri dan Ibunda Jam'iyah, serta Kakakku Slamet Febriyanto yang senantiasa memberikan perhatian, motivasi dan doa;
2. Guru-guruku dari sekolah dasar hingga perguruan tinggi;
3. Almamater Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

MOTO

" Keagungan ilmu dapat diraih dengan kesungguhan bukan kekayaan. Siapa bersungguh-sungguh mencari sesuatu pasti akan menemukannya. Seseorang akan mendapatkan sesuatu yang dicarinya sesuai dengan usaha yang dilakukannya. Sejauh mana usahamu sekian pula tercapai cita-citamu"

(Beberapa Sya'ir dalam Kitab Taklimul Muta'allim, karya Syaikh Al Zarnuji)

A. Ma'ruf Asrori. 2012. *Etika Belajar Bagi Penuntut Ilmu*. Surabaya: "Al-Miftah" Surabaya.

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

nama : Ahmad Rico Santoso

NIM : 141810101007

menyatakan dengan sesungguhnya bahwa skripsi yang berjudul "Penggabungan Algoritma *Reversed Vigenere Encryption* dengan Modifikasi Algoritma *Skipjack* pada Penyandian Citra RGB" adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, Juli 2018

Yang menyatakan,

Ahmad Rico Santoso
NIM 141810101007

SKRIPSI

PENGGABUNGAN ALGORITMA *REVERSED VIGENERE ENCRYPTION* DENGAN MODIFIKASI ALGORITMA *SKIPJACK* PADA PENYANDIAN CITRA RGB

Oleh:

Ahmad Rico Santoso

NIM. 141810101007

Pembimbing

Dosen Pembimbing Utama : Abduh Riski, S.Si., M.Si.

Dosen Pembimbing Anggota : Ahmad Kamsyakawuni, S.Si., M.Kom.

PENGESAHAN

Skripsi berjudul "Penggabungan Algoritma *Reversed Vigenere Encryption* dengan Modifikasi Algoritma *Skipjack* pada Penyandian Citra RGB" telah diuji dan disahkan pada:

hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember

Tim Penguji:

Ketua,

Abduh Riski, S.Si., M.Si.
NIP 199004062015041001

Anggota I,

Ahmad Kamsyakawuni, S.Si., M.Kom.
NIP 197211291998021001

Anggota II,

Bagus Juliyanto, S.Si., M.Si.
NIP 198007022003121001

Anggota III,

Ikhsanul Halikin, S.Pd., M.Si.
NIP 198610142014041001

Mengesahkan

Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Jember

Drs. Sujito., Ph.D.
NIP 196102041987111001

RINGKASAN

Penggabungan Algoritma *Reversed Vigenere Encryption* dengan Modifikasi Algoritma *Skipjack* pada Penyandian Citra RGB; Ahmad Rico Santoso, 141810101007; 2018: 72 halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Perkembangan teknologi dan informasi pada saat ini berkembang sangat pesat dengan berbagai fitur-fitur terbaru. Perkembangan teknologi dan informasi ini masih perlu diperhatikan terutama masalah keamanan data dan informasi. Pengamanan data dan informasi penting dilakukan untuk mencegah bocornya suatu pesan atau informasi kepada orang yang tidak berhak menerima. Oleh karena itu, pengamanan suatu data perlu diperhatikan dan dilakukan dengan menggunakan suatu teknik yang dinamakan dengan kriptografi. Kriptografi merupakan suatu ilmu dan seni untuk melindungi atau menyembunyikan pesan/informasi agar tidak mudah diketahui oleh orang yang tidak berhak menerima pesan/informasi..

Pada penelitian ini, pesan yang digunakan dalam penyandian adalah pesan dalam bentuk citra RGB. Pesan citra disandikan menggunakan metode gabungan algoritma *Reversed Vigenere Encryption* dengan modifikasi algoritma *Skipjack*. Algoritma *Reversed Vigenere Encryption* merupakan algoritma *vigenere cipher* pada umumnya namun kunci yang digunakan pada proses enkripsi dan dekripsi dilakukan transposisi kebalikan (*permutation reversed*). Adapun pada algoritma *Skipjack* dilakukan modifikasi pada *F-Table*, jumlah putaran pada proses enkripsi-dekripsi dan penggunaan aturan pada proses enkripsi-dekripsi. Setiap elemen-elemen dalam *F-Table* diubah penempatannya secara acak agar kerahasiaan proses enkripsi dan dekripsi tetap terjaga. Jumlah putaran pada algoritma Skipjack diringkas dari yang semula 32 putaran menjadi 8 putaran untuk mempercepat proses enkripsi dan dekripsi. Adapun penggunaan aturan enkripsi menggunakan *rule A* pada 4 putaran pertama dan mengunakan *rule B* pada 4 putaran kedua

sedangkan penggunaan aturan dekripsi menggunakan *rule* B^{-1} pada 4 putaran pertama dan menggunakan *rule* A^{-1} pada 4 putaran kedua.

Proses enkripsi pada citra menggunakan metode yang diusulkan menghasilkan *cipherimage* yang berbeda secara signifikan terhadap citra aslinya. Proses dekripsi pada *cipherimage* dapat dikembalikan sesuai dengan bentuk citra aslinya tanpa mengubah nilai-nilai *pixels* dari *plainimage*. Hasil enkripsi pada citra dianalisis keamanannya menggunakan analisis histogram dan analisis diferensial. Hasil analisis histogram yang didapatkan adalah nilai-nilai *pixels* pada *cipherimage* menghasilkan penyebaran nilai-nilai *pixels* secara merata. Hal ini dapat dikatakan bahwa hasil enkripsi pada citra dapat tahan dan kuat terhadap serangan-serangan kriptoanalisis tipe statistik. Berdasarkan data penelitian 10 citra yang telah diuji, hasil analisis diferensial didapatkan nilai NPCR mendekati 100% yaitu sebesar 99,5870% hingga 99,6399% sedangkan nilai UACI didapatkan sebesar 29,1768% hingga 36,4841%. Berdasarkan hasil analisis histogram dan analisis diferensial, *Chiperimage* yang dihasilkan dari proses enkripsi menggunakan metode yang diusulkan dapat tahan terhadap serangan-serangan kriptoanalisis tipe statistik dan serangan diferensial.

PRAKATA

Puji syukur ke hadirat Allah Swt. atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "Penggabungan Algoritma *Reversed Vigenere Encryption* dengan Modifikasi Algoritma *Skipjack* pada Penyandian Citra RGB". Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan strata satu (S1) pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Abduh Riski S.Si., M.Si., dan Ahmad Kamsyakawuni, S.Si., M.Kom., selaku Dosen Pembimbing yang telah memberikan bimbingan dan bantuan dalam penyempurnaan skripsi ini;
2. Bagus Juliyanto, S.Si., M.Si., dan Ikhsanul Halikin, S.Pd., M.Si., selaku Dosen Penguji yang telah memberikan kritik dan saran yang membangun dalam penyempurnaan skripsi ini;
3. Kosala Dwija Purnama, S.Si., M.Si., selaku selaku Dosen Pembimbing Akademik yang telah membimbing dalam pemilihan matakuliah;
4. Ayahanda Samsuri dan Ibunda Jam'iyah serta kakakku Slamet Febriyanto yang telah memberikan dukungan, motivasi dan doa;
5. Seluruh teman-teman "EXTREME" 2014 dan teman-teman "Kuda Perjaka 14" yang telah memberikan motivasi serta dukungannya;
6. Sahabat-sahabati PMII Rayon FMIPA Universitas Jember yang telah memberikan motivasi dan pengalamannya;
7. Semua pihak yang tidak dapat disebutkan satu per satu.

Penulis menerima segala kritik dan saran yang bersifat membangun dari semua pihak demi kesempurnaan penulisan skripsi ini. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, Juli 2018

Penulis

DAFTAR ISI

Halaman

DAFTAR TABEL

Halaman

DAFTAR GAMBAR

DAFTAR LAMPIRAN

Halaman

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Seiring dengan perkembangan teknologi dan informasi, hal yang paling populer dalam proses perkembangannya adalah komputer dan jaringan. Perkembangan komputer dan jaringan pada saat ini berkembang sangat pesat dengan berbagai fitur-fitur terbaru. Perkembangan teknologi dan informasi ini masih perlu diperhatikan terutama masalah keamanan data dan informasi. Pengamanan data dan informasi penting dilakukan untuk mencegah bocornya suatu pesan atau informasi kepada orang yang tidak berhak menerima. Oleh karena itu, pengamanan suatu data perlu diperhatikan dan dilakukan dengan menggunakan suatu teknik yang dinamakan kriptografi.

Kriptografi merupakan suatu ilmu dan seni untuk melindungi atau menyembunyikan pesan/informasi agar tidak mudah diketahui oleh orang lain atau orang yang tidak berhak menerima pesan/informasi dengan cara mengubah pesan asli menjadi kode-kode yang sulit dimengerti maknanya. Dalam kriptografi terdapat algoritma-algoritma untuk menyandikan suatu data/informasi yaitu algoritma kriptografi klasik dan algoritma kriptografi modern. Algoritma kriptografi klasik merupakan algoritma pada masa lampau yang biasanya digunakan untuk menyusun strategi perang, bersifat sederhana dan berbasis substitusi karakter ke karakter. Sedangkan algoritma kriptografi modern merupakan algoritma yang umumnya digunakan pada masa saat ini dan biasanya pada proses penyandiannya berbasis angka atau bit-string. Pada algoritma kriptografi modern biasanya memiliki tingkat keamanan yang cukup kuat dibandingkan dengan tingkat keamanan pada algoritma kriptografi klasik.

Pada penelitian ini, penyandikan pesan/informasi dalam bentuk citra RGB. Pesan atau informasi yang disandikan menggunakan gabungan algoritma kriptografi klasik dan algoritma kriptografi modern, yaitu algoritma *Reversed Vigenere Encryption* dan modifikasi algoritma *Skipjack*. Sebelumnya Sengupta (2013) telah menggunakan algoritma *Reversed Vigenere Encryption* untuk mendesain suatu sistem keamanan pada sistem *cloud computing*. Pada penelitian

tersebut, algoritma *Reversed Vigenere Encryption* pada dasarnya sama dengan algoritma *Vigenere Cipher*, hanya saja kunci pada algoritma ini dibalik atau dilakukan transposisi kebalikan (*permutation reversed*), sedangkan pada *Vigenere Cipher* masih terdapat kelemahan yaitu diulangnya kunci secara terus-menerus sehingga menimbulkan enkripsi yang sama pada potongan plainteksnya sehingga mudah dipecahkan menggunakan metode kasiski. Oleh karena itu, pada penelitian ini algoritma *Reversed Vigenere Encryption* perlu diperkuat keamanannya dengan cara menggabungkannya dengan algoritma kriptografi lain. Algoritma yang akan digabungkan pada penelitian ini adalah algoritma *Skipjack* dengan memodifikasi beberapa bagian pada algoritma *Skipjack*.

Bagian algoritma *Skipjack* yang akan dimodifikasi adalah *F-Table*, jumlah putaran pada proses enkripsi-dekripsi dan penggunaan aturan pada proses enkripsi-dekripsi. Setiap elemen-elemen dalam *F-Table* akan diubah penempatannya secara acak agar kerahasiaan proses enkripsi dan dekripsi tetap terjaga. Adapun jumlah putaran pada algoritma *Skipjack* akan diringkas dari yang semula 32 putaran menjadi 8 putaran untuk mempercepat proses enkripsi dan dekripsi. Adapun penggunaan aturan enkripsi menggunakan *rule A* pada 4 putaran pertama dan mengunakan *rule B* pada 4 putaran kedua sedangkan penggunaan aturan dekripsi menggunakan *rule* B^{-1} pada 4 putaran pertama dan menggunakan *rule* A^{-1} pada 4 putaran kedua.

Algoritma skipjack merupakan algoritma kriptografi modern yang dikembangkan oleh *National Security Agency* (NSA) di Amerika Serikat untuk menjamin keamanan komunikasi via telepon. Algoritma *Skipjack* merupakan algoritma yang menggunakan 2 operasi dasar matematika yaitu teknik operasi XOR dan permutasi pada kunci. Algoritma Skipjack mengenkripsi plaintext 64 bit menjadi ciphertext 64 bit dengan jumlah putaran sebanyak 32 putaran. Kunci yang digunakan pada algoritma skipjack sepanjang 80 bit atau 10 karakter. Salah satu keunggulan algoritma Skipjack adalah usaha-usaha analisis kriptografi (kriptanalisis) tidak bergantung pada kerahasiaan algoritmanya (Hartono, 2013). Pada penelitian ini, penulis mengajukan gabungan algoritma *Reversed Vigenere Encryption* dengan modifikasi algoritma *Skipjack* yang akan digunakan pada

penyandian pesan berupa citra RGB. Penyandian pesan citra RGB menggunakan dua macam kunci yaitu kunci 1 dan kunci 2. Kunci 1 digunakan pada algoritma *Reversed Vigenere Encryption* sedangkan kunci 2 digunakan pada modifikasi algoritma *Skipjack*. Penggabungan algoritma *Reversed Vigenere Encryption* dengan modifikasi algoritma *Skipjack* ini dilakukan dengan harapan agar tingkat keamanan pada penyandian pesan citra RGB memiliki tingkat keamanan yang kuat.

1.2 Rumusan Masalah

Perumusan masalah yang akan dibahas pada penelitian kali ini adalah:

a. Bagaimanakah cara mengenkripsikan citra RGB menggunakan gabungan algoritma *Reversed Vigenere Encryption* dengan modifikasi algoritma *Skipjack*?
b. Bagaimanakah cara mendekripsikan citra RGB menggunakan gabungan algoritma *Reversed Vigenere Encryption* dengan modifikasi algoritma *Skipjack*?
c. Bagaimanakah hasil analisis keamanan pada penyandian citra RGB menggunakan gabungan algoritma *Reversed Vigenere Encryption* dengan modifikasi algoritma *Skipjack* menggunakan analisis histogram dan analisis diferensial?
d. Bagaimanakah perbandingan antara hasil enkripsi citra menggunakan algoritma *Reversed Vigenere Encryption* dengan hasil enkripsi citra menggunakan gabungan algoritma *Reversed Vigenere Encryption* dan modifikasi algoritma *Skipjack*?

1.3 Batasan Masalah

Batasan masalah pada penelitian ini adalah objek yang dienkripsi berupa citra RGB dan kunci yang digunakan pada penyandian menggunakan dua kunci berupa karakter yaitu kunci 1 dan kunci 2. Adapun kunci 2 memiliki panjang karakter berukuran 10 karakter karena kunci 2 digunakan pada algoritma *Skipjack*.

1.4 Tujuan Penelitian

Tujuan yang akan dicapai pada penelitian ini adalah:

a. Mengenkripsikan citra RGB menggunakan gabungan algoritma *Reversed Vigenere Encryption* dengan modifikasi algoritma *Skipjack*.
b. Mendekripsikan citra RGB menggunakan gabungan algoritma *Reversed Vigenere Encryption* dengan modifikasi algoritma *Skipjack*.
c. Menganalisis keamanan hasil enkripsi pada penyandian citra RGB menggunakan gabungan algoritma *Reversed Vigenere Encryption* dengan modifikasi algoritma *Skipjack* melalui analisis histogram dan analisis diferensial.
d. Membandingkan hasil enkripsi citra menggunakan algoritma *Reversed Vigenere Encryption* dengan hasil enkripsi citra menggunakan gabungan algoritma *Reversed Vigenere Encryption* dan modifikasi algoritma *Skipjack*

1.5 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah:

a. Mengetahui proses enkripsi dan dekripsi citra RGB menggunakan gabungan algoritma *Reversed Vigenere Encryption* dengan modifikasi algoritma *Skipjack*.
b. Mengetahui hasil analisis keamanan pada enkripsi citra RGB menggunakan gabungan algoritma *Reversed Vigenere Encryption* dengan modifikasi algoritma *Skipjack*.
c. Mengetahui perbandingan antara hasil enkripsi citra menggunakan algoritma *Reversed Vigenere Encryption* dengan hasil enkripsi citra menggunakan gabungan algoritma *Reversed Vigenere Encryption* dan modifikasi algoritma *Skipjack*
d. Sebagai bahan studi dan objek literatur bagi penulis dan pembaca untuk bidang teknologi dan informasi.

BAB 2. TINJAUAN PUSTAKA

2.1 Citra

Secara harfiah, citra merupakan gambar pada bidang dua dimensi. Ditinjau berdasarkan sudut pandang matematis, citra merupakan fungsi kontinu dari intensitas cahaya pada bidang dua dimensi. Sumber cahaya menerangi objek kemudian objek memantulkan kembali sebagian dari berkas cahaya tersebut. Pantulan cahaya ini selanjutnya ditangkap oleh alat-alat optik seperti mata pada manusia, kamera, pemindai (*scanner*) dan sebagainya sehingga bayangan objek yang disebut dengan citra tersebut terekam. Secara umum, citra terdiri dari dua macam, yaitu citra diam dan citra bergerak.

(a) lena (b) kapal

Gambar 2.1 (a) citra lena dan (b) citra kapal

Gambar 2.1 merupakan contoh dari citra diam. Citra diam adalah citra tunggal yang tidak bergerak sedangkan citra bergerak adalah rangkaian citra diam yang ditampilkan secara beruntun sehingga memberikan kesan pada mata kita sebagai gambar yang bergerak. Setiap citra didalam rangkaian itu disebut sebagai *frame*. Gambar-gambar yang tampak pada film atau video pada hakikatnya terdiri dari beberapa ratus atau ribuan *frame* (Munir, 2002).

2.1.1 Model Citra

Secara matematis, fungsi intensitas cahaya pada bidang dua dimensi disimbolkan dengan $f(x, y)$ dalam hal ini:

(x, y) : koordinat pada bidang dua dimensi

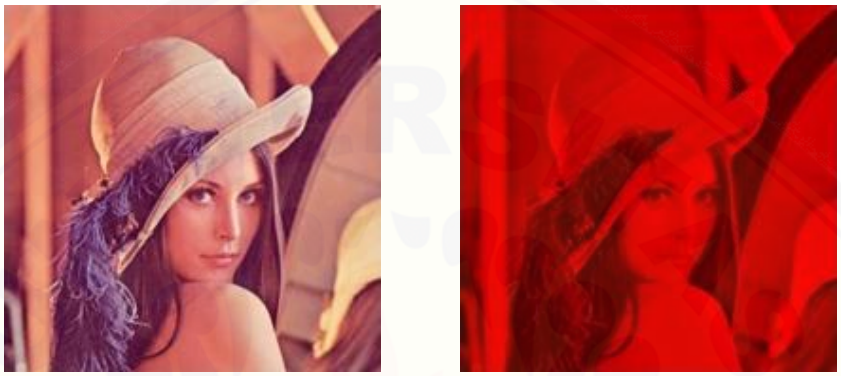
$f(x, y)$: intensitas cahaya pada titik (x, y)

Gambar 2.2 memperlihatkan cara menentukan koordinat titik pada suatu citra. Sistem koordinat yang diacu pada citra adalah sistem koordinat Cartesian dalam hal ini sumbu mendatar menyatakan sumbu x dan sumbu tegak menyatakan sumbu y.

Gambar 2.2 Cara menentukan koordinat titik pada citra

Intensitas f dari gambar hitam putih (x, y) disebut sebagai derajat keabuan (*grey level*) yang mana derajat keabuannya bergerak dari hitam ke putih sedangkan citranya disebut citra hitam putih atau citra monokrom. Derajat keabuan memiliki rentang nilai dari l_{min} sampai l_{max} atau dapat ditulis $l_{min} < f < l_{max}$. Selang (l_{min}, l_{max}) disebut skala keabuan. Selang (l_{min}, l_{max}) biasa digeser untuk alasan-alasan praktis menjadi selang [0, L] yang mana nilai intensitas 0 menyatakan hitam dan nilai intensitas L menyatakan putih. Sebagai contoh citra dengan skala keabuan [0, 255] yang mana nilai intensitas 0 menyatakan hitam dan nilai intensitas 255 menyatakan putih sehingga nilai antara 0 sampai 255 menyatakan warna keabuan yang terletak antara hitam dan putih. Citra hitam putih disebut sebagai citra satu kanal karena warnanya ditentukan oleh satu fungsi intensitas saja sedangkan citra

berwarna dikenal sebagai citra spektral atau citra RGB karena warna pada citra disusun atas 3 komponen warna yang disebut komponen RGB yaitu merah (*red*), hijau (*green*) dan biru (*blue*). Intensitas suatu titik pada citra berwarna atau citra spektral merupakan kombinasi dari 3 intensitas yaitu derajat keabuan merah ($f_{merah}(x,y)$), derajat keabuan hijau ($f_{hijau}(x,y)$) dan derajat keabuan biru ($f_{biru}(x,y)$) (Munir, 2002).



(a) Citra RGB (b) Kanal Merah

(c) Kanal Hijau (d) Kanal Biru

(Sumber: https://pemrogramanmatlab.com/2017/07/26/pengolahan-citra-digital/)

Gambar 2.3 Citra RGB beserta kanal warnanya

2.1.2 Digitalisasi Citra

Suatu citra agar dapat diolah menggunakan komputer digital maka citra harus direpresentasikan secara numerik atau nilai-nilai diskrit. Representasi citra dari fungsi kontinu menjadi nilai-nilai diskrit disebut *digitalisasi*. Citra yang dihasilkan pada digitalisasi disebut sebagai citra digital. Secara umum, citra digital berbentuk persegi panjang dan ukuran dimensinya dinyatakan sebagai

tinggi × lebar atau lebar × panjang. Citra digital yang berukuran N × M dinyatakan dengan matriks berukuran N baris dan M kolom sebagai berikut:

$$f(x,y) = \begin{bmatrix} f(0,0) & \cdots & f(0,M) \\ \vdots & \ddots & \vdots \\ f(N-1,0) & \cdots & f(N-1,M-1) \end{bmatrix}$$

Indeks baris (x) dan indeks kolom (y) menyatakan koordinat suatu titik pada citra sedangkan $f(x,y)$ merupakan intensitas (derajat keabuan) pada titik (x, y). Masing-masing elemen pada citra digital (elemen matriks) disebut sebagai *image element* atau *pixel*. Jadi, jika citra berukuran N × M maka citra tersebut memiliki NM buah *pixel* (Munir, 2002).

2.2 Kriptografi

Kriptografi merupakan suatu ilmu atau teknik yang digunakan untuk menyembunyikan suatu pesan/informasi agar tidak dapat diketahui oleh pihak yang tidak berhak mendapatkan informasi tersebut. Secara bahasa kriptografi berasal dari bahasa yunani yaitu "crypto" dan "graphia". Kata "crypto" berarti rahasia dan "graphia" berarti tulisan. Sehingga menurut terminologi, kriptografi merupakan suatu ilmu dan seni untuk menjaga keamanan atau kerahasiaan pesan ketika pesan dikirim dari suatu tempat ketempat yang lain (Ariyus, 2006).

Secara umum, komponen kriptografi terdiri dari beberapa komponen, yaitu:

a. Plainteks, sering disebut sebagai cleartext atau teks asli. Teks asli merupakan pesan yang memiliki makna yang diproses menggunakan algoritma kriptografi untuk menjadi teks kode.
b. Enkripsi, merupakan cara pengamanan data atau pesan asli (plainteks) sehingga terjaga kerahasiaannya. Proses enkripsi mengubah plainteks ke bentuk teks kode menggunakan algoritma yang dapat mengkodekan data yang kita inginkan.
c. Cipherteks, merupakan suatu pesan yang telah melalui proses enkripsi. Pesan yang ada berupa teks kode sehingga tidak bisa dibaca karena berupa karakter-karakter yang tidak mempunyai makna.
d. Dekripsi, merupakan kebalikan dari enkripsi. Pesan yang telah dienkripsi dikembalikan kebentuk asalnya yaitu plainteks.

e. Kunci, merupakan kunci yang digunakan untuk melakukan enkripsi ataupun dekripsi. Ada dua jenis kunci yaitu kunci simetris dan kunci asimetris. Kunci simetris merupakan kunci yang digunakan pada enkripsi sama dengan kunci yang digunakan pada dekripsi. Sedangkan kunci asimetris merupakan kunci yang digunakan pada enkripsi berbeda dengan kunci yang digunakan pada dekripsi. Pada kunci asimetris terdapat dua jenis kunci yaitu kunci umum *(public key*) dan kunci rahasia (*private key*). Kunci umum merupakan kunci yang dapat dipublikasikan atau boleh semua orang mengetahui sedangkan kunci rahasia merupakan kunci yang dirahasiakan atau hanya boleh diketahui oleh satu orang (Ariyus, 2008).

2.3 Teknik Transposisi

Teknik transposisi merupakan teknik memindahkan posisi karakter teks asli ke posisi teks lain tanpa mengubah nilai aslinya. Salah satu contoh sederhana dari teknik transposisi adalah teknik transposisi columnar. Teknik transposisi columnar mengubah karakter teks asli dengan cara menulis karakter teks asli dengan orientasi baris dengan panjang karakter yang sama kemudian teks sandi didapatkan dengan menulis ulang teks sesuai kolom yang disepakati sebelumnya.

Sebagai contoh, teks asli adalah “JURUSAN MATEMATIKA”, maka dengan menulis tabel yang terdiri dari 6 kolom dengan orientasi baris didapatkan:

Kunci	:	4	2	1	6	3	5
Teks asli	:	J	U	R	U	S	A
		N	M	A	T	E	M
		A	T	I	K	A	X

String X digunakan untuk mengisi sel kosong pada tabel. Selanjutnya tulis teks sandi sesuai dengan urutan berdasarkan kunci dengan orientasi kolom sehingga didapatkan teks sandi:

RAIUMTSEAJNAAMXUTK

(Sadikin, 2012).

Ada beberapa teknik-teknik transposisi lain, seperti zig-zag, segitiga, spiral, dan diagonal. Berikut adalah contoh-contoh dari teknik transposisi pada zig-zag, segitiga, spiral dan diagonal:

a. *Zig-zag*: memasukkan teks asli dengan pola zig-zag seperti contoh dibawah ini:

			A						G						A						T						I					X
		Y		S				N		B				J		R				P		O				F		C				A	
	A				E		A				E		A				K		I				G		A				I		R		
S						D						L						R						R						T				

Teks kode dari teknik ini dengan membaca dari baris atas ke baris bawah

“AGATIXYSNBJRPOFCAAEAEAKIGAIRSDLRRT”

b. *Segitiga*: masukkan teks asli dengan pola segitiga dan dibaca dari atas kebawah.

					S					
				A	Y	A				
			B	E	L	A	J			
		R	K	R	I	P	T	O		
	G	R	A	F	I	C	I	T	R	
A	X	X	X	X	X	X	X	X	X	X

Teks kodenya adalah:

“AGXRRXBKAXAERFXSYLIIXAAPCXJTIXOTXRXX”

c. *Spiral*: teks asli dimasukkan secara spiral dan dapat dibaca dari atas kebawah.

S	A	Y	A	S	E
I	P	T	O	G	D
R	R	A	X	R	A
K	T	X	X	A	N
R	I	C	I	F	G
A	J	A	L	E	B

Teks kodenya adalah:

“SIRKRAAPRTIJYTAXCAAOXXILSGRAFEEDANGB”

d. *Diagonal*: Dengan menggunakan pola ini teks asli dimasukkan dengan cara diagonal.

S	D	L	R	R	T
A	A	A	I	A	R
Y	N	J	P	F	A
A	G	A	T	I	X
S	B	R	O	C	X
E	E	K	G	I	X

Teks kodenya adalah

"SDLRRTAAAIARYNJPFAAGATIXSBROCXEEKGIX"

(Ariyus, 2008).

2.4 Vigenere Cipher

Vigenere cipher merupakan teknik enkripsi pada kriptografi klasik yang diperkenalkan oleh diplomat Perancis, yaitu Blaise de Vigenere pada Abad 16 pada tahun 1586. Sebelumnya Giovon Batista telah memperkenalkan untuk pertama kali pada tahun 1553 seperti yang terdapat dalam buku *La Cifra del Sig.* Algoritma ini baru dikenal luas setelah 200 tahun kemudian dan dinamakan kode *Vigenere* untuk digunakan oleh tentara Konfederasi (*Confederate Army*) pada Perang Sipil Amerika. Kode *Vigenere* berhasil dipecahkan oleh Babbage dan Kasiski pada pertengahan Abad 19. Pada teknik substitusi *vigenere* setiap teks kode bisa memiliki banyak kemungkinan teks asli. Teknik dari substitusi *vigenere* cipher bisa dilakukan dengan dua cara yaitu dengan menggnakan angka dan menggunakan huruf (Ariyus, 2008).

2.4.1 Vigenere Cipher Menggunakan Angka

Teknik substitusi Vigenere dengan menggunakan angka dilakukan dengan menukarkan huruf dengan angka. Adapun tabel substitusi Vigenere menggunakan angka adalah sebagai berikut:

Tabel 2.1 Substitusi Vigenere Cipher Menggunakan Angka

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Jika kunci yang digunakan adalah 6 kode yaitu K = (2, 8, 15, 7, 4, 17) dan teks aslinya “This cryptosystem is not secure” maka hasil enkripsi yang didapatkan adalah sebagai berikut:

T	H	I	S	C	R	Y	P	T	O	S	Y	S	T
19	7	8	18	2	17	24	15	19	14	18	24	18	19
2	8	15	7	4	17	2	8	15	7	4	17	2	8
21	15	23	25	6	8	0	23	8	21	22	15	20	27

E	M	I	S	N	O	T	S	E	C	U	R	E
4	12	8	18	13	14	19	18	4	2	20	17	4
15	7	4	17	2	8	15	7	4	17	2	8	15
19	19	12	9	15	22	8	25	8	19	22	15	19

Plainteks : This cryptosystem is not secure

Kunci : (2, 8, 15, 7, 4, 17)

Cipherteks : 21 15 23 25 6 8 0 23 8 21 22 15 20 27 19 19 12 9 15 22 8 25 8 19 22 15 19 atau jika diubah kebentuk huruf cipherteks yang dihasilkan adalah VPXZGIAXIVWPUBTTMJPWIZITWZT

(Ariyus, 2006).

Kunci yang digunakan pada *vigenere cipher* dibuat berulang sepanjang plainteks sehingga jumlah huruf pada kunci akan sama dengan jumlah huruf pada plainteks. Pergeseran setiap huruf pada plainteks akan ditentukan oleh huruf pada kunci yang mempunyai posisi yang sama dengan huruf pada plainteks. Adapun fungsi enkripsi dan dekripsi pada *vigenere cipher* adalah seperti pada Persamaan (2.1) dan Persamaan (2.2) berikut:

$$Ci = E(Pi) = (Pi+Ki) \bmod 26 \quad (2.1)$$

$$Pi = D(Ci) = (Ci-Ki) \bmod 26 \quad (2.2)$$

yang mana Ci = Cipherteks, Pi = Plainteks dan Ki = Kunci (Hallim dkk, 2010).

2.4.2 Vigenere Cipher Menggunakan Huruf

Teknik substitusi *vigenere cipher* menggunakan huruf bisa digunakan tabel sebagai berikut:

Tabel 2.2 Substitusi *Vigenere Cipher* Menggunakan Huruf

Kunci \ Plainteks	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	B	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Plainteks : SAYA BELAJAR KRIPTOGRAFI CITRA

Kunci : SAYA BELAJAR KRIPTOGRAFI CITRA

Cipherteks : KAWA CIWASAI UIQEMCMIAKQ EQMIA

Cara menentukan cipherteks pada system ini, pada tabel bisa dilihat pada posisi horizontal merupakan plainteks dan pada posisi vertical kunci, jika plainteks huruf K maka lihat posisi letak huruf K pada plainteks tabel dan posisi huruf K pada posisi kunci, jika sudah menemukan tarik garis lurus kebawah dari plainteks dan

garis lurus kesamping dari posisi kunci sehingga menemukan huruf U, maka huruf U yang akan menjadi cipherteks dan begitu seterusnya (Ariyus, 2006).

2.4.3 Reversed Vigenere Encryption

Reversed Vigenere Encryption merupakan algoritma kriptografi klasik seperti *Vigenere cipher* pada umumnya tetapi kunci yang digunakan pada algoritma dilakukan transposisi kebalikan (*permutation reversed*) terlebih dahulu. Sebagai contoh yaitu jika terdapat kunci yang akan digunakan adalah "KEAMANAN" maka kunci tersebut dilakukan transposisi kebalikan (*permutation reversed*) terlebih dahulu sehingga kunci akhir yang akan digunakan adalah "NANAMAEK" (Sengupta dkk., 2013).

2.4.4 Metode Kasiski

Metode Kasiski merupakan metode kriptanalisis yang digunakan untuk memecahkan kode *vigenere chiper*. Metode Kasiski diperkenalkan pertama kali oleh Friedrich Kasiski pada tahun 1863. Metode Kasiski digunakan untuk menemukan panjang kunci pada enkripsi *vigenere cipher*. Adapun langkah-langkah pada metode kasiski adalah sebagai berikut:

1. Temukan beberapa karakter *cipherteks* yang sama atau berulang
2. Hitung jarak antara *cipherteks* yang sama ke *cipherteks* yang sama lainya
3. Hitung semua faktor (pembagi) dari jarak tersebut.
4. Tentukan irisan dari himpunan faktor pembagi tersebut.
5. Nilai yang muncul didalam irisan menyatakan angka yang muncul pada semua faktor pembagidari jarak-jarak tersebut. Nilai irisan yang diperoleh merupakan panjang kunci yang mungkin digunakan.

Contoh penggunaan metode Kasiski adalah sebagai berikut:

Cipherteks:

LJVBQ STNEZ LQMED **LJV**MA MPKAU FAVAT **LJV**DA YYVNF JQLNP **LJV**HK VTRNF **LJV**CM LKETA **LJV**HU YJVSF KRFTT WEFUX VHZNP

Cari *cipherteks* yang sama atau berulang yaitu LJV

Hitung jarak antara *cipherteks* yang sama:

Jarak LJV ke-1 dengan LJV ke-2 = 15

Jarak LJV ke-2 dengan LJV ke-3 = 15

Jarak LJV ke-3 dengan LJV ke-4 = 15

Jarak LJV ke-4 dengan LJV ke-5 = 10

Jarak LJV ke-5 dengan LJV ke-6 = 10

Faktor pembagi 15 = {3, 5, 15}

Faktor pembagi 10 = {2, 5, 10}

Irisan dari kedua faktor pembagi tersebut adalah 5, jadi kemungkinan panjang kunci adalah 5 karakter (Munir, 2017).

2.5 Algoritma Skipjack

Algoritma *skipjack* merupakan algoritma kriptografi modern yang dikembangkan oleh National Security Agency (NSA) di Amerika Serikat untuk mengamankan komunikasi via telepon. Algoritma *skipjack* mengenkripsi plainteks sebanyak 64 bit menjadi cipherteks sebanyak 64 bit dengan jumlah putaran sebanyak 32 putaran. Kunci yang digunakan pada algoritma *skipjack* berjumlah 10 karakter (80 bit). Proses enkripsi pada algoritma *skipjack* menggunakan dua buah *rule* yaitu *rule* A dan *rule* B sedangkan pada proses dekripsi *rule* yang digunakan berkebalikan dengan *rule* pada proses enkripsi yaitu menggunakan dua buah *rule* A^{-1} dan *rule* B^{-1}. Operasi-operasi dasar yang digunakan pada algoritma *skipjack* yaitu menggunakan teknik operasi XOR dan permutasi. Adapun tabel kebenaran dari teknik operasi XOR adalah sebagai berikut:

Tabel 2.3 Teknik Operasi XOR

Bilangan 1	Bilangan 2	Bilangan 1 ⊕ Bilangan 2
0	0	0
0	1	1
1	0	1
1	1	0

(Sumber: Yohandri, 2013)

Operasi permutasi dilakukan menggunakan potongan plainteks, kunci rahasia dan suatu tabel substitusi yang disebut sebagai *F-Table* (Tabel 2.5). Nilai-nilai yang terdapat dalam *F-Table* berupa bilangan-bilangan heksadesimal.Berikut *F-Table* yang digunakan pada saat proses permutasi:

Tabel 2.4 *F-Table* Algoritma Skipjack

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	A3	D7	09	83	F8	48	F6	F4	B3	21	15	78	99	B1	AF	F9
1x	E7	2D	4D	8A	CE	4C	CA	2E	52	95	D9	1E	4E	38	44	28
2x	0A	DF	02	A0	17	F1	60	68	12	B7	7A	C3	E9	FA	3D	53
3x	96	84	6B	BA	F2	63	9A	19	7C	AE	E5	F5	F7	16	6A	A2
4x	39	B6	7B	0F	C1	93	81	1B	EE	B4	1A	EA	D0	91	2F	B8
5x	55	B9	DA	85	3F	41	BF	E0	5A	58	80	5F	66	0B	D8	90
6x	35	D5	C0	A7	33	06	65	69	45	00	94	56	6D	98	9B	76
7x	97	FC	B2	C2	B0	FE	DB	20	E1	EB	D6	E4	DD	47	4A	1D
8x	42	ED	9E	6E	49	3C	CD	43	27	D2	07	D4	DE	C7	67	18
9x	89	CB	30	1F	8D	C6	8F	AA	C8	74	DC	C9	5D	5C	31	A4
Ax	70	88	61	2C	9F	0D	2B	87	50	82	54	64	26	7D	03	40
Bx	34	4B	1C	73	D1	C4	FD	3B	CC	FB	7F	AB	E6	3E	5B	E5
Cx	AD	04	23	9C	14	51	22	F0	29	79	71	7E	FF	8C	0E	E2
Dx	0C	EF	BC	72	75	6F	37	A1	EC	D3	8E	62	8B	86	10	E8
Ex	08	77	11	BE	92	4F	24	C5	32	36	9D	CF	F3	A6	BB	AC
Fx	5E	6C	A9	13	57	25	B5	E3	BD	A8	3A	01	05	59	2A	46

F-Table digunakan untuk mengkonversi hasil fungsi permutasi. Sebagai contoh hasil fungsi permutasi dari F(ED) = A6 (Hartono, 2013).

2.5.1 Algoritma Pengolahan Kunci

Algoritma skipjack memiliki panjang kunci rahasia sebanyak 80 bit. Kunci rahasia dibagi beberapa bagian menjadi 8 bit subkunci yang akan digunakan sebagai proses enkripsi dan dekripsi. Subkunci yang diperoleh dinyatakan dalam bentuk cv_0, cv_1, cv_2, cv_3, cv_4, cv_5, cv_6, cv_7, cv_8, cv_9 yang disebut sebagai *cryptovariable*.

Berikut langkah-langkah dalam pengolahan kunci rahasia:

a. Kunci rahasia diubah dalam bentuk heksadesimal.

b. Kunci yang telah diubah kedalam bentuk heksadesimal dibagi menjadi 10 bagian dengan masing-masing subkunci berukuran 8 bit dengan ketentuan sebagai berikut:

cv_0 : bit 1 sampai bit 8; cv_1 : bit 9 sampai bit 16

cv_2 : bit 17 sampai bit 24; cv_3 : bit 25 sampai bit 32

cv_4 : bit 33 sampai bit 40; cv_5 : bit 41 sampai bit 48

cv_6 : bit 49 sampai bit 56; cv_7 : bit 57 sampai bit 64

cv_8 : bit 65 sampai bit 72; cv_9 : bit 73 sampai bit 80

Bit 1 merupakan bit yang diperoleh dari posisi bit paling tinggi/paling depan (MSB, *Most Significant Bit*). Adapun contoh pengolahan kunci pada algoritma skipjack adalah sebagai berikut:

Kunci rahasia = MATEMATIKA

Kunci diubah dalam bentuk heksadesimal = 4D4154454D4154494B41

Selanjutnya kunci yang diubah kedalam bentuk heksadesimal dibagi menjadi 10 bagian dengan masing-masing 8 bit:

cv_0 = 4D; cv_1 = 41; cv_2 = 54; cv_3 = 45; cv_4 = 4D

cv_5 = 41; cv_6 = 54; cv_7 = 49; cv_8 = 4B; cv_9 = 41

(Hartono, 2013).

2.5.2 Algoritma Permutasi

Permutasi pada algoritma skipjack merupakan bagian terpenting dalam proses enkripsi dan dekripsi. Permutasi pada proses enkripsi disebut sebagai permutasi G yang dilakukan pada *rule* A dan *rule* B sedangkan pada proses dekripsi dilakukan pada *rule*A^{-1} dan *rule* B^{-1} dandisebut sebagai permutasi G^{-1}. Algoritma permutasi memerlukan masukan nilai-nilai *cryptovariable* dan potongan plainteks. Potongan plainteks yang diambil adalah seperempat bagian dari blok plainteks dalam bentuk heksadesimal yang berukuran 16 bit. Berikut ini adalah langkah-langkah dari proses permutasi G dan permutasi G^{-1}:

a. Untuk permutasi G, G(*Word* = $g_1 \| g_2$) = $g_5 \| g_6$ yang mana g_1 merupakan byte pertama dari *Word* (*high byte*) dan g_2 merupakan merupakan byte kedua dari

Word (*low byte*). Hasil dari permutasi G adalah gabungan dari g_5 dan g_6. Rumus umum yang digunakan dalam permutasi G adalah sebagai berikut:

$$g_i = F(g_{i-1} \oplus cv_{4k+i-3}) \oplus g_{i-2}, \quad (2.3)$$

dimana $3 \leq i \leq 6$, dengan i awal = 3 dan $i \in N$; *k* merupakan nilai putaran enkripsi, untuk putaran enkripsi pertama k = 0. F merupakan fungsi substitusi pada *F-Table* sedangkan cv_{4k+i-3} merupakan *cryptovariable* dengan indeks $4k + i - 3$.

Berdasarkan rumus umum permutasi G maka diperoleh nilai g_3, g_4, g_5, dan g_6 sebagai berikut:

$g_3 = F(g_2 \oplus cv_{4k}) \oplus g_1$; $g_4 = F(g_3 \oplus cv_{4k+1}) \oplus g_2$

$g_5 = F(g_4 \oplus cv_{4k+2}) \oplus g_3$; $g_6 = F(g_5 \oplus cv_{4k+3}) \oplus g_4$

b. Untuk permutasi G^{-1}, G^{-1}(*Word* $= g_5 \parallel g_6$) $= g_1 \parallel g_2$ yang mana g_5 merupakan byte pertama dari *Word* (*high byte*) dan g_6 merupakan merupakan byte kedua dari *Word* (*low byte*). Hasil dari permutasi G^{-1} adalah gabungan dari g_1 dan g_2. Rumus umum yang digunakan dalam permutasi G^{-1} adalah sebagai berikut:

$$g_{i-2} = F(g_{i-1} \oplus cv_{4(k-1)+i-3}) \oplus g_i, \quad (2.4)$$

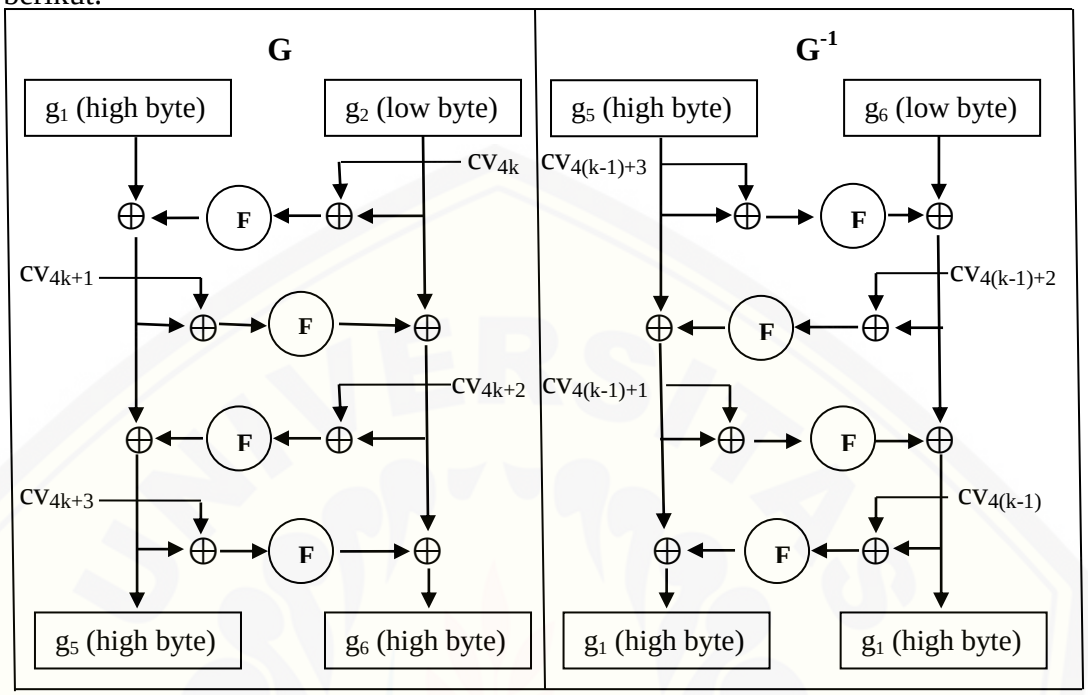
dimana $3 \leq i \leq 6$, dengan i awal = 6 dan $i \in N$; *k* merupakan nilai putaran enkripsi, untuk putaran enkripsi pertama *k* = 0. F merupakan fungsi substitusi pada *F-Table* sedangkan $cv_{4(k-1)+i-3}$ merupakan *cryptovariable* dengan indeks $4(k-1) + i - 3$.

Berdasarkan rumus umum permutasi G^{-1} maka diperoleh nilai g_1, g_2, g_3, dan g_4 sebagai berikut:

$g_1 = F(g_2 \oplus cv_{4(k-1)}) \oplus g_3$; $g_2 = F(g_3 \oplus cv_{4(k-1)+1}) \oplus g_4$

$g_3 = F(g_4 \oplus cv_{4(k-1)+2}) \oplus g_5$; $g_4 = F(g_5 \oplus cv_{4(k-1)+3}) \oplus g_6$

Adapun diagram dari algoritma permutasi G dan permtasi G^{-1} adalah sebagai berikut:



Gambar 2.4 Diagram permutasi G dan permtasi G^{-1}

(Hartono, 2013).

2.5.3 Enkripsi Algoritma Skipjack

Algoritma skipjack mengenkripsi plainteks berukuran 64 bit menjadi cipherteks berukuran 64 bit dengan proses enkripsi sebanyak 32 putaran. 8 putaran pertama menggunakan *rule A*, 8 putaran kedua menggunakan *rule B*, 8 putaran ketiga menggunakan *rule A* dan 8 putaran keempat menggunakan *rule B*. Karakter-karakter plainteks yang dienkripsi dikonversi kedalam bentuk heksadesimal sesuai dengan nilai-nilai yang terdapat dalam ASCII (*American Standard Code for Information Interchange*). Karakter-karakter plainteks yang dikonversi kedalam bentuk heksadesimal kemudian dibagi menjadi 4 bagian yang disebut sebagai *Word* yang dinyatakan sebagai W_1^0, W_2^0, W_3^0, W_4^0 yang mana setiap *Word* berukuran 16 bit. Cipherteks yang dihasilkan adalah W_1^{32}, W_2^{32}, W_3^{32}, W_4^{32}. Pada proses enkripsi terdapat beberapa ketetapan yaitupada putaran pertama $k = 0$ dan *counter* = 1 yang mana pada putaran selanjutnya k dan *counter* ditambah

satu. Adapun langkah-langkah enrkripsi algoritma skipjack adalah sebagai berikut:

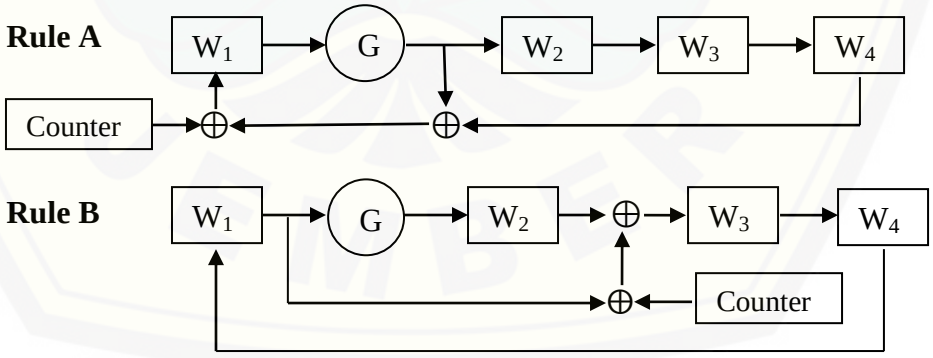
Langkah-langkah enkripsi pada *rule A*:

a. Mencari permutasi G dengan input W_1^k.
b. W_1^{k+1} merupakan hasil dari operasi XOR antara hasil permutasi G, W_4^k, dan *counter*.
c. W_2^{k+1} merupakan hasil dari permutasi G.
d. $W_3^{k+1} = W_2^k$.
e. $W_4^{k+1} = W_3^k$.
f. *counter* dan *k* ditambah satu.

Langkah-langkah enkripsi pada *rule B*:

a. Mencari permutasi G dengan input W_1^k.
b. $W_1^{k+1} = W_4^k$
c. W_2^{k+1} merupakan hasil dari permutasi G.
d. W_3^{k+1} merupakan hasil dari operasi XOR antara W_1^k, W_2^k dan *counter*.
e. $W_4^{k+1} = W_3^k$.
f. *counter* dan *k* ditambah satu.

Adapun diagram dari proses enkripsi *rule A* dan *rule B* adalah sebagai berikut:



Gambar 2.5 Diagram enkripsi *rule A* dan *rule B*

(Hartono, 2013).

2.5.4 Dekripsi Algoritma Skipjack

Proses dekripsi merupakan kebalikan dari proses enkripsi yang mana cipherteks akan diubah menjadi pesan aslinya yaitu plainteks. Cipherteks yang didekripsi diubah terlebih dahulu ke bentuk heksadesimal. Cipherteks yang telah dikonversi kebentuk heksadesimal selanjutnya dibagi menjadi 4 bagian yang disebut sebagai *Word* dan dinyakakan dalam bentuk W_1^{32}, W_2^{32}, W_3^{32}, W_4^{32} dengan masing-masing *Word* berukuran 16 bit. Proses dekripsi dilakukan sebanyak 32 putaran menggunakan dua buah *rule* yaitu *rule* A^{-1} dan *rule* B^{-1} dengan ketentuan dekripsi yaitu 8 putaran pertama menggunakan *rule* B^{-1}, *8* putaran kedua menggunakan *rule* A^{-1}, 8 putaran ketiga menggunakan *rule* B^{-1}, dan 8 putaran keempat menggunakan *rule* B^{-1}.Adapun ketetapan yang terdapat pada proses dekripsi yaitu k = 32 dan *counter* = 32. Plainteks yang didapatkan dari proses dekripsi adalah W_1^0, W_2^0, W_3^0, W_4^0.Adapun langkah-langkah dekripsi algoritma skipjack adalah sebagai berikut:

Langkah-langkah dekripsi pada *rule* A^{-1}:

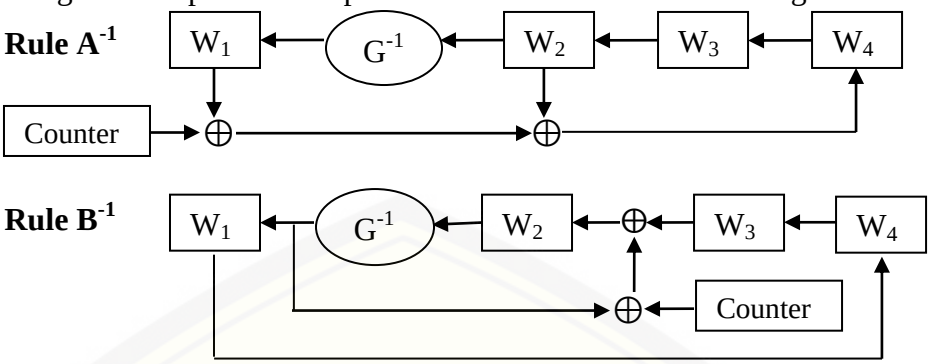
1. Mencari permutasi G^{-1} dengan input W_2^k.
2. W_1^{k-1} merupakan hasil daripermutasi G^{-1}.
3. $W_2^{k-1} = W_3^k$.
4. $W_3^{k-1} = W_4^k$.
5. W_4^{k-1} merupakan hasil dari operasi XOR antara W_1^k, W_2^k dan *counter*.
6. *counter* dan *k* dikurangi satu.

Langkah-langkah dekripsi pada *rule* B^{-1}:

1. Mencari permutasi G^{-1} dengan input W_2^k.
2. W_1^{k-1} merupakan hasil dari permutasi G^{-1}.
3. W_2^{k-1} merupakan hasildari operasi XOR antara hasil dari permutasi G^{-1}, W_3^k, dan *counter*.
4. $W_3^{k-1} = W_4^k$.
5. $W_4^{k-1} = W_1^k$.

g. *counter* dan *k* dikurangi satu.

Adapun diagram dari proses dekripsi *rule* A^{-1}dan *rule* B^{-1} adalah sebagai berikut:



Gambar 2.6 Diagram dekripsi *rule* A^{-1}dan *rule* B^{-1}

(Hartono, 2013).

2.6 Analisis Histogram

Analisis histogram merupakan analisis yang digunakan untuk memperkirakan keamanan dan ketahanan hasil enkripsi dari serangan-serangan kriptanalisis tipe statistik. Pada analisis histogram suatu citra yang terenkripsi harus memiliki penyebaran enkripsi pada nilai-nilai *pixel* di setiap saluran warna secara merata agar penyerang tidak dapat mengekstrak informasi statistik dari frekuensi nilai-nilai *pixel* di setiap saluran warna (Boriga *et al*., 2014).

2.7 Analisis Diferensial

Analisis diferensial digunakan untuk menguji pengaruh perubahan setiap *pixel* pada citra yang terenkripsi. Terdapat dua Indikator pengukuran yang umum digunakan pada analisis ini yaitu *Number of Pixels Change Rate* (NPCR) dan *Unified Average Changing Intensity* (UACI). *Number of Pixels Change Rate* (NPCR) merupakan persentase banyaknya *pixels* yang berubah pada citra asli ketika dienkripsi sedangkan *Unified Average Changing Intensity* (UACI) merupakan persentase perubahan warna terpadu pada citra asli ketika dienkripsi melalui selisih antara nilai-nilai *pixels* pada citra asli dengan citra hasil enkripsi. Adapun perhitungan NPCR didefinisikan sebagai berikut:

$$NPCR = \left(\frac{1}{m \times n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1} d_{i,j}\right) \times 100\ \% \qquad (2.5)$$

yang mana m dan $n$adalah lebar dan tinggi citra sedangkan $d_{i,j}$ ditentukan sebagai berikut:

$$d_{i,j} = \begin{cases} 0, & jika\ c_{i,j}^{(1)} = c_{i,j}^{(2)} \\ 1 & jika\ c_{i,j}^{(1)} \neq c_{i,j}^{(2)} \end{cases}$$

yang mana $c_{i,j}^{(1)}$ dan $c_{i,j}^{(2)}$ merupakan nilai derajat keabuan dari baris *i* dan kolom *j* dari citra $c^{(1)}$ dan citra $c^{(2)}$.

Sedangkan perhitungan UACI didefinisikan sebagai berikut:

$$UACI = \left(\frac{1}{m \times n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \frac{\left| c_{i,j}^{(1)} - c_{i,j}^{(2)} \right|}{255} \right) \times 100\% \qquad (2.6)$$

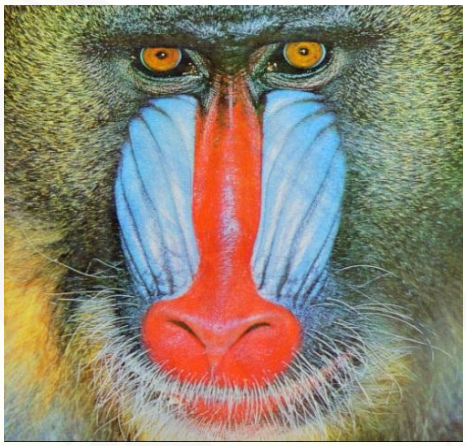
Secara teori, nilai minimum yang ideal pada indikator NPCR adalah sebesar 99,6094% dan pada indikator UACI adalah sebesar 33,4635% (Kwok *et al*., 2007) sedangkan menurut Boriga dkk., (2014) nilai pada indikator NPCR dapat dikatakan tahan terhadap serangan diferensial pada nilai minimal sebesar 98,87% dan pada indikator UACI sebesar minimal 32,17%.

BAB 3. METODE PENELITIAN

3.1 Data Penelitian

Data yang digunakan pada penelitian kali ini adalah citra RGB berdimensi 128 × 128 *pixels* yang disebut sebagai *plainimage*. Data yang akan diuji sebanyak 10 citra. Berikut adalah data-data yang akan digunakan pada penelitian:

Gambar 3.1 Citra Paprika



Gambar 3.2 Citra Babon

Gambar 3.3 Citra Lena

Gambar 3.4 Citra Anak Perempuan

(Sumber: www.http://informatika.stei.itb.ac.id/~rinaldi.munir/.../CitraUji.htm)

(Sumber: https://pemrogramanmatlab.files.wordpress.com/2016/09/lena.jpg)

Gambar 3.5 Citra Buah

Sumber: (http://www.sehatfresh.com/buah...remaja/)

Gambar 3.6 Citra Kampung Jodipan

Sumber: (sekilaskendari.blogspot.co.id/...html)

Gambar 3.7 Citra Pelangi

Sumber: (http://bobo.grid.id/Sains/Iptek/...Pelangi)

Gambar 3.8 Citra Bunga Tulip

Sumber: (https://ilmubudidaya.com/cara...tulip)

Gambar 3.9 Citra Pramuka

Gambar 3.10 Citra Kawah Ijen

3.2 Langkah Penelitian

Adapun langkah-langkah pada penelitian kali ini adalah sebagai berikut:

a. Studi Literatur

Pada tahap ini dilakukan pengumpulan literatur dan mempelajari teori-teori yang berkaitan dengan citra dan kriptografi khususnya tentang teori yang berkaitan dengan algoritma *Reversed Vigenere Encryption* dan Algoritma *Skipjack*.

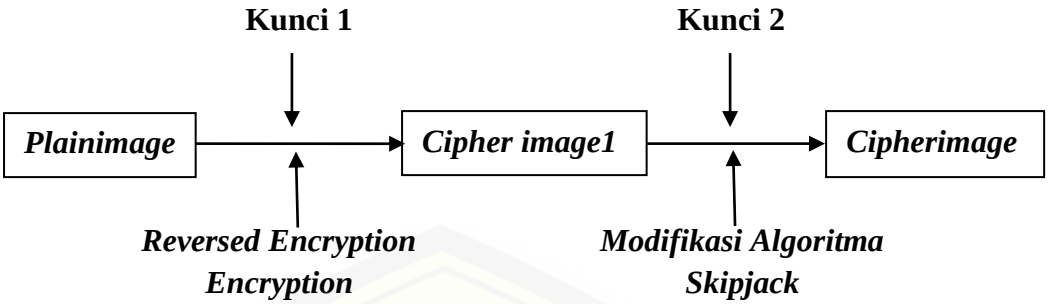
b. Penentuan Model Modifikasi Algoritma *Skipjack*

Pada tahap ini dilakukan penentuan model modifikasi algoritma *Skipjack* untuk digunakan pada proses enkripsi dan dekripsi citra RGB.

c. Percobaan Enkripsi dan Dekripsi Citra RGB Menggunakan Gabungan Algoritma *Reversed Vigenere Encryption* dan Modifikasi Algoritma *Skipjack*

Pada tahap ini percobaan dilakukan dengan mengenkripsi data penelitian berupa citra RGB menggunakan gabungan algoritma *Reversed Vigenere Encryption* dengan Modifikasi Algoritma *Skipjack* yang telah ditentukan. Hasil enkripsi kemudian didekripsi menggunakan metode yang diusulkan untuk memeriksa apakah proses enkripsi dan dekripsi pada citra RGB dapat dilakukan. Metode penggabungan algoritma *Reversed Vigenere Encryption* dengan modifikasi algoritma *Skipjack* menggunakan dua buah kunci yang disebut sebagai kunci 1 dan kunci 2. Kunci 1 digunakan pada algoritma *Reversed Vigenere Encryption* sedangkan kunci 2 digunakan pada modifikasi algoritma *Skipjack*. Adapun langkah-langkah enkripsi pada tahap ini adalah sebagai berikut:

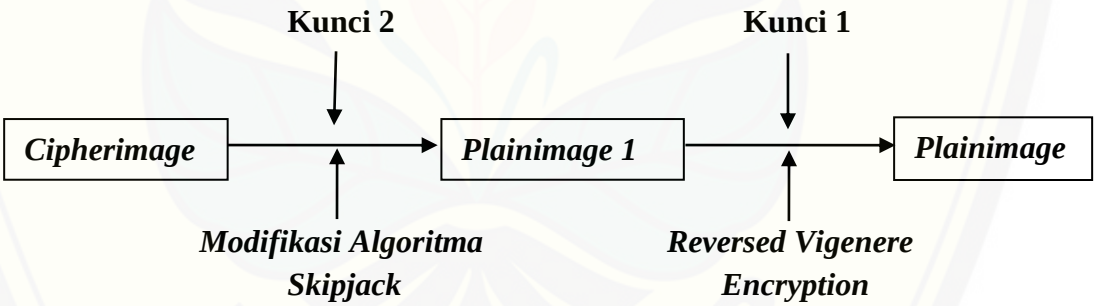
1. *Planimage* dienkripsi menggunakan Algoritma *Reversed Vigenere Encryption* menggunakan kunci 1 berupa karakter.
2. Hasil pada langkah pertama kemudian dienkripsi menggunakan modifikasi algoritma *Skipjack*. Adapun kunci yang digunakan pada langkah kedua menggunakan kunci 2 berupa karakter yang berukuran 10 karakter. Hasil keluaran pada tahap ini merupakan hasil akhir dari proses enkripsi yaitu berupa *cipherimage*.



Gambar 3.11 Proses enkripsi pada Gabungan algoritma *Reversed Vigenere Encryption* dengan Modifikasi Algoritma *Skipjack*

Adapun langkah-langkah untuk tahap dekripsi adalah sebagai berikut:

1. lakukan dekripsi pada *cipherimage 1* menggunakan modifikasi algoritma *Skipjack* dengan kunci 2 berupa karakter yang berukuran 10 karakter
2. hasil dekripsi pada langkah pertama kemudian didekripsi kembali menggunakan algoritma *Reversed Vigenere Encryption*. Adapun kunci yang digunakan pada langkah kedua menggunakan kunci 1 berupa karakter. Hasil keluaran pada tahap ini merupakan hasil akhir proses dekripsi yaitu pesan asli atau biasa disebut sebagai *plainimage*.



Gambar 3.12 Proses dekripsi pada Gabungan algoritma *Reversed Vigenere Encryption* dengan Modifikasi Algoritma *Skipjack*

d. Pembuatan Program Aplikasi Enkripsi dan Dekripsi Citra RGB

langkah selanjutnya adalah pembuatan program aplikasi enkripsi dan dekripsi citra menggunakan MATLAB R2015b sesuai dengan algoritma yang telah dibuat.

e. Uji Coba Program Aplikasi Enkripsi dan Dekripsi Citra RGB

Uji coba program dilakukan dengan menguji citra RGB pada penelitian untuk proses enkripsi dan dekripsi menggunakan program yang telah dibuat pada

aplikasi MATLAB R2015b agar dapat dilakukan analisis hasil enkripsi dan dekripsi.

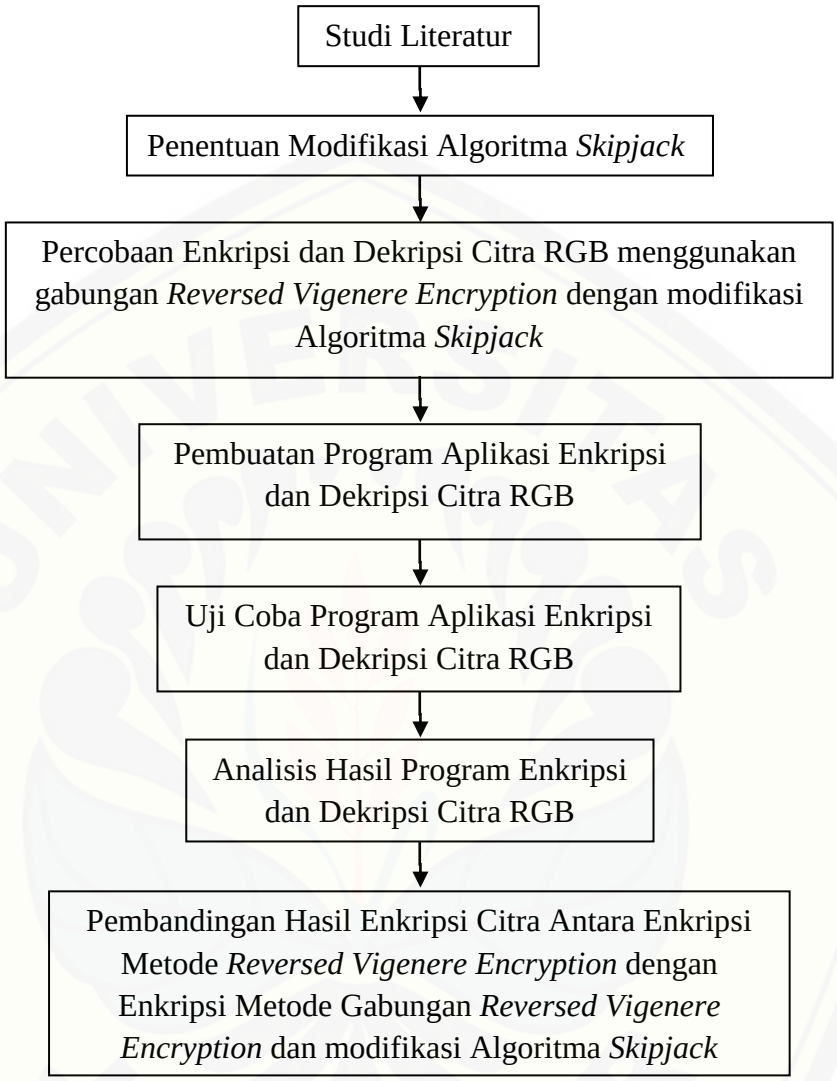
f. Analisis Hasil Program Enkripsi dan Dekripsi Citra RGB

Pada tahap ini hasil akhir dari proses enkripsi dan dekripsi citra pada program simulasi MATLAB R2015b akan dianalisis menggunakan analisis histogram dan analisis diferensial.

g. Pembandingan Hasil Enkripsi Citra Antara Enkripsi Metode *Reversed Vigenere Encryption* dengan Enkripsi Metode Gabungan *Reversed Vigenere Encryption* dan modifikasi algoritma *Skipjack*

Pada tahap ini hasil enkripsi citra menggunakan metode yang diajukan akan dibandingkan dengan hasil enkripsi citra menggunakan metode algoritma *Reversed Vigenere Encryption.* Pembandingan dilakukan untuk mengetahui apakah hasil enkripsi citra menggunakan metode yang diusulkan akan menghasilkan *cipherimage* yang lebih baik atau lebih buruk dari hasil enkripsi citra sebelum dilakukan penggabungan menggunakan modifikasi algoritma *Skipjack.*

Berikut adalah skema langkah-langkah pada penelitian:



(Gambar 3.13 Skema langkah-langkah penelitian)

BAB 5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat diambil beberapa kesimpulan sebagai berikut:

a. Proses Enkripsi *plainimage* menggunakan gabungan algoritma *Reversed Vigenere Encryption* dengan modifikasi algoritma *Skipjack* menghasilkan *cipherimage* yang berbeda secara signifikan terhadap citra aslinya.
b. Proses Dekripsi *cipherimage* menggunakan gabungan algoritma *Reversed Vigenere Encryption* dengan modifikasi algoritma *Skipjack* dapat dilakukan dengan baik tanpa mengubah nilai-nilai *pixels* dari citra aslinya.
c. *Chiperimage* yang dihasilkan pada metode gabungan algoritma *Reversed Vigenere Encryption* dengan modifikasi algoritma *Skipjack* dapat tahan dan kuat terhadap serangan-serangan kriptoanalisis tipe statistik. Hal ini didasarkan pada hasil analisis histogram pada *cipherimage* yang memiliki penyebaran nilai-nilai *pixels* yang tersebar secara merata. Metode gabungan algoritma *Reversed Vigenere Encryption* dengan modifikasi algoritma *Skipjack* juga dapat tahan terhadap serangan diferensial. Hal ini didadasarkan pada nilai NPCR yang diperoleh telah memenuhi batas nilai minimal NPCR.
d. Berdasarkan pembahasan yang telah disampaikan pada Sub subab 4.2.5, hasil enkripsi pada metode gabungan algoritma *Reversed Vigenere Encryption* dengan modifikasi algoritma *Skipjack* menghasilkan *cipherimage* yang lebih baik dan unggul daripada hasil enkripsi pada metode algoritma *Reversed Vigenere Encryption* jika ditinjau dari hasil enkripsi dan hasil analisis histogram sedangkan pada hasil analisis diferensial hasil enkripsi pada metode algoritma *Reversed Vigenere Encryption* memiliki nilai NPCR dan UACI yang lebih besar daripada nilai NPCR dan UACI pada metode gabungan yang diusulkan.

5.2 Saran

Adapun saran yang dapat diberikan untuk penelitian selanjutnya adalah:

a. Menerapkan kunci pada algoritma *Reversed Vigenere Encryption* berupa citra agar penyerang sulit menduga kunci yang digunakan untuk proses enkripsi dan dekripsi.
b. Menyelidiki dan mementukan bagaimana modifikasi algoritma *Skipjack* yang tepat agar nilai UACI yang dihasilkan pada metode gabungan dapat mencapai diatas batas nilai minimal UACI.

DAFTAR PUSTAKA

Ariyus, D. 2006. *Kriptografi Keamanan Data dan Komunikasi*. Yogyakarta: Graha Ilmu.

Ariyus, D. 2008. *Pengantar Ilmu Kriptografi*. Yogyakarta: c.v Andi Offset.

Boriga, R.E., A.C. Dăscălescu dan A.V. Diaconu. 2014. A New Fast Image Encryption Scheme Based on 2D Chaotic Maps. https://pdfs.semanticscholar.org/3c7e/a5908fe266ef743260fcd3bb98992238a6fc.pdf [Diakses pada 12 Maret 2018].

Hallim, A., I. U. Nadhori dan Setiawardhana. 2010. Pembuatan Perangkat Lunak Media Pembelajaran Kriptografi Klasik. http://repo.pens.ac.id/444/1/873.pdf [Diakses pada 9 April 2018].

Hartono. 2013. Aplikasi Pengamanan Data Menggunakan Metode Skipjack. http://dosen.publikasistmikibbi.lppm.org/permalink/000099.pdf [Diakses pada 9 April 2018].

Kwok, H.S. dan W.K.S. Tang. A Fast Image Encryption System Based on Chaotics Maps with Finite Precision Representation. https://pdfs.semanticscholar.org/1c97/58e931426b892d75cf640917d88aa87d05d4.pdf [Diakses pada 24 Maret 2018].

Munir, R. 2002. *Diktat Kuliah Pengolahan Citra*. Bandung: Departemen Teknik Informatika ITB.

Munir, R. 2017. *Kriptanalisis Bahan Kuliah IF4020 Kriptografi*. Bandung: Departemen Teknik Informatika ITB.

Sadikin, R. 2012. *Kriptografi Untuk Keamanan Jaringan*. Yogyakarta: C.V Andi Offset.

Sengupta, N dan J. Holmes. 2013. Designing of Cryptography Based Security System for Cloud Computing. *International Conference on Cloud & Ubiquitous Computing & Emerging Technologies*. 20: 53.

Yohandri, 2013. *Elektronika Digital*. Padang: Universitas Negeri Padang.

LAMPIRAN

LAMPIRAN A. Skrip Program Enkripsi dan Dekripsi pada Aplikasi MATLAB R2015b

a. Skrip program pada pembentukan kunci algoritma *reversed vigenere encryption*

```
key=get(handles.edit2,'string');
if isempty(key)
    errordlg('Maaf, Kunci Belum Diisi');
else
keydouble=double(key);
keyrev=keydouble(end:-1:1);
if length(keydouble)>m*n
    kunci=keyrev(1:m*n);
else
    kunci=[];
    for a=1:floor(m*n/length(keyrev))
        kunci=[kunci keyrev];
    end
    kunci=[kunci keyrev(1:mod(m*n,length(keyrev)))];
end
matkey=kunci(1:n);
for i=2:m
    matkey=[matkey;kunci((i-1)*n+(1:n))];
end
```

b. Skrip program pada enkripsi algoritma *reversed vigenere encryption*

```
for i=1:3
   imchip(:,:,i)=uint8(mod(double(implain(:,:,i))+matkey,256));
end
```

c. Skrip program pada *F-Table*

```
TableF={'F6','40','C6','29','EE','08','42','06','2C','31','69','43','7F','95','
84','6A'

'7E','B0','D1','4D','BA','E3','FF','48','82','78','C1','4C','D7','86','36','AF'

'DC','35','34','DB','B3','C0','72','19','85','B9','17','F1','4A','C2','50','92'

'5A','CD','18','D5','A3','6F','76','93','80','2A','CE','E7','DE','EF','ED','C4'

'20','61','91','0E','49','DA','C5','15','60','A0','7A','AD','5B','24','F9','FD'

'03','54','81','B8','BD','2E','10','12','7C','75','71','1A','F7','88','3A','6B'

'1F','83','14','F4','E0','77','5E','32','01','E2','28','4F','9A','E9','AB','22'

'66','9C','BF','8A','5C','D2','90','39','CC','D9','56','7D','33','BE','0B','55'

'A1','99','F5','F8','D6','D3','AC','A6','47','65','2B','AE','1C','21','97','46'

'13','F0','45','B5','16','2D','C9','3B','E5','9D','C8','A7','FC','26','0D','74'

'89','8E','57','05','B4','E4','70','B2','BB','6D','0C','B1','51','64','B7','79'
```

```
'4E','D0','C3','FA','09','8F','2F','BC','07','44','EC','27','CF','C7','CA','8C'

'52','9B','7B','A4','67','58','D4','04','68','0F','D8','E8','AA','3C','3F','3D'

'6C','53','98','1B','00','CB','A2','EB','9F','25','A8','F2','B6','23','3E','1D'

'02','A5','73','0A','96','30','8D','9E','5F','37','87','E6','F3','FB','41','11'

'DD','FE','94','59','63','8B','DF','E1','38','EA','1E','6E','A9','4B','62','5D'
};
```

d. Skrip program pada pembacaan *F-Table*

```
function permut=FTable(hex,TableF)
hexa='0123456789ABCDEF';
for i=1:size(hex,1);
    p1=find(hexa==hex(i,1));
    p2=find(hexa==hex(i,2));
    permut(i,:)=TableF{p1,p2};
end
```

e. Skrip program pada enkripsi algoritma *skipjack*

```
for ii=1:o
   for i=1:m
      for j=1:n/8
        imchip(i,(j-1)*8+(1:8),ii)= uint8(Permutasi(double(
        imchip(i,(j-1)*8+(1:8),ii)),Key,8,TableF));
      end
   end
end
```

f. Skrip program pada enkripsi *skipjack* menggunakan *rule A* dan *rule B*

```
function Cipher=Permutasi(Plain,Key,Putaran,TableF)
Plainhex=dec2hex(Plain);
Keyhex=dec2hex(double(Key));
W=Plainhex;
for put=1:Putaran
    if (put<=Putaran/2) %Rule A
        g(1,:)=W(1,:);
        g(2,:)=W(2,:);
        for k=3:6
g(k,:)=binary2hex(xor2(hex2binary(FTable(binary2hex(xor2(
hex2binary(g(k-1,:)),hex2binary(Keyhex(mod(4*(put-1)+k-
3,10)+1,:)))),TableF)),hex2binary(g(k-2,:))));
        end
        temp1=[g(5,:);g(6,:)];
        temp2=W(7:8,:);
        W(7:8,:)=W(5:6,:);
        W(5:6,:)=W(3:4,:);
        W(3:4,:)=[g(5,:);g(6,:)];
W(1,:)=binary2hex(xor2(xor2(hex2binary(temp1(1,:)),hex2binary(
temp2(1,:))),'00000000'));
W(2,:)=binary2hex(xor2(xor2(hex2binary(temp1(2,:)),hex2binary(
temp2(2,:))),hex2binary(dec2hex(put))));
    else  %Rule B
        g(1,:)=W(1,:);
        g(2,:)=W(2,:);
```

```
        for k=3:6
g(k,:)=binary2hex(xor2(hex2binary(FTable(binary2hex(xor2(
hex2binary(g(k-1,:)),hex2binary(Keyhex(mod(4*(put-1)+k-
3,10)+1,:)))),TableF)),hex2binary(g(k-2,:))));
        end
        temp1=W(1:2,:);
        temp2=W(3:4,:);
        temp3=W(7:8,:);
        W(7:8,:)=W(5:6,:);
        W(1:2,:)=temp3;
        W(3:4,:)=[g(5,:);g(6,:)];
W(5,:)=binary2hex(xor2(xor2(hex2binary(temp1(1,:)),hex2binary(
temp2(1,:))),'00000000'));
W(6,:)=binary2hex(xor2(xor2(hex2binary(temp1(2,:)),hex2binary(
temp2(2,:))),hex2binary(dec2hex(put))));
    end
end
Cipher=hex2dec(W)';
```

g. Skrip program pada dekripsi algoritma *reversed vigenere encryption*

```
for i=1:3
   dechip(:,:,i)=uint8(mod(double(dechip(:,:,i))-matkey,256));
end
```

h. Skrip program pada dekripsi algoritma *skipjack*

```
for ii=1:o
   for i=1:m
      for j=1:n/8
         dechip(i,(j-1)*8+(1:8),ii)=uint8(Permutasi2(double
         (imchip(i,(j-1)*8+(1:8),ii)),Key,8,TableF));
      end
    end
end
```

i. Skrip program pada dekripsi *skipjack* menggunakan *rule* A^{-1} dan *rule* B^{-1}

```
function Plain=Permutasi2(Cipher,Key,Putaran,TableF)
Cipherhex=dec2hex(Cipher);
Keyhex=dec2hex(double(Key));
W=Cipherhex;
for put=1:Putaran
    if put>Putaran/2 %Rule A1
        g(2,:)=W(3,:);
        g(1,:)=W(4,:);
        for k=3:6
g(k,:)=binary2hex(xor2(hex2binary(FTable(binary2hex(xor2(
hex2binary(g(k-1,:)),hex2binary(Keyhex(mod(4*(Putaran-put)+6-
k,10)+1,:)))),TableF)),hex2binary(g(k-2,:))));
        end
        temp1=W(1:2,:);
        temp2=W(3:4,:);
        W(3:4,:)=W(5:6,:);
        W(5:6,:)=W(7:8,:);
```

```
        W(1:2,:)=[g(6,:);g(5,:)];
W(7,:)=binary2hex(xor2(xor2(hex2binary(temp1(1,:)),hex2binary(
temp2(1,:))),'00000000'));
W(8,:)=binary2hex(xor2(xor2(hex2binary(temp1(2,:)),hex2binary(
temp2(2,:))),hex2binary(dec2hex(Putaran-put+1))));
    else %Rule B1
        g(2,:)=W(3,:);
        g(1,:)=W(4,:);
        for k=3:6
g(k,:)=binary2hex(xor2(hex2binary(FTable(binary2hex(xor2(
hex2binary(g(k-1,:)),hex2binary(Keyhex(mod(4*(Putaran-put)+6-
k,10)+1,:)))),TableF)),hex2binary(g(k-2,:))));
        end
        temp1=W(5:6,:);
        W(5:6,:)=W(7:8,:);
        W(7:8,:)=W(1:2,:);
        W(1:2,:)=[g(6,:);g(5,:)];
W(3,:)=binary2hex(xor2(xor2(hex2binary(g(6,:)),hex2binary(temp1
(1,:))),'00000000'));
W(4,:)=binary2hex(xor2(xor2(hex2binary(g(5,:)),hex2binary(temp1
(2,:))),hex2binary(dec2hex(Putaran-put+1))));
    end
end
Plain=hex2dec(W)';
```

LAMPIRAN B. Skrip Program Analisis Histogram dan Analisis Diferensial pada Aplikasi MATLAB R2015b

a. Skrip program pada analisis histogram dan analisis diferensial

```
radio1=get(handles.radiobutton1,'value');
radio2=get(handles.radiobutton2,'value');
radio3=get(handles.radiobutton5,'value');
if radio1==1
    implain=get(handles.axes1,'UserData');
    [m,n,o]=size(implain);
    if mod(n,8)~=0
        implain(:,:,1)=[implain(:,:,1) ones(m,8-mod(n,8))*255];
        implain(:,:,2)=[implain(:,:,2) ones(m,8-mod(n,8))*255];
        implain(:,:,3)=[implain(:,:,3) ones(m,8-mod(n,8))*255];
    end
    imchip=get(handles.axes2,'UserData');
    [m,n,o]=size(imchip);
    axes(handles.axes4);
    imhist(implain(:,:,1));
    hold on
    imhist(implain(:,:,2));
    imhist(implain(:,:,3));
    hold off
    axes(handles.axes5);
    imhist(imchip(:,:,1));
    hold on
    imhist(imchip(:,:,2));
    imhist(imchip(:,:,3));
    hold off
    dij=double(implain)-double(imchip);
    dij(dij~=0)=1;
    npcr=sum(sum(sum(dij)))/(m*n*o)*100;
    set(handles.text12,'string',num2str(npcr))
    cij=abs(double(implain)-double(imchip));
    uaci=sum(sum(sum(cij/255)))/(m*n*o)*100;
    set(handles.text15,'string',num2str(uaci));
elseif radio2==1
    imchip=get(handles.axes1,'UserData');
    dechip=get(handles.axes3,'UserData');
    [m,n,o]=size(imchip);
    axes(handles.axes4);
    imhist(imchip(:,:,1));
    hold on
    imhist(imchip(:,:,2));
    imhist(imchip(:,:,3));
    hold off
    axes(handles.axes6);
    imhist(dechip(:,:,1));
    hold on
    imhist(dechip(:,:,2));
    imhist(dechip(:,:,3));
    hold off
    dij=double(imchip)-double(dechip);
    dij(dij~=0)=1;
```

```
    npcr=sum(sum(sum(dij)))/(m*n*o)*100;
    set(handles.text24,'string',num2str(npcr))
    cij=abs(double(imchip)-double(dechip));
    uaci=sum(sum(sum(cij/255)))/(m*n*o)*100;
    set(handles.text25,'string',num2str(uaci));
elseif radio3==1
    implain=get(handles.axes1,'UserData');
    [m,n,o]=size(implain);
    if mod(n,8)~=0
        implain(:,:,1)=[implain(:,:,1) ones(m,8-mod(n,8))*255];
        implain(:,:,2)=[implain(:,:,2) ones(m,8-mod(n,8))*255];
        implain(:,:,3)=[implain(:,:,3) ones(m,8-mod(n,8))*255];
    end
    imchip=get(handles.axes2,'UserData');
    dechip=get(handles.axes3,'UserData');
    [m,n,o]=size(imchip);
    axes(handles.axes4);
    imhist(implain(:,:,1));
    hold on
    imhist(implain(:,:,2));
    imhist(implain(:,:,3));
    hold off
    axes(handles.axes5);
    imhist(imchip(:,:,1));
    hold on
    imhist(imchip(:,:,2));
    imhist(imchip(:,:,3));
    hold off
    axes(handles.axes6);
    imhist(dechip(:,:,1));
    hold on
    imhist(dechip(:,:,2));
    imhist(dechip(:,:,3));
    hold off
    %plainimage ke cipherimage
    dij=double(implain)-double(imchip);
    dij(dij~=0)=1;
    npcr=sum(sum(sum(dij)))/(m*n*o)*100;
    set(handles.text12,'string',num2str(npcr))
    cij=abs(double(implain)-double(imchip));
    uaci=sum(sum(sum(cij/255)))/(m*n*o)*100;
    set(handles.text15,'string',num2str(uaci));
    %cipherimage ke plainimage
    dij=double(imchip)-double(dechip);
    dij(dij~=0)=1;
    npcr=sum(sum(sum(dij)))/(m*n*o)*100;
    set(handles.text24,'string',num2str(npcr))
    cij=abs(double(imchip)-double(dechip));
    uaci=sum(sum(sum(cij/255)))/(m*n*o)*100;
    set(handles.text25,'string',num2str(uaci));
    %plainimage ke hasil dekripsi
    dij=double(implain)-double(dechip);
    dij(dij~=0)=1;
    npcr=sum(sum(sum(dij)))/(m*n*o)*100;
    set(handles.text30,'string',num2str(npcr))
    cij=abs(double(implain)-double(dechip));
```

```
    uaci=sum(sum(sum(cij/255)))/(m*n*o)*100;
    set(handles.text31,'string',num2str(uaci));
end
```

LAMPIRAN C. Nilai NPCR dan UACI dari *Plainimage* ke Hasil Dekripsi

No	Data Penelitian	Nilai NPCR	Nilai UACI
1	Citra Paprika	0 %	0 %
2	Citra Babon	0 %	0 %
3	Citra Lena	0 %	0 %
4	Citra Anak Perempuan	0 %	0 %
5	Citra Buah	0 %	0 %
6	Citra Kampung Jodipan	0 %	0 %
7	Citra Pelangi	0 %	0 %
8	Citra Bunga Tulip	0 %	0 %
9	Citra Pramuka	0 %	0 %
10	Citra Kawah Ijen	0 %	0 %