IMPLEMENTASI *VIGENERE CIPHER* PADA PENYANDIAN CITRA BERBASIS PEMBANGKITAN KUNCI ALGORITMA *ADVANCED ENCRYPTION STANDARD* (AES)

SKRIPSI

Oleh
Eriska Yuniar Putri
NIM 151810101047

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2018

IMPLEMENTASI *VIGENERE CIPHER* PADA PENYANDIAN CITRA BERBASIS PEMBANGKITAN KUNCI ALGORITMA *ADVANCED ENCRYPTION STANDARD* (AES)

SKRIPSI

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

Oleh
Eriska Yuniar Putri
NIM 151810101047

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2018

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Kedua Orang tuaku tersayang, Ayahanda Kartono dan Ibunda Hertris Setyo Indah, serta adikku Fahreza Hertanto Putra yang senantiasa memberi dukungan dan do'a;
2. Tunanganku Ahmad Hoirul Anwar yang senantiasa memberikan motivasi dan do'a;
3. Guru-guru dan dosen dari sekolah dasar hingga perguruan tinggi;
4. Almamater Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

MOTO

Seseorang yang bertindak tanpa ilmu ibarat bepergian tanpa petunjuk. Dan sudah banyak yang tahu kalau orang seperti itu sekiranya akan hancur, bukan selamat.
(Hasan Al Basri)

Basri, H. A. 2017 dalam Andhini, Z. A. E. 2018. Analisis Motivasi Orang Tua Menyekolahkan Anak Ke Perguruan Tinggi Di Desa Brangkal Kecamatan Karanganom Kabupaten Klaten. *Skripsi*. Surakarta: Program Studi Pendidikan Akuntansi Fakultas Keguruan Dan Ilmu Pendidikan, Universitas Muhammadiyah Surakarta.

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

nama : Eriska Yuniar Putri

NIM : 151810101047

menyatakan dengan sesungguhnya bahwa skripsi yang berjudul "Implementasi *Vigenere Cipher* Pada Penyandian Citra Berbasis Pembangkitan Kunci Algoritma *Advanced Encryption Standard* (AES)" 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, 27 Desember 2018

Yang menyatakan,

Eriska Yuniar Putri

NIM 151810101047

SKRIPSI

IMPLEMENTASI *VIGENERE CIPHER* PADA PENYANDIAN CITRA BERBASIS PEMBANGKITAN KUNCI ALGORITMA *ADVANCED ENCRYPTION STANDARD* (AES)

Oleh:

Eriska Yuniar Putri

NIM. 151810101047

Pembimbing

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

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

PENGESAHAN

Skripsi berjudul "Implementasi *Vigenere Cipher* Pada Penyandian Citra Berbasis Pembangkitan Kunci Algoritma *Advanced Encryption Standard* (AES)" telah diuji dan disahkan pada:

hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Jember

Tim Penguji:

Ketua,	Anggota I,
Abduh Riski, S.Si., M.Si.	Ahmad Kamsyakawuni, S.Si., M.Kom.
NIP 199004062015041001	NIP 197211291998021001
Anggota II,	Anggota III,
Kiswara Agung Santoso, S.Si., M.Kom.	Kusbudiono, S.Si., M.Si.
NIP 197209071998031003	NIP 197704302005011001

Mengesahkan
Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Jember

Drs. Sujito., Ph.D.
NIP 196102041987111001

RINGKASAN

Implementasi *Vigenere Cipher* Pada Penyandian Citra Berbasis Pembangkitan Kunci Algoritma *Advanced Encryption Standard* (AES); Eriska Yuniar Putri, 151810101047; 2018: 64 Halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Perkembangan teknologi komputer yang semakin canggih maka diperlukan teknik yang aman untuk mengirimkan suatu data maupun informasi. Salah satu cara untuk menjaga keamanan dan kerahasiaan suatu data maupun informasi adalah dengan teknik enkripsi dan dekripsi. Teknik enkripsi dan dekripsi dikenal dan dipelajari dalam bidang ilmu kriptografi. Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan.

Pada penelitian ini bertujuan untuk meningkatkan keamanan pada penyandian pesan maka penelitian ini membahas tentang *Vigenere Cipher* untuk menyandikan suatu citra pada pembangkitan kuncinya dengan mengadopsi dari konsep algoritma AES yaitu dengan menambahan perilaku *Shift Rows*, *Shift Column*, dan operator XOR pada proses penyandian citra menggunakan *Vigenere Cipher*. Proses enkripsi menggunakan *Vigenere Cipher* hanya terdiri dari satu putaran, sehingga hasil dari enkripsi masih terlihat polanya dan mudah untuk diduga citra aslinya. Proses enkripsi citra menggunakan *Vigenere*-AES terdiri dari 10 putaran menghasilkan citra yang terlihat acak (tidak berpola). Proses dekripsi menggunakan *Vigenere Cipher* dan proses dekripsi citra menggunakan *Vigenere*-AES juga berhasil mengembalikan *chiper image* menjadi *plain image* awal.

Berdasarkan data penelitian yang terdiri dari 8 citra yang telah diuji, hasil dari uji histogram menghasilkan histogram yang lebih seragam menggunakan *Vigenere*-AES dibandingkan hasil histogram yang hanya menggunakan *Vigenere Cipher*, terlihat juga dari perhitungan X^2 bahwa hasil yang peroleh *Vigenere*-AES lebih kecil dibandingkan perhitungan yang dihasilkan menggunakan *Vigenere Cipher*. Hasil nilai NPCR yang diperoleh menggunakan *Vigenere Cipher* adalah sebesar 100% dan nilai UACI yang diperoleh sebesar 42,2972% hingga 48,5928%.

Sedangkan nilai NPCR yang diperoleh menggunakan *Vigenere*-AES adalah sebesar 99,6023% hingga 99,6189% dan nilai UACI yang diperoleh adalah sebesar 28,6491% hingga 35,1975%. Berdasarkan hasil yang telah diperoleh bahwa hasil dari nilai NPCR dan UACI *Vigenere Cipher* dapat dikatakan lebih baik dibandingkan dengan hasil nilai NPCR dan UACI *Vigenere*-AES. Secara numerik, *Vigenere Cipher* lebih baik dari *Vigenere*-AES tetapi secara visual hasil dari enkripsi *Vigenere*-AES lebih baik dari *Vigenere Cipher* karena citra yang dihasilkan oleh *Vigenere*-AES terlihat acak (tidak berpola). Tidak berkorelasinya antara uji numerik dengan tampilan visual karena uji diferensial kurang baik pada kasus kali ini. Hasil nilai koefisien korelasi menggunakan *Vigenere*-AES mendekati nol dan lebih kecil dibandingkan dengan nilai koefisien korelasi menggunakan *Vigenere Cipher*, itu artinya *Vigenere*-AES lebih kuat terhadapat serangan statistik dibandingkan dengan *Vigenere Cipher*.

Berdasarkan perbandingan antara hasil perhitungan dari histogram, NPCR, UACI, dan koefisien korelasi. Tingkat keamanan hasil penyandian citra menggunakan *Vigenere Cipher* berbasis pembangkitan kunci Algoritma AES menghasilkan nilai yang lebih mendekati batas indikator aman, sehingga dapat disimpulkan bahwa penyandian citra menggunakan *Vigenere Cipher* berbasis pembangkitan kunci Algoritma AES lebih kuat dibandingkan dengan hasil penyandian citra menggunakan *Vigenere Cipher*.

PRAKATA

Puji syukur ke hadirat Allah Swt. atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "Implementasi *Vigenere Cipher* Pada Penyandian Citra Berbasis Pembangkitan Kunci Algoritma *Advanced Encryption Standard* (AES)". 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. Kiswara Agung Santoso, S.Si., M.Kom., dan Kusbudiono, S.Si., M.Si., selaku Dosen Penguji yang telah memberikan kritik dan saran yang membangun dalam penyempurnaan skripsi ini;
3. Kusbudiono, S.Si., M.Si., selaku selaku Dosen Pembimbing Akademik yang telah membimbing dalam pemilihan matakuliah;
4. Ayahanda Kartono dan Ibunda Hertris Setyo Indah serta adikku Fahreza Hertanto Putra yang telah memberikan dukungan dan doa;
5. Tunanganku Ahmad Hoirul Anwar yang telah memberikan motivasi dan do'a;
6. Seluruh teman-teman "SIGMA" 2015 dan teman-teman "Istri Sholeha" yang telah memberikan motivasi serta dukungannya;
7. Sahabat-sahabati UKM SPORA FMIPA Universitas Jember yang telah memberikan motivasi dan pengalamannya;
8. 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, Desember 2018

Penulis

DAFTAR ISI

DAFTAR TABEL

Halaman

DAFTAR GAMBAR

DAFTAR LAMPIRAN

Halaman

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Seiring dengan pesatnya perkembangan teknologi informasi dan komunikasi yang terjadi saat ini tentu diperlukan teknik yang aman untuk mengirimkan suatu pesan. Salah satu cara untuk menjaga keamanan dan kerahasiaan suatu data maupun informasi adalah dengan teknik enkripsi dan dekripsi. Teknik ini berguna untuk membuat pesan, data, maupun informasi tidak dapat dibaca atau dimengerti oleh orang lain, kecuali untuk penerima yang berhak dan mengetahui teknik dekripsinya. Teknik enkripsi dan dekripsi dikenal dan dipelajari dalam bidang ilmu kriptografi.

Kriptografi berasal dari bahasa Yunani, *crypto* dan *graphia*. *Chypto* berarti rahasia (*secret*) dan *graphia* berarti tulisan (*writing*). Menurut terminologinya, kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat lain (Ariyus, 2008). Kriptografi mendukung kebutuhan dari dua aspek keamanan informasi, yaitu perlindungan terhadap kerahasiaan data informasi (*secrecy*) dan perlindungan terhadap pemalsuan dan pengubahan informasi yang tidak diinginkan (*authenticity*) (Bhaudhayana dan Widiartha, 2015).

Menurut era kemunculannya, kriptografi dapat diklasifikasikan menjadi dua macam yaitu kriptografi klasik dan kriptografi modern. Salah satu kriptografi klasik adalah *Vigenere Cipher*. *Vigenere Cipher* menggunakan tabel huruf yang digunakan untuk melakukan enkripsi dan dekripsi. *Vigenere Cipher* biasanya digunakan untuk mengenkripsi sebuah pesan yang berupa teks dan masih menggunakan metode substitusi, sehingga penyandian data yang dikirimkan lebih mudah untuk dipecahkan oleh orang-orang yang tidak berhak menerima pesan tersebut. Metode *Vigenere Cipher* telah dapat dilakukan kriptanalisis terhadap panjang kunci dengan metode Kasiski. Kelamahan *Vigenere* dipecahkan dengan memanfaatkan *Vigenere* menggunakan kunci yang sama berulang-ulang sehingga menghasilkan potongan *ciphertext* yang sama untuk *plaintext* yang sama (Pradipta, 2016).

Perkembangan teknologi komputer yang semakin canggih, banyak metode-metode yang diperluas penggunaannya, seperti halnya kriptografi klasik yang salah satunya masih menggunakan metode substitusi saat ini diperluas dengan menggunakan operasi matematika yang lebih aman dalam penyandian data, sehingga terciptalah kriptografi modern. Salah satu kriptografi modern yaitu algoritma *Advanced Encryption Standard* (AES) merupakan algoritma kriptografi yang cukup aman untuk melindungi data atau informasi yang bersifat rahasia. Algoritma AES menggunakan panjang kunci yang bervariasi, yaitu 128 bit, 192 bit, dan 256 bit. Perbedaan panjang kunci ini yang nantinya mempengaruhi jumlah putaran pada algoritma AES (Bhaudhayana dan Widiartha, 2015).

Beberapa penelitian sebelumnya yang berkaitan yaitu Luthfi (2011) melakukan penelitian yang berjudul Enkripsi Citra Bitmap Melalui Substitusi Warna Menggunakan *Vigenere Cipher*. Pada penelitian tersebut masih mengenkripsi sebuah teks yang disisipkan terhadap citra digital, sehingga citra yang telah dienkripsi masih dapat terlihat polanya. Hanifah (2012) melakukan penelitian dengan judul Aplikasi Algoritma Rijndael Dalam Pengamanan Citra Digital. Penelitian tersebut menggunakan Algoritma Rijndael dalam penyandian citra *grayscale* saja dan hanya menggunakan citra yang berukuran 128 bit. Saputra (2014) melakukan penelitian dengan judul Aplikasi Kriptografi Suara Menggunakan Algoritma *Advanced Encryption Standard* (AES). Pada penelitian tersebut membahas tentang bagaimana teknik kriptografi suara diimplementasikan menggunakan algoritma AES.

Berdasarkan penelitian sebelumnya, maka penulis tertarik untuk melakukan penelitian yang bertujuan untuk meningkatkan keamanan pada penyandian pesan. Penulis mengajukan *Vigenere Cipher* untuk menyandikan suatu citra pada pembangkitan kuncinya dengan mengadopsi dari konsep algoritma AES yaitu dengan menambahan perilaku *Shift Rows, Shift Column,* dan operator XOR pada proses penyandian citra menggunakan *Vigenere Cipher*.

1.2 Rumusan Masalah

Berdasarkan latar belakang maka dapat dibuat rumusan masalah sebagai berikut:

a. Bagaimana proses enkripsi dan dekripsi citra menggunakan *Vigenere Cipher*?
b. Bagaimana proses enkripsi dan dekripsi citra menggunakan *Vigenere Cipher* berbasis pembangkitan kunci Algoritma AES?
c. Bagaimana perbandingan tingkat keamanan hasil penyandian citra menggunakan *Vigenere Cipher* dengan hasil penyandian citra menggunakan *Vigenere Cipher* berbasis pembangkitan kunci Algoritma AES?

1.3 Tujuan Penelitian

Tujuan yang ingin dicapai dari penelitian ini adalah sebagai berikut:

a. Melakukan proses enkripsi dan dekripsi citra menggunakan *Vigenere Cipher*.
b. Melakukan proses enkripsi dan dekripsi citra menggunakan *Vigenere Cipher* berbasis pembangkitan kunci Algoritma AES.
c. Membandingkan tingkat keamanan hasil penyandian citra menggunakan *Vigenere Cipher* dengan hasil penyandian citra menggunakan *Vigenere Cipher* berbasis pembangkitan kunci Algoritma AES

1.4 Manfaat Penelitian

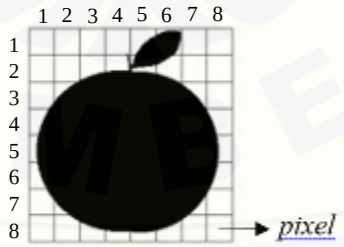
Manfaat dari penelitian ini adalah sebagai berikut:

a. Mengetahui proses enkripsi dan dekripsi citra menggunakan *Vigenere Cipher*.
b. Mengetahui proses enkripsi dan dekripsi citra menggunakan *Vigenere Cipher* berbasis pembangkitan kunci Algoritma AES.
c. Mengetahui perbandingan tingkat keamanan hasil penyandian citra menggunakan *Vigenere Cipher* dengan hasil penyandian citra menggunakan Vigenere Cipher berbasis pembangkitan kunci Algoritma AES?
d. Sebagai tambahan pengetahuan dalam mengkaji permasalahan yang berkaitan dengan penyisipan pesan berupa citra menggunakan *Vigenere Cipher* berbasis pembangkitan kunci Algoritma AES dengan bantuan *software* MATLAB.

BAB 2. TINJAUAN PUSTAKA

2.1 Citra

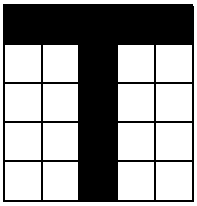
Citra adalah representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpanan. Citra digital adalah citra yang dapat diolah oleh komputer. Istilah citra digital sangat popular pada masa sekarang. Banyak peralatan elektronik, misalnya *scanner*, kamera digital, mikroskop digital, dan pembaca sidik jari (*fingerprint reader*), yang menghasilkan citra digital juga sangat popular digunakan oleh pengguna untuk mengolah foto. Beberapa contoh aplikasi yang menyajikan berbagai fitur untuk memanipulasi citra digital, yaitu *Adobe Photoshop* dan GIMP (*GNU Image Manipulation Program*). Citra digital merupakan citra yang diambil berdasarkan kuantisasi tertentu sehingga citra digital ini terbentuk dari *pixel-pixel* yang besarnya tergantung pada besar kecilnya nilai (besarnya derajat keabuan) (Sari dkk, 2017). Setiap *pixel* merepresentasikan warna atau tingkat keabuan pada satu titik di dalam citra. Nilai *x* pada titik koordinat (*x, y*) merupakan sumbu mendatar (horisontal) yang menunjukkan kolom dari suatu *pixel* dalam citra sedangkan *y* (sumbu vertikal) menunjukkan baris dari suatu *pixel* (Hakim, 2014).



Gambar 2.1 Representasi Citra Digital

Berdasarkan jenisnya, citra digital dibagi menjadi tiga (Mu'mi, 2017) yaitu :

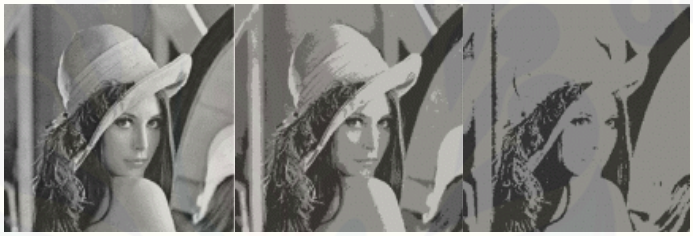
a. Citra biner, yaitu citra yang hanya memiliki dua warna yaitu hitam dan putih. Tiap piksel dalam citra biner direpresentasikan dalam bit 0 dan bit 1 yang mana bit 0 artinya putih dan bit 1 artinya hitam.



1	1	1	1	1
0	0	1	0	0
0	0	1	0	0
0	0	1	0	0
0	0	1	0	0

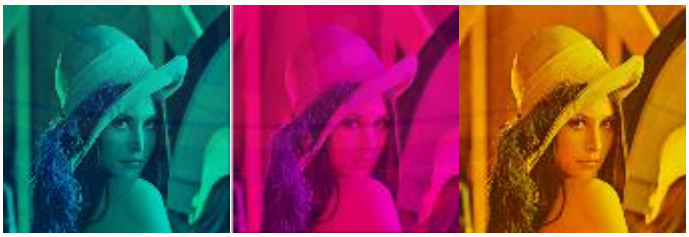
Gambar 2.2 Contoh Citra Biner

b. Citra *grayscale*, yaitu citra yang memerlukan nilai batas keabuan sebagai nilai patokan. *Pixel* dengan derajat keabuan lebih besar dari nilai batas akan diberi nilai 1 dan sebaliknya piksel dengan derajat keabuan lebih kecil dari nilai batas akan diberi nilai 0. Citra *grayscale* disimpan dalam format 8 bit untuk setiap *sample pixel*, yang memungkinkan sebanyak 256 intensitas. Semakin besar jumlah bit warna yang disediakan di memori, maka semakin halus gradasi warna yang terbentuk.



Gambar 2.3 Contoh Citra *Grayscale*

c. Citra warna, yaitu citra yang setiap warna dasarnya memiliki intensitas tersendiri dengan nilai minimum nol (0) dan nilai maksimum 255 (8 bit). Citra warna memiliki warna dasar RGB (*Red, Green, Blue*) atau CMYK (*Cyan, Magenta, Yellow, Black*). Pada umumnya citra warna yang digunakan adalah citra RGB.



Gambar 2.4 Dekomposisi Layer Citra RGB

Gambar 2.5 Dekomposisi Layer Citra CMYK

2.2 Konversi Bilangan

Berdasarkan basisnya, bilangan dibagi menjadi beberapa macam bilangan antara lain adalah bilangan desimal dan bilangan heksadesimal. Bilangan desimal merupakan bilangan yang memiliki basis sebanyak 10, yaitu 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Sedangkan bilangan heksadesimal merupakan bilangan yang memiliki basis sebanyak 16, yaitu 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Bilangan desimal dan bilangan heksadesimal dapat dikonversi menjadi basis bilangan yang lainnya. Penulisan dasar bilangan desimal dan bilangan heksadesimal dapat dilihat pada Tabel 2.1.

Tabel 2.1 Konversi Bilangan Desimal dan Bilangan Heksadesimal

Bilangan Desimal	Bilangan Heksadesimal
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	A
11	B
12	C
13	D
14	E
15	F

Salah satu langkah untuk melakukan konversi bilangan-bilangan tersebut, yaitu merubah bilangan heksadesimal ke biner dapat dilakukan melalui beberapa

tahap. Pertama ubah bilangan ke biner kemudian jumlahkan tiap bobot digit biner yang memiliki bit 1.

Contoh : Ubah bilangan $A85_{16}$ ke desimal

Solusi : Ubah bilangan ke biner

$A = 1010_2$ $8 = 1000_2$ $5 = 0101_2$

Jadi $A85_{16} = 101010000101_2$

Jumlahkan tiap bobot bilangan biner yang memiliki bit 1

$$2^{11} + 2^9 + 2^7 + 2^2 + 2^0 = 2048 + 512 + 128 + 4 + 1 = 2693_{10}$$

Jadi hasil konversi bilang heksadesimal $A85_{16}$ ke bilangan desimal adalah 2693_{10} (Yohandri, 2013).

2.3 Operasi XOR

Pada penelitian ini terdapat beberapa transformasi yang membutuhkan operasi XOR didalamnya. *p* dan *q* adalah sebuah proposisi yang bernilai benar jika salah satu dari *p* dan *q* adalah benar dan bernilai salah jika keduanya bernilai benar atau salah.

Tabel 2.2 Tabel Kebenaran Operasi XOR

P	*q*	$p \oplus q$
Benar	Benar	Salah
Benar	Salah	Benar
Salah	Benar	Benar
Salah	Salah	Salah

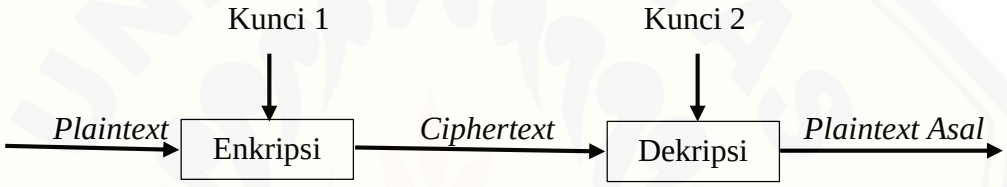
Pada pemrosesan suatu algoritma, unit terkecil adalah bit yang merupakan elemen dalam {0,1}. Pernyataan yang salah dinyatakan dalam bit 0 dan pernyataan yang benar dinyatakan dalam bit 1. Tabel 2.2 dapat direpresentasikan dalam bit menjadi seperti dalam Tabel 2.3.

Tabel 2.3 Operasi XOR dalam Representasi Bit

P	*q*	$p \oplus q$
1	1	0
1	0	1
0	1	1
0	0	0

2.4 Kriptografi

Kriptografi berasal dari bahasa Yunani, *crypto* dan *graphia*. *Chypto* berarti rahasia (*secret*) dan *graphia* berarti tulisan (*writing*). Ada beberapa definisi kriptografi yang menyatakan bahwa kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti maknanya. Definisi ini mungkin cocok pada masa lalu dimana kriptografi digunakan untuk keamanan komunikasi penting seperti komunikasi dikalangan militer, diplomat, dan mata-mata. Namun saat ini kriptografi lebih sekedar *privacy*, tetapi juga untuk tujuan data *integrity*, *autentication*, dan *non-repudiation* (Mu'mi, 2017).



Gambar 2.6 Proses Enkripsi dan Dekripsi Pesan

2.4.1 Terminologi Kriptografi

Beberapa istilah (terminologi) dalam keriptografi dapat dijelaskan sebagai berikut (Ariyus, 2008).

a. *Plaintext*: Teks-asli yang diproses menggunakan algoritma kriptografi untuk menjadi *ciphertext* (teks-kode). Teks-asli ini merupakan pesan yang ditulis atau diketik dengan memiliki makna.
b. *Ciphertext*: suatu pesan yang telah melalui proses enkripsi. Pesan yang ada pada teks-kode ini tidak bisa dibaca karena berupa karakter-karakter yang tidak mempunyai makna (arti).
c. Enkripsi: proses untuk menyandikan *plaintext* menjadi *ciphertext*.
d. Dekripsi: proses pengurai sandi dari *ciphertext* menjadi *plaintext*.
e. Kunci (*key*): parameter yang digunakan untuk mentransformasi proses pengenkripsian dan pendekripsian pesan.
f. Pesan: dapat berupa data atau informasi yang dikirim atau yang disimpan di dalam media perekaman.

g. *Cryptanalysis*: Kriptanalisis bisa diartikan sebagai analisis kode atau suatu ilmu untuk mendapatkan teks-asli tanpa harus mengetahui kunci yang sah secara wajar.

2.4.2 Tujuan Kriptografi

Terdapat empat tujuan mendasar dari kriptografi yang menerapkan aspek keamanan informasi, yaitu:

a. Kerahasiaan

Kerahasiaan adalah layanan yang digunakan untuk menjaga isi dari informasi dari siapapun kecuali yang memiliki kunci rahasia atau otoritas untuk membuka informasi yang telah disandikan.

b. Integritas Data

Berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk dapat menjaga integritas data, suatu sistem harus memiliki kemampuan untuk mendeteksi manipulasi data yang dilakukan pihak-pihak yang tidak berhak, antara lain penyisipan, penghapusan, dan pendistribusian data lain ke dalam data yang asli.

c. Otentifikasi

Berhubungan dengan identifikasi, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri. Informasi yang dikirimkan harus diotentikasi keasliannya, isi datanya, waktu pengiriman dan lain sebagainya.

d. *Non*-repudiasi

Non-repudiasi merupakan usaha untuk mencegah terjadinya penyangkalan terhadap pengirim atau terciptanya suatu informasi oleh yang mengirimkan atau membuat.

2.5 *Vigenere Cipher*

Vigenere Cipher termasuk dalam cipher abjad-majemuk (*polyalpabetic substitution cipher*). Algoritma tersebut baru dikenal luas pada 200 tahun kemudian yang oleh penemunya *cipher* tersebut kemudian dinamakan *Vigenere Cipher*.

Vigenere Cipher menggunakan tabel *Vigenere* untuk melakukan enkripsi. Setiap baris di dalam tabel menyatakan huruf-huruf *ciphertext* yang diperoleh dengan Caesar Cipher (Syawal dkk, 2016).

Kunci : $K = k_1 k_2 \ldots k_m$

Keterangan:

k_i untuk $1 \leq i \leq m$ menyatakan jumlah pergeseran pada huruf ke-i

Pada Tabel 2.4 *Vigenere Cipher*, kolom paling kiri menyatakan huruf-huruf kunci, dan baris paling atas menyatakan *plaintext*. Sedangkan karakter-karakter lainnya menunjukkan karakter *ciphertext*. Setiap baris di dalam tabel menyatakan huruf-huruf *ciphertext* yang diperoleh dengan Caesar Cipher. Pergeseran huruf menjadi *ciphertext* ditentukan oleh nilai desimal dari huruf kunci yang bersangkutan.

Tabel 2.4 *Vigenere Cipher*

		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
kunci	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	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	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

Algoritma enkripsi dan dekripsi pada *Vigenere Cipher* memiliki beberapa karakteristik, yaitu :

a. Hanya menampung 26 huruf alfabet, sedangkan tanda baca lain tidak dapat terbaca.
b. Panjang kunci yang diterima harus sama dengan panjang *plaintext* (P_i), jika panjang kunci lebih pendek daripada panjang *plaintext*, maka kunci diulang secara periodik.

Contoh 1:

Plaintext : THISPLAINTEXT

Kunci : SONY

Dalam hal ini kunci "SONY" diulang sepanjang *plaintext* sesuai dengan karakteristik yang telah dijelaskan. Maka kunci yang digunakan menjadi sebagai berikut:

Plaintext : THISPLAINTEXT

Kunci : SONYSONYSONYS

Setelah terdapat kunci dan *plaintext* yang sesuai dengan karakteristik *Vigenere Cipher*, maka dilakukan langkah selanjutnya, yaitu proses enkripsi. Pada proses enkripsi menggunakan Tabel 2.4.

Contoh 2:

Plaintext : THISPLAINTEXT

Kunci : SONYSONYSONYS

Untuk menghasilkan *ciphertext*, yaitu dengan cara melihat huruf pada *plaintext* dan huruf pada kunci pada Tabel 2.4. Dimana huruf pada *plaintext* dilihat pada baris paling atas pada tabel, sedangkan huruf pada kunci dilihat pada kolom paling kiri pada tabel. Sehingga didapatkan *ciphetext* seperti pada dibawah ini:

Ciphertext : LVVQHZNGFHRVL

Enkripsi pada *Vigenere Cipher* juga dapat dituliskan secara matematis, dengan menggunakan penjumlahan dan operasi modulus seperti pada persamaan (2.1) :

$$C_i = (p_i + k_i)\ mod\ z \qquad (2.1)$$

Keterangan :

C_i = huruf ke-i pada *chipertext*

p_i = huruf ke-i pada *plaintext*

k_i = huruf ke-i pada kata kunci

mod = operasi modulus (sisa pembagian)

z = jumlah karakter

Sebelum melakukan enkripsi secara matematis maka huruf pada kunci dan *plaintext* harus di konversi kedalam bilangan dengan menggunakan tabel 2.5.

Tabel 2.5 Konversi pada *Vigenere Cipher*

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

Contoh 3:

Berdasarkan Contoh 2, huruf pada *plaintext* dan huruf pada kunci di konversi kedalam bentuk bilangan menggunakan Tabel 2.5 kemudian dihitung menggunakan persamaan (2.1) karena huruf terdiri dari 26 karakter, maka *z* yang digunakan adalah 26. Sehingga diperoleh perhitungan seperti dibawah ini:

(19 + 18) mod 26 = 11

(7 + 14) mod 26 = 21

Dengan cara yang sama untuk perhitungan selanjutnya, sehingga hasil enkripsi yang diperoleh sebagai berikut:

Tabel 2.6 Hasil Enkripsi *Vigenere Cipher*

Plaintext	19	7	8	18	15	11	0	8	13	19	4	23	19
Kunci	18	14	13	24	18	14	13	24	18	14	13	24	18
Ciphertext	11	21	21	16	7	25	13	6	5	7	17	21	11

Bilangan yang sama tidak selalu dienkripsi menjadi bilangan *ciphetext* yang sama pula. Contoh: bilangan *plaintext* 19 dapat dienkripsi menjadi 7 atau 11, dan bilangan *ciphertext* 21 dapat merepresentasikan bilangan *plaintext* 7, 8, dan 23. Hal di atas merupakan karakteristik dari *cipher* abjad majemuk: setiap bilangan *ciphetext* dapat memiliki kemungkinan banyak bilangan *plaintext*. Pada cipher substitusi sederhana, setiap bilangan *ciphetext* selalu menggantikan bilangan *plaintext* tertentu.

Dekripsi pada *Vigenere Cipher* juga dapat dituliskan secara matematis, dengan menggunakan pengurangan dan operasi modulus seperti pada persamaan (2.2) :

$$P_i = (c_i - k_i)\ mod\ z \qquad (2.2)$$

Keterangan :

P_i = huruf ke-i pada *plaintext*

c_i = huruf ke-i pada *chipertext*

k_i = huruf ke-i pada kata kunci

mod = operasi modulus (sisa pembagian)

z = jumlah karakter

Sebelum melakukan dekripsi secara matematis maka huruf pada kunci dan *plaintext* harus di konversi kedalam bilangan dengan menggunakan Tabel 2.5.

Contoh 4:

Seperti langkah yang terdapat pada Contoh 3, *ciphertext* dan kunci akan hitungan menggunakan persamaan (2.2) karena huruf terdiri dari 26 karakter, maka z yang digunakan adalah 26. Sehingga doperoleh perhitungan seperti dibawah:

(11 - 18) mod 26 = 19

(21 - 14) mod 26 = 7

Dengan cara yang sama untuk perhitungan selanjutnya, sehingga hasil dekripsi yang diperoleh sebagai berikut:

Tabel 2.7 Hasil Dekripsi *Vigenere Cipher*

Ciphertext	11	21	21	16	7	25	13	6	5	7	17	21	11
Kunci	18	14	13	24	18	14	13	24	18	14	13	24	18
Plaintext	19	7	8	18	15	11	0	8	13	19	4	23	19

2.6 *Advanced Encryption Standard* (AES)

Algoritma AES merupakan algoritma chiper yang aman untuk melindungi data atau informasi yang bersifat rahasia. AES dipublikasikan oleh NIST (*National Institute of Standard and Technology*) pada tahun 2001 yang digunakan untuk menggantikan algoritma DES yang sudah dianggap kuno dan mudah dibobol.

Input dan *output* dari algoritma AES terdiri dari urutan data sebesar 128 bit. Urutan data yang sudah terbentuk dalam satu kelompok 128 bit tersebut disebut juga sebagai blok data atau plaintext yang nantinya akan dienkripsi menjadi *ciphertext*. *Cipher key* dari AES terdiri dari *key* dengan panjang 128 bit, 192 bit, atau 256 bit. Pada penelitian ini menggunakan *Cipher* key dengan panjang 128 bit. Perbedaan panjang kunci akan mempengaruhi jumlah *round* yang akan diimplementasikan pada algoritma AES ini.

Tabel 2.8 Jumlah Putaran pada Algoritma AES

Tipe	Panjang Kunci	Jumlah Putaran
AES-128	128 bit	10
AES-192	192 bit	12
AES-256	256 bit	14

2.6.1 *Key Schedule*

Tahap *key schedule* bertujuan membangun 10 sub-kunci yang akan digunakan pada tiap iterasi tahap enkripsi dan dekripsi. Di awal akan dimasukkan suatu kunci yang disebut *Key*, yang seterusnya akan dilakukan ekspansi terhadap *Key* tersebut dengan menggunakan hasil dari iterasi sebelumnya untuk diproses dalam iterasi berikutnya.

Terlebih dahulu ditetapkan panjang kunci adalah 128 bit, jika kunci yang digunakan kurang dari 128 bit, maka dilakukan penambahan bit-bit "0" pada berisan depan kunci, sehingga panjangnya mencapai 128 bit. Jika kunci yang digunakan lebih dari 128 bit, maka akan diambil 128 bit pertama sebagai *Key*.

Pada *Key* ini dilakukan partisi blok 128 bit dari kunci menjadi 16 nilai masing-masing dua karakter heksadesimal dan direpresentasikan ke dalam matriks 4×4, disebut matriks kunci. Kolom pertama, kedua, ketiga, dan keempat dari matriks kunci tersebut merepresentasikan *word* ke-1 (A_1), *word* ke-2 (A_2), *word* ke-3 (A_3), *word* ke-4 (A_4) yang masing-masing berukuran 4 *byte* atau 32 bit.

Pada kasus ini akan dibangkitkan 40 *word* dengan menggunakan *Key* sebagai *input* kunci, yang selanjutnya akan dipakai pada tiap iterasi pada proses enkripsi dan dekripsi. Untuk menghasilkan kolom pertama pada *Round Key* 1 (B_1), maka prosesnya terdiri dari beberapa operasi yang berurutan yaitu:

a. Operasi *RotWord*

Operasi perputaran 8 bit dengan cara pergeseran kolom secara siklik ke atas. Contoh: *Input word* (a_0, a_1, a_2, a_3) menjadi (a_1, a_2, a_3, a_0).

b. Operasi *SubWord*

Operasi substitusi 8 bit pada hasil *RotWord* dengan nilai dari S-Box. S-Box pada proses ini adalah S-Box sama seperti dengan S-Box yang dipakai pada transformasi *SubByte* pada proses enkripsi.

c. Proses XOR dengan nilai R-Con

Hasil *SubWord* dengan suatu nilai konstan R-Con yang bersesuain tiap round. Dimana R-Con yang bersesuaian yaitu:

$$\begin{bmatrix} 01 & 02 & 04 & 08 & 10 & 20 & 40 & 80 & 1B & 36 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \end{bmatrix}$$

d. Proses XOR antara *word* hasil proses sebelumnya dengan A_1

Untuk menghasilkan *word* ke-2 pada *Round Key* 1 (B_2) cukup dilakukan proses XOR antara B_1 dengan A_2. Prosess yang sama untuk mendapatkan *word* ke-3 dan ke-4 dari *Round Key* 1. Ulangi semua proses diatas untuk mendapatkan sub-kunci selanjutnya (Hanifah, 2012).

2.6.2 Proses Enkripsi Algoritma AES

Proses enkripsi algoritma AES terdiri dari 4 jenis transformasi *bytes*, yaitu *SubBytes, ShiftRows, Mixcolumns, dan AddRoundKey*. Pada awal proses enkripsi, *input* yang telah dicopykan ke dalam *state* akan mengalami transformasi *byte AddRoundKey*. Setelah itu, *state* akan mengalami transformasi *SubBytes, ShiftRows, MixColumns, dan AddRoundKey* secara berulang-ulang sebanyak Nr. Proses ini dalam algoritma AES disebut sebagai *round function*. *Round* yang terakhir agak berbeda dengan *round-round* sebelumnya dimana pada *round* terakhir, *state* tidak mengalami transformasi *MixColumns*. Algoritma enkripsi dapat dilihat pada skema berikut ini :

a. *AddRoundKey*

Dalam *initial round*, transformasi *AddRoundKey* dilakukan terhadap kunci utama. Sedangkan dalam 10 *round* yang lain, proses *AddRoundKey* dilakukan terhadap kunci putaran (*round key*). Proses *AddRoundKey* didefinisikan sebagai operasi XOR antara *array state* dengan *round key*. Operasi XOR dilakukan pada masing-masing *byte* dalam *array* sehingga menghasilkan nilai baru pada *array* hasil dengan ukuran *array* hasil sama dengan ukuran *array state* awal dan *array key*, yaitu sebesar 4×4. Hasil untuk masing-masing baris dan kolom pada *array state* hasil diperoleh dari hasil operasi XOR antara *array state* awal dengan *array key* untuk baris dan kolom yang sama.

b. *SubBytes*

Transformasi *SubBytes* memetakan setiap *byte* dari *array state* dengan menggunakan tabel substitusi S-Box. Tabel S-Box dapat dilihat pada berikut.

Tabel 2.9 Tabel S-Box

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

c. *ShiftRows*

Transformasi *ShiftRows* melakukan pergeseran secara siklik (*wrapping*) pada 3 baris terakhir dari *array state*. Jumlah pergeseran bergantung pada nilai baris (r). Baris $r = 1$ digeser sejauh 8 bit, baris $r = 2$ digeser sejauh 16 bit, dan baris $r = 3$ digeser sejauh 24 bit. Baris $r = 0$ tidak digeser.

d. *MixColumns*

Transformasi *MixColumns* mengalikan setiap kolom dari *array state* dengan polinom $a(x)$ mod $(x^4 + 1)$.

2.6.3 Proses Dekripsi Algoritma AES

Transformasi *cipher* dapat dibalikkan dan diimplementasikan dalam arah yang berlawanan untuk menghasilkan *inverse cipher* yang mudah dipahami untuk algoritma AES. Transformasi *byte* yang digunakan pada *invers cipher* adalah *InvShiftRows, InvSubBytes, InvMixColumns, dan AddRoundKey*. Algoritma dekripsi dapat dilihat pada skema berikut ini :

a. *InvShiftRows*

   InvShiftRows adalah transformasi *byte* yang berkebalikan dengan transformasi *ShiftRows*. Pada transformasi *InvShiftRows*, dilakukan pergeseran bit ke kanan sedangkan pada *ShiftRows* dilakukan pergeseran bit ke kiri.

b. *InvSubBytes*

   InvSubBytes juga merupakan transformasi *byte* yang berkebalikan dengan transformasi *SubBytes*. Pada *InvSubBytes*, tiap elemen pada *state* dipetakan dengan menggunakan tabel *Inverse* S-Box.

c. *InvMixColumns*

   Setiap kolom dalam *state* dikalikan dengan matrik perkalian dalam AES. Hasil dari perkalian matriks tersebut, setiap *byte* dalam kolom *array state* akan digantikan dengan nilai baru (Bhaudhayana dan Widiartha, 2015).

2.7 Analisis Histogram

Analisis histogram dapat mencerminkan informasi dari penyebaran nilai *pixel* pada suatu citra, analisis histogram ini digunakan untuk memperkirakan keamanan pesan yang telah dienkripsi dari serangan kriptanalisis. Histogram suatu citra yang dihasilkan dari pengenkripsian harus memiliki nilai-nilai *pixel* disetiap saluran warna yang tersebar secara seragam agar mampu menjaga keamanan pesan dari serangan statistik. Semakin seragam hasil dari analisis histogram maka semakin kuat keamanan dari pesan yang telah dienkripsi.

Untuk menganalisis keseragaman histogram dari gambar yang terenkripsi, maka dapat menggunakan pengujian X^2, dimana semakin kecil hasil dari X^2 maka tingkat keseragaman dalam histogram semakin merata dan hasil dari pengenkripsian semakin aman, sedangkan semakin besar hasil dari X^2 maka tingkat

keseragaman dalam histogram semakin tidak merata dan hasil dari pengenkripsian semakin tidak aman. Nilai dari X^2 untuk gambar yang terenkripsi dari dimensi $m \times n$ diberikan formula seperti pada persamaan (2.3).

$$X^2 = \sum_{i=0}^{255} \frac{(v_i - v_0)^2}{v_0} \tag{2.3}$$

dimana v_i adalah frekuensi yang diamati dari nilai keabuan $i(0 \leq i \leq 255)$ dan v_0 adalah frekuensi yang diharapkan dari sebuah nilai keabuan i, jadi $v_0 = \frac{m \times n}{256}$ (Boriga dkk, 2014).

2.8 Analisis Diferensial

Analisis diferensial digunakan untuk mengevaluasi kekuatan algoritma pengenkripsian citra dari serangan diferensial. Analisis diferensial dapat ditentukan dengan dua indikator pengukuran yang biasa digunakan, yaitu *Number of Pixels Change Rate* (*NPCR*) dan *Unifer Average Changing Intensity* (UACI). NPCR digunakan untuk mengetahui berapa banyak *pixel* yang berbeda dari dua buah citra, sedangkan UACI berfokus pada interval perbedaan nilai *pixel* dari kedua citra. Perhitungan *NPCR* didefinisikan seperti pada persamaan (2.4).

$$NPCR = \left(\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \sum_{k=0}^{o-1} \frac{d_{i,j,k}}{T} \right) \times 100\% \tag{2.4}$$

dimana T merupakan jumlah total *pixel* di *cipher image*. Untuk menghitung T maka diperlukan m, n, dan o yang melambangkan lebar, tinggi, dan kedalaman citra. Sedangakan $d_{i,j,k}$ melambangkan derajad keabuan dan ditentukan sebagai berikut:

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

dimana $c_{i,j,k}^{(1)}$ dan $c_{i,j,k}^{(2)}$ melambangkan nilai keabuan dari baris i, kolom j, dan kanal k dari citra $c^{(1)}$ (*plain image*) dan $c^{(2)}$(*cipher image*).

Sedangkan perhitungan UACI didefinisikan seperti pada persamaan (2.5).

$$UACI = \left(\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \sum_{k=0}^{o-1} \frac{\left| c_{i,j,k}^{(1)} - c_{i,j,k}^{(2)} \right|}{F \times T} \right) \times 100\% \qquad (2.5)$$

dimana F menunjukkan nilai *pixel* terbesar yang kompatibel dengan format *cipher image* (Wu, 2011). Batas minimal indikator NPCR sebesar 99,609375% dan batas minimal UACI sebesar 33,463541% untuk citra *grayscale* dan RGB, maka *cipher image* dikatakan baik apabila memenuhi batas minimal dari indikator NPCR dan UACI (Boriga dkk, 2014). Secara visual, *cipher image* dikatakan baik apabila sangat "berbeda" dengan citra aslinya dan terlihat acak.

2.9 Analisis Koefisien Korelasi

Analisis statistik seperti faktor koefisien korelasi digunakan untuk mengukur hubungan antara dua variabel, yaitu *plain image* dan *cipher image*. Faktor ini menunjukkan sejauh mana algoritma enkripsi yang diusulkan sangat aman dalam serangan statistik. Oleh karena itu, *cipher image* harus sepenuhnya berbeda dari *plain image*. Koefisien korelasi diukur dengan persamaan (2.6).

$$CorrCoef(\mathrm{x},\mathrm{y}) = \frac{\sum_{i=1}^{n}(x_i - \mu(x))(y_i - (\mu(y))}{\sigma(x)\sigma(y)} \qquad (2.6)$$

di mana $\mu(x)$ dan $\mu(y)$ adalah rata-rata dari masing-masing *x* dan *y* diperoleh dari persamaan (2.7).

$$\mu(x) = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \text{dan} \qquad \mu(y) = \frac{1}{n}\sum_{i=1}^{n} y_i \qquad (2.7)$$

x dan *y* adalah variabel dari *plain image* dan *cipher image*.

Standar deviasi (σ) digunakan untuk mengetahui seberapa dekat sebaran data dengan nilai rata-ratanya. Berikut adalah persamaan (2.8) tentang standar deviasi untuk masing-masing *x* dan *y*.

$$\sigma(x) = \sqrt{\sum_{i=1}^{n}(x_i - \mu(x))^2} \quad \text{dan} \quad \sigma(y) = \sqrt{\sum_{i=1}^{n}(y_i - \mu(y))^2} \qquad (2.8)$$

Jika koefisien korelasi sama dengan *satu*, itu berarti *plain image* dan *cipher image* adalah identik. Jika korelasi koefisien sama dengan *nol*, itu berarti *cipher image* benar-benar berbeda dari *plain image* (yaitu enkripsi yang baik) (Mousa dkk, 2013).

BAB 3. METODE PENELITIAN

3.1 Data Penelitian

Data yang digunakan dalam penelitian kali ini adalah citra RGB dan citra *grayscale* yang digunakan sebagai *plain image*. Data yang digunakan untuk pengujian pada penelitian ini sebanyak 8 citra. Berikut ini adalah data-data yang digunakan pada penelitian.

Gambar 3.1 Citra 1

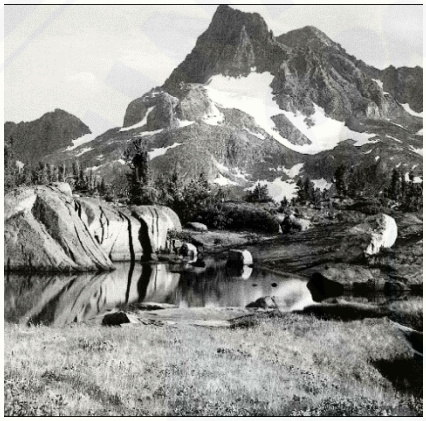
Gambar 3.2 Citra 2

Gambar 3.3 Citra 3

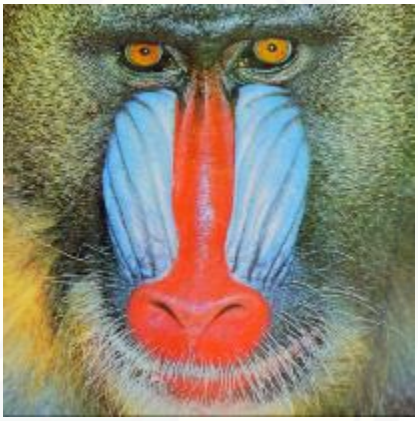
Gambar 3.4 Citra 4

Gambar 3.5 Citra 5

Gambar 3.6 Citra 6



Gambar 3.7 Citra 7



Gambar 3.8 Citra 8

(http://informatika.stei.itb.ac.id/~rinaldi.munir/Koleksi/Citra%20Uji/CitraUji.htm)

3.2 Langkah Penelitian

Langkah-langkah pada penelitian ini adalah sebagai berikut:

a. Studi Literatur

Pada tahap studi literatur penulis mengumpulkan literatur yang berkaitan dengan *Vigenere Cipher* dan Algorima AES.

b. Percobaan Enkripsi dan Dekripsi Citra Menggunakan *Vigenere Cipher*

Pada tahap percobaan enkripsi dan dekripsi citra menggunakan *Vigenere Cipher*, penulis melakukan penelitian dengan mencoba perhitungan secara manual dan melakukan penelitian menggunakan program. Data yang dienkripsi pada perhitungan manual menggunakan *plain image* yang dibangkitkan secara

acak dengan menggunakan *software* MATLAB dan menggunakan kunci berupa teks yang telah di proses dengan *key schedule*.

Langkah-langkah enkripsi dan dekripsi pada tahap percobaan enkripsi dan dekripsi citra menggunakan *Vigenere Cipher* adalah sebagai berikut :

1) Menyiapkan *plain image*.
2) Menyiapkan kunci yang telah di ubah ke dalam bentuk desimal menggunakan Tabel ASCII dengan panjang kunci sesuai panjang *plain image* yang digunakan.
3) Melakukan proses enkripsi *Vigenere Cipher* dengan menggunakan *plain image* dan kunci yang telah dihasilkan pada langkah kedua.
4) Melakukan proses dekripsi *Vigenere Cipher* dengan menggunakan *cipher image* dan kunci yang telah dihasilkan pada langkah kedua.

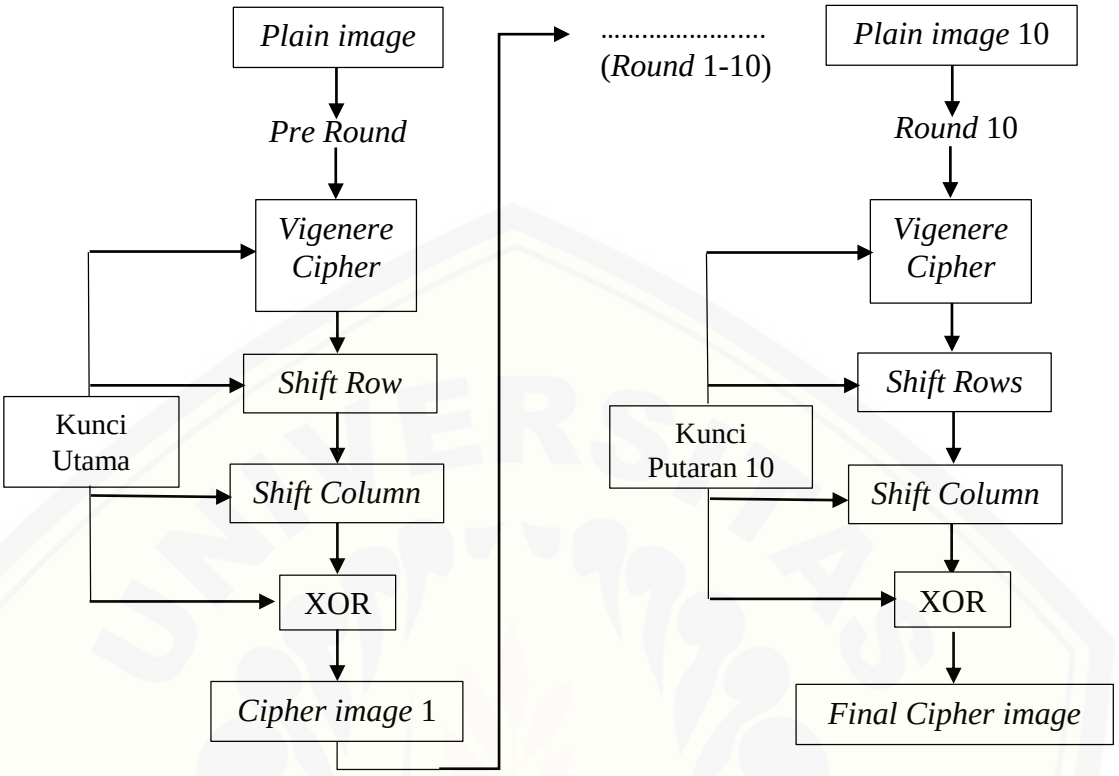
Gambar 3.9 Proses Enkripsi dan Dekripsi *Vigenere Cipher*

c. Percobaan Enkripsi dan Dekripsi Citra Menggunakan *Vigenere Cipher* Berbasis Pembangkit Kunci Algoritma AES

Pada tahap percobaan enkripsi dan dekripsi citra menggunakan *Vigenere Cipher* berbasis pembangkit kunci algoritma AES, penulis melakukan penelitian dengan mencoba perhitungan secara manual dan melakukan penelitian menggunakan program. Data yang dienkripsi pada perhitungan manual menggunakan *plain image* yang dibangkitakn secara acak dengan menggunakan *software* MATLAB dan menggunakan kunci berupa teks yang telah di proses dengan *key schedule*.

Langkah-langkah enkripsi pada tahap percobaan enkripsi dan dekripsi citra menggunakan *Vigenere Cipher* berbasis pembangkit kunci algoritma AES adalah sebagai berikut :

1) Menyiapkan *plain image*.
2) Melakukan pembangkitan kunci menggunakan *key schedule* pada kunci yang telah diubah kedalam bentuk desimal menggunakan Tabel ASCII.
3) Melakukan proses *Vigenere Cipher* dengan menggunakan *plain image* dan kunci yang telah dibangkitkan pada langkah kedua sesuai dengan putaran.
4) Hasil dari langkah ketiga kemudian di pergeseran baris (*shift row*). Setiap baris digeser ke kanan pada saat baris ganjil dan setiap baris digeser ke kiri pada saat baris genap. Pergeseran dilakukan sejumlah *index* karakter kunci ditambah baris ke-*i*.
5) Hasil dari langkah keempat kemudian di pergeseran kolom (*shift column*). Setiap kolom digeser ke bawah pada saat kolom ganjil dan setiap kolom digeser ke atas pada saat kolom genap. Pergeseran dilakukan sejumlah *index* karakter kunci ditambah kolom ke-*i*.
6) Hasil dari langkah keempat kemudian dilakukan operasi XOR dengan kunci sesuai dengan putaran.
7) Melakukan langkah pertama sampai langkah kelima sebanyak 10 kali putaran dengan menggunakan kunci sesuai dengan putaran dan menggunakan *plain image* yang telah dihasilkan dari putaran sebelumnya. *Output* dari tahap ini adalah hasil akhir dari pengenkripsian yaitu *cipher image.*



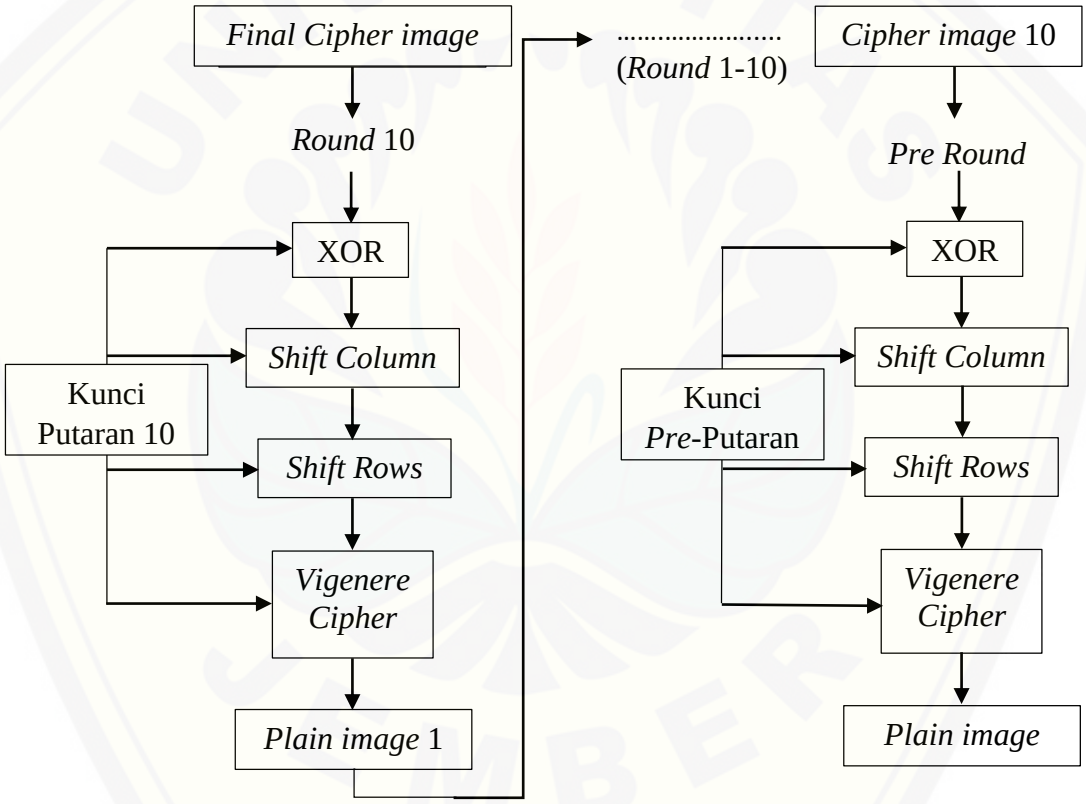
Gambar 3.10 Proses enkripsi pada *Vigenere Cipher* Berbasis Pembangkit Kunci Algoritma AES

Langkah-langkah dekripsi pada tahap percobaan enkripsi dan dekripsi citra menggunakan *Vigenere Cipher* berbasis pembangkit kunci algoritma AES adalah sebagai berikut :

1) Menyiapkan *cipher image* yang telah dihasilkan pada tahap pengekripsian.
2) Menyiapkan kunci yang telah diubah kedalam bentuk desimal menggunakan Tabel ASCII dan digunakan sesuai dengan putaran.
3) Melakukan operasi XOR dengan menggunakan kunci yang sesuai dengan putaran.
4) Hasil dari langkah kedua kemudian di pergeseran kolom (*shift column*). Setiap kolom digeser ke atas pada saat kolom ganjil dan setiap kolom digeser ke bawah pada saat kolom genap. Pergeseran dilakukan sejumlah *index* karakter kunci ditambah kolom ke-*i*.
5) Hasil dari langkah ketiga kemudian di pergeseran baris (*shift row*). Setiap baris digeser ke kanan pada saat baris ganjil dan setiap baris digeser ke kiri

pada saat baris genap. Pergeseran dilakukan sejumlah *index* karakter kunci ditambah baris ke-*i*.

6) Setelah mendapatkan hasil dari langkah ketiga maka dilakukan proses *Vigenere Cipher* dengan menggunakan kunci sesuai dengan putaran.
7) Melakukan langkah pertama sampai langkah keempat sebanyak 10 kali putaran dengan menggunakan kunci sesuai dengan putaran dan menggunakan *cipher image* yang telah dihasilkan dari putaran sebelumnya. *Output* dari tahap ini adalah hasil akhir dari pendekripsian yaitu *plain image*.



Gambar 3.11 Proses dekripsi pada *Vigenere Cipher* Berbasis Pembangkit Kunci Algoritma AES

d. Pembuatan Program Aplikasi Enkripsi dan Dekripsi

Pembuatan program enkripsi dan dekripsi pada citra menggunakan *software* MATLAB R2015b sesuai dengan algoritma yang digunakan pada penelitian ini.

e. Analisis Hasil

Analisis hasil dilakukan setelah mengenkripsi data menggunakan *Vigenere Cipher* dan menggunakan *Vigenere Cipher* berbasis pembangkitan kunci algoritma AES kemudian dihitung hasil histogram, diferensial, dan koefisien korelasi. Dilanjutkan dengan membandingkan hasil perhitungan dari histogram, NPCR, UACI, dan koefisien korelasi. Sehingga dapat dianalisis pengaruh pertambahan pembangkitan kunci algoritma AES terhadap peningkatan keamanan *cipher image* yang dihasilkan.

Skema langkah-langkah pada penelitian :

Gambar 3.12 Skema langkah-langkah penelitian

BAB 5. PENUTUP

5.1 Kesimpulan

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

a. Proses enkripsi menggunakan *Vigenere Cipher* menghasilkan *cipher image* yang masih terlihat polanya, sehingga kurang aman dalam serangan kriptanalisis. Proses dekripsi citra menggunakan *Vigenere Cipher* dapat dapat mengembalikan *cipher image* kedalam citra aslinya.
b. Proses enkripsi citra menggunakan *Vigenere Cipher* berbasis pembangkitan kunci algoritma AES dapat menghasilkan *cipher image* yang terlihat acak, sehingga aman dalam serangan kriptanalisis. Proses dekripsi citra menggunakan *Vigenere Cipher* berbasis pembangkitan kunci algoritma AES dapat mengembalilkan *cipher image* kedalam citra aslinya.
c. Berdasarkan perbandingan antara hasil perhitungan dari histogram, NPCR, UACI, dan koefisien korelasi. Tingkat keamanan hasil penyandian citra menggunakan *Vigenere Cipher* berbasis pembangkitan kunci Algoritma AES menghasilkan nilai yang lebih mendekati batas indikator aman, sehingga dapat disimpulkan bahwa penyandian citra menggunakan *Vigenere Cipher* berbasis pembangkitan kunci Algoritma AES lebih kuat dibandingkan dengan hasil penyandian citra menggunakan *Vigenere Cipher*.

5.2 Saran

Adapun saran yang perlu diperhatikan untuk penelitian lebih lanjut adalah:

a. Menerapkan algoritma AES-192, AES-256, atau algoritma kriptografi modern yang lainnya untuk dibandingkan dengan *Vigenere Cipher* atau algoritma kriptografi klasik yang lainnya.
b. Menerapkan kunci berupa citra dalam implementasi *Vigenere Cipher* pada penyandian citra berbasis pembangkitan kunci algoritma AES agar penyerang sulit menduga kunci yang akan digunakan untuk proses enkripsi dan dekripsi.

DAFTAR PUSTAKA

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

Bhaudhayana, G. W., dan I. M. Widiartha. 2015. Implementasi Algoritma Kriptografi AES 256 Dan Metode Steganografi LSB Pada Gambar Bitmap. *Jurnal Ilmiah*, 8(2):15-25.

Boriga, R. E., A. C. Dăscălescu, dan A. V. Diaconu. 2014. A New Fast Image Encryption Scheme Based on 2D Chaotic Maps. *IAENG International Journal of Computer Science*, 41(4):1-10.

Cahyadi, T. 2012. Implementasi Steganografi LSB Dengan Enkripsi Vigenere Cipher Pada Citra JPEG. *Transient*, 1(4):1-8.

Hakim, L. 2014. Aplikasi Dan Implementasi Secret Sharing Menggunakan Kriptografi Visual Pada Citra Biner. *Jurnal Universitas Brawijaya*, 2(5):1-4.

Hanifah, F. 2012. Aplikasi Algoritma Rijndael Dalam Pengamanan Citra Digital. *Skripsi*. Depok: Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Indonesia.

Luthfi, A. 2011. Enkripsi Citra Bitmap Melalui Substitusi Warna Menggunakan Vigenere Cipher. Bandung: Program Studi Teknik Informatika, Institut Teknologi Bandung.

Mousa, A., O. S. F. Allah., dan E. S. M. Nigm. 2013. Security Analysis of Reverse Encryption Algorithm for Databases. *International Journal of Computer Applications (0975 – 8887)*, 66(14):19-27.

Mu'mi, N. F. A. 2017. Steganografi Citra Menggunakan Kriptografi Hybrid Playfair Cipher Dan Caesar Cipher. *Skripsi*. Makassar: Jurusan Matematika, Fakultas Matematika Dan Ilmu Pegetahuan Alam, Universitas Negeri Makassar.

Pradipta, G. A. 2016. Penerapan Kombinasi Metode Enkripsi Vigenere Chiper Dan Transposisi Pada Aplikasai Client Server Chatting. *Jurnal Sistem Dan Informatika*, 10(2):119-127.

Saputra, R. 2014. Aplikasi Kriptografi Suara Menggunakan Algoritma Advanced Encryption Standard (AES). *Skripsi*. Semarang: Jurusan Ilmu Komputer/ Informatika, Fakultas Sains Dan Matematika, Universitas Diponegoro.

Sari, J. I., Sulindawaty, dan H. T. Sihotang. 2017. Implementasi Penyembunyian Pesan Pada Citra Digital Dengan Menggabungkan Algoritma Hill Cipher Dan Metode Least Significant Bit (LSB). *Mantik Penusa*, 1(2):1-8.

Syawal, M. F., D. C. Fikriansyah, dan N. Agani. 2016. Implementasi Teknik Steganografi Menggunakan Algoritma Vigenere Cipher Dan Metode LSB. *Jurnal TICOM*, 4(3):91-99.

Wu, Yue., J. P. Noonan., dan S. Agaian. 2011. NPCR and UACI Randomness Tests for Image Encryption. *Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications* (JSAT), 31-38.

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

LAMPIRAN

LAMPIRAN A. Skrip Program Pembentukan *Key Schedule* pada MATLAB R2015b

```
function sub=subkey(key,iter,SBoxd)
RotW(:,1)=key([2,3,4,1],4);
SubW(:,1)=SBoxd(RotW(:,1)+1);
SubWbin=dec2binary(SubW);
key1(:,1)=key(:,1);
key1bin=dec2binary(key1);
if iter==9
    RCon=deci2hex(27);
elseif iter==10
    RCon=deci2hex(54);
else
    RCon=deci2hex(2^(iter-1));
end
mid={RCon;'00';'00';'00'};
midbin=hex2binary(mid);
for i=1:4
    temp{i,1}=xor2(xor2(SubWbin{i,1},midbin{i}),key1bin{i,1});
end
for i=2:4
    key2(:,1)=key(:,i);
    key2bin=dec2binary(key2);
    for j=1:4
        temp{j,i}=xor2(temp{j,i-1},key2bin{j,1});
    end
end
sub=binary2dec(temp);
```

LAMPIRAN B. Skrip Program Enkripsi dan Dekripsi pada MATLAB R2015b

a. Skrip program enkripsi pada proses *Vigenere Cipher*

```
%VIGENERE ENKRIPSI
function cipherimage=EncryptVigenere(plainimage,key0)
[m,n,o]=size(plainimage);
Kunci=double(key0);
Key=repmat(Kunci,1,ceil(m*n/length(Kunci)));
KeyV=Key(1:m*n);
KunciV=reshape(KeyV,n,m)';
for i=1:o
    cipherimage(:,:,i)=mod(plainimage(:,:,i)+KunciV,256);
end
```

b. Skrip program dekripsi pada proses *Vigenere Cipher*

```
%VIGENERE DEKRIPSI
function plainimage=DecryptVigenere(cipherimage,key0)
[m,n,o]=size(cipherimage);
Kunci=double(key0);
Key=repmat(Kunci,1,ceil(m*n/length(Kunci)));
KeyV=Key(1:m*n);
KunciV=reshape(KeyV,n,m)';
for i=1:o
    plainimage(:,:,i)=mod(double(cipherimage(:,:,i))-
KunciV,256);
end
```

LAMPIRAN C. Skrip Program Enkripsi dan Dekripsi *Vigenere*-AES pada MATLAB R2015b

a. Skrip program enkripsi pada proses *Vigenere*-AES

```
%VIGENERE-AES ENKRIPSI
function cipherimage=EncryptVigAES(plainimage,key0)
%SIZE IMAGE
[m,n,o]=size(plainimage);
%SBOX HEXA
SBox={'63','7C','77','7B','F2','6B','6F','C5','30','01','67','
2B','FE','D7','AB','76'

'CA','82','C9','7D','FA','59','47','F0','AD','D4','A2','AF','9
C','A4','72','C0'

'B7','FD','93','26','36','3F','F7','CC','34','A5','E5','F1','7
1','D8','31','15'

'04','C7','23','C3','18','96','05','9A','07','12','80','E2','E
B','27','B2','75'

'09','83','2C','1A','1B','6E','5A','A0','52','3B','D6','B3','2
9','E3','2F','84'

'53','D1','00','ED','20','FC','B1','5B','6A','CB','BE','39','4
A','4C','58','CF'

'D0','EF','AA','FB','43','4D','33','85','45','F9','02','7F','5
0','3C','9F','A8'

'51','A3','40','8F','92','9D','38','F5','BC','B6','DA','21','1
0','FF','F3','D2'

'CD','0C','13','EC','5F','97','44','17','C4','A7','7E','3D','6
4','5D','19','73'

'60','81','4F','DC','22','2A','90','88','46','EE','B8','14','D
E','5E','0B','DB'

'E0','32','3A','0A','49','06','24','5C','C2','D3','AC','62','9
1','95','E4','79'

'E7','C8','37','6D','8D','D5','4E','A9','6C','56','F4','EA','6
5','7A','AE','08'

'BA','78','25','2E','1C','A6','B4','C6','E8','DD','74','1F','4
B','BD','8B','8A'

'70','3E','B5','66','48','03','F6','0E','61','35','57','B9','8
6','C1','1D','9E'

'E1','F8','98','11','69','D9','8E','94','9B','1E','87','E9','C
E','55','28','DF'
```

```
'8C','A1','89','0D','BF','E6','42','68','41','99','2D','0F','B
0','54','BB','16'};
%SBOX DECIMAL
SBoxd=hex2deci(SBox)';
%UBAH KUNCI JADI MATRIKS 4x4
key0d=reshape(double(key0),4,4);
%BENTUK SUB KUNCI
sub=createsubkey(key0d,SBoxd);
%KUNCI UTAMA
Kunci=double(key0);
Key=repmat(Kunci,1,ceil(m*n/length(Kunci)));
KeyV=Key(1:m*n);
KunciV=reshape(KeyV,n,m)';
for i=1:o
    %VIGENERE
    cipherimage(:,:,i)=mod(plainimage(:,:,i)+KunciV,256);
    %SHIFTROW
    cipherimage(:,:,i)=shrow(cipherimage(:,:,i),Kunci);
    %SHIFTCOLUMN
    cipherimage(:,:,i)=shcol(cipherimage(:,:,i),Kunci);
    %XOR
    cipherimage(:,:,i)=bitxor(cipherimage(:,:,i),KunciV);
end
for putaran=1:10
    %KUNCI TIAP PUTARAN
    Kunci=reshape(sub{putaran}',1,16);
    Key=repmat(Kunci,1,ceil(m*n/length(Kunci)));
    KeyV=Key(1:m*n);
    KunciV=reshape(KeyV,n,m)';
    for i=1:o
        %VIGENERE
        cipherimage(:,:,i)=mod(cipherimage(:,:,i)+KunciV,256);
        %SHIFTROW
        cipherimage(:,:,i)=shrow(cipherimage(:,:,i),Kunci);
        %SHIFTCOLUMN
        cipherimage(:,:,i)=shcol(cipherimage(:,:,i),Kunci);
        %XOR
        cipherimage(:,:,i)=bitxor(cipherimage(:,:,i),KunciV);
    end
end
```

b. Skrip program dekripsi pada proses *Vigenere*-AES

```
%VIGENERE-AES DEKRIPSI
function plainimage=DecryptVigAES(cipherimage,key0)
%SIZE IMAGE
[m,n,o]=size(cipherimage);
%SBOX HEXA
SBox={'63','7C','77','7B','F2','6B','6F','C5','30','01','67','
2B','FE','D7','AB','76'

'CA','82','C9','7D','FA','59','47','F0','AD','D4','A2','AF','9
C','A4','72','C0'
```

```
'B7','FD','93','26','36','3F','F7','CC','34','A5','E5','F1','7
1','D8','31','15'

'04','C7','23','C3','18','96','05','9A','07','12','80','E2','E
B','27','B2','75'

'09','83','2C','1A','1B','6E','5A','A0','52','3B','D6','B3','2
9','E3','2F','84'

'53','D1','00','ED','20','FC','B1','5B','6A','CB','BE','39','4
A','4C','58','CF'

'D0','EF','AA','FB','43','4D','33','85','45','F9','02','7F','5
0','3C','9F','A8'

'51','A3','40','8F','92','9D','38','F5','BC','B6','DA','21','1
0','FF','F3','D2'

'CD','0C','13','EC','5F','97','44','17','C4','A7','7E','3D','6
4','5D','19','73'

'60','81','4F','DC','22','2A','90','88','46','EE','B8','14','D
E','5E','0B','DB'

'E0','32','3A','0A','49','06','24','5C','C2','D3','AC','62','9
1','95','E4','79'

'E7','C8','37','6D','8D','D5','4E','A9','6C','56','F4','EA','6
5','7A','AE','08'

'BA','78','25','2E','1C','A6','B4','C6','E8','DD','74','1F','4
B','BD','8B','8A'

'70','3E','B5','66','48','03','F6','0E','61','35','57','B9','8
6','C1','1D','9E'

'E1','F8','98','11','69','D9','8E','94','9B','1E','87','E9','C
E','55','28','DF'

'8C','A1','89','0D','BF','E6','42','68','41','99','2D','0F','B
0','54','BB','16'};
%SBOX DECIMAL
SBoxd=hex2deci(SBox)';
%UBAH KUNCI JADI MATRIKS 4x4
key0d=reshape(double(key0),4,4);
%BENTUK SUB KUNCI
sub=createsubkey(key0d,SBoxd);
for putaran=1:10
    %KUNCI TIAP PUTARAN
    Kunci=reshape(sub{11-putaran}',1,16);
    Key=repmat(Kunci,1,ceil(m*n/length(Kunci)));
    KeyV=Key(1:m*n);
    KunciV=reshape(KeyV,n,m)';
    for i=1:o
        %XOR
```

```
        cipherimage(:,:,i)=bitxor(cipherimage(:,:,i),KunciV);
        %SHIFTCOLUMN
        cipherimage(:,:,i)=shcol2(cipherimage(:,:,i),Kunci);
        %SHIFTROW
        cipherimage(:,:,i)=shrow2(cipherimage(:,:,i),Kunci);
        %VIGENERE
        cipherimage(:,:,i)=mod(cipherimage(:,:,i)-KunciV,256);
    end
end
%KUNCI UTAMA
Kunci=double(key0);
Key=repmat(Kunci,1,ceil(m*n/length(Kunci)));
KeyV=Key(1:m*n);
KunciV=reshape(KeyV,n,m)';
for i=1:o
    %XOR
    cipherimage(:,:,i)=bitxor(cipherimage(:,:,i),KunciV);
    %SHIFTCOLUMN
    cipherimage(:,:,i)=shcol2(cipherimage(:,:,i),Kunci);
    %SHIFTROW
    cipherimage(:,:,i)=shrow2(cipherimage(:,:,i),Kunci);
    %VIGENERE
    plainimage(:,:,i)=mod(double(cipherimage(:,:,i))-
KunciV,256);
end
```

LAMPIRAN D. Tabel ASCII

Dec	Hex	Char	Dec	Hex	Char
00	00	NUL	32	20	SP
01	01	SOH	33	21	!
02	02	STX	34	22	″
03	03	ETX	35	23	#
04	04	EOT	36	24	$
05	05	ENQ	37	25	%
06	06	ACK	38	26	&
07	07	BEL	39	27	’
08	08	BS	40	28	(
09	09	HT	41	29	)
10	0A	LF	42	2A	*
11	0B	VT	43	2B	+
12	0C	FF	44	2C	,
13	0D	CR	45	2D	-
14	0E	SO	46	2E	.
15	0F	SI	47	2F	/
16	10	DLE	48	30	0
17	11	DC1	49	31	1
18	12	DC2	50	32	2
19	13	DC3	51	33	3
20	14	DC4	52	34	4
21	15	NAK	53	35	5
22	16	SYN	54	36	6
23	17	ETB	55	37	7
24	18	CAN	56	38	8
25	19	EM	57	39	9
26	1A	SUB	58	3A	:
27	1B	ESC	59	3B	;
28	1C	FS	60	3C	<
29	1D	GS	61	3D	=
30	1E	RS	62	3E	>
31	1F	US	63	3F	?

Dec	Hex	Char
64	40	@
65	41	A
66	42	B
67	43	C
68	44	D
69	45	E
70	46	F
71	47	G
72	48	H
73	49	I
74	4A	J
75	4B	K
76	4C	L
77	4D	M
78	4E	N
79	4F	O
80	50	P
81	51	Q
82	52	R
83	53	S
84	54	T
85	55	U
86	56	V
87	57	W
88	58	X
89	59	Y
90	5A	Z
91	5B	[
92	5C	\
93	5D	]
94	5E	^
95	5F	_

Dec	Hex	Char
96	60	‘
97	61	a
98	62	b
99	63	c
100	64	d
101	65	e
102	66	f
103	67	g
104	68	h
105	69	i
106	6A	j
107	6B	k
108	6C	l
109	6D	m
110	6E	n
111	6F	o
112	70	p
113	71	q
114	72	r
115	73	s
116	74	t
117	75	u
118	76	v
119	77	w
120	78	x
121	79	y
122	7A	z
123	7B	{
124	7C	|
125	7D	}
126	7E	~
127	7F	DEL