PENGAMANAN *IMAGE* DENGAN MODIFIKASI ALGORITMA *ELECTRONIC CODE BOOK* (ECB)

SKRIPSI

Oleh
Melinda Asti
NIM 161810101074

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2020

PENGAMANAN *IMAGE* DENGAN MODIFIKASI ALGORITMA *ELECTRONIC CODE BOOK* (ECB)

SKRIPSI

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

Oleh
Melinda Asti
NIM 161810101074

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2020

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Kedua Orang tua tersayang, Almarhum Ayahanda Drijanto dan Ibunda Endang Kasrotin yang senantiasa memberi dukungan dan do'a;
2. Kedua kakakku Makmur Witono dan Miftha Firlani serta adikku Mega Dzaky Arkananta yang senantiasa memberi motivasi dan do'a;
3. Guru-guru dan dosen dari sekolah dasar hingga perguruan tinggi;
4. Almamater Jurusan Matematika 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)

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

nama : Melinda Asti

NIM : 161810101074

menyatakan dengan sesungguhnya bahwa skripsi yang berjudul "Pengamanan *Image* dengan Modifikasi Algoritma *Electronic Code Book* (ECB)" 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, Januari 2020

Yang menyatakan,

Melinda Asti

NIM 161810101074

SKRIPSI

PENGAMANAN *IMAGE* DENGAN MODIFIKASI ALGORITMA *ELECTRONIC CODE BOOK* (ECB)

Oleh
Melinda Asti
NIM 161810101074

Pembimbing

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

Dosen Pembimbing Anggota : Dr. Kiswara Agung Santoso, S.Si., M.Kom.

PENGESAHAN

Skripsi berjudul "Pengamanan *Image* dengan Modifikasi Algoritma *Electronic Code Book* (ECB)" telah diuji dan disahkan pada:

hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember

Tim Penguji:

Ketua,	Anggota I,
Ahmad Kamsyakawuni, S.Si., M.Kom.	Dr. Kiswara Agung Santoso, S.Si., M.Kom.
NIP. 197211291998021001	NIP. 197209071998031003
Anggota II,	Anggota III,
Ikhsanul Halikin, S.Pd., M.Si.	Kusbudiono, S.Si., M.Si.
NIP. 198610142014041001	NIP. 197704302005011001

Mengesahkan

Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Jember

Drs. Achmad Sjaifullah, M.Sc., Ph.D.

NIP. 195910091986021001

RINGKASAN

Pengamanan *Image* dengan Modifikasi Algoritma *Electronic Code Book* (ECB); Melinda Asti, 161810101074; 2020: 88 Halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Perkembangan teknologi dan informasi saat ini semakin canggih, salah satunya yaitu pada pengiriman pesan. Pengiriman pesan biasanya rentan terhadap pengaksesan oleh pihak ketiga. Hal ini menyebabkan isi pesan yang dikirim dapat diketahui oleh pihak ketiga. 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.

Penelitian ini bertujuan untuk meningkatkan keamanan pada pengamanan data maka penelitian ini membahas pengamanan *image* menggunakan modifikasi *Electronic Code Book* (ECB). Modifikasi *Electronic Code Book* (ECB) terdapat 2 tambahan operasi XOR setelah proses enkripsi *Electronic Code Book* (ECB), yaitu yang pertama diXOR-kan dengan kunci MSB dan yang kedua diXOR-kan dengan kelipatan kuncinya. Proses enkripsi menggunakan modifikasi *Electronic Code Book* (ECB), pixel pada *plain image* yang bernilai sama jika di enkripsi hasilnya tidak selalu bernilai sama. Sehingga citra yang dihasilkan terlihat acak (tidak berpola). Proses dekripsi berhasil mengembalikan *chiperimage* menjadi *plainimage* awal.

Berdasarkan data penelitian yang terdiri dari 8 citra yang telah diuji, hasil dari uji histogram menghasilkan histogram yang lebih seragam menggunakan modifikasi *Electronic Code Book* (ECB) dibandingkan hasil histogram yang hanya menggunakan *Electronic Code Book* (ECB), terlihat juga dari perhitungan X^2 bahwa hasil yang peroleh modifikasi *Electronic Code Book* (ECB) lebih kecil dibandingkan perhitungan yang dihasilkan menggunakan *Electronic Code Book* (ECB). Hasil nilai NPCR dan UACI *Electronic Code Book* (ECB) dapat dikatakan lebih baik dibandingkan dengan hasil nilai NPCR dan UACI modifikasi

Electronic Code Book (ECB). Hasil nilai koefisien korelasi menggunakan modifikasi *Electronic Code Book* (ECB) mendekati nol dan lebih kecil dibandingkan dengan nilai koefisien korelasi menggunakan *Electronic Code Book* (ECB), itu artinya modifikasi *Electronic Code Book* (ECB) lebih kuat terhadap serangan statistik dibandingkan dengan *Electronic Code Book* (ECB). Berdasarkan perbandingan antara hasil perhitungan dari histogram, NPCR, UACI, dan koefisien korelasi, tingkat keamanan hasil pengamanan image menggunakan modifikasi *Electronic Code Book* (ECB) menghasilkan nilai yang lebih mendekati batas indikator aman, sehingga dapat disimpulkan bahwa pengamanan citra menggunakan modifikasi *Electronic Code Book* (ECB) lebih kuat dibandingkan dengan hasil penyandian citra menggunakan *Electronic Code Book* (ECB).

PRAKATA

Puji syukur ke hadirat Allah SWT. atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "Pengamanan *Image* dengan Modifikasi Algoritma *Electronic Code Book* (ECB)". 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. Ahmad Kamsyakawuni, S.Si., M.Kom. dan Dr. Kiswara Agung Santoso, S.Si., M.Kom., selaku dosen pembimbing yang telah memberikan bimbingan dan bantuan dalam penyempurnaan skripsi ini;
2. Ikhsanul Halikin, S.Pd., M.Si. dan Kusbudiono, S.Si., M.Si., selaku dosen penguji yang telah memberikan kritik dan saran yang membangun dalam penyempurnaan skripsi ini;
3. Ikhsanul Halikin, S.Pd., M.Si., selaku selaku Dosen Pembimbing Akademik (DPA) yang telah membimbing dalam pemilihan matakuliah;
4. Almarhum Ayahanda Drijanto dan Ibunda Endang Kasrotin yang telah memberikan dukungan dan doa;
5. Kedua kakak saya Makmur Witono dan Miftha Firlani serta adik saya Mega Dzaky Arkananta yang senantiasa memberi motivasi dan do'a;
6. Seluruh teman-teman "Misdirection" 2016 dan teman-teman yang telah memberikan motivasi serta dukungannya;
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, Januari 2020

Penulis

DAFTAR ISI

DAFTAR TABEL

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR LAMPIRAN

Halaman

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi dan informasi saat ini semakin canggih, salah satunya yaitu pada pengiriman pesan. Pengiriman pesan biasanya rentan terhadap pengaksesan oleh pihak ketiga. Hal ini menyebabkan isi pesan yang dikirim dapat diketahui oleh pihak ketiga.

Solusi untuk menjaga keamanan dan kerahasiaan suatu data maupun informasi adalah teknik enkripsi dan dekripsi. Teknik ini membuat pesan, data dan informasi tidak dapat dimengerti oleh orang lain kecuali pengirim dan penerimanya. Penerima harus mengetahui teknik deskripsi pesan agar isi pesan dapat dibaca dan dimengerti. Ilmu yang mempelajari tentang teknik enkripsi dan deskripsi pesan, data, maupun informasi disebut 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). Kriptografi dibagi menjadi 2 yaitu kriptografi klasik dan kriptografi modern. Salah satu contoh kriptografi modern yaitu *Electronic Code Book* (ECB).

Electronic Code Book (ECB) merupakan metode kriptografi modern yang digunakan untuk mengenkripsi dan mendekripsi teks, citra dan lainnya. Citra terbentuk dari beberapa pixel yang di dalam pixelnya terdiri dari beberapa bit. Bit dibagi menjadi 2 yaitu *Least Significant Bit* (LSB) dan *Most Significant Bit* (MSB). LSB merupakan 4 bit yang letaknya paling kanan sedangkan MSB merupakan 4 bit yang letaknya paling kiri dari sebuah pixel (Gabriel, 2012).

Beberapa penelitian sebelumnya yang berkaitan yaitu Mufid (2010) melakukan penelitian yang berjudul Teknik Enkripsi dan Deskripsi Menggunakan

Algoritma *Electronic Code Book* (ECB). Penelitian tersebut hanya dilakukan dengan teknik enkripsi dan deskripsi teks. Hasil dari penelitian ini adalah *plainteks* yang sama jika dienkripsi akan menghasilkan *cipher teks* yang sama. Hutabalian (2014) melakukan penelitian yang berjudul Perancangan Perangkat Lunak Pengamanan File Menggunakan Algoritma *Electronic Code Book* (ECB). Penelitian tersebut dilakukan dengan teknik enkripsi dan deskripsi teks secara manual dan juga pembuatan program untuk mengenkripsi plainteks dalam jumlah besar. Hasil dari penelitian ini adalah *plainteks* yang sama jika dienkripsi akan menghasilkan *cipher teks* yang sama juga. Wahyuni (2017) melakukan penelitian yang berjudul Implementasi Steganografi dalam Menyembunyikan Pesan Teks dengan Metode MSB (*Most Significan Bit*). Hasil dari penelitian tersebut adalah pesan dapat disembunyikan ke dalam gambar, namun hasil gambar yang telah disisipkan pesan akan berbeda dengan gambar sebelumnya.

Berdasarkan penelitian sebelumnya, maka penulis tertarik untuk melakukan penelitian yang bertujuan untuk meningkatkan keamanan pada penyandian gambar dengan mengenkripsi suatu gambar dengan Algoritma *Electronic Code Book* (ECB). Penulis mengajukan penelitian tentang Pengamanan *Image* Menggunakan Modifikasi Algoritma *Electronic Code Book* (ECB).

1.2 Rumusan Masalah

Berdasarkan latar belakang maka dibuat rumusan masalah yaitu:

a. Bagaimana hasil proses pengamanan *image* dengan algoritma *Electronic Code Book* (ECB)?
b. Bagaimana hasil proses pengamanan *image* dengan modifikasi algoritma *Electronic Code Book* (ECB)?
c. Bagaimana perbandingan tingkat keamanan hasil pengamanan image menggunakan *Electronic Code Book* (ECB) dengan hasil penyandian citra menggunakan modifikasi *Electronic Code Book* (ECB) ?

1.3 Tujuan Penelitian

Tujuan yang ingin dicapai dari penelitian ini adalah sebagai berikut:

a. Melakukan proses pengamanan *image* dengan algoritma *Electronic Code Book* (ECB).
b. Melakukan proses pengamanan *image* dengan modifikasi algoritma *Electronic Code Book* (ECB).
c. Membandingkan tingkat keamanan hasil pengamanan *image* menggunakan *Electronic Code Book* (ECB) dengan hasil penyandian citra menggunakan modifikasi *Electronic Code Book* (ECB).

1.4 Manfaat Penelitian

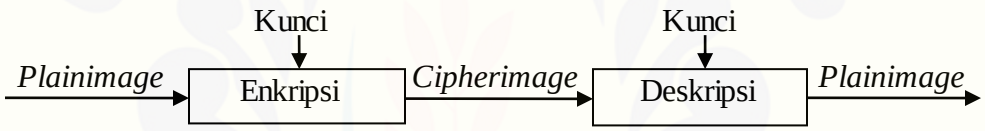
Manfaat dari penelitian ini adalah sebagai berikut:

a. Mengetahui hasil proses pengamanan *image* dengan algoritma *Electronic Code Book* (ECB).
b. Mengetahui hasil proses pengamanan *image* dengan modifikasi algoritma *Electronic Code Book* (ECB).
c. Mengetahui perbandingan tingkat keamanan hasil pengamanan *image* menggunakan *Electronic Code Book* (ECB) dengan hasil penyandian citra menggunakan modifikasi *Electronic Code Book* (ECB).

BAB 2. TINJAUAN PUSTAKA

2.1 Kriptografi

Kriptografi berasal dari kata kripto dan grafi. Kripto berarti menyembunyikan, dan grafi yaitu ilmu. Kriptografi (*cryptography*) adalah suatu ilmu yang mempelajari suatu sistem penyandian untuk menjamin kerahasiaan dan keamanan data. Orang yang melakukan disebut *Criptographer*. Kriptografi merupakan ilmu yang mempelajari teknik matematika yang berhubungan dengan aspek keamanan informasi, seperti kerahasiaan data, keabsahan data, dan integritas data serta autentikasi data (Menezes dkk, 1996). Dapat disimpulkan bahwa kriptografi merupakan ilmu yang mempelajari tentang penyandian data untuk menjamin suatu kerahasiaan, keamanan, keabsahan dan integritas data.



Gambar 2.1 Proses Enkripsi dan Deskripsi

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

a. *Plaimage*: Gambar asli yang diproses menggunakan algoritma kriptografi untuk menjadi *cipherimage*.
b. *Cipherimage*: suatu gambar yang telah melalui proses enkripsi. Gambar yang ada pada *cipherimage* ini tidak berpola tidak mempunyai makna (arti).
c. Enkripsi: proses untuk menyandikan *plainimage* menjadi *cipherimage*.
d. Dekripsi: proses pengurai sandi dari *cipherimage* menjadi *plainimage*.
e. Kunci (*key*): parameter yang digunakan untuk mentransformasi proses pengenkripsian dan pendekripsian gambar.
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 gambar asli tanpa harus mengetahui kunci yang sah secara wajar.

Algoritma Kriptografi dibagi menjadi 2 yaitu kriptografi klasik dan kriptografi modern. Kriptografi Modern memiliki tingkat kesulitan yang kompleks (Prayudi, 2005), dan kekuatan kriptografinya ada pada *key* atau kuncinya (Wirdasari, 2008). Algoritma ini menggunakan pengolahan simbol biner karena berjalan mengikuti operasi komputer digital. Operasi biner yang sering digunakan dalam kriptografi adalah operasi XOR Sehingga membutuhkan dasar berupa pengetahuan terhadap matematika untuk menguasainya (Sadikin, 2012).

2.2 Operasi XOR

Operasi XOR merupakan operasi yang memuat benar atau salah. Jika salah satu dari p dan q bernilai benar, maka operasi XOR bernilai benar. Sebaliknya, jika keduanya bernilai benar atau salah, maka operasi tersebut bernilai salah.

Tabel 2.1 Tabel Kebenaran Operasi XOR

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

Unit terkecil dari suatu citra adalah bit. Elemen dari operasi XOR terdiri dari 1 dan 0. Bit 0 adalah pernyataan salah dan bit 1 adalah pernyataan benar. Tabel 2.2 merupakan tabel kebenaran operasi XOR dalam representasi bit.

Tabel 2.2 Operasi XOR dalam Representasi Bit

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

2.3 Kode ASCII

Kode ASCII (*American Standard Code of Information Interchange)* merupakan suatu standar internasional dalam kode huruf dan simbol, contohnya 124 adalah untuk karakter "|". Kode ASCII selalu digunakan oleh komputer dan alat komunikasi lain untuk menunjukkan teks. Sedangkan fungsi dari kode ASCII ialah digunakan untuk mewakili karakter-karakter angka maupun huruf didalam komputer, sebagai contoh dapat dilihat pada karakter 1, 2, 3, A, B, C, dan sebagainya. Pada citra, kode ASCII digunakan untuk mewakili nilai-nilai pixel dalam bentuk desimal ataupun biner. Kode ASCII sebenarnya memiliki komposisi bilangan biner sebanyak 8 bit. Dimulai dari 0000 0000 hingga 1111 1111. Total kombinasi yang dihasilkan sebanyak 256, dimulai dari kode 0 hingga 255 dalam sistem bilangan desimal (Mukodim, 1994).

2.4 Algoritma *Electronic Code Book* (ECB)

Untuk setiap blok *plaintext Pi*, dienkripsi secara individual dan independen menjadi blok *ciphertext Ci*. Secara matematis, enkripsi dengan algoritma ECB dinyatakan sebagai

$$Ci = Ek\ (\ Pi\ ) \qquad (2.1)$$

sedangkan dekripsi sebagai

$$Pi = Dk\ (Ci\ ) \qquad (2.2)$$

Dalam hal ini, *K* adalah kunci dan *Pi* dan *Ci* masing-masing blok *plaintext* dan *ciphertext* ke-*i*.

Pada teks, *plaintext* dienkripsi dengan kunci menggunakan algoritma ECB menghasilkan *ciphertext*. Sehingga tiap *plaintext* yang sama akan menghasilkan *ciphertext* yang selalu sama pula. Pada citra, nilai pixel *plainimage* dienkripsi dengan kunci menggunakan algoritma ECB menghasilkan *cipherimage*. Sehingga tiap nilai pixel *planimage* yang sama akan menghasilkan nilai pixel *cipherimage* yang selalu sama pula. Sifat dari algoritma ECB ini adalah sederhana dan efisien. Algoritma ECB banyak digunakan karena sederhana dan efisien namun tingkat keamanannya baik.

Istilah "code book" di dalam ECB muncul dari fakta bahwa karena blok *plaintext* yang sama selalu dienkripsi menjadi blok *ciphertext* yang sama, maka secara teoritis dimungkinkan membuat buku kode *plaintext* dari *ciphertext* yang berkoresponden. Namun semakin besar ukuran blok, semakin besar pula ukuran buku kodenya. Misalkan jika blok berukuran 64 bit, maka buku kode terdiri dari $2^{64} - 1$ buah buku kode, yang berarti terlalu besar untuk disimpan dan setiap kunci mempunyai buku kode yang berbeda.

2.4.1 Proses Enkripsi *Electronic Code Book* (ECB)

Untuk membuat enkripsi (perubahan *plaintext* ke dalam *ciphertext*) dengan menggunakan algoritma *Electronic Code Book* (ECB) berdasarkan persamaan (2.1) dapat dilakukan dengan langkah-langkah sebagai berikut :

a. *Plaintext* yang dimasukkan berupa data teks. Misalkan dimasukkan sebuah *plaintext* :

   MSI10

   Teks/ASCII : MSI10 dapat dirubah menjadi bentuk desimal atau biner dengan hasil seperti dibawah ini.

ASCII	: M	S	I	1	0
Desimal	: 77	83	73	49	48
Biner	: 01001101	01010011	01001001	00110001	00110000

b. Bagi *plaintext* menjadi blok-blok yang berukuran 4 bit (*plaintext* sudah dibinerkan).

Plaintext	: 01001101 01010011 01001001 00110001 00110000
Blok 4 bit	: 0100 1101 0101 0011 0100 1001 0011 0001 0011 0000

c. Tentukan kunci (*K*) yang akan digunakan dalam enkripsi. Contoh enkripsi ini menggunakan kunci heksadesimal (4 bit).

   Heksadesimal : B

   Biner : 1011

d. Gunakan fungsi enkripsi (*E*) dengan meng XOR-kan *plaintext Pi* dengan *K*.

Blok Biner : 0100 1101 0101 0011 0100 1001 0011 0001 0011 0000

Kunci : 1011 1011 1011 1011 1011 1011 1011 1011 1011 1011

XOR : 1111 0110 1110 1000 1111 0010 1000 1010 1000 1011

e. Geser hasil XOR tersebut 1 bit ke kiri tiap blok bit sehingga menghasilkan seperti ini.

Geser 1 bit ke kiri :1111 1100 1101 0001 1111 0100 0001 0101 0001 0111

f. Maka akan menghasilkan cipherteks seperti dibawah ini. Dalam contoh ini *ciphertext* di konversi dalam bentuk heksadesimal.

Chipertext : 11111100 11010001 11110100 00010101 00010111

Heksadesimal : FC D1 F4 15 17

2.4.2 Proses Deskripsi *Electronic Code Book* (ECB)

Untuk membuat deskripsi (perubahan *ciphertext* kedalam *plaintext*) dengan menggunakan algorithma *Electronic Code Book* (ECB) berdasarkan persamaan (2.2) dapat dilakukan dengan langkah-langkah sebagai berikut :

a. Geser *ciphertext* 1 bit ke kanan tiap blok bit sehingga terjadi seperti dibawah ini

Ciphertext sebelum digeser.

1111 1100 1101 0001 1111 0100 0001 0101 0001 0111

Ciphertext setelah digeser.

1111 0110 1110 1000 1111 0010 1000 1010 1000 1011

b. *Cipherteks* blok-blok biner hasil pergeseran tersebut dilakukan operasi fungsi enkripsi E dengan cara mengXOR-kan *Ci* tersebut dengan kunci *K*.

Perhatikan hasil operasi tersebut dibawah ini.

Hasil pergeseran: 1111 0110 1110 1000 1111 0010 1000 1010 1000 1011

Kunci : 1011 1011 1011 1011 1011 1011 1011 1011 1011 1011

XOR : 0100 1101 0101 0011 0100 1001 0011 0001 0011 0000

Maka hasilnya sudah sama dengan plainteks awal.

Plaintext : 0100 1101 0101 0011 0100 1001 0011 0001 0011 0000

Biner : 01001101 01010011 01001001 00110001 00110000

ASCII : M S I 1 0

(Mufid, 2010)

2.5 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).

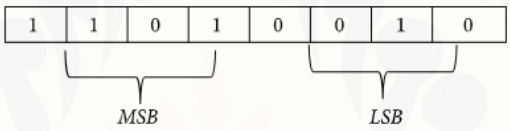
Citra RGB terdiri dari tiga bidang citra yang saling lepas, masing-masing terdiri dari warna mrah, hijau, dan biru. Suatu warna dispesifikasikan sebagai campuran sejumlah komponen warna utama. Monitor komputer dan televisi memakai RGB. Sorotan electron menghasilkan sinyal merah, hijau, biru yang dikombinasikan untuk menghasilkan berbagai warna yang dilihat pada layar.

Citra *grayscale* merupakan citra digital yang terdiri atas warna abu-abu, warna hitam pada bagian yang intensitas terlemah dan warna putih pada intensitas terkuat. Citra *grayscale* berbeda dengan citra ”hitamputih”, dimana pada konteks

komputer, citra hitam putih hanya terdiri atas 2 warna saja yaitu ”hitam” dan ”putih” saja. Pada citra *grayscale* warna bervariasi antara hitam dan putih, tetapi variasi warna diantaranya sangat banyak (Munir, 2004).

2.6 Pixel

Pixel merupakan bagian dari suatu citra. Pixel gambar dapat dikonversi menjadi 8 digit biner (bit). Bit pertama ke keempat disebut *Least Significant Bit* (LSB) di mana perubahan nilai bit pada posisi ini tidak berdampak pada gambar. Bit kelima hingga kedelapan disebut *Most Significant Bit* (MSB), di mana perubahan nilai bit pada posisi ini memiliki efek pada gambar. Gambar 2.1 menunjukkan posisi bit MSB dan LSB.



Gambar 2.2 MSB dan LSB

Sebagai contoh, misalkan tiga pixel yang berdekatan (sembilan bytes) dengan kode RGB berikut :

00110101	11010110	11101010
11110100	00111001	11100001
01110001	10010001	11100001

Pesan yang akan disisipkan adalah karakter “R”, yang nilai binernya adalah “**01010010**”. Pesan akan disisipkan dengan menggunakan metode MSB, maka akan dihasilkan citra hasil dengan urutan bit sebagai berikut:

00110101	**1**1010110	**0**1101010
11110100	**0**0111001	**0**1100001
11110001	**0**0010001	11100001

Pada contoh di atas, dapat dilihat bahwa sebagian MSB (bit ke-8) yang ada pada citra asal (original) digantikan dengan bit dari pesan yang akan disisipkan. Sama seperti metode LSB, pada metode MSB saat penyisipan pesan ada pixel yang berubah dari piksel asal, ada yang tidak berubah sama sekali (Gabriel, 2012).

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 mengetahui perbedaan *plainimage* dengan *cipherimage*. 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).

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

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 derajat 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 untuk mengetahui berapa banyak pixel yang berbeda antara *plainimage* dengan *cipherimage* yaitu sebesar 99,609375% dan batas minimal UACI pada interval perbedaan nilai pixel antara *plain image* dan *cipher image* yaitu 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).

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

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

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

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*.

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

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 ini dipilih berdasarkan dua jenis citra yaitu citra RGB dan citra grayscale yang digunakan sebagai *plainimage*. Data yang digunakan untuk pengujian pada penelitian ini sebanyak 8 citra. Gambar 3.1 sampai 3.8 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

3.2 Langkah Penelitian

Langkah-langkah pada penelitian ini adalah sebagai berikut:

a. Studi Literatur

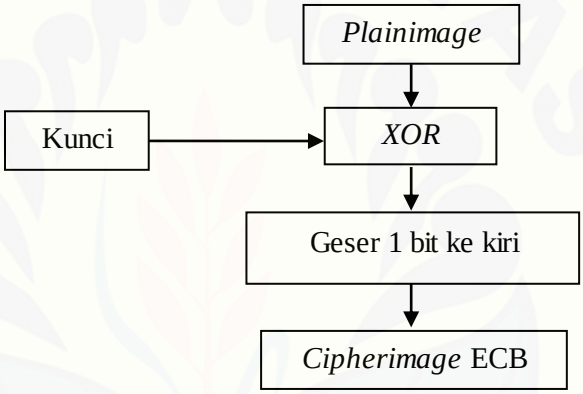
Penulis mengumpulkan literatur yang berkaitan dengan Algoritma *Electronic Code Book* (ECB) dan citra dari berbagai sumber seperti buku dan internet.

b. Percobaan Pengamanan *Image* Menggunakan *Electronic Code Book* (ECB)

Penulis melakukan penelitian dengan mencoba perhitungan secara manual. Data yang dienkripsi berupa *plainimage*. Penulis juga melakukan penelitian menggunakan software MATLAB.

Langkah-langkah enkripsi gambar menggunakan *Electronic Code Book* (ECB) adalah sebagai berikut :

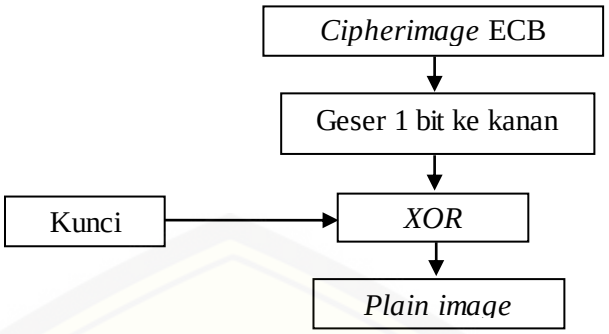
1) Menyiapkan *plain image* dan kunci berupa satu karakter yang dikonversi ke dalam bentuk biner. Bagi *plainimage* dalam bentuk biner menjadi blok-blok yang berukuran 4 bit.
2) Kunci diulang sebanyak bit pada *plain image* kemudian dilakukan operasi XOR dengan *plainimage*.
3) Hasil dari operasi XOR digeser tiap blok bit sebanyak satu bit ke kiri dengan blok bit berisi masing-masing 4 bit. Output dari pengenkripsian ini yaitu *cipherimage ECB*.



Gambar 3.9 Proses Enkripsi *Electronic Code Book* (ECB)

Langkah-langkah deskripsi gambar menggunakan *Electronic Code Book* (ECB) adalah sebagai berikut :

1) Menyiapkan *cipherimage ECB* dan kunci berupa teks yang dikonversi ke dalam bentuk biner. Bagi *cipherimage* dalam bentuk biner menjadi blok-blok yang berukuran 4 bit.
2) Melakukan pergeseran tiap blok bit sebanyak satu bit ke kanan dengan blok bit berisi masing-masing 4 bit pada *cipherimage ECB*.
3) Kunci diulang sebanyak bit pada *cipherimage ECB* kemudian dilakukan operasi XOR dengan *cipherimage ECB* hasil pergeseran. Output dari tahap ini adalah hasil akhir dari pengenkripsian yaitu *plainimage*.



Gambar 3.10 Proses Deskripsi *Electronic Code Book* (ECB)

c. Percobaan Pengamanan *Image* Menggunakan Modifikasi *Electronic Code Book* (ECB)

Penulis melakukan penelitian dengan mencoba perhitungan secara manual. Data yang dienkripsi berupa *plainimage*. Penulis juga melakukan penelitian menggunakan software MATLAB.

Langkah-langkah enkripsi gambar menggunakan *Electronic Code Book* (ECB) adalah sebagai berikut :

1) Menyiapkan *plainimage* dan kunci berupa satu karakter. *Plainimage* dan kunci dikonversi ke dalam bentuk biner. Bagi *plainimage* dalam bentuk biner menjadi blok-blok yang berukuran 4 bit.
2) Kunci diulang sebanyak bit pada *plainimage* kemudian dilakukan operasi XOR dengan *plainimage*.
3) Hasil dari operasi XOR digeser tiap blok bit sebanyak satu bit ke kiri dengan blok bit berisi masing-masing 4 bit. Output dari pengenkripsian ini yaitu *cipherimage ECB.*
4) Kemudian kunci dalam bentuk biner sebanyak 8 bit dibagi menjadi 4 blok bit dengan masing-masing berisi 2 bit.

   Contoh:

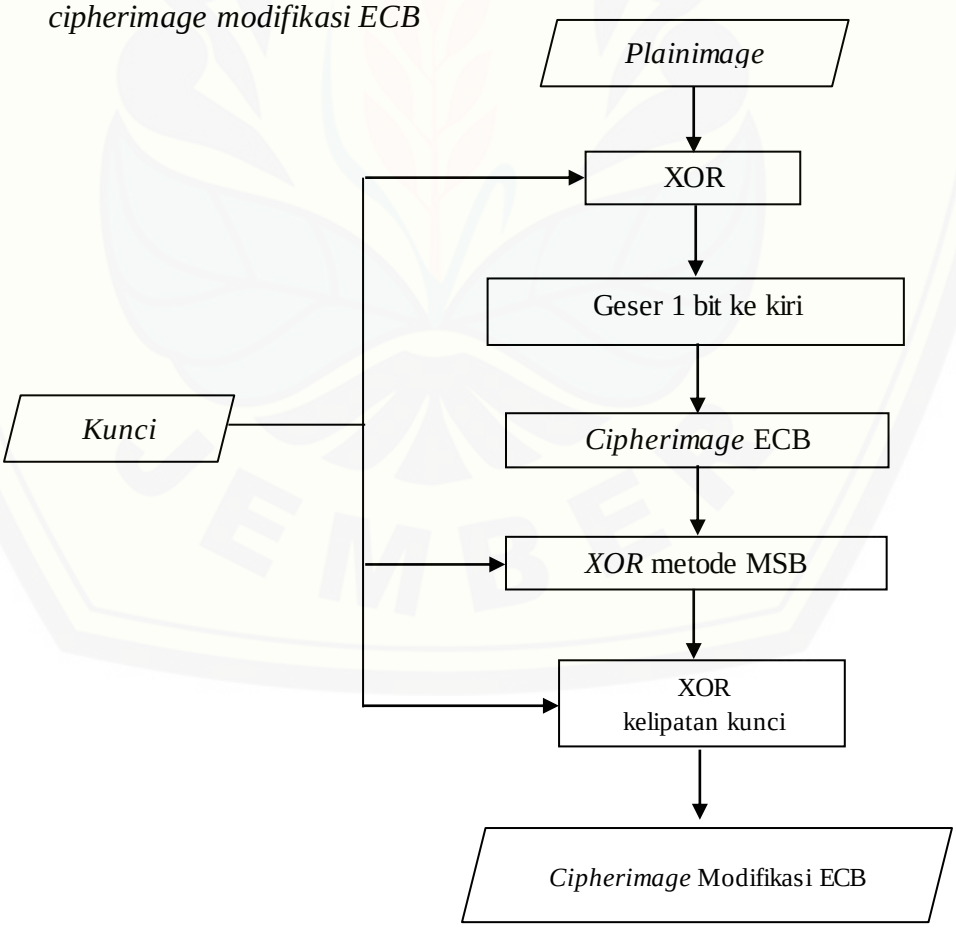
   M = 77 = 01001101 = 01 00 11 01

   Setelah itu masing-masing blok bit kunci yang berisi 2 bit, disisipkan 6 bit dibelakangnya dengan “000000” sehingga berjumlah 8 bit dan hasilnya merupakan kunci MSB.

   01000000 **00**000000 **11**000000 **01**000000

Cipherimage ECB di XOR kan dengan perulangan kunci MSB tersebut menggunakan metode MSB menghasilkan *cipherimage*.

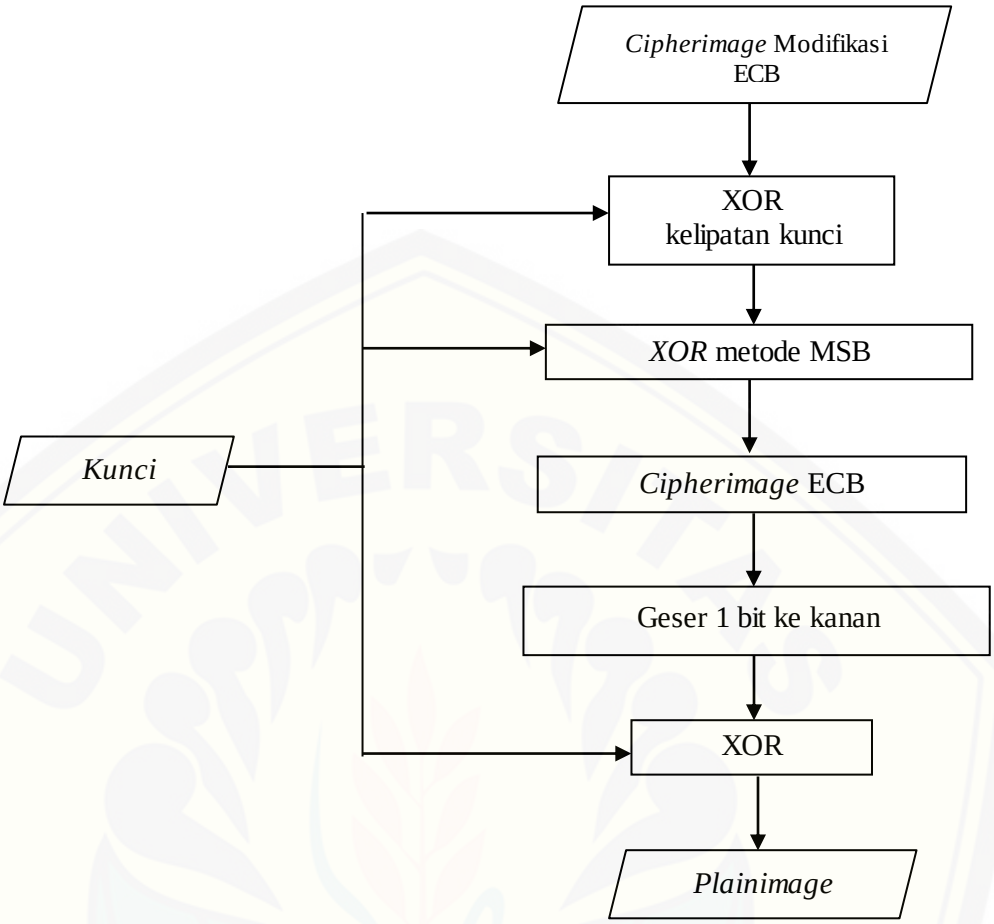
5) Hasil dari operasi XOR *cipherimage ECB* dengan kunci MSB di XOR-kan dengan kelipatan kunci. Misalkan kunci M dikonversi ke desimal menjadi 77. Kelipatan dari 77 yaitu 77, 154, 231, 308, 385, dan seterusnya. Pada citra, pixel yang terdiri dari 8 bit memiliki nilai pixel antara 0 sampai 255 berjumlah 256. Sehingga pada citra digunakan operasi modulo 256 untuk mencari kelipatan kunci dengan perhitungan sebagai berikut : 77 mod 256 = 77, 154 mod 256 = 256, 231 mod 256 = 231, 308 mod 256 = 52, 385 mod 256 = 129. *Cipherimage* MSB di XOR kan dengan perulangan kelipatan kunci modulo 256 sebanyak bit pada *image* MSB menghasilkan *cipherimage modifikasi ECB*.
6) Output dari tahap ini adalah hasil akhir dari pengenkripsian yaitu *cipherimage modifikasi ECB*



Gambar 3.9 Proses Enkripsi Modifikasi *Electronic Code Book* (ECB)

Langkah-langkah deskripsi gambar menggunakan *Electronic Code Book* (ECB) adalah sebagai berikut :

1) Menyiapkan *cipherimage* dan kunci berupa satu karakter. *Plainimage* dan kunci dikonversi ke dalam bentuk biner
2) Misalkan kunci M dikonversi ke desimal menjadi 77. Kelipatan dari 77 yaitu 77, 154, 231, 308, 385, dan seterusnya. Pada citra, pixel yang terdiri dari 8 bit memiliki nilai pixel antara 0 sampai 255 berjumlah 256. Sehingga pada citra digunakan operasi modulo 256 untuk mencari kelipatan kunci dengan perhitungan sebagai berikut : 77 mod 256 = 77, 154 mod 256 = 256, 231 mod 256 = 231, 308 mod 256 = 52, 385 mod 256 = 129. *Cipherimage* di XOR kan dengan perulangan kelipatan kunci modulo 256 sebanyak bit pada *cipherimage* menghasilkan *cipherimage MSB*.
3) Kemudian kunci dalam bentuk biner sebanyak 8 bit dibagi menjadi 4 blok bit dengan masing-masing berisi 2 bit.
   Contoh:
   M = 77 = 01001101 = 01 00 11 01
   Setelah itu masing-masing blok bit kunci yang berisi 2 bit, disisipkan 6 bit dibelakangnya dengan "000000" sehingga berjumlah 8 bit dan hasilnya merupakan kunci MSB.
   01000000 **00**000000 **11**000000 **01**000000
   Cipherimage MSB di XOR kan dengan perulangan kunci MSB tersebut menggunakan metode MSB menghasilkan *cipherimage ECB*.
4) Bagi *cipherimage ECB* dalam bentuk biner menjadi blok-blok yang berukuran 4 bit. Melakukan pergeseran tiap blok bit sebanyak satu bit ke kanan dengan blok bit berisi masing-masing 4 bit pada *cipherimage ECB*.
5) Kunci utama diulang sebanyak bit pada *cipher image ECB* kemudian diXOR-kan dengan *cipherimage ECB* hasil pergeseran.
6) Output dari tahap ini adalah hasil akhir dari pengenkripsian yaitu *plainimage*.



Gambar 3.10 Proses Deskripsi Modifikasi *Electronic Code Book* (ECB)

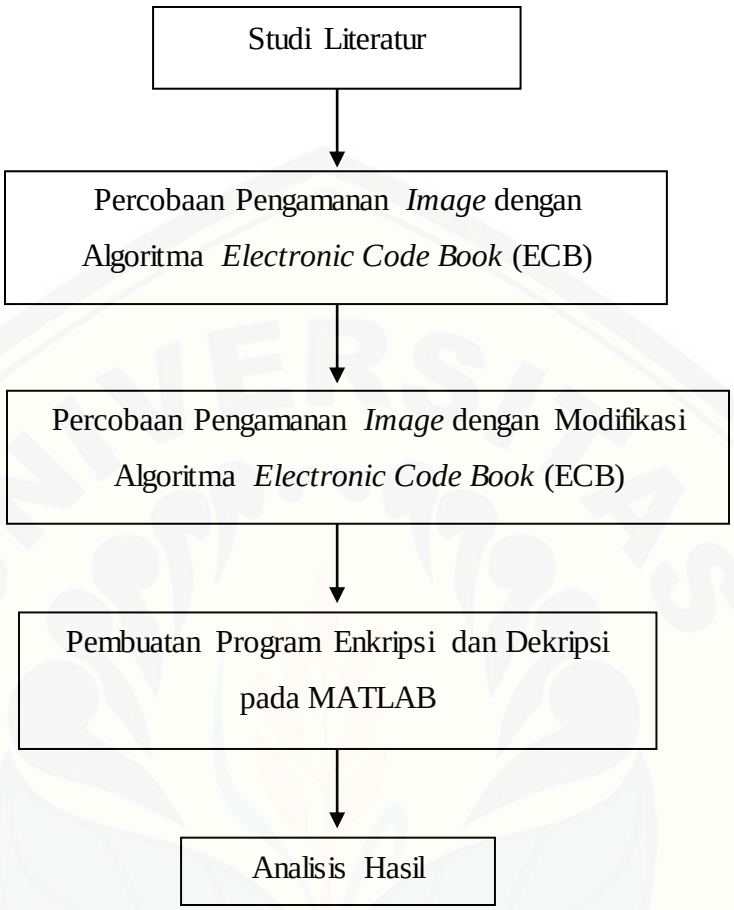
d. Pembuatan Program Aplikasi Enkripsi dan Dekripsi

Pembuatan program aplikasi enkripsi dan dekripsi pada citra menggunakan software MATLAB R2015a sesuai dengan algoritma yang digunakan pada penelitian ini.

e. Analisis Hasil

Analisis hasil dilakukan setelah mengenkripsi data menggunakan *Electronic Code Book* (ECB) dan menggunakan modifikasi *Electronic Code Book* (ECB) kemudian dihitung hasil histogram, diferensial, dan koefisien korelasi. Dilanjutkan dengan membandingkan hasil perhitungan dari histogram, NPCR, UACI, dan koefisien korelasi. Sehingga dapat dianalisis pengaruh modifikasi pada *Electronic Code Book* (ECB) terhadap peningkatan keamanan *cipherimage* yang dihasilkan.

Skema langkah-langkah pada penelitian :



Gambar 3.11 Skema langkah-langkah pada penelitian

BAB 5. PENUTUP

5.1 Kesimpulan

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

a. Proses enkripsi menggunakan *Electronic Code Book* (ECB) menghasilkan *cipherimage* yang masih terlihat polanya, sehingga kurang aman dalam serangan kriptanalisis. Proses dekripsi citra menggunakan *Electronic Code Book* (ECB) dapat dapat mengembalikan *cipherimage* kedalam citra aslinya.
b. Proses enkripsi citra menggunakan Modifikasi *Electronic Code Book* (ECB) dapat menghasilkan *cipherimage* yang terlihat acak, sehingga aman dalam serangan kriptanalisis. Proses dekripsi citra menggunakan Modifikasi *Electronic Code Book* (ECB) dapat mengembalilkan *cipherimage* kedalam citra aslinya.
c. Berdasarkan perbandingan antara hasil perhitungan dari histogram, NPCR, UACI, dan koefisien korelasi. Tingkat keamanan hasil penyandian citra menggunakan Modifikasi *Electronic Code Book* (ECB) menghasilkan nilai yang lebih mendekati batas indikator aman, sehingga dapat disimpulkan bahwa penyandian citra menggunakan Modifikasi *Electronic Code Book* (ECB) lebih kuat dibandingkan dengan hasil penyandian citra menggunakan *Electronic Code Book* (ECB).

5.2 Saran

Adapun saran yang perlu diperhatikan untuk penelitian lebih lanjut adalah:

a. Menerapkan algoritma kriptografi modern yang lainnya untuk dibandingkan dengan Algoritma *Electronic Code Book* (ECB) atau algoritma kriptografi lainnya.
b. Menerapkan kunci lebih dari satu karakter pada Algoritma *Electronic Code Book* (ECB) agar penyerang sulit menduga kunci yang akan digunakan untuk proses enkripsi dan dekripsi.

DAFTAR PUSTAKA

Aristia. 2013. *Simulasi dan Analisis Steganografi Citra Digital Menggunakan Metode AES dan BCH Code*. Bandung: Institut Teknologi Telkom.

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

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.

Gabriel. 2012. An enhanced Least Significant Bit Steganographic Method for Information Hiding. *Journal of Information Engineering and Applications* Vol 2 No.9.

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

Hutabalian, R.T. 2014. Perancangan Perangkat Lunak Pengamanan File Menggunakan Algoritma *Electronic Code Book* (ECB). *Jurnal Iilmiah*. 2(1): 98-104.

Menezes, Alfred J., Paul C van Oorschot, dan Scott A. Vanstone. 1996. *Handbook of Applied Cryptography*. CRC Press.

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.

Mufid, Ahmad. 2010. Teknik Enkripsi dan Deskripsi Menggunakan Algoritma *Electronic Code Book* (ECB). *Jurnal Teknik*, 6(1): 21 – 25.

Mukodim, D. 1994. *Pengantar Bahasa Rakitan*. Jakarta: Gunadarma.

Munir. 2006. *Kriptografi*. Bandung: Informatika.

Prayudi, Yudi, Idham Halik. 2005. Studi Analisis Algoritma Rivest Code 6 (RC6) dalam Enkripsi Deskripsi Data. *Prosiding Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*, Yogyakarta.

Sadikin, R. 2012. *Kriptografi untuk Keamanan Jaringan dan Implementasinya dalam Bahasa Java*. Yogyakarta: Penerbit Andi.

Santoso, K.A., Fatmawati, H.Suprajitno. 2018. On Max-Plus Algebra and Its Application on Image Steganography. *The Scientific World Journal*, 1-9.

Wahyuni, S.R. 2017. Implementasi Steganografi dalam Menyembunyikan Pesan Teks dengan Metode MSB (Most Significant Bit). *Jurnal Informatika dan Teknologi Jaringan*, vol 1, No. 2.

Wirdasari, D. 2008. Prinsip Kerja Kriptografi dalam Mengamankan Informasi. *Jurnal SAINTIKOM*, 5(2): 174-184.

LAMPIRAN

LAMPIRAN A. Skrip Program Pembentukan Kunci pada MATLAB R2015a

a. Skrip program kunci pada proses *Electronic Code Book* (ECB)

```
%Kunci Utama
 KeyM = repmat(Key,m,n); %Matriks kunci
 KeyB = dec2bin(Key,8); %Kunci Biner
```

b. Skrip program kunci pada proses Modifikasi *Electronic Code Book* (ECB)

```
%Kunci MSB
 for i = 1:4
     Key4(1,i) = bin2dec([KeyB(2*i-1:2*i) '000000']);
 end
 KeyMN = repmat(Key4,1,ceil(m*n/4));
 KeyMSB = reshape(KeyMN(1:m*n),n,m)';

%Kelipatan Kunci
if mod(Key,2) == 0
     KeyK = Key+1;
 else
     KeyK = Key;
 end
 KeyKl = reshape(mod(KeyK*(1:m*n),256),n,m)';
```

LAMPIRAN B. Skrip Program Enkripsi dan Dekripsi *Electronic Code Book* (ECB) pada MATLAB R2015a

a. Skrip program enkripsi pada proses *Electronic Code Book* (ECB)

```
%ENKRIPSI ECB
% Bitshift Pattern
 shiftEn = importdata('shiftEn4b2b.mat');
 shiftDe = importdata('shiftDe4b2b.mat');

 if rb1 == 1 % Enkripsi ECB
     Image1 = Image;
     Image2 = Image;
     Image3 = Image;
     for i = 1 : o
         %XOR
         Image1(:,:,i) = bitxor(Image(:,:,i),KeyM);
         %bitshift
         Image2(:,:,i) = shiftEn(Image1(:,:,i)+1);
     end
     Enkripsi1 = Image2;
     axes(handles.axes3);
     imshow(uint8(Enkripsi1));
     set(handles.axes3,'UserData',Enkripsi1);
     if o == 3
                set(handles.popupmenu2,'string',{'Kanal
   RGB','Kanal Red','Kanal Green','Kanal Blue'},'value',1);
     elseif o == 1
                set(handles.popupmenu2,'string','Kanal
   Grayscale','value',1);
     end
     axes(handles.axes4);
     Histogram(Enkripsi1,1);
```

b. Skrip program dekripsi pada proses *Electronic Code Book* (ECB)

```
%DEKRIPSI ECB
elseif rb2 == 1 % Dekripsi ECB
     Image1 = Image;
     Image2 = Image;
     Image3 = Image;
     for i = 1 : o
         %bitshift
         Image2(:,:,i) = shiftDe(Image(:,:,i)+1);
         %XOR
         Image3(:,:,i) = bitxor(Image2(:,:,i),KeyM);
     end
     Dekripsi1 = Image3;
     axes(handles.axes3);
     imshow(uint8(Dekripsi1));
     set(handles.axes3,'UserData',Dekripsi1);
     if o == 3
                set(handles.popupmenu2,'string',{'Kanal
   RGB','Kanal Red','Kanal Green','Kanal Blue'},'value',1);
     elseif o == 1
```

```
            set(handles.popupmenu2,'string','Kanal
Grayscale','value',1);
  end
  axes(handles.axes4);
  Histogram(Dekripsi1,1);
```

LAMPIRAN C. Skrip Program Enkripsi dan Dekripsi Modifikasi *Electronic Code Book* (ECB) pada MATLAB R2015a

a. Skrip program enkripsi pada proses Modifikasi *Electronic Code Book* (ECB)

```
%ENKRIPSI MODIFIKASI ECB
elseif rb3 == 1 % Enkripsi ECB Modif
     Image1 = Image;
     Image2 = Image;
     Image3 = Image;
     Image4 = Image;
     for i = 1 : o
         %XOR
         Image1(:,:,i) = bitxor(Image(:,:,i),KeyM);
         %bitshift
         Image2(:,:,i) = shiftEn(Image1(:,:,i)+1);
         %MSB
         Image3(:,:,i) = bitxor(Image2(:,:,i),KeyMSB);
         %Modif
         Image4(:,:,i) = bitxor(Image3(:,:,i),KeyKl);
     end
     Enkripsi2 = Image4;
     axes(handles.axes3);
     imshow(uint8(Enkripsi2));
     set(handles.axes3,'UserData',Enkripsi2);
     if o == 3
                set(handles.popupmenu2,'string',{'Kanal
  RGB','Kanal Red','Kanal Green','Kanal Blue'},'value',1);
       elseif o == 1
                set(handles.popupmenu2,'string','Kanal
  Grayscale','value',1);
       end
       axes(handles.axes4);
       Histogram(Enkripsi2,1);
```

b. Skrip program dekripsi pada proses Modifikasi *Electronic Code Book* (ECB)

```
%DEKRIPSI MODIFIKASI ECB
elseif rb4 == 1 % Dekripsi ECB Modif
     Image1 = Image;
     Image2 = Image;
     Image3 = Image;
     Image4 = Image;
     for i = 1 : o
         %Modif
         Image1(:,:,i) = bitxor(Image(:,:,i),KeyKl);
         %MSB
         Image2(:,:,i) = bitxor(Image1(:,:,i),KeyMSB);
         %bitshift
         Image3(:,:,i) = shiftDe(Image2(:,:,i)+1);
         %XOR
         Image4(:,:,i) = bitxor(Image3(:,:,i),KeyM);
     end
     Dekripsi2 = Image4;
```

```
    axes(handles.axes3);
    imshow(uint8(Dekripsi2));
    set(handles.axes3,'UserData',Dekripsi2);
    if o == 3
              set(handles.popupmenu2,'string',{'Kanal
  RGB','Kanal Red','Kanal Green','Kanal Blue'},'value',1);
    elseif o == 1
              set(handles.popupmenu2,'string','Kanal
  Grayscale','value',1);
    end
    axes(handles.axes4);
    Histogram(Dekripsi2,1);
end
```

LAMPIRAN D. Skrip Program Analisis Keamanan pada MATLAB R2015a

a. Skrip program Analisis Keamanan

```
Input = get(handles.axes1,'UserData');
Output = get(handles.axes3,'UserData');
if ~isempty(Input) && ~isempty(Output)
   [m1,n1,o1] = size(Input);
   [m2,n2,o2] = size(Output);
   if m1==m2 && n1==n2 && o1==o2
     NPCR = NPCRcal(Input,Output);
     set(handles.text6,'string',[num2str(NPCR) ' %']);
     UACI = UACIcal(Input,Output);
     set(handles.text7,'string',[num2str(UACI) ' %']);
     corr = CoefCorr(Input,Output);
     set(handles.text8,'string',num2str(corr));
     X2 = X2cal(Output);
     set(handles.text10,'string',num2str(X2));
   end
 end
```

b. Skrip program Analisis Histogram

```
function X2=X2cal(Image)
[m,n,o]=size(Image);
v0=m*n/256;
X2=0;
for i=1:o
    h=imhist(uint8(Image(:,:,i)));
   X2=X2+sum((h-v0).^2/v0);
end
```

c. Skrip program Analisis Diferensial

```
function npcr=NPCRcal(plainimage,cipherimage)
[m,n,o]=size(plainimage);
dij=plainimage-cipherimage;
dij(dij~=0)=1;
npcr=sum(sum(sum(dij)))/(m*n*o)*100;

function uaci=UACIcal(plainimage,cipherimage)
[m,n,o]=size(plainimage);
uaci=sum(sum(sum((abs(plainimage-cipherimage))
/255)))/(m*n*o)*100;
```

d. Skrip program Analisis Koefisien Korelasi

```
function corr=CoefCorr(plainimage,cipherimage)
[m,n,o]=size(plainimage);
mup=sum(sum(sum(plainimage)))/(m*n*o);
muc=sum(sum(sum(cipherimage)))/(m*n*o);
sigp=sqrt(sum(sum(sum((plainimage-mup).^2))));
sigc=sqrt(sum(sum(sum((cipherimage-muc).^2))));
corr=sum(sum(sum((plainimage-mup).*(cipherimage-
muc))))/(sigp*sigc);
```

LAMPIRAN E. Tabel ASCII

Desimal	Octa	Heksadesimal	Biner	Simbol
0	000	00	00000000	NULL
1	001	01	00000001	SOH
2	002	02	00000010	STX
3	003	03	00000011	ETX
4	004	04	00000100	EOT
5	005	05	00000101	ENQ
6	006	06	00000110	ACK
7	007	07	00000111	BEL
8	010	08	00001000	BS
9	011	09	00001001	HT
10	012	0A	00001010	LF
11	013	0B	00001011	VT
12	014	0C	00001100	FF
13	015	0D	00001101	CR
14	016	0E	00001110	SO
15	017	0F	00001111	SI
16	020	10	00010000	DLE
17	021	11	00010001	DC1
18	022	12	00010010	DC2
19	023	13	00010011	DC3
20	024	14	00010100	DC4
21	025	15	00010101	NAK
22	026	16	00010110	SYN
23	027	17	00010111	ETB
24	030	18	00011000	CAN
25	031	19	00011001	EM
26	032	1A	00011010	SUB
27	033	1B	00011011	ESC
28	034	1C	00011100	FS
29	035	1D	00011101	GS
30	036	1E	00011110	RS
31	037	1F	00011111	US
32	040	20	00100000	(Spasi)
33	041	21	00100001	!
34	042	22	00100010	"
35	043	23	00100011	#
36	044	24	00100100	$
37	045	25	00100101	%
38	046	26	00100110	&
39	047	27	00100111	'
40	050	28	00101000	(
41	051	29	00101001	)

42	052	2A	00101010	*
43	053	2B	00101011	+
44	054	2C	00101100	,
45	055	2D	00101101	-
46	056	2E	00101110	.
47	057	2F	00101111	/
48	060	30	00110000	0
49	061	31	00110001	1
50	062	32	00110010	2
51	063	33	00110011	3
52	064	34	00110100	4
53	065	35	00110101	5
54	066	36	00110110	6
55	067	37	00110111	7
56	070	38	00111000	8
57	071	39	00111001	9
58	072	3A	00111010	:
59	073	3B	00111011	;
60	074	3C	00111100	<
61	075	3D	00111101	=
62	076	3E	00111110	>
63	077	3F	00111111	?
64	100	40	01000000	@
65	101	41	01000001	A
66	102	42	01000010	B
67	103	43	01000011	C
68	104	44	01000100	D
69	105	45	01000101	E
70	106	46	01000110	F
71	107	47	01000111	G
72	110	48	01001000	H
73	111	49	01001001	I
74	112	4A	01001010	J
75	113	4B	01001011	K
76	114	4C	01001100	L
77	115	4D	01001101	M
78	116	4E	01001110	N
79	117	4F	01001111	O
80	120	50	01010000	P
81	121	51	01010001	Q
82	122	52	01010010	R
83	123	53	01010011	S
84	124	54	01010100	T
85	125	55	01010101	U

86	126	56	01010110	V
87	127	57	01010111	W
88	130	58	01011000	X
89	131	59	01011001	Y
90	132	5A	01011010	Z
91	133	5B	01011011	[
92	134	5C	01011100	\
93	135	5D	01011101	]
94	136	5E	01011110	^
95	137	5F	01011111	_
96	140	60	01100000	`
97	141	61	01100001	a
98	142	62	01100010	b
99	143	63	01100011	c
100	144	64	01100100	d
101	145	65	01100101	e
102	146	66	01100110	f
103	147	67	01100111	g
104	150	68	01101000	h
105	151	69	01101001	i
106	152	6A	01101010	j
107	153	6B	01101011	k
108	154	6C	01101100	l
109	155	6D	01101101	m
110	156	6E	01101110	n
111	157	6F	01101111	o
112	160	70	01110000	p
113	161	71	01110001	q
114	162	72	01110010	r
115	163	73	01110011	s
116	164	74	01110100	t
117	165	75	01110101	u
118	166	76	01110110	v
119	167	77	01110111	w
120	170	78	01111000	x
121	171	79	01111001	y
122	172	7A	01111010	z
123	173	7B	01111011	{
124	174	7C	01111100	|
125	175	7D	01111101	}
126	176	7E	01111110	~
127	177	7F	01111111	
128	200	80	10000000	€
129	201	81	10000001	•

130	202	82	10000010	‚
131	203	83	10000011	ƒ
132	204	84	10000100	„
133	205	85	10000101	…
134	206	86	10000110	†
135	207	87	10000111	‡
136	210	88	10001000	ˆ
137	211	89	10001001	‰
138	212	8A	10001010	Š
139	213	8B	10001011	‹
140	214	8C	10001100	Œ
141	215	8D	10001101	•
142	216	8E	10001110	Ž
143	217	8F	10001111	•
144	220	90	10010000	•
145	221	91	10010001	‘
146	222	92	10010010	’
147	223	93	10010011	“
148	224	94	10010100	”
149	225	95	10010101	•
150	226	96	10010110	–
151	227	97	10010111	—
152	230	98	10011000	˜
153	231	99	10011001	™
154	232	9A	10011010	š
155	233	9B	10011011	›
156	234	9C	10011100	œ
157	235	9D	10011101	•
158	236	9E	10011110	ž
159	237	9F	10011111	Ÿ
160	240	A0	10100000	
161	241	A1	10100001	¡
162	242	A2	10100010	¢
163	243	A3	10100011	£
164	244	A4	10100100	¤
165	245	A5	10100101	¥
166	246	A6	10100110	¦
167	247	A7	10100111	§
168	250	A8	10101000	¨
169	251	A9	10101001	©
170	252	AA	10101010	ª
171	253	AB	10101011	«
172	254	AC	10101100	¬
173	255	AD	10101101	

174	256	AE	10101110	®
175	257	AF	10101111	¯
176	260	B0	10110000	°
177	261	B1	10110001	±
178	262	B2	10110010	²
179	263	B3	10110011	³
180	264	B4	10110100	´
181	265	B5	10110101	µ
182	266	B6	10110110	¶
183	267	B7	10110111	·
184	270	B8	10111000	¸
185	271	B9	10111001	¹
186	272	BA	10111010	º
187	273	BB	10111011	»
188	274	BC	10111100	¼
189	275	BD	10111101	½
190	276	BE	10111110	¾
191	277	BF	10111111	¿
192	300	C0	11000000	À
193	301	C1	11000001	Á
194	302	C2	11000010	Â
195	303	C3	11000011	Ã
196	304	C4	11000100	Ä
197	305	C5	11000101	Å
198	306	C6	11000110	Æ
199	307	C7	11000111	Ç
200	310	C8	11001000	È
201	311	C9	11001001	É
202	312	CA	11001010	Ê
203	313	CB	11001011	Ë
204	314	CC	11001100	Ì
205	315	CD	11001101	Í
206	316	CE	11001110	Î
207	317	CF	11001111	Ï
208	320	D0	11010000	Ð
209	321	D1	11010001	Ñ
210	322	D2	11010010	Ò
211	323	D3	11010011	Ó
212	324	D4	11010100	Ô
213	325	D5	11010101	Õ
214	326	D6	11010110	Ö
215	327	D7	11010111	×
216	330	D8	11011000	Ø
217	331	D9	11011001	Ù

218	332	DA	11011010	Ú
219	333	DB	11011011	Û
220	334	DC	11011100	Ü
221	335	DD	11011101	Ý
222	336	DE	11011110	Þ
223	337	DF	11011111	ß
224	340	E0	11100000	à
225	341	E1	11100001	á
226	342	E2	11100010	â
227	343	E3	11100011	ã
228	344	E4	11100100	ä
229	345	E5	11100101	å
230	346	E6	11100110	æ
231	347	E7	11100111	ç
232	350	E8	11101000	è
233	351	E9	11101001	é
234	352	EA	11101010	ê
235	353	EB	11101011	ë
236	354	EC	11101100	ì
237	355	ED	11101101	í
238	356	EE	11101110	î
239	357	EF	11101111	ï
240	360	F0	11110000	ð
241	361	F1	11110001	ñ
242	362	F2	11110010	ò
243	363	F3	11110011	ó
244	364	F4	11110100	ô
245	365	F5	11110101	õ
246	366	F6	11110110	ö
247	367	F7	11110111	÷
248	370	F8	11111000	ø
249	371	F9	11111001	ù
250	372	FA	11111010	ú
251	373	FB	11111011	û
252	374	FC	11111100	ü
253	375	FD	11111101	ý
254	376	FE	11111110	þ
255	377	FF	11111111	ÿ