ENKRIPSI CITRA DIGITAL MENGGUNAKAN *VIGENERE CIPHER* DAN *LOGISTIC MAP*

SKRIPSI

Disusun oleh:

GONDO SUWIRYO

NIM. 0810963044

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN

PROGRAMSTUDITEKNIK INFORMATIKA

PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER

UNIVERSITASBRAWIJAYA

MALANG

2012

ENKRIPSI CITRA DIGITAL MENGGUNAKAN *VIGENERE CIPHER* DAN *LOGISTIC MAP*

SKRIPSI

Sebagaisalahsatusyaratuntukmemperoleh
GelarSarjanadalambidangIlmuKomputer

Disusun oleh:
GONDO SUWIRYO
NIM. 0810963044

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
PROGRAMSTUDITEKNIK INFORMATIKA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITASBRAWIJAYA
MALANG
2012

ENKRIPSI CITRA DIGITAL MENGGUNAKAN *VIGENERE CIPHER* DAN *LOGISTIC MAP*

SKRIPSI

Sebagaisalahsatusyaratuntukmemperoleh
GelarSarjanadalambidangIlmuKomputer

Disusun oleh:
GONDO SUWIRYO
NIM. 0810963044

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I,	**Dosen Pembimbing II,**
Edy Santoso, S.Si., M.Kom.	**Lailil Muflikhah S.Kom, M.Sc.**
NIP. 197404142003121004	**NIP. 19741113 200501 2 001**

LEMBAR PENGESAHAN SKRIPSI

ENKRIPSI CITRA DIGITAL MENGGUNAKAN *VIGENERE CIPHER* DAN *LOGISTIC MAP*

SKRIPSI

Sebagaisalahsatusyaratuntukmemperoleh
GelarSarjanadalambidangIlmuKomputer

Disusun oleh:

GONDO SUWIRYO
NIM. 0810963044

Setelah dipertahankan di depan Majelis Penguji
pada tanggal 28 Desember 2012
dan dinyatakan memenuhi syarat untuk memperoleh
gelar Sarjana dalam bidang Ilmu Komputer

Penguji,	**Penguji,**
Drs. Marji, M.T.	**Drs. Achmad Ridok, M.Kom.**
NIP. 19670801 199203 1 001	**NIP. 19680825 199403 1 002**

Penguji,

Budi Darma Setiawan, S.Kom., M.Cs.
NIK.84101506110090

Mengetahui
Ketua Program Studi Teknik Informatika

Drs. Marji, M.T.
NIP. 19670801 199203 1 001

PERNYATAAN ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah SKRIPSI ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata didalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku. (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, Desember 2012

Mahasiswa,

Gondo Suwiryo

NIM. 0810963044

KATA PENGANTAR

Puji syukur ke hadirat Allah SWT yang telah melimpahkanrahmat dan hidayah-Nya kepada penulis, sehingga penulis dapatmenyelesaikan skripsi yang berjudul "*Enkripsi Citra Digital* Menggunakan *Vigenere Cipher* Dan *Logistic Map*".

Skripsi ini adalah sebagai salah satu syarat untukmenyelesaikan studi dan memperoleh gelar Sarjana Komputer di Universitas Brawijaya.Banyak pihak yang berperan atas terselesaikannya skripsi ini. Atas bantuan yang telah diberikan, penulisingin menyampaikan penghargaan dan ucapan terima kasihkepada :

1. Edy Santoso, SSi., M.Kom,selaku dosen pembimbing utama dan Lailil Muflikhah, S.Kom., M.Sc.selaku dosen pembimbing pendamping yang bijaksana dan sabar dalam membimbing dan menyalurkan ilmu kepada penulis dalam penyusunan skripsi ini.
2. Drs. Marji, MT, selaku Ketua Program Studi Teknik Informatika. Terima kasih atas bimbingan yang telah diberikan.
3. Ir. Sutrisno, MT, selaku Ketua Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
4. Segenap bapak dan ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada penulis selama menempuh pendidikan di Program Studi Teknik InformatikaProgram Teknologi Informasi dan Ilmu KomputerUniversitas Brawijaya.
5. Segenap staf dan karyawan di ProgramStudiTeknik Informatika Program Teknologi Informasi dan Ilmu KomputerUniversitas Brawijaya.
6. Kedua orang tua dan segenap keluarga yang telah mendoakan dan memberi dukungan sehingga penulis dapat menyelesaikan skripsi ini dengan baik.
7. Teman-teman Ilmu Komputer 2008 yang telah banyak memberikan bantuan, dorongan serta motivasi sehingga skripsi ini dapat terselesaikan.
8. Dan semua pihak yang telah membantu dalam penyusunan skripsi ini yang tidak dapat penulis sebutkan satu persatu.

Penulis sadari bahwa masih banyak kekurangan dalam laporanini, oleh karena itu penulis sangat menghargai saran dan kritik yang sifatnya membangun demi perbaikan penulisan dan mutu isi tugasakhir ini untuk kelanjutan penelitian serupa di masa mendatang.Sebuah harapan tulus semoga skripsi ini dapat bermanfaatsebesar-besarnya.

Malang, Desember 2012

Penulis

ABSTRAK

Gondo Suwiryo. 2012. :*Enkripsi Citra Digital* Menggunakan *Vigenere Cipher* Dan *Logistic Map*.

Dosen Pembimbing :Edy Santoso, SSi., M.Kom, dan Lailil Muflikhah, S.Kom, M.Sc.

Citra merupakan representasi dari sebuah fakta yang terjadi di dunia pada saat tertentu. Pada suatu saat, citra dapat menjadi aset berharga yang tidak boleh dilihat selain oleh orang yang bersangkutan sehingga dibutuhkan sebuah metode untuk melindungi kerahasiaan, keutuhan dan keaslian citra. Kriptografi dapat dimanfaatkan untuk menjamin keamanan suatu citra dengan mengamankan suatu informasi (*plainimage*) dengan menyembunyikan citra sehingga tampak terlihat tidak seperti aslinya (*cipherimage*).

Vigenere cipher adalah teknik enkripsi yang merupakansalah satu algoritma kriptografi yang digunakan untuk penyandian teks. Namun dalam penelitian ini pemakaian *vigenere cipher* diperluas dari teks ke citra *bitmap* 24-bit. Selain itu, metode ini akan digabungkan dengan salah satu teori *chaos* (acak atau random) yaitu *logistic map*. *Logistic map* tersebut digunakan untuk membangkitkan bilangan secara acak.

Bilangan acak ini kemudian akan dimanfaatkan sebagai kunci dalam melakukan proses enkripsi dan dekripsi.Hasil percobaan menunjukkan pada resolusi 1024x768 tingkat keamanannya masih kurang baik karena hasil enkripsi (*cipherimage*) masih terlihat pola aslinya, namun pada resolusi 125x122 menghasilkan *cipherimage* dengan tingkat keamanan yang lebih baik karena hasil *enkripsi* (*cipherimage*) tidak terlihat pola aslinya.

KataKunci:Citra, *Enkripsi*, Kriptografi, *Logistic Map*,*Vigenere cipher*

ABSTRACT

Gondo Suwiryo. 2012. :*Enkripsi Citra Digital* Menggunakan *Vigenere Cipher* Dan *Logistic Map*.

Advisor :Edy Santoso, SSi., M.Kom and Lailil Muflikhah, S.Kom, M.Sc

The imageis a representation ofa fact that ishappeningin the worldat any given moment. At one point, the image canbe avaluableassetthat should not beseenexcept by theperson concernedand so we needa methodto protect the confidentiality, integrityandauthenticity ofthe image. Cryptographycan be usedtoensure the safety ofanimagewith thesecuringof information(plainimage) by hidingthe imageso itlookslooksnothing likethe original(cipherimage).

Vigenerecipherencryptionis a techniquethatis one of thecryptographic algorithmsusedfortext encoding. However,in this studythe use ofVigenereciphertext tobe extendedfrom24-bitbitmapimage. In addition,this methodwill be coupledwith oneof chaos(random) thelogisticmap.Logisticmapis usedtogeneraterandomnumbers.

Random numberis then usedas a keyin the process ofencryptionanddecryption.Hasilexperimentsshow1024x768 resolutionlevel of securityis still not gooddue tothe encrypted(cipherimage) still looksoriginalpattern, butat a resolution of125x122generatecipherimagewitha betterlevel of securitybecausethe encrypted(cipherimage) invisible originalpattern.

Keywords:Image, *Encryption, Cryptography, LogisticMap,Vigenerecipher*

DAFTAR ISI

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR TABEL

Halaman

DAFTAR SOURCE CODE

Halaman

BABI
PENDAHULUAN

1.1.LatarBelakang

Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat yang lain [ARI-05].Dunia kriptografi saat ini telah menerapkan berbagai metode untuk penyandian data berupa multimedia salah satunya adalah citra.Citra digunakan dalam berbagai bidang seperti seni, hiburan, iklan, pendidikan serta pelatihan.

Saat ini dengan bertambahnya penggunaan teknik digital bagi transmisi dan penyimpanan citra, masalah mendasar untuk melindungi kerahasiaan, keutuhan dan keaslian citra memang perlu diperhatikan.Hal ini dikarenakan kerahasiaan suatu informasi sangatlah penting dan bersifat pribadi.

Teknik kriptografi dapat dimanfaatkan untuk menjamin keamanan suatu citra. Salah satu teknik yang dapat dimanfaatkan adalah enkripsi dan dekripsi atau dengan kata lain mengamankan suatu informasi *(plainimage)* dengan menyembunyikan citra sehingga tampak terlihat tidak seperti aslinya *(cipherimage)*.Enkripsi citra dilakukan dengan cara merubah informasi warna pada tiap pixel dari citra tersebut. Dengan berubahnya warna-warna di setiap pixel citra digital, maka bentuk asli citra tersebut tidak dapat diketahui lagi.

Vigenere cipher merupakan pengembangan dari *Caesar cipher*. *Caesar cipher* adalah metode enkripsi paling awal. Metode ini termasuk metode cipher subtitusi dimana mengganti setiap karakter di dalam alfabet dengan karakter yang terletak pada tiga posisi berikutnya di dalam susunan alfabet[MUN-06].Kelebihan metode ini dibanding metode Caesar dan metode metode monoalfabetik lainnya adalah sandi ini tidak begitu rentan terhadap metode pemecahan sandi yang disebut analisis frekuensi.*Vigenere cipher* adalah salah satu metode enkripsi yang digunakan untuk penyandian teks. Namun oleh I Ketut Gede Suhartana (2009) pemakaian*vigenere cipher* diperluas dari teks ke citra bitmap 24-bit.

Metode *Vigenere cipher*akan lebih baik jika digunakan pembangkitan kunci untuk penyandiannya. Agar tingkat keamanan dan keakuratan dalam proses

enkripsi dan dekripsi tersebut menjadi lebih baik. Jika kunci itu semakin acak maka *cipherimage* yang dihasilkan akan semakin kuat.Maka diperlukan sebuah metode yang dapat digunakan untuk pembangkitan dan pengacakan kunci.

Salah satu teori chaos yaitu *Logistic Map* digunakan Ratna Ekasari Prihandini (2007) untuk pembangkitan kunci secara acak. Bilangan acak ini kemudian akan dimanfaatkan sebagai kunci dalam melakukan proses enkripsi.
Berdasarkan latar belakang diatas maka dalam skripsi ini diambil judul "**Enkripsi Citra Digital Menggunakan *Vigenere Cipher* Dan *Logistic Map*".**

1.2. Rumusan Masalah

Permasalahanyangakandijadikanobyekpenelitianpadatugasakhiriniadalah:

1. Bagaimana mengimplementasikan metode *Vigenere Cipher* dan *Logistic Map* pada citra digital untuk melakukan *enkripsi* dan *dekripsi*?
2. Bagaimana tingkat akurasi korelasi dan *entropy* dari hasil enkripsi metode *Vigenere Cipher* dan *Logistic Map* pada citra digital?

1.3.BatasanMasalah

Batasanmasalahdalampenulisantugasakhiriniadalah:

1. Proses *enkripsi* dan *dekripsi* citra digital hanya menggunakan metode *Vigenere Cipher* dan *Logistic Map*.
2. Data yang digunakan berupa citra digital dengan *format bitmap* (*.bmp) 24 bit dengan ukuran resolusi maksimal 1024x768.
3. Kualitas kelayakan didasarkan pada perubahan nilai korelasi dan *entropy* hasil *enkripsi* citra.
4. Aplikasi yang akan dihasilkan berbasis bahasa pemrogaman java.

1.4.Tujuan

Tujuandaripenulisantugasakhiriniadalah:

1. Menghasilkan perangkat lunak dengan teknik kriptografi yaitu metode *Vigenere Cipher* dan *Logistic Map* untuk enkripsi dan dekripsi pada citra digital.
2. Mengetahui tingkat akurasi korelasi dan *entropy dari* hasil dekripsi metode

Vigenere Cipher dan *Logistic Map* pada citra digital.

1.5.Manfaat

Manfaat penelitian adalah didapatkan aplikasi untuk keamanan data berupa citra digital dengan cara melakukan enkripsi dan dekripsi sekaligus mendapatkan pembelajaran pada bidang kriptografi.

1.6.MetodologiPenelitian

Metodologiyangdigunakandalampenulisantugasakhiriniadalahsebagaiberikut:

1. StudiLiteratur
   Mempelajarimetode*vigenere cipher*dan*logistic map.*
2. PendefinisiandanAnalisisMasalah
   Mendefinisikandanmenganalisismasalahuntukmemperolehsolusiyangtepat.
3. PerencanaandanPembuatanPerangkatLunak
   Merencanakanpembuatan perangkat lunak menggunakan Netbeans7.1.
4. PengujiandanAnalisaPerangkatLunak
   Mengujicobaprogramdanmenganalisahasiloutputdariprogram.
5. PengambilanKesimpulan
   Mengambilkesimpulandenganmelihathasiloutputprogram.

1.7.SistematikaPenulisan

Tugasakhirinidisusunberdasarkansistematikapenulisansebagaiberikut:

1. BABIPENDAHULUAN
   Babiniberisitentanglatarbelakangpenulisantugasakhir,rumusanmasalah,batasanmasalah,tujuanpenulisan,manfaatpenulisan,metodologiyangdigunakansertasistematika penulisan.
2. BABIITINJAUANPUSTAKA
   Babiniberisitentangteori-teori,metode,dankonsepyangdipakaipenulisdalampembuatantugasakhir.
3. BABIIIMETODOLOGIDANPERANCANGAN
   Bab ini berisi mengenai metodologi serta perancangan sistem untuk *enkripsi* citramenggunakan metode *Vigenere Cipher* dan *Logistic Map*dan

perancangan antarmuka.

4. BABIVHASILDANPEMBAHASAN
   Babiniberisipenjelasanimplementasidanrancanganyangtelahdiuraikanpada BabIIIdanhasilpengujianyangdilakukan.Babinijugaberisitentangpengujianprogramyangtelahdibuat.
5. BABVKESIMPULANDANSARAN
   Babiniberisikesimpulandansaranyangdiperolehdaripembahasanmateridanpengujianaplikasiyangdiharapkanbermanfaatuntukpengembanganpenelitianlebihlanjut.

BABII
TINJAUANPUSTAKA

2.1. Kriptografi

2.1.1 Pengertian Kriptografi

Kriptografi berasal dari bahasa Yunani, *crypto* dan *graphia*.*Crypto* berarti *secret* (rahasia) dan *graphia* berarti *writing* (tulisan). Menurut terminologinya, kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat yang lain.Dalam perkembangannya, kriptografi juga digunakan untuk mengidentifikasi pengiriman pesan dan tanda tangan digital dan keaslian pesan dengan sidik jari digital [Dony Ariyus-05].

Kriptografi telah menjadi salah satu cara yang sering digunakan dalam melawan atau menghalangi penyerangan terhadap keamanan dan privasi seseorang, menjamin integritas data dan kerahasiaan data, dan memberikan kepercayaan pada global *ecommerce*. Kriptografi telah menjadi hal utama dalam menyediakan kebutuhan keamanan digital pada era komunikasi digital modern saat ini. Tujuan kriptografi adalah untuk menjamin aspek kerahasiaan, integritas data, autentikasi, dan non-repudansi pada semua komunikasi dan pertukaran informasi [KIZ-05].

Di dalam kriptografi kita akan sering menemukan berbagai istilah atau terminology. Beberapa istilah yang harus diketahui yaitu :

1. Pesan, plaintext, dan cipherteks

   Pesan (message) adalah data atau informasi yang dapat dibaca dan dimengerti maknanya. Nama lain untuk pesan adalah (plaintext) atau teks jelas (cleartext). Agar pesan tidak dapat dimengerti maknanya oleh pihak lain yang tidak berkepentingan, maka pesan perlu disandikan kebentuk lain yang tidak dapat dipahami. Bentuk pesan yang tersandi disebut cipherteks (*ciphertext*) atau kriptogram (*cryptogram*).*Cipherteks* harus dapat ditransformasikan kembali menjadi *plaintext* semula agar dapat diterima dan bisa dibaca.

2. Pengirim dan Penerima

   Komunikasi data melibatkan pertukaran pesan antara dua entitas.Pengirim (*sender*) adalah entitas yang mengirim pesan kepada entitas lainnya.Penerima (*receiver*) adalah entitas yang menerima pesan. Pengirim tentu menginginkan pesan dapat dikirm secara aman, yaitu pengirim yakin bahwa pihak lain tidak dapat membaca isi pesan yang dikirim. Solusinya adalah dengan cara

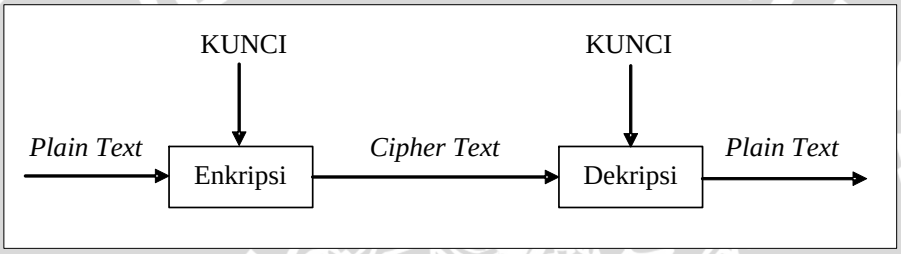
menyandikan pesan menjadi *cipherteks*.

3. *Enkripsi* dan *Dekripsi*

   Proses menyandikan plainteks menjadi cipherteks disebut enkripsi (*encryption*) atau *enciphering*. Sedangkan proses mengembalikan cipherteks menjadi plainteks disebut dekripsi (*decryption*) atau *deciphering*.

4. *Cipher* dan kunci

   Algoritma kriptogarfi disebut juga cipher, yaitu aturan untuk enkripsi dan dekripsi, atau fungsi matematika yang digunakan untuk enkripsi dan dekripsi. Gambar 2.1. memperlihatkan gambaran umum proses kriptografi.Beberapa cipher memerlukan algoritma yang berbeda untuk *enciphering* dan *deciphering*.



Gambar 2.1 Gambaran umum proses kriptografi

5. Sistem kriptografi

   Kriptografi membentuk sebuah sistem yang dinamakan sistem Kriptografi.Sistem kriptografi (*cryptosystem*) adalah kumpulan yang terdiri dari algoritma kriptografi, semua plainteks dan cipherteks yang mungkin, dan kunci.Di dalam kriptografi, cipher hanyalah salah satu komponen saja.

6. Penyadap

   Penyadap (*eavesdropper*) adalah orang yang mencoba menangkap pesan selama ditransmisikan. Tujuan penyadap adalah untuk mendapatkan informasi sebanyak - banyaknya mengenai system kriptografi yang digunakan untuk berkomunikasi dengan maksud untuk memecahkan cipherteks. Nama lain penyadap :*enemy*, *adversary*, *intruder*, *interceptor*, *bad guy*.

7. Kriptanalisis dan kriptologi

   Kriptografi berkembang sedemikian rupa sehingga melahirkan bidang yang berlawanan yaitu kriptanalisis.Kriptanalisis (*cryptanalysis*) adalah ilmu dan seni untuk memecahkan cipherteks menjadi plainteks tanpa mengetahui kunci yang

digunakan.Pelakunya disebut kriptanalis [ARI-08].

2.1.2 Sejarah Kriptografi

Sejarah kriptografi ini dimulai pada zaman romawi kuno.Suatu saat Julius Caesar ingin mengirimkan sebuah pesan rahasia melalui seorang kurir kepada seorang jenderal dimedan perang. Karena tidak ingin pesan tersebut bocor ditengah jalan, Julius Caesar kemudian mengacak pesan tersebut sehingga pesan tidak dapat diketahui oleh orang lain kecuali dirinya dan jenderalnya. Julius Caesar melakukan pergeseran huruf sehingga mengganti semua susunan alfabet. huruf A diganti dengan hufuf D, huruf B diganti dengan huruf E, C diganti dengan F dan seterusnya, sehingga pesan tersebut berubah menjadi suatu sandi dan tidak terbaca. Proses tersebut dinamakan proses enkripsi. Pesan awal sebelum dilakukan penyandian disebut dengan *plaintext*,sedangkan pesan hasil penyandian disebut dengan *chipertext*[PRA-12].

Metode kriptografi berkembang sesuai perkembangan zaman dan kebutuhan.Pada abad ke-15, Leo Battista Alberti menemukan metode roda kode (*wheel cipher*) yang kemudian dikembangkan menjadi alat enkripsi dan dekripsi hingga saat ini.Penggunaan alat ini aman karena bersifat fleksibel yang memungkinkan dilakukanya berbagai perubahan.Bentuk roda kode terdiri dari dua buah potongan silindris yaitu silindris dalam dan luar. Masing-masing silindris memiliki label seluruh alphabet yang tidak harus terurut dan sama. Silindris luar merupakan alphabet untuk teks asli dan silindris luar untuk teks kode.

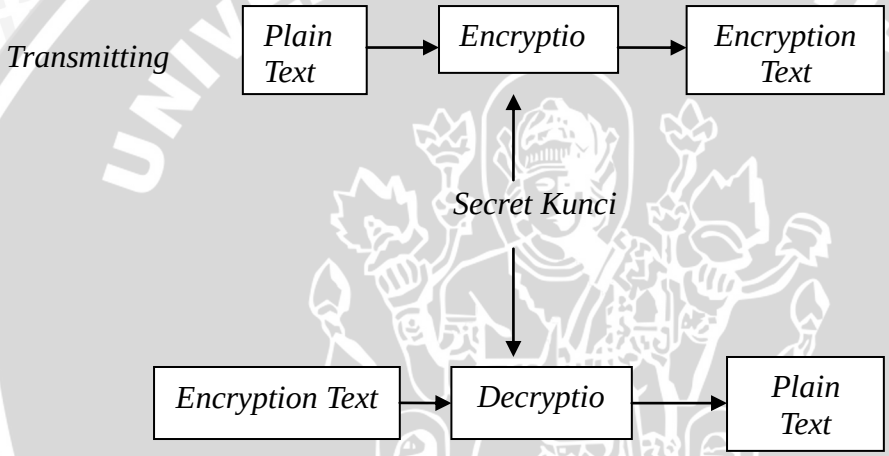
Perkembangan selanjutnya adalah pada saat Perang Dunia II, pada saat itu Jerman menggunakan mesin rotor yang disebut enigma melakukan enkripsi dan dekripsi pesan.Enigma merupakan mesin kripto berbasis rotor.Mesin rotor yang dibuat oleh Alexander Koch pada tahun 1919 dikembangkan dan dipatenkan oleh Arthur Scheribus. Mesin rotor yang dikembangkan oleh Scheribus diberi namaenigma yang kemudian akan dikomersialkan. Pada tahun 1926 Angkatan Laut Jerman memodifikasi sederhana enigma.Kemudian pada tahun 1930, Enigma versi militer telah berhasil dibangun [ARI-08].

2.1.3 Jenis Kriptografi

Terdapat dua jenis teknik kriptografi berdasar jenis kuncinya, yaitu:

a) Kriptografi Simetris

Kriptografi simetris disebut juga sebagai kriptografi konvensional, yaitu teknik yang menggunakan kunci yang sama untuk proses enkripsi dan dekripsinya. Gambar 2.2. memperlihatkanskema kriptografisimetris yanghanya membutuhkan satu buah kunci yang sama. Keamanan kriptografi simetris tergantung pada kuncinya.kriptografi simetris sering juga disebut teknik kunci rahasia, teknik kunci tunggal atau teknik satu kunci. Dua kategori yang termasuk pada teknik simetris ini adalah metode *block cipher* dan *stream cipher*[KUR-04].



Gambar 2.2 Skema Kriptografi Simetris

Kelebihan teknik kriptografi simetris adalah:

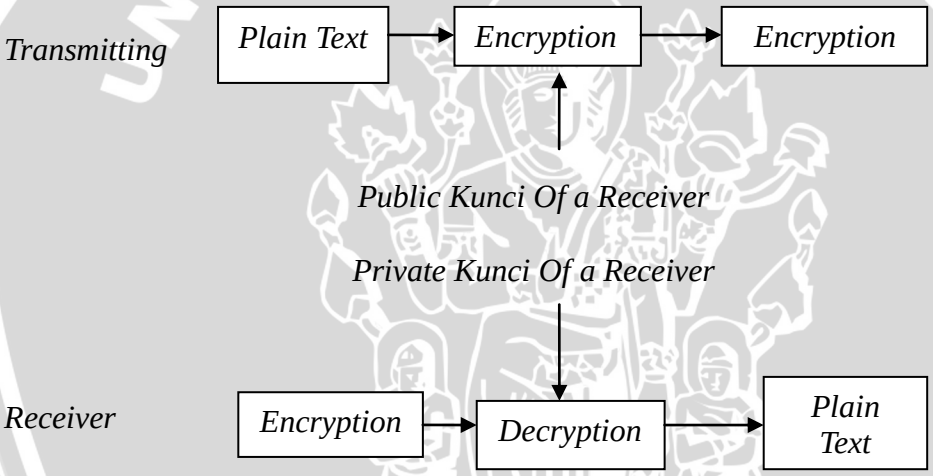
- Algoritma ini dirancang sehingga proses enkripsi/dekripsi membutuhkan waktu yang singkat.
- Ukuran kunci relatif lebih pendek.
- Algoritmanya bisa menghasilkan *cipher* yang lebih kuat.
- Autentikasi pengiriman pesan langsung diketahui dari *ciphertext* yang diterima, karena kunci hanya diketahui oleh pengirim dan penerima pesan saja.

Kelemahan teknik kriptografi simetris adalah:

- Kunci harus dikirim melalui saluran yang aman. Kedua entitas yang berkomunikasi harus menjaga kerahasiaan kunci ini.
- Kunci harus sering diubah, mungkin pada setiap sesi komunikasi [MUR-04].

b) Kriptografi Asimetris

Kriptografi asimetris atau biasa disebut kriptografi kunci publik dirancang sedemikian sehingga kunci yang digunakan untuk mengenkripsi dan mendekripsi berbeda. Gambar 2.3 memperlihatkan skema kriptografi asimetris yang menggunakan dua buah kunci.Sehingga kunci dekripsi tidak dapat dihitung dari kunci enkripsi.Kriptografi tersebut disebut *public-key* karena kunci enkripsi dapat dibuat secara *public*.Orang asing dapat menggunakan kunci enkripsi tersebut untuk mengenkripsi sebuah pesan, tetapi hanya orang tertentu dengan kunci dekripsi sepadan dapat mendekripsi pesan tersebut.Dalam sistem inikunci enkripsi sering disebut *public key* sedangkan kunci dekripsi sering disebut *private key* [KUR-04].



Gambar 2.3 Skema Kriptografi Asimetris

Kelebihanteknik kriptografi asimetri adalah:

- Hanya *Private key* yang harus benar-benar rahasia/aman.
- Sangat jarang untuk perlu merubah *public key* dan *private key*.

Kelemahanteknik kriptografi asimetri adalah:

- Ukuran kunci lebih besar dari pada algoritma kunci simetri.

Tidak adanya jaminan bahwa *public key* benar-benar aman [MUN-04].

2.1.4 Kriptografi Klasik

Sebelum komputer ada, kriptografi dilakukan dengan menggunakan pensil

dan kertas. Algoritma kr iptografi (cipher) yang digunakan saat itu, dinamakan juga algoritma klasik, adalah berbasis karakter, yaitu enkripsi dan dekripsi dilakukan pada setiap karakter pesan. Semua algoritma klasik termasuk ke dalam sistrm kriptografi simetris dan digunakan jauh sebelum kriptografi kunci publik ditemukan.

Kriptogarfi klasik memiliki beberapa ciri :

- Berbasis karakter
- Menggunakan pena dan kertas saja, belum ada computer
- Termasuk ke dalam kriptografi kunci simetris.

Tiga alasan mempelajari algoritma klasik :

- Memahami konsep dasar kriptografi
- Dasar algoritma kriptografi modern
- Memahami kelemahan sistem kode.

[ARI-08].

2.2. *Vigenere Cipher*

Vigenere Cipher merupakan algoritma kriptografi klasik. Operasi pada algoritma kriptografi klasik berbasis pada operasi karakter, sedangkan operasi pada algoritma kriptografi modern berbasis pada operasi bit. Dalam kriptografi klasik, Vigenere Cipher termasuk ke dalam cipher substitusi abjad majemuk, yang terbuat dari sejumlah cipher abjad tunggal, masing-masing dengan kunci yang berbeda.*Vigenere Cipher* telah berkali-kali diciptakan ulang dengan cukup bervariasi.Namun, metode aslinya digambarkan oleh Giovan Batista Belaso pada tahun 1553 seperti tertulis di dalam bukunya *La Cifra del Sig*. Giovan Batista Belaso.Meskipun demikian, *Vigenere Cipher* dipopulerkan oleh Blaise de Vigenere pada tahun 1586.

Vigenere Cipher menggunakan Bujur Sangkar *Vigenere* (Tabel 2.1) untuk melakukan enkripsi. Pada bujur sangkar tersebut, kolom paling kiri menyatakan huruf-huruf kunci, dan baris paling atas menyatakan plainteks

Tabel 2.1 Bujur Sangkar *Vigener*

	PLAIN TEXT									
K		A	B	C	D	.....	W	X	Y	Z

E Y	a	A	B	C	D	.....	W	X	Y	Z
	b	B	C	D	E	.....	X	Y	Z	A
	c	C	D	E	F	.....	Y	Z	A	B
	d	D	E	F	G	.....	Z	A	B	C
	e	E	F	G	H	.....	A	B	C	D
	f	F	G	H	I	.....	B	C	D	E
	g	G	H	I	J	.....	C	D	E	F
	h	H	I	J	K	.....	D	E	F	G
	i	I	J	K	L	.....	E	F	G	H
	j	J	K	L	M	.....	F	G	H	I
	k	K	L	M	N	.....	G	H	I	J
	l	L	M	N	O	.....	H	I	J	K
	m	M	N	O	P	.....	I	J	K	L
	n	N	O	P	Q	.....	J	K	L	M
	o	O	P	Q	R	.....	K	L	M	N
	p	P	Q	R	S	.....	L	M	N	O
	q	Q	R	S	T	.....	M	N	O	P
	r	R	S	T	U	.....	N	O	P	Q
	s	S	T	U	V	.....	O	P	Q	R
	t	T	U	V	W	.....	P	Q	R	S
	u	U	V	W	X	.....	Q	R	S	T
	v	V	W	X	Y	.....	R	S	T	U
	w	W	X	Y	Z	.....	S	T	U	V
	x	X	Y	Z	A	.....	T	U	V	W
	y	Y	Z	A	B	.....	U	V	W	X
	z	Z	A	B	C		V	W	X	Y

Bujur Sangkar *Vigenere* digunakan untuk mendapatkan cipherteks dengan menggunakan kunci yang telah ditentukan.Jika panjang kunci lebih pendek daripada panjang plainteks, maka kunci diulang penggunaannya (sistem periodik). Jika panjang kunci adalah

m, maka periodenya adalah m. Secara singkat, enkripsi dapat digambarkan sebagai berikut:

p (*plainteks*) : KRIPTOGRAFI

k (kunci) : LAMPIONLAMP

c (*cipherteks*) : VRUEBCTCARX

Penggunaan Bujur Sangkar *Vigenere* pada enkripsi serupa dengan penjumlahan (dalam desimal) plainteks dengan kunci, lalu modulo 26, sehingga dapat dirumuskan sebagai berikut:

Enkripsi: ci = E(pi) = (pi + ki) mod 26

Dekripsi: pi = D(ci) = (ci - ki) mod 26

Dekripsi *Vigenere Cipher* dengan menggunakan Bujur Sangkar *Vigenere* dilakukan dengan cara berkebalikan enkripsi, yaitu dengan menarik garis mendatar dari huruf kunci sampai ke huruf *cipherteks* yang dituju, kemudian dari huruf *cipherteks* tersebut, tarik garis vertikal ke atas sampai ke huruf plainteks [MUN-06].

2.2.1 Enkripsi Dengan *Vigenere Chiper* Yang Diperluas

Enkripsi dan Dekripsi dengan algoritma *Vigenere cipher* menggunakan kunci yang sama. Jika tidak maka proses dekripsi tidak akan mendapatkan image yang sama dengan aslinya. Kunci-kunci tersebut disebut dengan *Vigenere tableau*.Dalam implementasinya tabel tersebut dikembangkan dimana dengan nilai plain dari 0 sampai dengan 255. Tabel 2.2 dibawah adalah tabel *vigenere* yang telah dikembangkan :

Tabel 2.2 Bujur Sangkar *Vigenere* Yang Telah Dikembangkan

	PLAIN IMAGE PRIMER									
K		0	1	2	3	.....	252	253	254	255
E	0	0	1	2	3	.....	252	253	254	255
Y	1	1	2	3	4	.....	253	254	255	0
	2	2	3	4	5	.....	254	255	0	1
	3	3	4	5	6	.....	255	0	1	2
	4	4	5	6	7	.....	0	1	2	3
	5	5	6	7	8	.....	1	2	3	4
	6	6	7	8	9	.....	2	3	4	5
	7	7	8	9	10	.....	3	4	5	6
	8	8	9	10	11	.....	4	5	6	7
	9	9	10	11	12	.....	5	6	7	8
	10	10	11	12	13	.....	6	7	8	9
	11	11	12	13	14	.....	7	8	9	10
	12	12	13	14	15	.....	8	9	10	11
	.	.	.	.	.	.....	.	.	.	.
	.	.	.	.	.	.....	.	.	.	.
	.	.	.	.	.	.....	.	.	.	.
	.	.	.	.	.	.....	.	.	.	.
	.	.	.	.	.	.....	.	.	.	.
	.	.	.	.	.	.....	.	.	.	.
	.	.	.	.	.	.....	.	.	.	.
	250	251	252	253	254	.....	246	247	248	249
	251	252	253	254	255	.....	247	248	249	250

	252	253	254	255	0	.....	248	249	250	251
	253	254	255	0	1	.....	249	250	251	252
	254	255	0	1	2	.....	250	251	252	253
	255	0	1	2	3	.....	251	252	253	254

Keterangan Tabel 1:

- Angka pada baris pertama dengan arsiran adalah index nilai *pixel image* yang dikodekan (*plain image primer*).
- Angka pada kolom pertama dengan arsiran adalah kode kunci (*key*).
- Angka tanpa arsiran adalah hasil (cipher image primer).

Persamaan 2.1 merupakan rumus enkripsi yang digunakan untuk menghitung nilai *cipherimage*.

$$E_{ki}(a) = (a + k_i) \bmod 256 \qquad (2.1)$$

Keterangan :

$E_{ki}(a)$: *Cipher image primer*

a : *Index Plain image primer.*

k_i : *Penambahan posisi data dalam urutan nilai kunci.*

Sedangkan rumus yang digunakan untuk mendapatkan kembali *plainimage* yang telah terenkripsi (dekripsi) dinyatakan dengan persamaan 2.2.

$$E_{ki}(a) = (a - k_i) \bmod 256 \qquad (2.2)$$

Keterangan :

$E_{ki}(a)$: *Plain image primer*

a : *Index Cipher image primer.*

k_i : *Pengurangan posisi data dalam urutan nilai kunci.*[SUH-09].

2.3. Teori *Chaos*

2.3.1 Pengertian Teori *Chaos*

Secara matematik teori *chaos* menggambarkan tingkah laku dari sebuah system dinamik yang menunjukkan fenomena *chaos*.Salah satu karakteristik sistem chaosyaitu peka terhadap nilai awal (sering disebut dengan buterfly effect).Hasil darikesensitifan ini menunjukkan sebuah pertumbuhan yang eksponensial darikekacauan di dalam kondisi inisial, tingkah laku dari sistem chaotic ini munculsecara acak atau *random*.Ini tetap terjadi walaupun pada sistem ini bersifatdeterministik, yang berarti bahwa kejadian dinamik yang terjadi ditentukan oleh kondisi inisial mereka dan tidak mempunyai parameter acak.

Kelakuan seperti inidikenal dengan nama*deterministicchaos* atau *chaos*.Sistem *chaos* berguna untukpembangkitan bilangan acak dan bilangan acak dengan *chaos* tidak mempunyaiperiode.Hubungan yang sangat dekat antara chaos dan kriptografi membuat *chaos* sebagai dasar dari algoritma kriptografi yang aman untuk komunikasi dankriptografi.Teknik enkripsi berdasarkan chaos dianggap baik dalam praktek penggunaannya, karena didukung oleh 3 (tiga) teknik yaitu kecepatan, tingkat keamanan yang tinggi dan kompleksitasnya [SCA-98].

2.3.2 *Logistic Map* (Persamaan Logistik)

*Logistic Map*merupakan contoh pemetaan polinomial derajat dua, dan seringkali digunakan sebagai contoh bagaimana rumitnya sifat *chaos* (kacau) yang dapat muncul dari suatu persamaan yang sangat sederhana.Persamaan ini dipopulerkan oleh seorang ahli biologi yang bernama Robert May pada tahun 1976, melanjutkan persamaan logistik yang dikembangkan oleh Pierre Francois Verhulst.Secara matematis, persamaan logistik dapat dinyatakan dengan persamaan 2.3.

$$X_{n+1} = r\, X_n (1 - X_n) \qquad (2.3)$$

X_n : adalah nilai awal iterasi memiliki nilai antara 0 dan 1 ($0 \leq X \leq 1$), untuk tahun ke – n.

r : adalah laju pertumbuhan fungsi ($0 \leq X \leq 4$) .

Menurut [James Lampton] Fungsi konversi nilai chaos menjadi integer

dapat dilakukan dengan menggunakan fungsi pembulatan ke atas atau pembulatan ke bawah.Caranya, nilai *chaos*dikalikan dengan 10 berulangkali sampai ia mencapai panjang angka (size) yang diinginkan, selanjutnya potong hasil perkalian tersebut untuk mengambil bagian *integer*-nya saja. Secara matematis, nilai *chaos*x dikonversi ke *integer* dengan menggunakan persamaan 2.4.

$$T(x,size) = ||x*10^{count}|| \text{ , } x \neq 0 \qquad (2.4)$$

Keterangan :

T : Hasil konversi nilai chaos menjadi integer

X : Nilai *chaos*

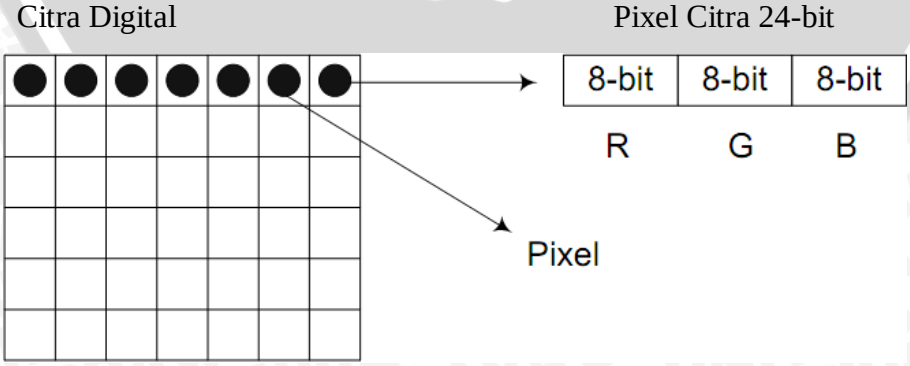
Size : Panjang angka

Count : Bilangan pangkat [LAM-TT].

2.4. Citra Digital

Citra adalah representasi atau deskripsi tentang suatu objek.Citra juga dapat diartikan sebagai objek pada bidang dua dimensi. Citra dapat direpresentasikan ke dalam citra analog dan citra digital. Citra analog dihasilkan dari sistem optik, misalnya mata manusia, kamera, sedangkan citra digital dihasilkan dari hasil digitisasi citra analog.

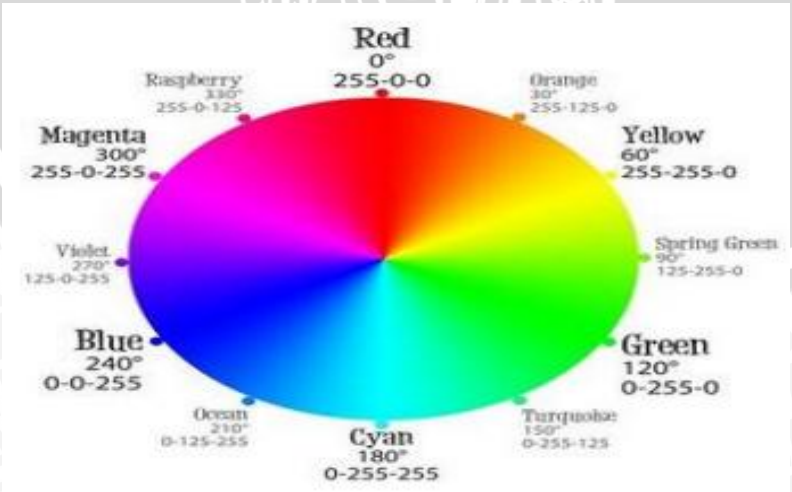
Citra digital adalah representasi numerik dari objek-objek. Citra digital dibentuk oleh sekumpulan angka dalam array dua dimensi. Tiap angka menggambarkan warna dari tiap titik dalam gambar sesuai dengan mode warna yang digunakan. Titik-titik ini disebut pixel yang merupakan singkatan dari picture element (elemen gambar). Gambar 2.4 menunjukkan citra digital.



Gambar 2.4 Citra Digital [MUN-92].

Citra digital dapat didefinisikan sebagai fungsi dua variabel, f(x,y), dimana x dan y adalah koordinat spasial dan nilai f(x,y) adalah intensitas citra pada koordinat tersebut, hal tersebut diilustrasikan pada Gambar 2.5. Teknologi dasar untuk menciptakan dan menampilkan warna pada citra digital berdasarkan pada penelitian bahwa sebuah warna merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau, dan biru (*Red, Green, Blue* - RGB).

RGB adalah suatu model warna yang terdiri dari merah, hijau, dan biru, digabungkan dalam membentuk suatu susunan warna yang luas. Setiap warna dasar, misalnya merah, dapat diberi rentang-nilai. Untuk monitor komputer, nilai rentangnya paling kecil = 0 dan paling besar = 255. Pilihan skala 256 ini didasarkan pada cara mengungkap 8 digit bilangan biner yang digunakan oleh mesin komputer. Dengan cara ini, akan diperoleh warna campuran sebanyak 256 x 256 x 256 = 1677726 jenis warna. Sebuah jenis warna, dapat dibayangkan sebagai sebuah vektor di ruang 3 dimensi yang biasanya dipakai dalam matematika, koordinatnya dinyatakan dalam bentuk tiga bilangan, yaitu komponen-x, komponen-y dan komponen-z. Misalkan sebuah vektor dituliskan sebagai r = (x,y,z). Untuk warna, komponen-komponen tersebut digantikan oleh komponen R(ed), G(reen), B(lue). Jadi, sebuah jenis warna dapat dituliskan sebagai berikut: warna = RGB(30, 75, 255). Putih = RGB (255,255,255), sedangkan untuk hitam= RGB(0,0,0) [AWC-96].



Gambar 2.5 Representasi Warna RGB Pada Citra Digital [AWC-96].

2.4.1 Struktur Citra Digital

Suatu citra digital berbentuk matriks, di mana elemen-elemen matriks dapat diakses melalui indeksnya, yaitu baris dan kolom. Gambar 2.6 menunjukan matriks citra digital berukuran NxM.Sebuah citra digital berukuran N x M, dengan keterangan sebagai berikut:

1. N = jumlah baris (panjang/tinggi matriks) → $0 \leq y \leq N-1$
2. M = jumlah kolom (lebar matriks) → $0 \leq x \leq M-1$
3. L = intensitas warna maksimal (derajat keabuan)
   → $0 \leq f(x,y) \leq L-1$

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

Gambar 2.6 Matriks citra digital berukuran NxM[MUN-04].

2.4.2 Citra Bitmap

Format BMP, disebut dengan *bitmap* adalah sebuah format citra yang digunakan untuk menyimpan citra bitmap digital. Pada citra berformat BMP (*bitmap*) yang tidak terkompresi, piksel citra disimpan dengan kedalaman warna 1, 4, 8, 16, 24, atau 32 bit per piksel. Terjemahan bebas bitmap adalah pemetaan bit. Artinya nilai intensitas piksel di dalam citra dipetakan ke sejumlah bit tertentu. Peta bit umumnya adalah 8, yang berarti setiap piksel panjangnya 8 bit. Delapan bit ini mempresentasikan nilai intensitas piksel. Dengan demikian ada sebanyak 2^8 =256 derajat keabuan, mulai dari 0 (00000000) sampai 255 (11111111).

Pada umumnya citra bitmap terdiri dari 4 blok data yaitu: *BMP header*, *Bit Information (DIB header)*, *Color Palette*, dan *Bitmap Data*. *BMP header* berisi informasi umum dari citra *bitmap* yang berada pada bagian awal file citra dan digunakan untuk mengidentifikasi citra.*Bit information* berisi informasi detail dari citra bitmap, yang akan digunakan untuk menampilkan citra pada layar. *Color palette* berisi informasi warna yang digunakan untuk indeks warna bitmap, dan

bitmap data berisi data citra yang sebenarnya, piksel per piksel.

Model ruang warna yang digunakan pada citra *bitmap* adalah RGB (*red, green, dan blue*).Sebuah ruang warna RGB dapat diartikan sebagai semua kemungkinan warna yang dapat dibuat dari tiga warna dasar *red*, *green*, dan *blue*. RGB sering digunakan di dalam sebagian besar aplikasi komputer karena dengan ruang warna ini tidak diperlukan transformasi untuk menampilkan informasi di layar monitor.[MUN-04].

Terdapat tiga macam citra dalam format BMP, adalah sebagai berikut:

1. **Citra biner**. Citra biner hanya memiliki dua nilai keabuan 0 dan 1. Oleh kerena itu 1 bit telah cukup untuk mempresentasikan nilai piksel.
2. **Citra hitam-putih** (*grayscale*).

Citra berwarna. Citra berwarna adalah citra yang lebih umum. Warna yang terlihat didalam citra bitmap merupakan kombinasi dari tiga komponen warna, yaitu : R (Red), G (Green) dan B (Blue). Pada citra 256 warna, setiap piksel memiliki panjang 8-bit, akan tetapi komponen RGBnya disimpan dalam tabel RGB yang disebut *palet.* Tabel 2.3 memperlihatkan panjang informasi palet untuk setiap versi *bitmap*.

Tabel 2.3 Panjang informasi *palet bitmap* berwarna

Citra *m* warna	*Palet bitmap*
Citra 16	64 *byte*
Citra 256	1024 *byte*
Citra 16,7 juta	0 *byte*

Format citra 8-*bit* dapat dilihat pada gambar 2.7. Format citra 4-*bit* (16 warna), hampir sama dengan format citra 8-*bit*. Pada citra 4-*bit* dan citra 8-*bit*, warna suatu piksel diacu dari tabel informasi palet *entry* ke-k (*k* merupakan nilai rentang 0-15 untuk citra 16 warna dan 0-155 untuk citra 256 warna). Sebagai contoh pada Gambar 2.7, piksel pertama bernilai 2, warna piksel pertama ini ditentukan oleh komponen RGB pada *palet* warna *entry* ke-2, yaitu R=14, G=13 dan B=16.piksel kedua serupa dengan piksel pertama. Piksel ketiga bernilai 1, warna ditentukan oleh komponen RGB pada *palet* warna *entry* ke-1, yaitu R=20, G=45 dan B=24. Demikian seterusnya untuk piksel-piksel lainnya. Khusus untuk

citra hitam-putih 8-*bit*, komponen R,G dan B suatu piksel bernilai sama dengan data bitmap piksel tersebut. Jadi piksel dengan nilai data bitmap 129, memiliki nilai R=129, G=129 dan B=129.

<header berkas>
<header*bitmap*>
<palet warna RGB>

	R	G	B
1	20	45	24
2	14	13	16
3	12	17	15
…			
256	46	78	25

<data *bitmap*>
2 2 1 1 1 3 5 . . .

Gambar 2.7 Format citra 8-bit[MUN-04].

Citra yang lebih kaya warna adalah citra 24-*bit*. Setiap piksel panjangnya 24-*bit*, karena setiap bit langsung menyatakan komponen warna merah (8-bit), komponen warna hijau (8-*bit*) dan komponen warna biru (8-*bit*). Citra 24-*bit* juga disebut citra 16 juta warna karena mampu menghasilkan 2^{24} =16.777.216 kombinasi warna. Contohnya seperti pada Gambar 2.8 berikut ini, dimana piksel pertama memiliki nilai R=20, G=19 dan B=21. Piksel kedua memiliki nilai R=24, G=23 dan B=24 dan demikian seterusnya.

<header berkas>
<header*bitmap*>
<data*bitmap*>
20 19 21 24 24 23 24 . . .

Gambar 2.8Format citra 24-bit [MUN-04].

2.5. Penyimpanan Citra

Menyimpan citra ke dalam media penyimpanan dalam bentuk digital memiliki bentuk yang beragam. Ada dua cara penyimpanan yang biasa dilakukan

oleh perangkat lunak yaitu bitmap dan *vector*. Dalam hal ini sering juga digunakan istilah program paint dan program *draw*.

Program paint atau program berbasis bitmap menyimpan citra sebagaimana ditampilkan di layar yaitu sebagai array dari *pixel-pixel*. Perubahan yang dilakukan pada citra dengan menggunakan program ini akan mengubah langsung tiap titik atau *pixel* pada citra. Kelebihan cara ini adalah kemudahannya untuk menampilkan gambar secara rinci dengan pola-pola yang kompleks atau gambar fotorealistik, yang tidak dapat dengan mudah direpresentasikan sebagai model matematika.

Program draw atau program berbasis vector menyimpan citra sebagai model matematika, dan setiap elemen citra disimpan secara terpisah. Perubahan yang dilakukan pada citra menggunakan program ini akan mengubah deskripsi matematika yang menyusun gambar dan program menghitung perubahan yang perlu pada warna-warna *pixel* secara tidak langsung. Kelebihan cara ini adalah kemampuannya untuk menciptakan gambar dalam resolusi yang berbeda tanpa kehilangan mutu gambar yang berarti[MUN-92].

2.6. Nilai Korelasi

Nilai korelasi digunakan untuk mengukur tingkat korelasi atau kemiripan antara dua buah piksel yang saling bertetanggaan pada citradigital. Pada korelasi mengukur tingkat kemiripan dengan membandingkannya secara horizontal, vertikal, diagonal kanan, dan diagonal kiri.Nilai korelasi dihitung dengan menggunakan Persamaan 2.5.

$$r = \frac{n\sum xy - \sum x \sum y}{\sqrt{[n\sum(x^2) - (\sum x)^2][n\sum(y^2) - (\sum y)^2]}} \quad (2.5)$$

Di mana :

r : Nilai korelasi

n : Jumlah pasangan piksel:

∑xy :Jumlah perkalian piksel x dengan piksel y

∑x : Jumlah data x

$\sum y$: Jumlah data y

$\sum x^2$: Jumlah kuadrat data x

$\sum y^2$: Jumlah kuadrat data *y*

Nilai korelasi berbanding terbalik dengan kualitas pada*chipherimage*. Semakin rendah nilai korelasi yang dimiliki *chipherimage* maka semakin baik kualitas yang dihasilkan dari proses enkripsi [YOU-08].

2.7. Nilai Entropy

Suatu citra digital bisa dipandang sebagai satu sumber piksel-piksel bebas. Nilai *entropy* ini memberikan batasan minimum banyaknya bit yang diperlukan untuk mengkodekan keluaran sumber informasi tersebut. *Entropi* digunakan untuk mengetahui keragaman dari intensitas citra. Nilai *entropy* besar untuk citra dengan transisi derajat keabuan merata dan bernilai kecil jika struktur citra tidak teratur. Sehingga semakin tinggi nilai *entropy* maka *cipherimage* yang dihasilkan semakin baik, dan sebaliknya jika nilai *entropy* semakin rendah maka *cipherimage* yang dihasilkan memiliki kualitas yang semakin rendah.Nilai *entropy* dihitung menggunakan Persamaan 2.6 [PRA-03].

$$He = -\sum_{k=0}^{G-1} P(k).\log2\,(P(k))_{(bit/symbol)} \quad (2.6)$$

Di mana :

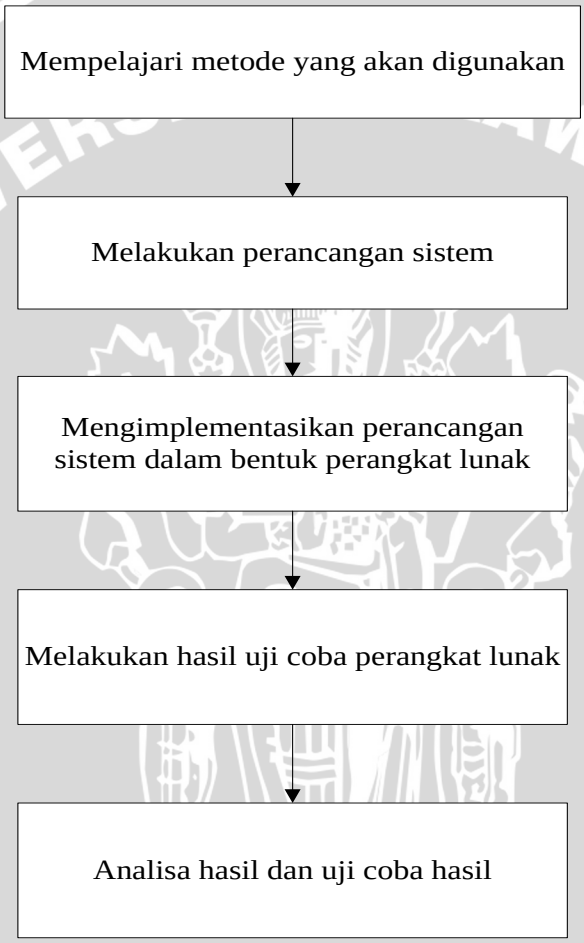
H_e : Nilai *entropy*

G : Derajat keabu-abuan citra masukkan (dari 0-255)

P(k) : Probabilitas symbol ke-k[PRA-03].

BAB III
METODE PENELITIAN

Bab ini berisi penjelasan mengenai metode dan langkah-langkah perancangan sistem yang dilakukan dalam penelitian. Langkah-langkah yang akan dilakukan ditunjukan pada gambar 3.



Gambar 3.1 Langkah-Langkah Penelitian

Berdasarkan gambar 3.1, langkah langkah yang dilakukan dalam penelitian ini adalah sebagai berikut :

1. Melakukan studi literatur mengenai karakteristik dan struktur citra digital dengan *format bitmap*, penggunaan metode *vigenere cipher*, dan juga

pembangkitan kunci secara acak menggunakan salah satu teori chaos yaitu *Logistic Map*.

2. Melakukan perancangan sistem.
3. Mengimplementasikan dperancangan sistem yang telah dibuat kebentuk perangkat lunak yang dapat melakukan proses enkripsi dan dekripsi citra digital.
4. Melakukan uji coba terhadap perangkat lunak yang telah dibuat.

Melakukan Menganalisa hasil uji dan mengevaluasi hasil tersebut.

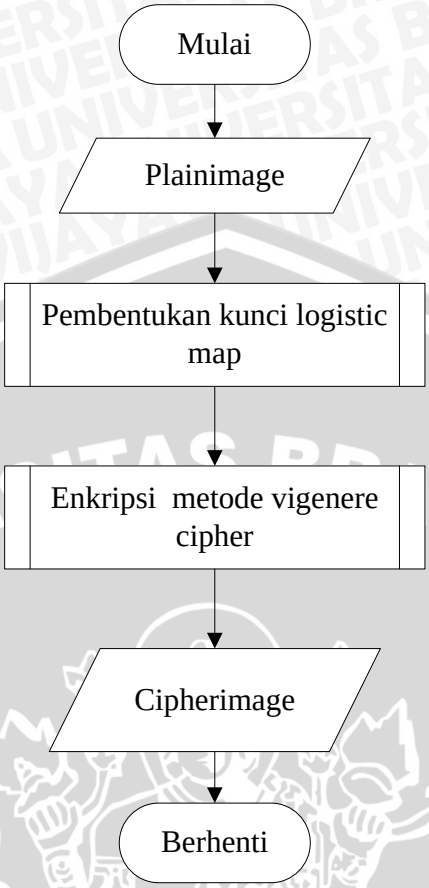
3.1.AnalisaUmum

3.1.1.DeskripsiUmumSistem

Perangkat lunak yang dibuat merupakan perangkat lunak yang digunakanuntuk menjaga keamanan data berupa citra digital melalui proses enkripsi agar citra yang disembunyikan tidak terlihat seperti aslinya dan juga proses pengembalian citra ke bentuk semula yaitu proses dekripsi. Perangkat lunak akan mengimplementasikan tehnik kriptografi dengan metode *vigenere cipher* dan juga teori chaos yaitu *Logistic Map*. Pada proses enkripsi untuk mengamankan citra digital, perangkat lunak akan memasukan kunci dan citra digital asli (*plainimage*) yang akan diubah menjadi *cipherimage*. Pembentukan kunci dilakukan dengan pembangkitan yaitu melakukan iterasi terhadap kunci yang telah dimasukan sehingga mendapatkan kunci yang acak dan bervariaasi.Proses – proses untuk melakukan enkripsi adalah sebagai berikut:

1. Memasukkan citra digital.
2. Pembentukan kunci dengan cara pembangkitan.
3. Enkripsi menggunakan metode *vigenere cipher*.
4. Penyusunan kembali piksel – piksel yang dienkripsi menjadi citra digital.

Flowchart proses enkripsi citra digital ditunjukkan pada Gambar 3.2.

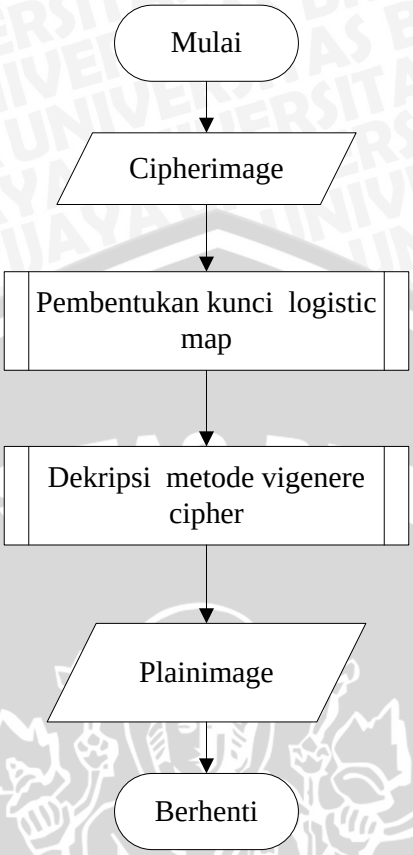


Gambar 3.2Flowchart Enkripsi Citra Digital

Untuk proses pengembalian data (dekripsi), perangkat lunak menerima inputan berupa *cipherimage*yang akan didekripsikan menjadi *plainimage* agar informasi yang ada sebelumnya dapat terlihat. Langkah-langkah proses dekripsi hampir sama dengan enkripsi, setiap piksel *cipherimage* akan disubtitusi dengan kunci yang telah dibangkitkan.Kemudianpiksel *cipherimage* hasil enkripsi disusun kembali membentuk *plainimage.*Proses – proses untuk melakukan dekripsi adalah sebagai berikut:

1. Memasukkan *cipherimage* hasil enkripsi metode *Vigenere Cipher.*
2. Pembentukan kunci dengan cara pembangkitan.
3. Dekripsi menggunakan metode *vigenere cipher*.
4. Penyusunan kembali piksel – piksel yang dienkripsi menjadi citra digital.

Flowchart proses dekripsi citra digital ditunjukkan pada Gambar 3.3.



Gambar 3.3*Flowchart* Dekripsi Citra Digital

3.1.2. Perancangan Sistem

Perangkat lunak yang akan dibuat memiliki tiga proses utama, yaitu proses pembentukan kunci, proses enkripsi dan proses dekripsi.

3.1.2.1 Proses Pembentukan Kunci

Pembentukan kunci dilakukan dengan cara membangkitkan menggunakan metode *logistic map*. Kunci digunakan untuk menjaga citra yang telah terenkripsi agar saat akan dikembalikan ke bentuk aslinya kata kunci yang dimasukan harus cocok. Sehingga inputan pada perangkat lunak ini selain citra digital adalah sejumlah karakter yang akan dibangkitkan menjadi kunci.

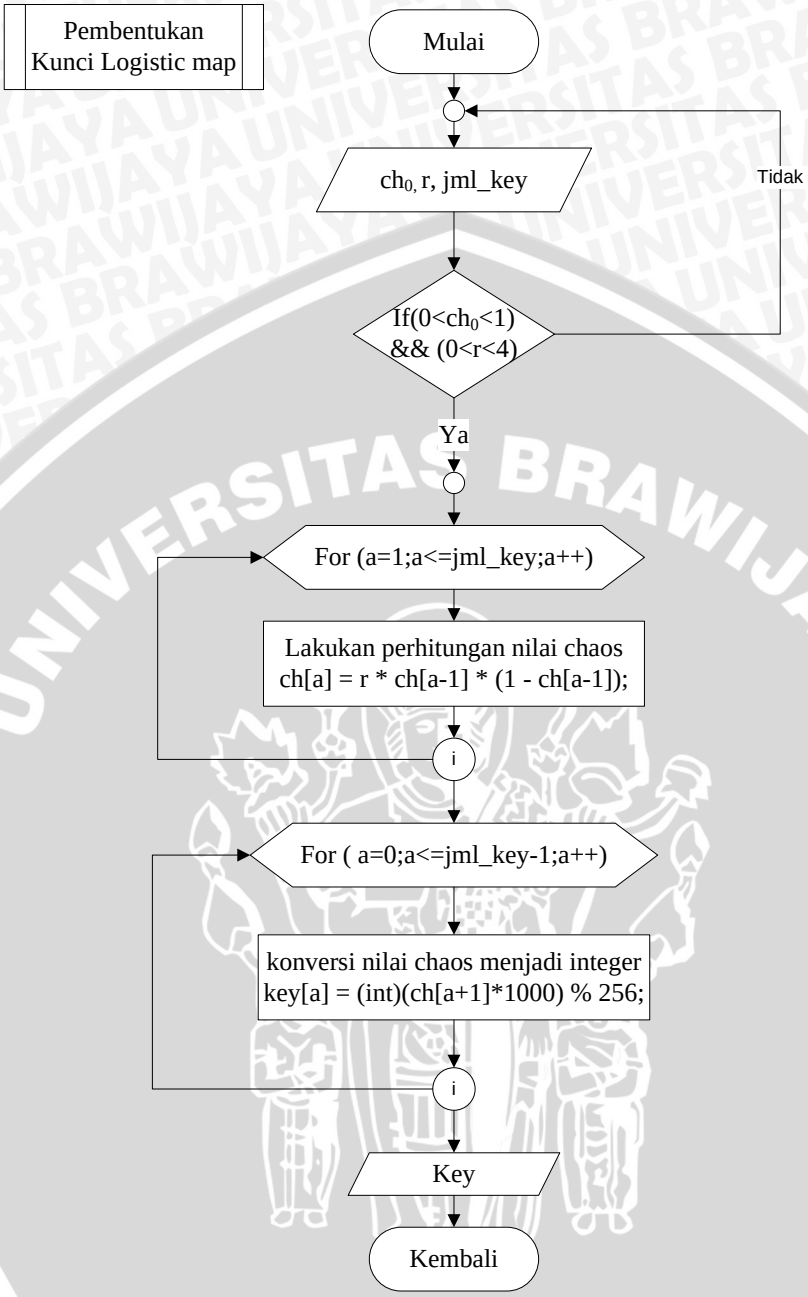
Langkah-langkah dalam proses pembentukan kunci adalah sebagai berikut:

1. Masukan nilai awal iterasi(X_0), laju pertumbuhan (r) dan jumlah kunci yang akan dibangkitkan.

2. Nilai awal iterasi memiliki nilai antara ($0 < x < 1$) dan nilai r antara ($0 < x < 4$).
3. Setelah didapatkan nilai $X_{0,}$ r, dan jumlah kunci maka hitung nilai chaos pada iterasi pertama menggunakan rumus

   $X_{n+1} = r\ X_n\ (1 - X_{n)})$, hasil dari nilai chaos pertama digunakan untuk menghitung nilai *chaos* pada iterasi kedua dan selanjutnya sampai jumlah kunci yang ingin dibangkitkan.
4. Karena nilai-nilai *chaos* tidak dapat langsung dioperasi-modulokan dengan *plainimage* karena masih berbentuk bilangan riil antara 0 dan 1 maka nilai-nilai *chaos* tersebut dikonversi ke nilai *integer* dengan fungsi pemotongan yang diusulkan James Lampton. Caranya adalah dengan mengalikan nilai *chaos* (x) dengan 10 berulangkali sampai ia mencapai panjang angka (*size*) yang diinginkan, selanjutnya potong hasil perkalian tersebut untuk mengambil bagian *integer*-nya saja.
5. Setelah itu akan didapatkan nilai – nilai kunci yang digunakan untuk melakukan proses enkripsi.

Berikut *flowchart* untuk pembentukan kunci yang ditunjukkan Gambar 3.4.



Gambar 3.4*Flowchart* Pembentukan Kunci

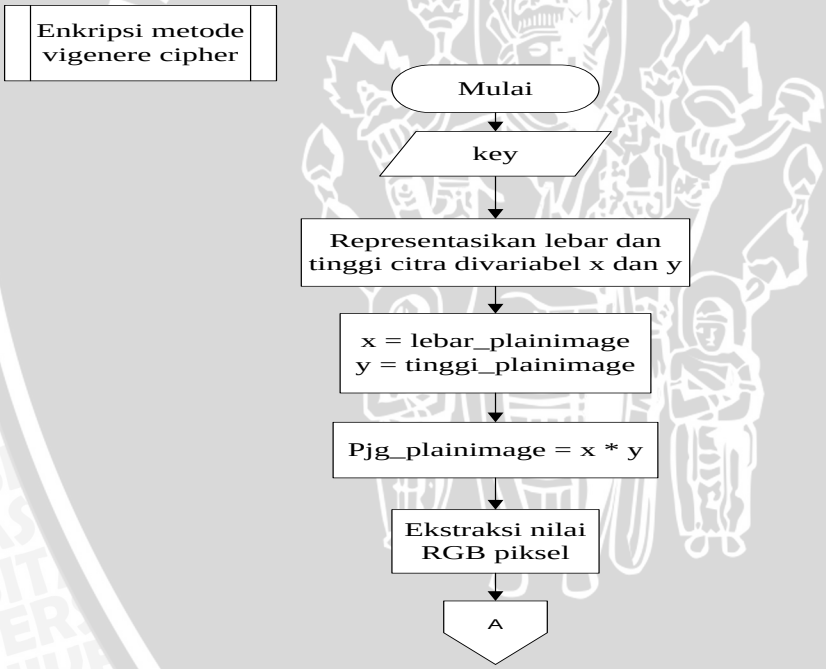
3.1.2.2 Proses Enkripsi Menggunakan Metode *Vigenere Cipher*

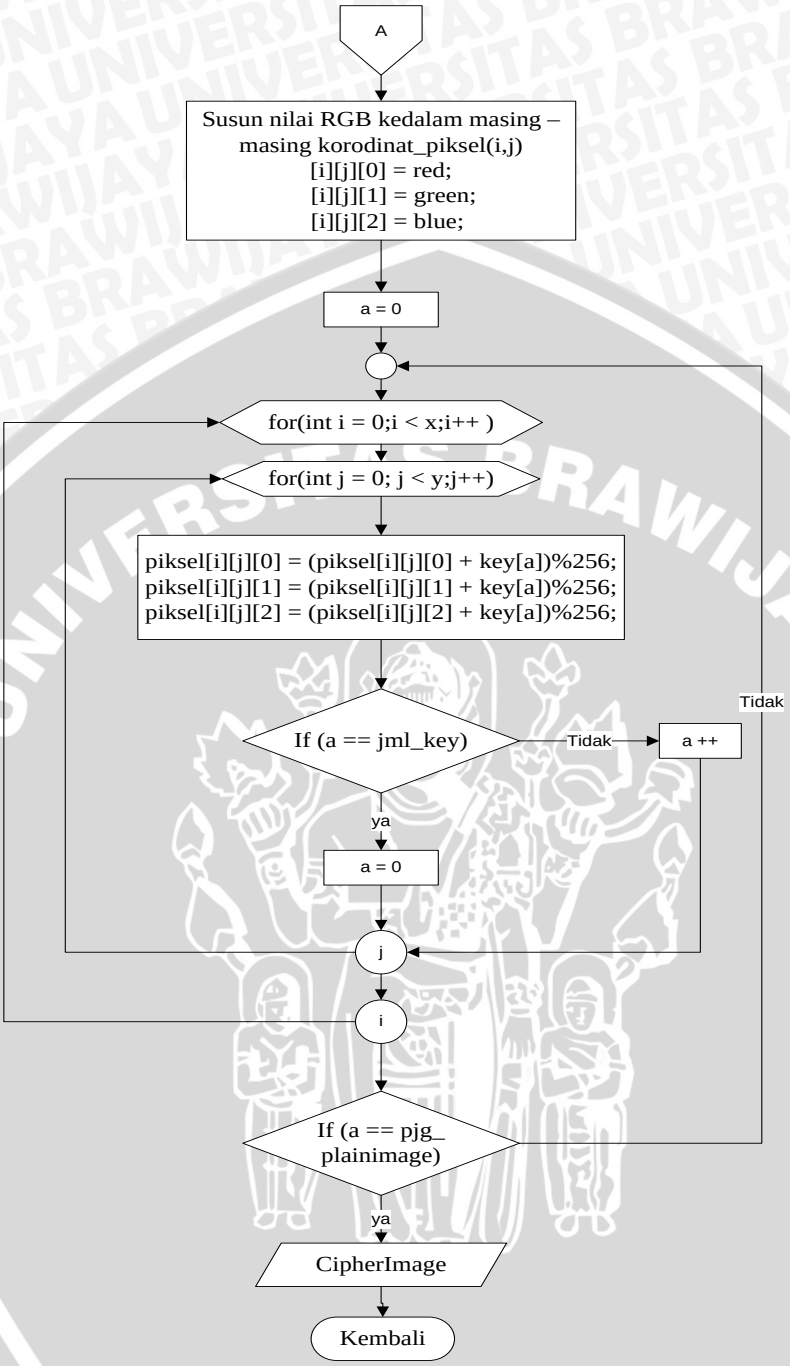
Pada proses ini, *plainimage* dienkripsi dengan menggunakan metode *Vigenere Cipher* untuk mendapatkan *cipherimage.* Langkah-langkah dalam proses enkripsi adalah sebagai berikut:

1. Baca *plainimage* dari inputan citradan kunciyang telah didapatkan dari hasil pembangkitan.

2. Representasikan lebar dan tinggi citra divariabel x dan y.
3. Lakukan ekstraksi nilai RGB piksel.
4. Kemudian susun nilai RGB kedalam masing – masing korodinat piksel (i,j) setelah itu disimpan menjadi array integer dan didapatkan data warna RGB dari semua piksel citra.
5. Setelah didapatkan data warna RGB dari semua piksel lalu setiap nilai RGB dalam koordinat piksel dienkripsi dengan nilai kunci. Jika semua data warna RGB telah dienkripsi dengan kunci maka akan didapatkan data warna RGB baru (*cipherimage*).
6. Hasil akhir dari proses enkripsi adalah berupa citra digital yang telah disembunyikan informasinya (*cipherimage)*.

Flowchart tahapan proses enkripsi ditunjukan Gambar 3.5.





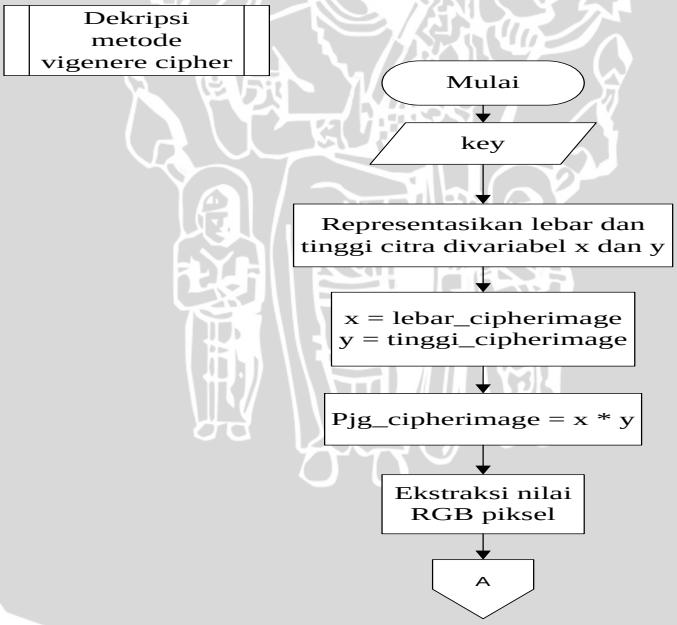
Gambar 3.5 *Flowchat* Enkripsi *Vigenere Cipher*

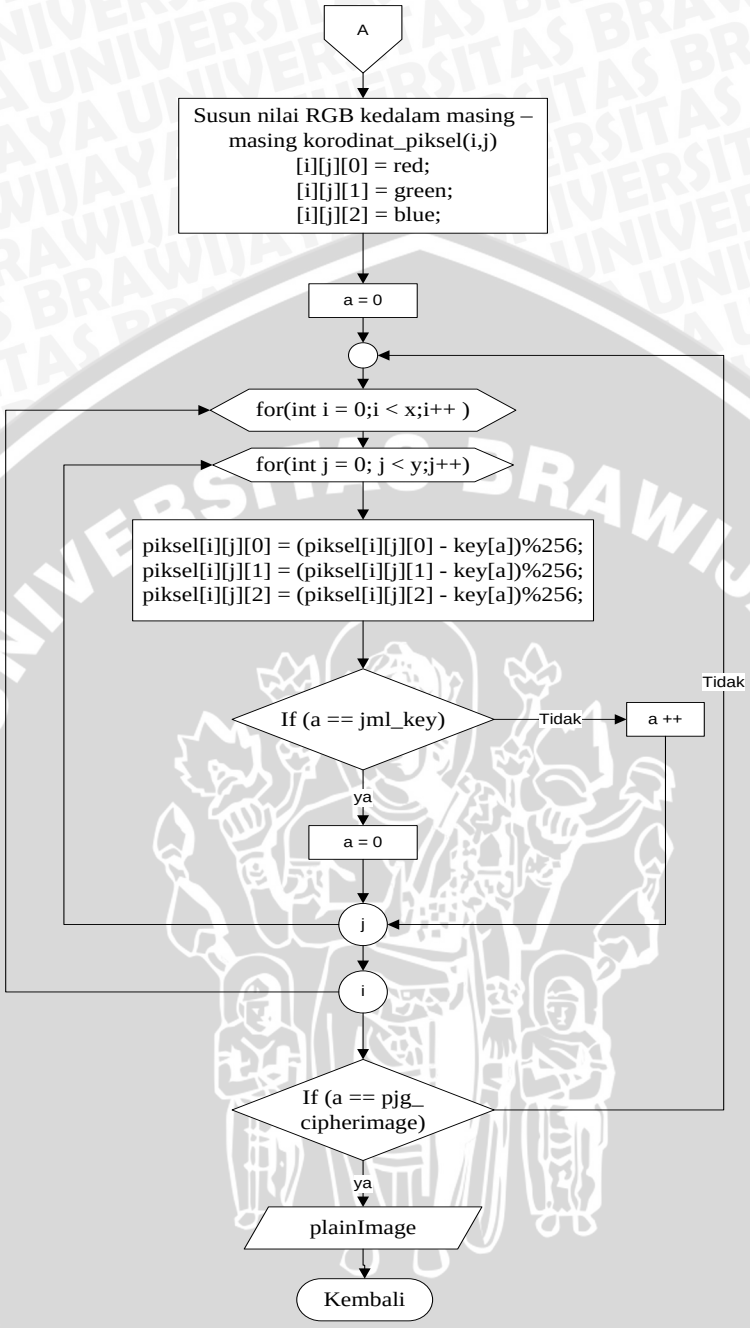
3.1.2.3 Proses Dekripsi Menggunakan Metode *Vigenere Cipher*

Pada proses dekripsi, hasil citra enkripsi atau *cipherimage*akan didenkripsi dengan menggunakan metode *Vigenere Cipher* untuk mendapatkan citra seperti semula (*plainimage*). Langkah-langkah dalam proses dekripsi adalah sebagai berikut:

1. Baca hasil citra enkripsi atau *cipherimage* dari inputan citradan kunciyang telah didapatkan dari hasil pembangkitan.
2. Representasikan lebar dan tinggi citra divariabel x dan y.
3. Lakukan ekstraksi nilai RGB piksel.
4. Kemudian susun nilai RGB kedalam masing – masing korodinat piksel (i,j) setelah itu disimpan menjadi array integer dan didapatkan data warna RGB dari semua piksel citra.
5. Setelah didapatkan data warna RGB dari semua piksel lalu setiap nilai RGB dalam koordinat piksel didekripsi dengan nilai kunci. Jika semua data warna RGB telah didekripsi dengan kunci yang sama seperti proses enkripsi maka akan didapatkan citra seperti semula (*plainimage*).
6. Hasil akhir dari proses dekripsi adalah berupa citra digital yang mempunyai informasi seperti semula (*plainimage)*.

Flowchart tahapan proses dekripsi ditunjukan Gambar 3.6.

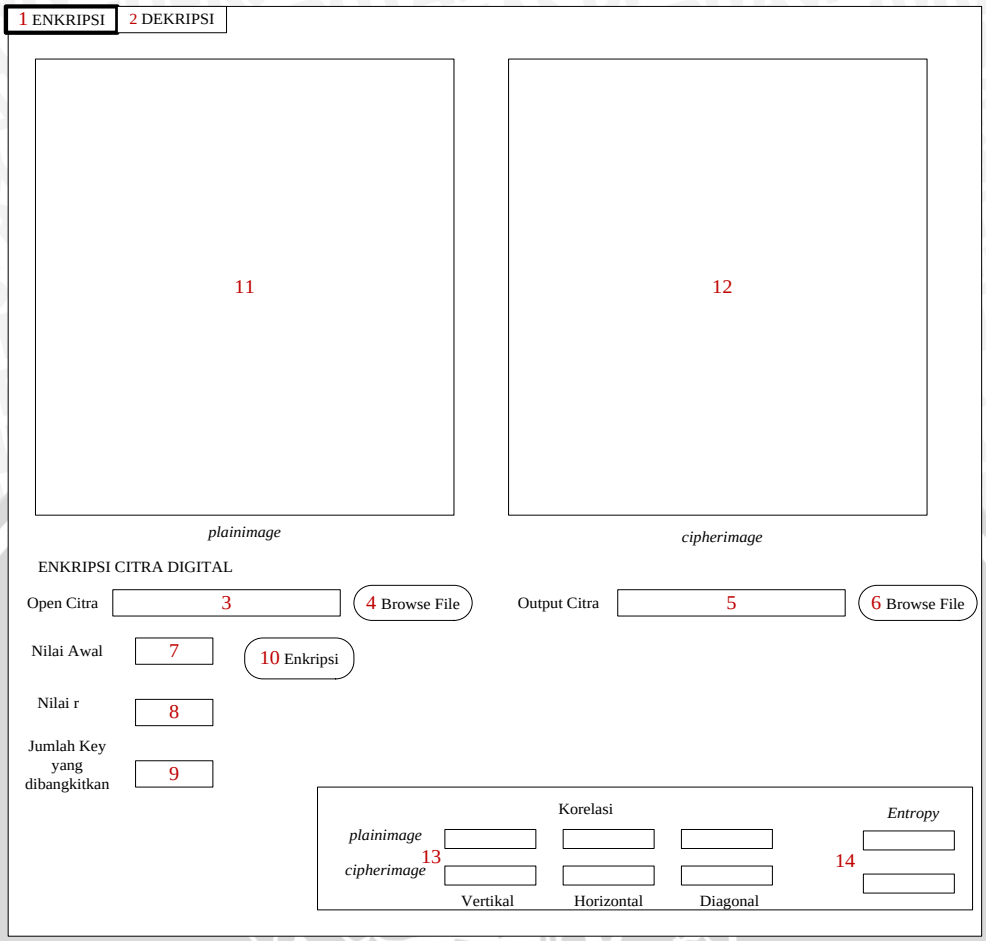




Gambar 3.6 *Flowchat* Dekripsi *Vigenere Cipher*

3.2.PerancanganAntarmuka

Antarmuka atau *interface* pada sistem ini terdiri dari dua bagian utama, yaitu bagian yang berfungsi untuk enkripsi citra digital dan bagian untuk dekripsi citra digital. Untuk bagian antarmuka enkripsi citra digital ditunjukan pada gambar 3.7



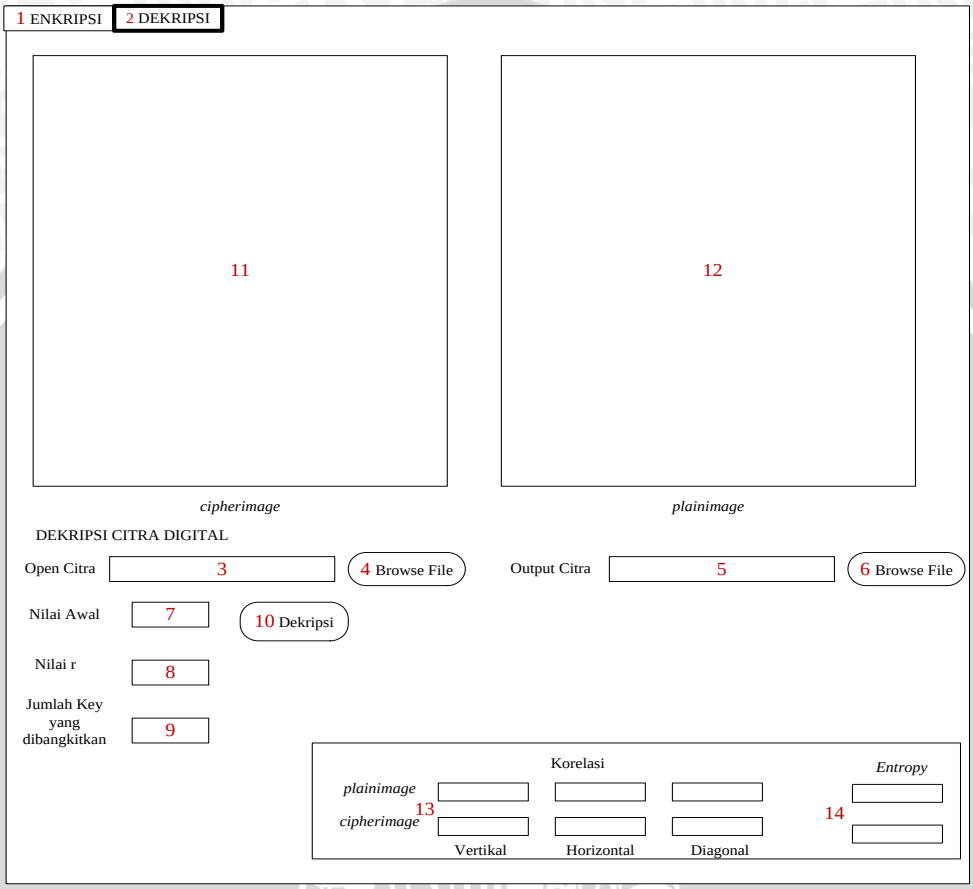
Gambar 3.7 Perancangan antarmuka enkripsi citra digital

Keterangan antarmuka enkripsi citra digital pada gambar 3.7 terdiri dari :

1. Tombol untuk menu enkripsi.
2. Tombol untuk menu dekripsi.
3. *Textfield* yang akan menunjukan direktori dari input citra digital.
4. Tombol untuk input citra digital.
5. *Textfield* yang akan menunjukan direktori dari output citra digital.
6. Tombol untuk output citra digital.
7. *Textfield* sebagai masukan nilai awal untuk pembentukan kunci.
8. *Textfield* sebagai masukan nilai r (laju pertumbuhan) untuk pembentukan kunci.
9. *Textfield* sebagai masukan jumlah kunci yang dibangkitkan.
10. Tombol mulai proses enkripsi .
11. Area untuk memuat citra asli untuk dienkripsi.

12. Area untuk memuat citra hasil enkripsi.
13. *Field* Nilai korelasi.
14. *Field* Nilai *entropy*

Sedangkan untuk bagian antarmuka dekripsi citra digital pada gambar 3.8.



Gambar 3.8 Perancangan antarmuka dekripsi citra digital

Keterangan antarmuka dekripsi citra digital pada gambar 3.8 terdiri dari :

1. Tombol untuk menu enkripsi.
2. Tombol untuk menu dekripsi.
3. *Textfield* yang akan menunjukan direktori dari input citra digital.
4. Tombol untuk input citra digital.
5. *Textfield* yang akan menunjukan direktori dari output citra digital.
6. Tombol untuk output citra digital.
7. *Textfield* sebagai masukan nilai awal untuk pembentukan kunci.

8. *Textfield* sebagai masukan nilai r (laju pertumbuhan) untuk pembentukan kunci.
9. *Textfield* sebagai masukan jumlah kunci yang dibangkitkan.
10. Tombol mulai proses dekripsi .
11. Area untuk memuat citra yang akan didekripsi.
12. Area untuk memuat citra hasil dekripsi.
13. *Field* Nilai korelasi.
14. *Field* Nilai *entropy.*

3.3 Perhitungan Manual

Pada perhitungan manual ini akan dilakukan dua proses perhitungan yaitu perhitungan pembentukan kunci dan perhitungan untuk enkripsi citra digital. Pada perhitungan manual ini contoh yang diambil adalah input berupa citra BMP 24 bit berukuran 4x4 piksel.

3.3.1 Perhitungan Proses Pembentukan Kunci

Pada perhitungan proses pembentukan kunci menggunakan persamaan $X_{n+1} = r\, X_n (1 - X_n)$ dan persamaan $T(x,size) = ||x*10^{count}||$, $x \neq 0$ yang diusulkan oleh James Lampton yang telah dijelaskan pada bab 2.3.2.

Berikut merupakan langkah-langkah pembentukan kunci:

1. Dilakukan inisialisasi nilai awal (*seed*) persamaan iterasi yaitu $X_0 = 0.75$, nilai r (laju pertumbuhan) = 3.8 , x_i (jumlah iterasi) = 9, dan nilai count = 3 untuk konversi nilai *chaos* ke integer.
2. Hitung nilai iterasi pertama sampai jumlah iterasi yang telah ditetapkan.
   - Iterasi pertama $X_0 = 0.75$

     $X_1 = r\, X_0 (1 - X_0)$

     $X_1 = (3.8) *(0.75)* (1 - 0.75)$

     $X_1 = (2.85)*(0.25)$

     $X_1 = 0.7125$
   - Iterasi kedua $X_1 = 0.7125$

     $X_2 = r\, X_1 (1 - X_1)$

$X_2 = (3.8) *(0.7125)* (1 – 0.7125)$

$X_2 = (2.7075)*(0.2875)$

$X_2 = 0.7784$

- Iterasi ketiga $X_2 = 0.7784$

  $X_3 = r\ X_2\ (1 – X_2)$

  $X_3 = (3.8) *(0.7784)* (1 – 0.7784)$

  $X_3 = (2.95792)*(0.2216)$

  $X_3 = 0.6554$

- Iterasi keempat $X_3 = 0.6554$

  $X_4 = r\ X_3\ (1 – X_3)$

  $X_4 = (3.8) *(0.6554)* (1 – 0.6554)$

  $X_4 = (2.49052)*(0.3446)$

  $X_4 = 0.858233192$

- Iterasi kelima $X_4 = 0.858233192$

  $X_5 = r\ X_4\ (1 – X_4)$

  $X_5 = (3.8) *(0.858233192)* (1 – 0.858233192)$

  $X_5 = (3.26128613)*(0.141766808)$

  $X_5 = 0.462342125$

- Iterasi keenam $X_5 = 0.462342125$

  $X_6 = r\ X_5\ (1 – X_5)$

  $X_6 = (3.8) *(0.462342125)* (1 – 0.462342125)$

  $X_6 = (1.756900075)*(0.537657875)$

  $X_6 = 0.944611161$

- Iterasi ketujuh $X_6 = 0.944611161$

  $X_7 = r\ X_3\ (1 – X_3)$

  $X_7 = (3.8) *(0.944611161)* (1 – 0.944611161)$

  $X_7 = (3.589522411)*(0.055388839)$

  $X_7 = 0.198819479$

- Iterasi kedelapan $X_7 = 0.198819479$

  $X_8 = r\ X_7\ (1 – X_7)$

  $X_8 = (3.8) *(0.198819479)* (1 – 0.198819479)$

  $X_8 = (0.75551402)*(0.801180521)$

$X_8 = 0.605303116$

- Iterasi kesembilan $X_8 = 0.605303116$

  $X_9 = r\ X_8\ (1 - X_8)$

  $X_9 = (3.8)$ *(= 0.605303116)* $(1 - 0.605303116)$

  $X_9 = (2.300151841)*(0.394696884)$

  $X_9 = 0.907862764$

3. Konversi nilai *chaos* ke integer agar barisan nilai chaotik dapat dipakai untuk enkripsi dan dekripsi.

   Nilai *chaos* yang didapatkan dari perhitungan iterasi adalah sebagai berikut :

   - $X_1 = 0.7125$
   - $X_2 = 0.7784$
   - $X_3 = 0.6554$
   - $X_4 = 0.858233192$
   - $X_5 = 0.462342125$
   - $X_6 = 0.944611161$
   - $X_7 = 0.198819479$
   - $X_8 = 0.605303116$
   - $X_9 = 0.907862764$

   Dari nilai *chaos* tersebut dikonversi ke integer.

   - $X_1 = 0.7125$

     $0.7125 \times 10^3 = 712.5$

     kemudian ambil bagian integer-nya dengan

     $||712.5|| = 712$

     kunci = 712 mod 256

     = 200

   - $X_2 = 0.7784$

     $0.7784 \times 10^3 = 778.4$

     kemudian ambil bagian integer-nya dengan

     $||778.4|| = 778$

     kunci = 778 mod 256

     = 10

   - $X_3 = 0.6554$

$0.6554 \times 10^3 = 655.4$

kemudian ambil bagian integer-nya dengan

||655.4|| = 655

kunci = 655 mod 256

= 143

- $X_4 = 0.858233192$

$0.858233192 \times 10^3 = 858.233192$

kemudian ambil bagian integer-nya dengan

||858.233192|| = 858

kunci = 858 mod 256

= 90

- $X_5 = 0.462342125$

$0.462342125 \times 10^3 = 462.342125$

kemudian ambil bagian integer-nya dengan

||462.342125|| = 462

kunci = 462 mod 256

= 206

- $X_6 = 0.944611161$

$0.944611161 \times 10^3 = 944.611161$

kemudian ambil bagian integer-nya dengan

||944.611161|| = 944

kunci = 944 mod 256

= 176

- $X_7 = 0.198819479$

$0.198819479 \times 10^3 = 198.819479$

kemudian ambil bagian integer-nya dengan

||198.819479|| = 198

kunci = 198 mod 256

= 198

- $X_8 = 0.605303116$

$0.605303116 \times 10^3 = 605.303116$

kemudian ambil bagian integer-nya dengan

||605.303116|| = 605

kunci = 605 mod 256

= 93

- X_9 = 0.907862764

  $0.907862764 \times 10^3 = 907.862764$

  kemudian ambil bagian integer-nya dengan

  ||907.862764|| = 907

  kunci = 907 mod 256

  = 139

4. Didapatkan kunci untuk melakukan enkripsi dan dekripsi yaitu sebagai berikut :

$k_1 = 200$	$k_6 = 176$
$k_2 = 10$	$k_7 = 198$
$k_3 = 143$	$k_8 = 93$
$k_4 = 90$	$k_9 = 139$
$k_5 = 206$	

3.3.2 Perhitungan Proses Enkripsi *Vigenere Cipher*

Citra uji yang digunakan dalam perhitungan manual proses enkripsi ini adalah citra BMP 24 bit 4x4piksel dan kunci yang telah dihasilkan dari proses pembentukan kunci.Citra uji yang digunakan ditunjukan pada gambar 3.9.

Gambar 3.9 Citra BMP 24 bit 4x4 *piksel*

Citra uji yang memiliki format BMP 24 bit dapat diketahui nilai RGB dari setiap pikselnya. Nilai RGB dari setiap piksel yang didapat kemudian dienkripsi. Berikut nilai RGB setiap piksel pada citra uji yang ditunjukan pada Tabel 3.1

Tabel 3.1 Nilai RGB Citra Uji

Piksel	R	G	B
Piksel (0.0)	155	141	125
Piksel (0.1)	172	158	137
Piksel (0.2)	127	136	136
Piksel (0.3)	105	93	111
Piksel (1.0)	161	154	143
Piksel (1.1)	89	85	99
Piksel (1.2)	126	113	109
Piksel (1.3)	97	91	98
Piksel (2.0)	83	62	82
Piksel (2.1)	118	99	108
Piksel (2.2)	152	146	143
Piksel (2.3)	73	84	106
Piksel (3.0)	211	194	169
Piksel (3.1)	190	186	168
Piksel (3.2)	197	184	159
Piksel (3.3)	78	74	90

Nilai RGB dari setiap piksel dienkripsi dengan cara subtitusi setiap nilai RGB dari semua piksel.

- Piksel (0.0) dengan $k_1 = 200$

  Red : 155 + 200 mod 256 = 99

Green : 141 + 200 mod 256 = 85

Blue : 125 + 200 mod 256 = 69

- Piksel (0.1) dengan $k_2 = 10$

  Red : 172 + 10 mod 256 = 182

  Green : 158 + 10 mod 256 = 168

  Blue : 137 + 10 mod 256 = 147

- Piksel (0.2) dengan $k_3 = 143$

  Red : 127 + 143 mod 256 = 14

  Green : 136 + 143 mod 256 = 23

  Blue : 136 + 143 mod 256 = 23

- Piksel (0.3) dengan $k_4 = 90$

  Red : 105 + 90 mod 256 = 195

  Green : 93 + 90 mod 256 = 183

  Blue : 111 + 90 mod 256 = 201

- Piksel (1.0) dengan $k_5 = 206$

  Red : 161 + 206 mod 256 = 111

  Green : 154 + 206 mod 256 = 104

  Blue : 143 + 206 mod 256 = 93

- Piksel (1.1) dengan $k_6 = 176$

  Red : 89 + 176 mod 256 = 9

  Green : 85 + 176 mod 256 = 5

  Blue : 99 + 176 mod 256 = 19

- Piksel (1.2) dengan $k_7 = 198$

  Red : 126 + 198 mod 256 = 68

  Green : 113 + 198 mod 256 = 55

  Blue : 109 + 198 mod 256 = 51

- Piksel (1.3) dengan $k_8 = 93$

  Red : 97 + 93 mod 256 = 190

  Green : 91 + 93 mod 256 = 184

  Blue : 98 + 93 mod 256 = 191

- Piksel (2.0) dengan $k_9 = 139$

  Red : 83 + 139 mod 256 = 222

Green : 62 + 139 mod 256 = 201

Blue : 82 + 139 mod 256 = 221

- Piksel (2.1) dengan $k_1 = 200$

  Red : 118 + 200 mod 256 = 62

  Green : 99 + 200 mod 256 = 43

  Blue : 98 + 200 mod 256 = 42

- Piksel (2.2) dengan $k_2 = 10$

  Red : 152 + 10 mod 256 = 162

  Green : 146 + 10 mod 256 = 156

  Blue : 143 + 10 mod 256 = 153

- Piksel (2.3) dengan $k_3 = 143$

  Red : 73 + 143 mod 256 = 216

  Green : 84 + 143 mod 256 = 227

  Blue : 106 + 143 mod 256 = 249

- Piksel (3.0) dengan $k_4 = 90$

  Red : 211 + 90 mod 256 = 45

  Green : 194 + 90 mod 256 = 28

  Blue : 169 + 90 mod 256 = 3

- Piksel (3.1) dengan $k_5 = 206$

  Red : 190 + 206 mod 256 = 140

  Green : 186 + 206 mod 256 = 136

  Blue : 168 + 206 mod 256 = 118

- Piksel (3.2) dengan $k_6 = 176$

  Red : 197 + 176 mod 256 = 117

  Green : 184 + 176 mod 256 = 104

  Blue : 159 + 176 mod 256 = 79

- Piksel (3.3) dengan $k_7 = 198$

  Red : 78 + 198 mod 256 = 20

  Green : 74 + 198 mod 256 = 16

  Blue : 90 + 198 mod 256 = 32

Perubahan nilai RGB dari setiap piksel dari citra digital hasil enkripsi

dapat dilihat pada tabel 3.2. Berikut merupakan perubahan yang terjadi setelah proses enkripsi, *plainimage* diubah ke dalam bentuk *cipherimage*.

Tabel 3.2 Perubahan Nilai RGB Citra Hasil Enkripsi

Piksel	***Plainimage***			***Cipherimage***		
	Nilai RGB			**Nilai RGB**		
	R	**G**	**B**	**R**	**G**	**B**
Piksel (0.0)	155	141	125	99	85	69
Piksel (0.1)	172	158	137	182	168	147
Piksel (0.2)	127	136	136	14	23	23
Piksel (0.3)	105	93	111	195	183	201
Piksel (1.0)	161	154	143	111	104	93
Piksel (1.1)	89	85	99	9	5	19
Piksel (1.2)	126	113	109	68	55	51
Piksel (1.3)	97	91	98	190	184	191
Piksel (2.0)	83	62	82	222	201	221
Piksel (2.1)	118	99	108	62	43	42
Piksel (2.2)	152	146	143	162	156	153
Piksel (2.3)	73	84	106	216	227	249
Piksel (3.0)	211	194	169	45	28	3
Piksel (3.1)	190	186	168	140	136	118

Piksel (3.2)	197	184	159	117	104	79
Piksel (3.3)	78	74	90	20	16	32

3.3.3 Perhitungan Proses Dekripsi *Vigenere Cipher*

Proses dekripsi merupakan kebalikan dari proses enkripsi dan menggunakan kunci yang sama juga. Nilai RGB dari setiap piksel didekripsi dengan cara subtitusi setiap nilai RGB dari semua piksel.

- Piksel (0.0) dengan $k_1 = 200$
  Red : 99 - 200 mod 256 = 155
  Green : 85 - 200 mod 256 = 141
  Blue : 69 - 200 mod 256 = 125
- Piksel (0.1) dengan $k_2 = 10$
  Red : 182 - 10 mod 256 = 172
  Green : 168 - 10 mod 256 = 158
  Blue : 147 - 10 mod 256 = 137
- Piksel (0.2) dengan $k_3 = 143$
  Red : 14 - 143 mod 256 = 127
  Green : 23 - 143 mod 256 = 136
  Blue : 23 - 143 mod 256 = 136
- Piksel (0.3) dengan $k_4 = 90$
  Red : 195 - 90 mod 256 = 105
  Green : 183 - 90 mod 256 = 93
  Blue : 201 - 90 mod 256 = 111
- Piksel (1.0) dengan $k_5 = 206$
  Red : 111 - 206 mod 256 = 161
  Green : 104 - 206 mod 256 = 154
  Blue : 93 - 206 mod 256 = 143
- Piksel (1.1) dengan $k_6 = 176$
  Red : 9 - 176 mod 256 = 89
  Green : 5 - 176 mod 256 =8 5

Blue : 19 - 176 mod 256 = 99

- Piksel (1.2) dengan $k_7 = 198$

  Red : 68 - 198 mod 256 = 126

  Green : 55 - 198 mod 256 = 113

  Blue :51 - 198 mod 256 = 109

- Piksel (1.3) dengan $k_8 = 93$

  Red : 190 - 93 mod 256 = 97

  Green : 184 - 93 mod 256 = 91

  Blue : 191 - 93 mod 256 = 98

- Piksel (2.0) dengan $k_9 = 139$

  Red : 222 - 139 mod 256 = 83

  Green : 201 - 139 mod 256 = 62

  Blue : 221 - 139 mod 256 = 82

- Piksel (2.1) dengan $k_1 = 200$

  Red : 62 - 200 mod 256 = 118

  Green : 43 - 200 mod 256 = 99

  Blue : 42 - 200 mod 256 = 98

- Piksel (2.2) dengan $k_2 = 10$

  Red : 162 - 10 mod 256 = 152

  Green : 156 - 10 mod 256 = 146

  Blue : 153 - 10 mod 256 = 143

- Piksel (2.3) dengan $k_3 = 143$

  Red : 216 - 143 mod 256 = 73

  Green : 227 - 143 mod 256 = 84

  Blue : 249 - 143 mod 256 = 106

- Piksel (3.0) dengan $k_4 = 90$

  Red : 45 - 90 mod 256 = 211

  Green : 28 - 90 mod 256 = 194

  Blue : 3 - 90 mod 256 = 169

- Piksel (3.1) dengan $k_5 = 206$

  Red : 140 - 206 mod 256 = 190

  Green : 136 - 206 mod 256 = 186

Blue : 118 - 206 mod 256 = 168

- Piksel (3.2) dengan $k_6 = 176$

  Red : 117 - 176 mod 256 = 197

  Green : 104 - 176 mod 256 = 184

  Blue : 79 - 176 mod 256 = 159

- Piksel (3.3) dengan $k_7 = 198$

  Red : 20 - 198 mod 256 = 78

  Green : 16 - 198 mod 256 = 74

  Blue : 32 - 198 mod 256 = 90

Setelah didekripsi didapatkan nilai RGB dari setiap piksel sseperti nilai RGB semula.Berikut merupakan perubahan yang terjadi setelah proses dekripsi, *cipherimage* diubah ke dalam bentuk *plainimage* dapat dilihat pada tabel 3.3

Tabel 3.3 Perubahan Nilai RGB Citra Hasil Dekripsi

Piksel	***Cipherimage***			***Plainimage***		
	Nilai RGB			**Nilai RGB**		
	R	**G**	**B**	**R**	**G**	**B**
Piksel (0.0)	99	85	69	155	141	125
Piksel (0.1)	182	168	147	172	158	137
Piksel(0.2)	14	23	23	127	136	136
Piksel (0.3)	195	183	201	105	93	111
Piksel (1.0)	111	104	93	161	154	143
Piksel (1.1)	9	5	19	89	85	99
Piksel (1.2)	68	55	51	126	113	109
Piksel (1.3)	190	184	191	97	91	98

Piksel (2.0)	222	201	221	83	62	82
Piksel (2.1)	62	43	42	118	99	108
Piksel (2.2)	162	156	153	152	146	143
Piksel (2.3)	216	227	249	73	84	106
Piksel (3.0)	45	28	3	211	194	169
Piksel (3.1)	140	136	118	190	186	168
Piksel (3.2)	117	104	79	197	184	159
Piksel (3.3)	20	16	32	78	74	90

3.4 Perancangan Uji Coba

Setelah sistem dibuat maka akan dilakukan pengujian terhadap sistem .Pengujian ini dimaksudkan agar dapat mengetahui kinerja dari sistem.Selain itu, sebagai bahan untuk mengevaluasi hasil dari implementasi analisa dan perancangan sistem.Bahan uji coba adalah citra dengan *format bitmap* 24 bit dengan ukuran resolusi beragam.

3.4.1 Pengujian Hasil Proses Enkripsi

Pada pengujian terhadap hasil proses enkripsi dilakukan antara lain dengan jumlah kunci yang dibangkitkan (jumlah iterasi) yang berbeda-beda pada setiap percobaan. Kemudian menghitung nilai korelasi dan *entropy* dari citra asli dan setiap percobaan yang didapatkan.Nilai korelasi yang rendah dan nilai *entropy* yang tinggi terhadap citra aslinya menunjukan kinerja perangkat lunak baik dalam menghasilkan *cipherimage*.Seperti yang telah dijelaskan pada subbab 2.6 dan 2.7.Rancangan pengujian nilai korelasi dan *entropy* ditunjukkan pada tabel 3.4.

Tabel 3.4 Rancangan pengujian Nilai Korelasi dan *Entropy*

Nama File					
Pengujian		Citra Asli	Percobaan		
			I	II	III
Korelasi	Horizontal				
	Vertikal				
	Diagonal				
	Rata-rata				
Entropy					

Keterangan:

- Citra Asli : citra asli yang belum dienkripsi.
- Percobaan I : citra yang telah dienkripsi dengan jumlah kunci 150 yang dibangkitkan dengan persamaan 2.3 pada subbab 2.3.2.
- Percobaan II : citra yang telah dienkripsi dengan jumlah kunci 350 yang dibangkitkan dengan persamaan 2.3 pada subbab 2.3.2.

Percobaan III : citra yang telah dienkripsi dengan jumlah kunci 500 yang dibangkitkan dengan persamaan 2.3 pada subbab 2.3.2.

BABIV
IMPLEMENTASI DAN PEMBAHASAN

4.1.Lingkungan Implementasi

Lingkungan implementasi dari rancangan enkripsi metode *Vigenere Cipher* dan *Logistic Map*pada citra digital yang telah dibahas di Bab III, meliputi lingkungan perangkat keras dan lingkungan perangkat lunak.

4.1.1.Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam tugas akhir ini adalah sebagai berikut.

1. *Processor* Intel® Dual-Core T4300@2.10Ghz
2. *Memory*2048 MB
3. *Harddisk* 320 GB
4. *Systemtype* 32-bit OS
5. *Monitor* 14'
6. *Keyboard*
7. *Mouse*

4.1.2.Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan dalam pengembangan enkripsi menggunakan metode *Vigenere Cipher* dan *Logistic Map*pada citra digital adalah :

1. Sistem operasi Windows 7 Ultimate 32-bit sebagai tempat aplikasi dijalankan.
2. Aplikasi dibangun dengan Netbeans 7.1

   JDK yang digunakan adalah JDK *version* 1.7.0_02

4.2. Implementasi Perangkat Lunak

Berdasarkan perancangan proses pada Bab III, maka pada subbab ini merupakan implementasi dalam pembuatan aplikasi yang menggunakan bahasa pemrograman JAVA. Berikut merupakan implementasi program dari proses-proses untuk enkripsi dan dekripsi citra digital dengan metode *Vigenere Cipher* dan *Logistic Map*.

4.2.1 Implementasi Proses Pengambilan Citra Digital

Implementasi program proses enkripsi diawali dengan mengambil citra digital yang akan dienkripsi. Citra digital ini berformat .bmp. File inputan harus berupa file citra digital jika file yang diinputkan bukan berupa file citra digital maka akan ada peringatan. Implementasi proses pembentukan kunci dapat dilihat pada *source code* 4.1.

```
1  private void BtnImageActionPerformed(java.awt.event.ActionEvent evt) {
2      JFileChooser chooser = new JFileChooser();
3      int wew = chooser.showOpenDialog(this);
4      if(wew == 0)
5          {
6  TxtImage.setText(chooser.getSelectedFile().toString());
7
8          ft = chooser.getTypeDescription(chooser.getSelectedFile());
9          ft = ft.substring(0,ft.lastIndexOf(" "));
10
11         fp = TxtImage.getText().substring(0,TxtImage.getText().lastIndexOf("\\"));
12
13             try
14                 {
15                     LblImageBefore.setIcon(new ImageIcon(ImageIO.read( new File(TxtImage.getText()))));
16                     BtnEncrypt.setEnabled(true);
17                     BtnDecrypt.setEnabled(true);
18                 } catch (Exception e) {
19                 JOptionPane.showMessageDialog(this, "This is not an image file");
20                 BtnEncrypt.setEnabled(false);
21                 BtnDecrypt.setEnabled(false);
```

Source Code* 4.1.Source Code*Pengambilan Citra Digital

4.2.2 Implementasi Proses Pembentukan Kunci

Setelah diambil file citra digital langkah selanjutnya sebelum melakukan enkripsi atau dekripsi adalah melakukan pembentukan kunci. Pada proses ini pengguna memasukkan dengan cara membangkitkan menggunakan metode *logistic map* sesuai dengan persamaan 2.3. Kunci digunakan untuk menjaga citra yang telah terenkripsi agar saat akan dikembalikan ke bentuk aslinya kata kunci yang dimasukan harus cocok. Sehingga inputan pada perangkat lunak ini selain citra digital adalah sejumlah karakter yang akan dibangkitkan menjadikunci. Pada proses ini pengguna memasukan nilai awal iterasi(X_0) antara ($0 < x < 1$),), laju pertumbuhan (r) antara ($0 < x < 4$) dan jumlah kunci yang akan dibangkitkan. Implementasi proses pembentukan kunci dapat dilihat pada *source code* 4.2.

```
1  public int[] ChaosNumberKey(double a, double b, int s)
2     {
3           double[] ch = new double[s];
4           int[] key = new int[s];
5           ch[0]= a;
6           for(int i = 1; i < s;i++) {
7                   ch[i] = b * ch[i-1] * (1 - ch[i-1]);
8                   ch[i] = (int) (ch[i] * 1000);
9                   ch[i] = ch[i] / 1000;}
10          for(int i = 0; i< s-1;i++) {
11              key[i] = (int)(ch[i+1]*1000) % 256;}
12          return key;
13 }
```

***Source Code* 4.2**.*Source Code* Pembentukan Kunci

4.2.3 Implementasi Proses Pembentukan Piksel Citra

Setelah didapatkannya nilai-nilai kunci yang akan digunakan untuk proses enkripsi. Kemudian saat tombol “Enkripsi” ditekan maka proses pertama yang akan dilakukan adalah pengambilan nilai RGB piksel citra dan diekstraksi kedalam RGB piksel citra. Implementasi pengambilan nilai RGB piksel citra dapat dilihat pada *source code* 4.3.

```
1   public byte[] ImgToByte (String a) throws FileNotFoundException, IOException
2       {
3           File file = new File(a);
4           FileInputStream fis = new FileInputStream(file);
5           ByteArrayOutputStream bos = new ByteArrayOutputStream();
6           byte[] buf = new byte[1024];
7
8           try {
9       for (int readNum; (readNum = fis.read(buf)) != -1;) {
10                      bos.write(buf, 0, readNum);
11                  }
12          } catch (Exception e) {}
13
14          byte[] bytes = bos.toByteArray();
15          return bytes;
16      }
17
18  public int[][][] ImgToPixels(BufferedImage img)
19      {
20          int[][][] a = new int[img.getWidth()][img.getHeight()][3];
21          int c = 0;
22          for(int i = 0;i < img.getWidth();i++)
23          {
24              for(int j = 0; j < img.getHeight();j++)
25              {
26
27                  int rgb = img.getRGB(i,j);
28
29                  int red = (rgb >> 16) & 0x000000FF;
30                  int green = (rgb >>8 ) & 0x000000FF;
31                  int blue = (rgb) & 0x000000FF;
32
33                  a[i][j][0] = red;
34                  a[i][j][1] = green;
35                  a[i][j][2] = blue;
36              }
```

```
37                  }
38                  return a;
39              }
40          public BufferedImage PixelsToImage(int[][][] a, int
            x,int y)
41              {
42                  BufferedImage img = new
            BufferedImage(x,y,BufferedImage.TYPE_INT_RGB);
43                  Color col;
44
45                  for(int i = 0; i<x;i++)
46                  {
47                      for(int j = 0; j<y;j++)
48                      {
49                          col = new
            Color(a[i][j][0],a[i][j][1],a[i][j][2]);
50                          img.setRGB(i, j, col.getRGB());
51                      }
52                  }
53                  return img;
54      }
```

Source Code* 4.3.Source Code*Pengambilan Nliai RGB piksel Citra

Pada *source code* 4.3 terdapat 3 *method* yaitu ImgToByte, ImgToPixels, dan PixelsToImage. Pada *method* ImgToByte ini digunakan untuk mengubah input string menjadi data byte. Input String inilah dipakai sebagai direkrori nama file. File diambil dan diubah menjadi byte array. Kemudian pada *method* ImgToPixels terdapat 3 *integer* sebagai red, green dan blue.Ke tiga *integer* tersebut mengambil data RGB dari setiap pixel citra digital.Setelah itu ke tiga *integer* dimasukkan menjadi satu *integer* array.Dan *method PixelsToImage* dipakai sebagai representasi panjang dan lebar citra digital.

4.2.4 Implementasi Proses Enkripsi

Setelah pembangkitan kunci telah terbentuk maka proses enkripsi dapat dilakukan. Kunci – kunci yang terbentuk itu akan digunakan untuk melakukan proses enkripsi. Implementasi enkripsi dapat dilihat pada *source code* 4.4

```
1
if(en==0) {
```

```
2   for(int i = 0;i < x;i++)      {
3   for(int j = 0; j < y;j++){
4   out.write("Pixel[" + i + ","+ j +"] " + "Key:" + b[bc]
    +"\r\n");
5   out.write("ORG R:" + a[i][j][0]+ " G:" + a[i][j][1] +" B:"+
    a[i][j][2] +"\r\n");
6   out.write("Encryption Red: " + a[i][j][0]+ " + " + b[bc] +"
    MOD 256 = ");
7   a[i][j][0] = (a[i][j][0] + b[bc])%256;
8   out.write(a[i][j][0] +"\r\n");
9   out.write("Encryption Green: " + a[i][j][1]+ " + " + b[bc] +"
    MOD 256 = ");
10  a[i][j][1] = (a[i][j][1] + b[bc])%256;
11  out.write(a[i][j][1] +"\r\n");
12  out.write("Encryption Blue: " + a[i][j][2]+ " + " + b[bc]
    +" MOD 256 = ");
13  a[i][j][2] = (a[i][j][2] + b[bc])%256;
14  out.write(a[i][j][2] +"\r\n" + "\r\n");  bc=(bc+1)%s;
```

Source Code* 4.4.Source Code* Enkripsi dengan metode *Vigenere Chiper*

Pada *source code* 4.4 yaitu proses enkripsi nilai-nilai RGB yang tersimpan dalam *integer* array. Setiap nilai RGB itu masing-masing akandienkripsi setiap warnanya yaitu dengan mensubtitusinya. Tiap warna akan dienkripsi dari *byte* array dan dimoduluskan 256 sehinga tidak melebihi batas warna yaitu 255.

4.2.5 Implementasi Proses Dekripsi

Proses dekripsi mengembalikan *cipherimage* menjadi ke bentuk semula atau *plainimage*. Pada proses ini mengambil masukan dari proses pembentukan kunci yang sama dari proses enkripsi. Implementasi dekripsi *cipherimage* dapat dilihat pada *source code* 4.5.

```
1   for(int i = 0;i < x;i++)
2               {
3                   for(int j = 0; j < y;j++)
4                   {
5                       out.write("Pixel[" + i + ","+ j +"] " +
```

```
"Key:" + b[bc] +"\r\n");
6                       out.write("ORG R:" + a[i][j][0]+ " G:" +
a[i][j][1] +" B:"+ a[i][j][2] +"\r\n");
7                       out.write("Deccryption Red: " +
a[i][j][0]+ " - " + b[bc] +" MOD 256 = ");
8                       a[i][j][0] = ((a[i][j][0] + 256 -
b[bc]))%256;
9                       out.write(a[i][j][0] +"\r\n");
10                      out.write("Deccryption Green: " +
a[i][j][1]+ " - " + b[bc] +" MOD 256 =");
11                      a[i][j][1] = ((a[i][j][1] + 256 -
b[bc]))%256;
12                      out.write(a[i][j][1] +"\r\n");
13                      out.write("Deccryption Blue: " +
a[i][j][2]+ " - " + b[bc] +" MOD 256 =");
14                      a[i][j][2] = ((a[i][j][2] + 256 -
b[bc]))%256;
15                      out.write(a[i][j][2] +"\r\n" + "\r\n");
16                      bc=(bc+1)%s;   } } }
17  out.write("=============================================\r\n"
);
18          out.close();
19          return a;}
20  for(int i = 0;i < x;i++)
21              {
22                  for(int j = 0; j < y;j++)
23                  {
24                      out.write("Pixel[" + i + ","+ j +"] " +
"Key:" + b[bc] +"\r\n");
25                      out.write("ORG R:" + a[i][j][0]+ " G:" +
a[i][j][1] +" B:"+ a[i][j][2] +"\r\n");
26                      out.write("Deccryption Red: " +
a[i][j][0]+ " - " + b[bc] +" MOD 256 = ");
27                      a[i][j][0] = ((a[i][j][0] + 256 -
b[bc]))%256;
28                      out.write(a[i][j][0] +"\r\n");
29                      out.write("Deccryption Green: " +
a[i][j][1]+ " - " + b[bc] +" MOD 256 =");
30                      a[i][j][1] = ((a[i][j][1] + 256 -
b[bc]))%256;
31                      out.write(a[i][j][1] +"\r\n");
32                      out.write("Deccryption Blue: " +
a[i][j][2]+ " - " + b[bc] +" MOD 256 =");
33                      a[i][j][2] = ((a[i][j][2] + 256 -
b[bc]))%256;
34                      out.write(a[i][j][2] +"\r\n" + "\r\n");
35                      bc=(bc+1)%s;   } } }
36  out.write("=============================================\r\n"
);
```

```
37          out.close();
38          return a;}
```

Source Code* 4.5.Source Code* Dekripsi dengan metode *Vigenere Chiper*

4.2.6 Implementasi Proses Perhitungan Korelasi dan *Entropy*

Nilai korelasi digunakan untuk mengetahui kemiripan antara dua buah piksel yang saling bertetanggaan pada sebuah citra.Pada korelasi mengukur tingkat kemiripan dengan membandingkannya secara horizontal, vertikal, diagonal kanan, dan diagonal kiri. Implementasi perhitungan nilai korelasi dapat dilihat pada *source code* 4.6.

```
1   rh = (NKor(xRed, yhRed) + NKor(xBlue, yhBlue) + NKor(xGreen, yhGreen)) / 3;
2           rv = (NKor(xRed, yvRed) + NKor(xBlue, yvBlue) + NKor(xGreen, yvGreen)) / 3;
3           rd = (NKor(xRed, ydRed) + NKor(xBlue, ydBlue) + NKor(xGreen, ydGreen)) / 3;
4           korelasi = (rh + rv + rd) / 3;
5           double[] kora = new double[4];
6           kora[0] = rh;
7           kora[1] = rv;
8           kora[2] = rd;
9           kora[3] = korelasi;
10          return kora;        }
```

Source Code* 4.6.Source Code* Perhitungan Nilai Korelasi

Nilai *Entropy* digunakan untuk mengukur keragaman intensitas warna sebuah citra. Nilai *entropy* besar untuk citra dengan transisi derajat keabuan merata dan bernilai kecil jika struktur citra tidak teratur. Nilai *entropy* dihitung sesuai dengan persamaan 2.6. Implementasi perhitungan nilai *entropy* dapat dilihat pada *source code* 4.7.

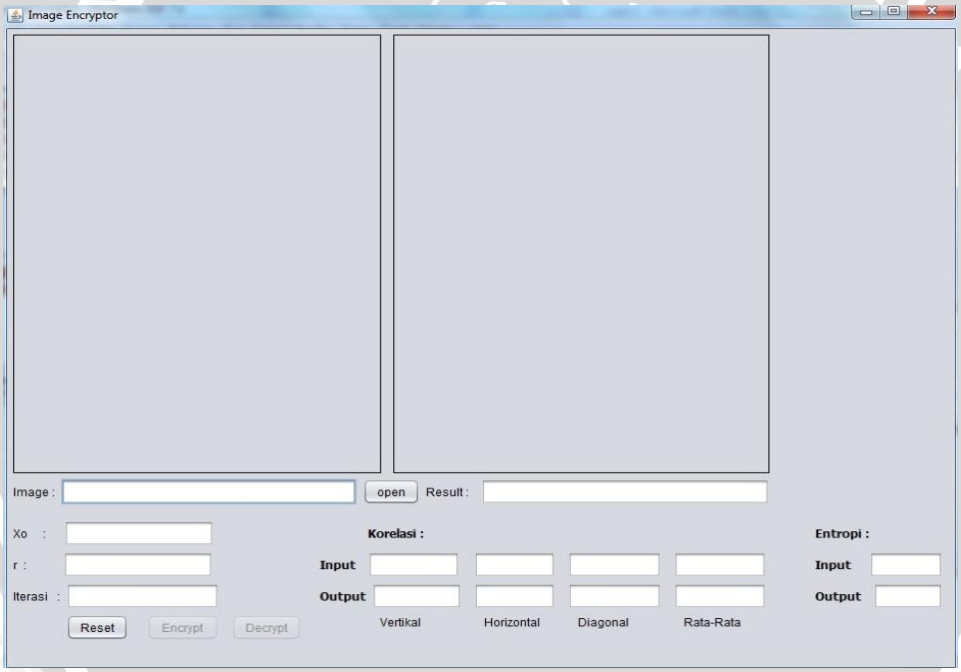
```
1   for(int i = 0;i<pRed.length;i++){
2               pRed[i] = pRed[i] / (x*y);
3               pGreen[i] = pGreen[i] / (x*y);
4               pBlue[i] = pBlue[i] / (x*y);}
5           for(int i =0;i<pRed.length;i++){
```

```
6           if(pRed[i]!=0)  {
7               phR = phR +
(pRed[i]*Math.log(pRed[i])/Math.log(2));}
8           if(pGreen[i]!=0) {
9               phG = phG +
(pGreen[i]*Math.log(pGreen[i])/Math.log(2)); }
10          if(pBlue[i]!=0){
11              phB = phB +
(pBlue[i]*Math.log(pBlue[i])/Math.log(2)); }}
12      return -(phR+phG+phB)/3; }
```

Source Code* 4.7.Source Code* Perhitungan Nilai *Entropy***

4.3. Implementasi Antarmuka

Implementasi antar muka pada aplikasi enkripsi citra digital Menggunakan *Vigenere Cipher* dan *Logistic Map* dapat dilihat pada gambar 4.1.

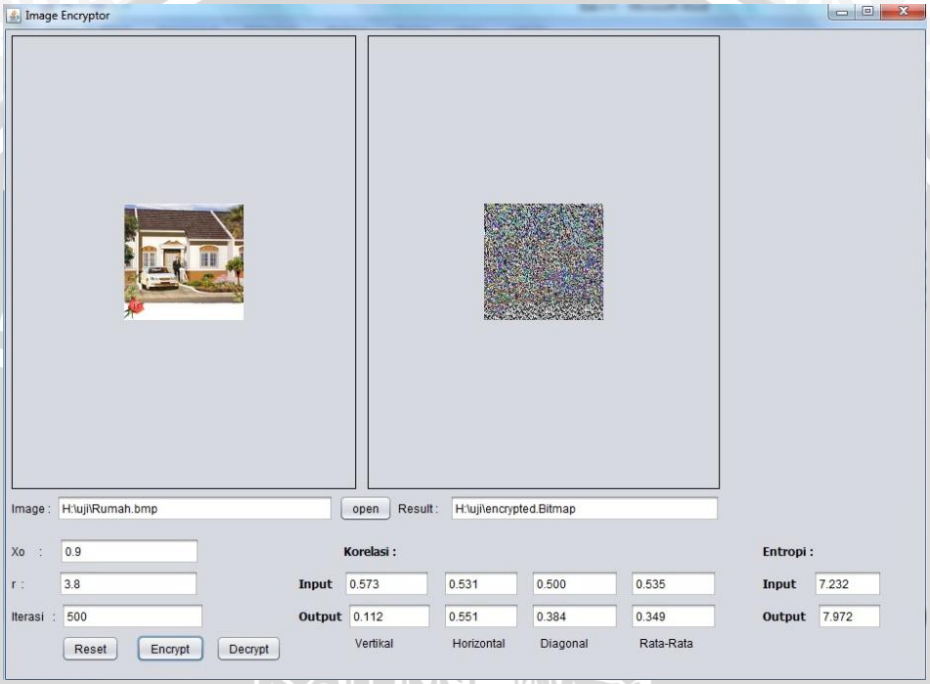


Gambar 4.1. Antarmuka

Implementasi program proses enkripsi citra digital diawali dengan dimasukannya citra digital lewat menu file – open, setelah diambilnya atau dimasukannya citra digital nilai-nilai pada X_0,r dan iterasi harus diisi. Kriteria nilai tersebut yaitu X_0 (nilai awal iterasi) antara ($0 < x < 1$), r(laju pertumbuhan) antara ($0 < x < 4$) dan itersi (jumlah kunci yang akan dibangkitkan) apabila nilai yang dimasukan tidak memenuhi kriteria diatas maka akan muncul peringatan,

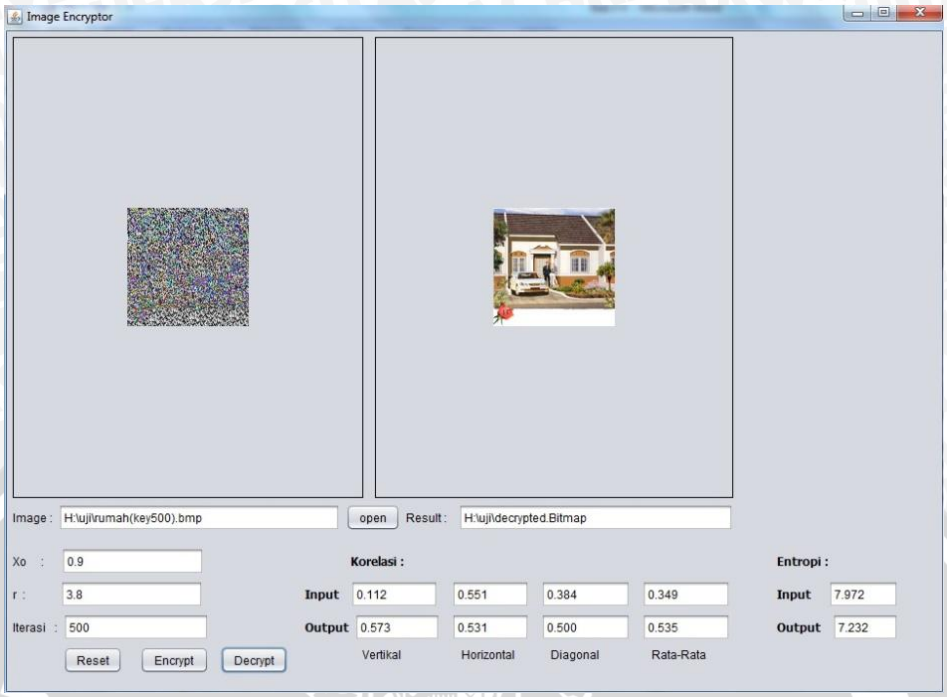
tetapi jika sudah maka proses enkripsi dapat dijalankan dengan menekan tombol enkripsi.

Setelah proses enkripsi selesai *cipherimage* tersimpan di lokasi yang sama dengan lokasi dari citra yang dimasukan dengan nama citra "encrypted". Field untuk nilai korelasi dan entropy akan terisi bersamaan dengan selesainya proses enkripsi. Untuk antar muka Proses enkripsi dapat dilihat pada Gambar 4.2.



Gambar 4.2. Antarmuka Proses enkripsi *plainimage*

Langkah untuk melakukan dekripsi citra hampir sama seperti pada langkah melakukan enkripsi, hanya saja nilai-nilai pada X_0,r dan iterasi harus sama dengan nilai-nilai pada saat proses enkripsi. Antar muka proses dekripsi dapat dilihat pada gambar 4.3



Gambar 4.3. Antarmuka Proses dekripsi *cipherimage*

4.4. Implementasi dan Pembahasan Uji Coba

Implementasi uji coba terhadap aplikasi enkripsi dan dekripsi citra digital menggunakan *Vigenere Cipher* dan *Logistic Map* ini mengacu kepada perancangan uji coba yang telah dibahas pada subbab 3.3.Sesuai dengan batasan masalah pada subbab 1.3, citra yang digunakan pada aplikasi ini adalah citra 24-bit.Daftar citra yang digunakan terlihat pada tabel 4.1.

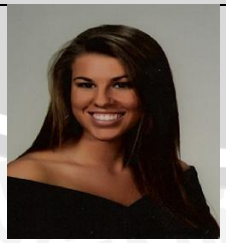
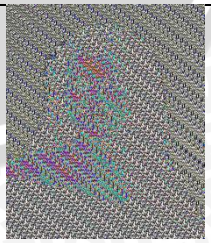
Tabel 4.1 Daftar Berkas Citra Bitmap 24 bit

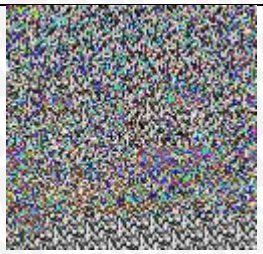
No	Nama File	Citra	Resolusi
1	car.bmp		1024x768
2	harimau.bmp		640x480

3	salena.bmp		227x329
4	rumah.bmp		125x122

Hasil enkripsi dengan nilai X_0 0.9, nilai r = 3.8 dan jumlah iterasi (jumlah kunci dibangkitkan) = 150 ditujukan pada tabel 4.2

Tabel 4.2 Hasil enkripsi Citra Bitmap 24 bit

No	Nama File	Citra (*plainimage*)	Resolusi	Citra (*cipherimage*)
1	car.bmp		1024x768	
2	harimau.bmp		640x480	
3	salena.bmp		227x329	

4	rumah.bmp		125x122	

4.4.1 Hasil dan Pembahasan Uji *Cipherimage*

Pada pengujian ini percobaan yang diberikan adalah iterasi yang memiliki jumlah yang berbeda dengan maksud untuk mengetahui pengaruh terhadap nilai korelasi dan *entropy* yang dihasilkan saat proses enkripsi. Percobaan yang diberikan dapat dilihat di tabel 4.3

Tabel 4.3 Kasus Pengujian

Percobaan	Jumlah Iterasi
Percobaan I	150
Percobaan II	350
Percobaan III	500

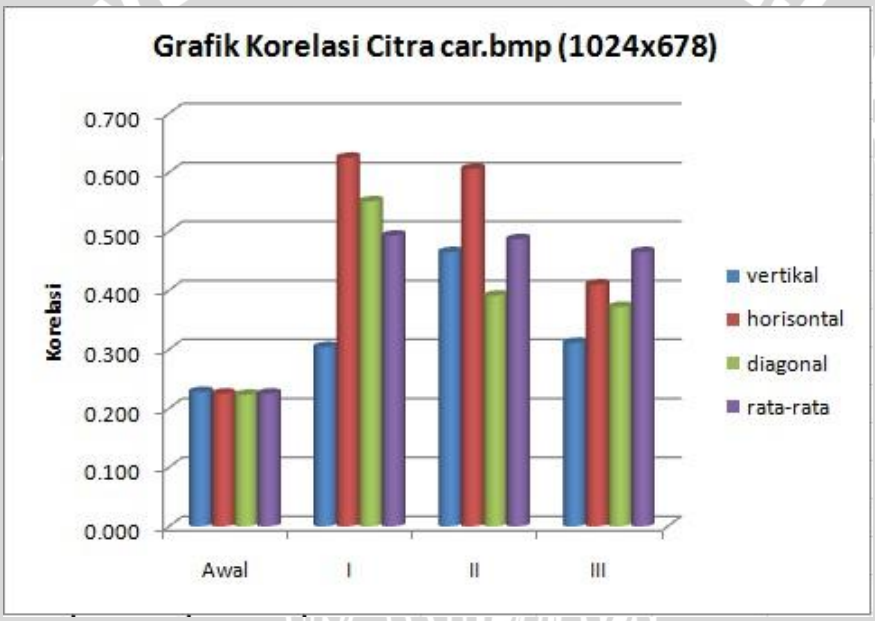
4.4.1.1 Hasil dan Pembahasan Uji Korelasi

Nilai korelasi digunakan untuk mengukur tingkat korelasi atau kemiripan antara dua buah piksel yang saling bertetanggaan pada citradigital. Jika koefisien dari nilai korelasi mendekati nol maka antara *chipherimage* dengan citra asli memiliki perbedaan, *chipherimage* tidak memiliki kemenarikan dan antar pikselnya yang saling bertetenggaan tidak terkait satu sama lain. Semakin rendah nilai korelasi yang dimiliki *chipherimage* maka semakin baik kualitas yang dihasilkan dari proses enkripsi. Hasil pengujian dari *cipherimage* yang dihasilkan oleh aplikasi ini ditunjukan oleh Tabel 4.4 – 4.8.Hasil korelasi uji *Cipherimage* citra car.bmp yang berukuran 1024x768dapat dilihat pada tabel 4.4:

Tabel 4.4 Hasil Pengujian *Cipherimage* car.bmp 1024x768

Pengujianke-1		car.bmp	Percobaan		
			I	II	III
Korelasi	Vertikal	0,228	0,304	0,465	0,311
	Horizontal	0,225	0,625	0,606	0,410
	Diagonal	0,223	0,551	0,391	0,372
	Rata-rata	0,225	0,493	0,487	0,465

Berdasarkan nilai korelasi yang didapatkan pada Tabel 4.4, grafik nilai korelasi pada citra harimau.bmp ukuran 1024x768 dapat dilihat pada gambar 4.4.



Gambar 4.4 Grafik Korelasi Citra car.bmp (1024x768)

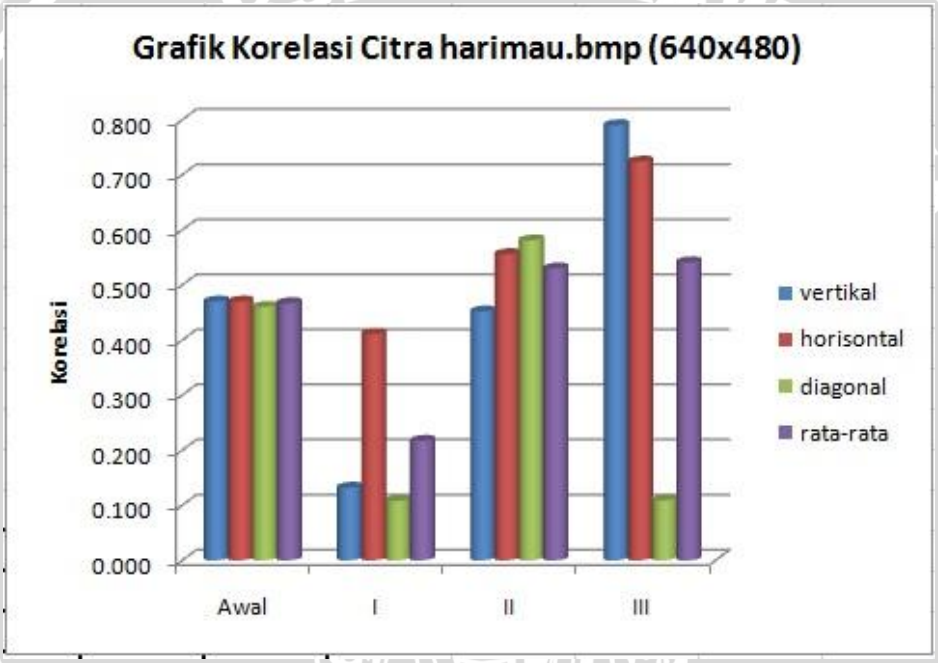
Berdasarkan grafik korelasi pada gambar 4.4 dapat dilihat mengalami kenaikan nilai korelasi pada c*ipherimage* jika dibandingkan dengan nilai korelasi citra awal.Dari percobaan I, II, dan III untuk citra digital dengan resolusi 1024 x 678 nilai korelasi antara piksel yang bertetanggaan arah vertikal, horizontal, diagonal semakin besar.

Hasil korelasi uji *Cipherimage* citra harimau.bmp yang berukuran 640 x 480dapat dilihat pada tabel 4.5:

Tabel 4.5 Hasil Pengujian *Cipherimage* harimau.bmp 640 x 480

Pengujianke-2		harimau.bmp	Percobaan		
			I	II	III
Korelasi	Vertikal	0,470	0,132	0,452	0,790
	Horizontal	0,470	0,411	0,556	0,724
	Diagonal	0,460	0,108	0,581	0,109
	Rata-rata	0,467	0,217	0,530	0,541

Berdasarkan nilai korelasi yang didapatkan pada Tabel 4.5, grafik nilai korelasi pada citra harimau.bmp ukuran 640 x 480 dapat dilihat pada gambar 4.5.



Gambar 4.5 Grafik Korelasi Citra harimau.bmp (640 x 480)

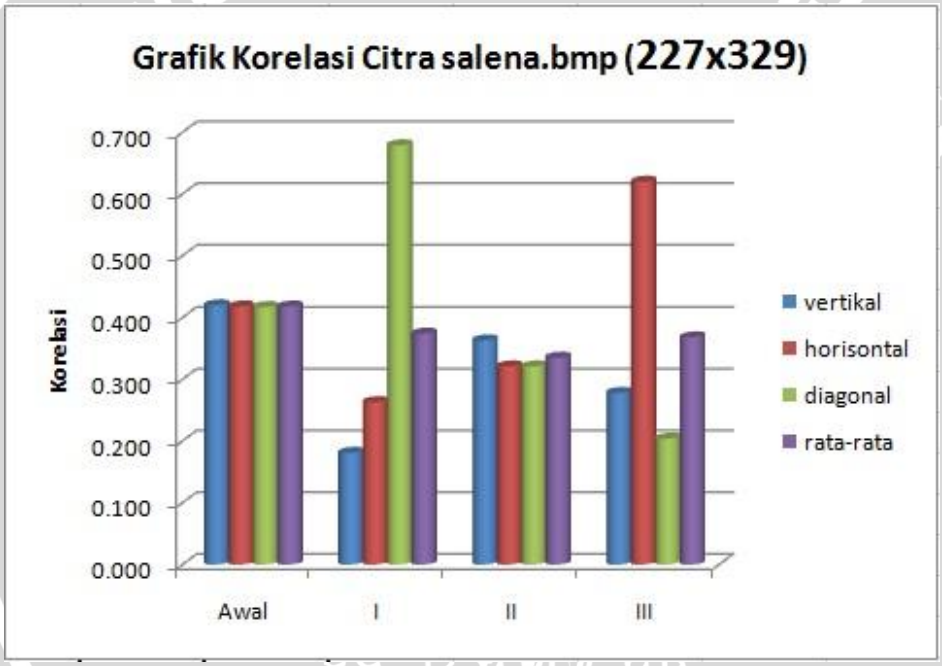
Berdasarkan grafik korelasi pada gambar 4.5 pada percobaan I mengalami penurunan nilai korelasi pada c*ipherimage* jika dibandingkan dengan nilai korelasi citra awal untuk semua nilai korelasi antara piksel yang bertetanggaan arah vertikal, horizontal, diagonal. Namun pada percobaan II dan III saat jumlah kunci ditambahkan nilai korelasi antara piksel yang bertetanggaan arah vertikal, horizontal, diagonal mengalami kenaikan kecuali pada nilai korelasi arah diagonal di percobaan III.

Hasil korelasi uji *Cipherimage* citra salena.bmp yang berukuran 227x329dapat dilihat pada tabel 4.6:

Tabel 4.6 Hasil Pengujian *Cipherimage* salena.bmp 227x329

Pengujianke-3		salena.bmp	Percobaan		
			I	II	III
Korelasi	Vertikal	0,420	0,181	0,363	0,278
	Horizontal	0,418	0,263	0,321	0,620
	Diagonal	0,417	0,679	0,321	0,204
	Rata-rata	0,418	0,374	0,335	0,368

Berdasarkan nilai korelasi yang didapatkan pada Tabel 4.6, grafik nilai korelasi pada citra salena.bmp ukuran 227x329 dapat dilihat pada gambar 4.6.



Gambar 4.6 Grafik Korelasi Citra salena.bmp (227x329)

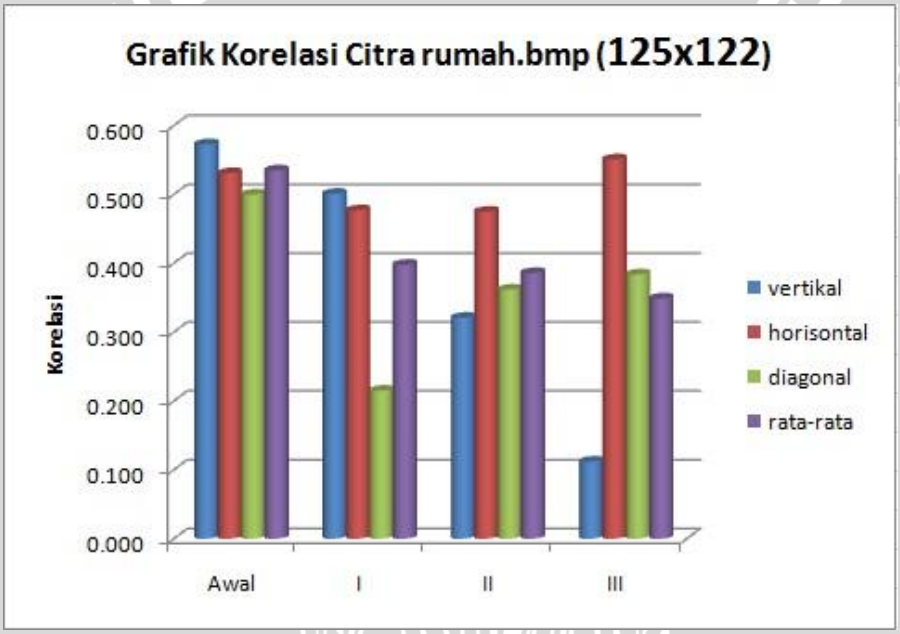
Berdasarkan grafik korelasi pada gambar 4.5 nilai korelasi rata-rata pada c*ipherimage* disemua percobaan untuk resolusi citra digital yang lebih kecil mengalami penurunan jika dibandingkan dengan nilai korelasi citra awal. Dan nilai korelasi antara piksel yang bertetanggaan arah vertikal, horizontal, diagonal juga mengalami penurunan kecuali pada nilai korelasi arah diagonal di percobaan II dan horizontal di percobaan III.

Hasil korelasi uji *Cipherimage* citra rumah.bmp yang berukuran 125x122dapat dilihat pada tabel 4.7:

Tabel 4.7 Hasil Pengujian *Cipherimage* rumah.bmp 125x122

Pengujianke-4		rumah.bmp	Percobaan		
			I	II	III
Korelasi	Vertikal	0,573	0,501	0,321	0,112
	Horizontal	0,531	0,477	0,475	0,551
	Diagonal	0,500	0,215	0,362	0,384
	Rata-rata	0,535	0,398	0,386	0,349

Berdasarkan nilai korelasi yang didapatkan pada Tabel 4.7, grafik nilai korelasi pada citra rumah.bmp ukuran 125x122 dapat dilihat pada gambar 4.7.



Gambar 4.7 Grafik Korelasi Citra rumah.bmp (125x122)

Berdasarkan grafik korelasi pada gambar 4.5 nilai korelasi rata-rata pada c*ipherimage* disemua percobaan untuk resolusi citra digital yang lebih kecil mengalami penurunan jika dibandingkan dengan nilai korelasi citra awal. Dan nilai korelasi antara piksel yang bertetanggaan arah vertikal, horizontal, diagonal juga mengalami penurunan kecuali pada nilai korelasi arah horizontal di percobaan III.

4.4.1.2 Hasil dan Pembahasan Uji *Entropy*

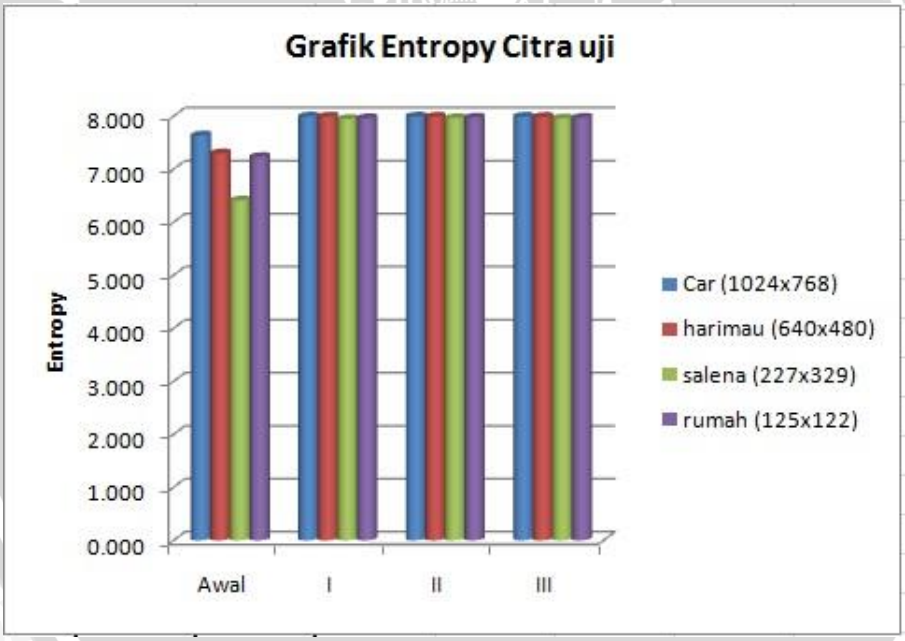
Nilai *entropy* digunakan untuk mengetahui keragaman intensitas dari sebuah citra. Semakin tinggi nilai *entropy* maka menunjukkan suatu citra memiliki

intensitas citra yang beragam dan transisi derajat keabuan yang merata. Tabel hasil pengujian *entropy* pada citra digital dapat dilihat pada tabel 4.8.

Tabel 4.8 Hasil Pengujian *Entropy* Citra Uji

Citra Uji		Citra Awal	Percobaan		
			I	II	III
Entropy	car.bmp	7,636	7,994	7,994	7,991
	harimau.bmp	7,299	7,992	7,990	7,986
	salena.bmp	6,412	7,949	7,965	7,958
	rumah.bmp	7,232	7,962	7,974	7,972

Berdasarkan nilai entropy yang didapatkan pada Tabel 4.8, grafik nilai entropy pada citra uji dapat dilihat pada gambar 4.8.



Gambar 4.8 Grafik*Entropy*Citra Uji

Pada grafik *entropy*citra uji nilai *entropy* pada c*ipherimage* untuk semua percobaan mengalami kenaikan jika dibandingkan dengan nilai *entropy*citra awal.Seperti yang dijelaskan pada subbab 2.7 yaitu semakin tinggi nilai *entropy* maka *cipherimage* yang dihasilkan semakin baik, dan sebaliknya jika nilai *entropy* semakin rendah maka *cipherimage* yang dihasilkan memiliki kualitas yang semakin rendah.

Secara umum dapat dilihat semakin kecil resolusinya nilai *entropynya* akan smeakin kecil. Jadi resolusi citra digital akan mempengaruhi perubahan nilai *entropy*. Sedangkan untuk perubahan jumlah kunci tidak begitu mempengaruhi perubahan nilai *entropy*.*Cipherimage* yang dihasilkan oleh perangkat lunak ini dapat dikatakan baik karena nilai *entropy* yang dihasilkan pada citra uji semuanya mengalami peningkatan dari nilai entropy citra digital awal. Hal ini menunjukkan bahwa *cipherimage* yang dihasilkan memiliki intensitas citra yang beragam dan transisi derajat keabuan yang merata dibandingkan dengan citra awal.

BABV
KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil pengujian dan analisis yang telah dilakukan, dapat diambil kesimpulan sebagai berikut :.

1. Metode *vigenere cipher* dapat diimplementasikan untuk melakukan enkripsi dan dekripsi pada citra digital. Langkah awal dengan pembangkitan kunci menggunakan *logistic map*.Setelah itu mengimplementasikan metode *vigenere cipher* untuk enkripsi atau dekripsi pada setiap nilai RGB di setiap piksel citra digital dan menghasilkan citra digital yang baru atau *cipherimage*.
2. Pada saat proses enkripsi nilai korelasi antar*piksel*yang bertetanggaan pada citra digital mengalami kenaikan untuk resolusi yang besar dan mengalami penurunan saat resolusi kecil. Sedangkan pada nilai *entropy* mengalami peningkatan, sehingga kelayakan kualitas cipherimage yang didasarkan pada nilai korelasi dan *entropy* dapat dikatakan cukup bagus.

5.2. Saran

Saran untuk pengembangan penelitian lebih lanjut berdasarkan hasil yang didapat dalam penelitian ini adalah :

1. Untuk meningkatkan kualitas dari cipherimage yang dihasilkan dapat digunakan penggabungan teknik ataupun pengembangan terhadap metode*vigenere cipher*.
2. Untuk mengimpementasikan metode*vigenere cipher* tidak terbatas pada data uji berupa file citra bmp 24-bit saja, tetapi dapat berupa file citra lainnya. Pengembangan penelitian ini juga dapat dilakukan dengan mengimpementasikan metode*vigenere cipher* pada file berupa video dan lainnya.

DAFTARPUSTAKA

[ARI-05] Ariyus, Dony. 2005. *Computer Security*. Penerbit Andi,Yogyakarta.

[ARI-08] Ariyus, Dony. 2008. *Pengantar Ilmu Kriptografi : Teori, Analisis, Implementasi.*Andi Offset,Yogyakarta.

[AWC-96] Awcock,G.W., dan R.Thomas.1996. *Applied Image Processing.*Mc Graw-Hill,Inc,Singapore .

[KIZ-05] Kizza, Joseph Mingga. 2005. *Computer Network Security*. Springer Science and Business Media, Inc,New York .

[KUR-04] Kurniawan, Yusuf. 2004. *Keamanan Internet dan Jaringan Komunikasi*.Informatika,Bandung.

[LAM-TT] Lampton, James. Tanpa Tahun. *Chaos Cryptography: Protecting Data Using Chaos.*Mississippi School for Mathematics and Science

[MUN-04] Munir, Renaldi. 2004. *Pengolahan Citra Digital Dengan Pendekatan Algoritmik.*Informatika,Bandung .

[MUN-06] Munir, R. 2006. *Kriptografi*.Penerbit Informatika,Bandung.

[MUR-92] Murni, A. dan S. Setiawan. 1992. *Pengantar Pengolahan Citra.*Elex Media Komputindo,Jakarta.

[PRA-03] Pratikaningtyas, Dhani,Imam Santoso ; dan Ajub Ajulian Z.2003. *Makalah Tugas Akhir : Klasifikasi Motif Batik Menggunakan Metode Transformasi Paket Wavelet.* Universitas Dipenogoro, Semarang.

[PRA-12] Pramudianti, Shinta. 2012. Kriptografi.
http://shinta-pramudianti.ugm.ac.id/2012/02/23/pengenalan-kriptografi/. Diakses tanggal 19 Maret 2012.

[PRI-TT] Prihandini, Ratna Ekasari, Tanpa Tahun. *Perbandingan Penggunaan Teknik Pembangkitan BilanganRandom Keystream Generator dengan Teknik Chaospada Stream cipher*. Jurnal Tugas Akhir Jurusan Teknik, ITB, Bandung.

[SCA-98] Scaringer, J., 1998.*Fast Encrytption of image data using chaotic Kolmogrov flow, J. Electronic Eng 7* (6), pp 318-437.

[SUH-09] Suhartana, I Ketut Gede, 2009.*Pengamanan Image True Color 24 Bit Menggunakan Algoritma Vigenere Cipher Dengan Penggunaan Kunci Bersama*. Jurnal Tugas Akhir Jurusan Teknik Informatika Universitas Udayana, Bali.

[YOU-08] Younes, M. Ali Bani, dan Jantan, Aman. 2008. *Image Encryption Using Block-Based Transformation Algorithm.IAENG International Journal of Computer Science.*