Aplikasi Kriptografi File Citra Digital Menggunakan Algoritma Triple DES (*Triple Data Encryption Standard*)

SKRIPSI

Disusun Oleh :

Mega Andria Dinni
08.18.901

PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2012

LEMBAR PERSETUJUAN

Aplikasi Kriptografi File Citra Digital Menggunakan Algoritma Triple DES (*Triple Data Encryption Standard*)

SKRIPSI

Disusun dan diajukan untuk melengkapi dan memenuhi persyaratan guna mencapai gelar Sarjana Informatika Strata Satu (S-1)

Disusun Oleh:

Mega Andria Dinni

NIM : 08.18.901

Diperiksa dan Disetujui,

Dosen Pembimbing I	Dosen Pembimbing II
Joseph Dedy Irawan, ST, MT.	**Michael Ardita, ST, MT.**
NIP.197404162005021002	**NIP. 103 1000 434**

Mengetahui,

Ketua Program Studi Teknik Informatika S-1

Joseph Dedy Irawan, ST, MT.

NIP.197404162005021002

PROGRAM STUDI TEKNIK INFORMATIKA

INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI

2012

SURAT PERNYATAAN

Nama Mahasiswa	: Mega Andria Dinni
Tempat / tanggal Lahir	: Pasuruan, 07 November 1987
NIM	: 0818901
Alamat	: Perumahan Bumi Citra Fajar Sekawan Indah Blok-B No.03 Kel.Bulusidokare Kec.Sidoarjo

Menyatakan bahwa karya skripsi saya yang berjudul :

"Aplikasi Kriptografi File Citra Digital Menggunakan Algoritma Triple DES(Triple Data Encryption Standard)" Adalah bukan merupakan karya tulis orang lain, baik sebagian maupun keseluruhan, kecuali dalam bentuk kutipan yang kami sebutkan sumbernya. Demikian surat pernyataan ini kami buat dengan sebenar-benarnya dan apabila pernyataan ini tidak benar, kami bersedia mendapatkan sanksi akademis.

Malang, 9 Agustus 2012

Yang Menyatakan,

Mega Andria Dinni

ABSTRAKSI

Aplikasi Kriptografi File Citra Digital Menggunakan Algoritma Triple DES (*Triple Data Encryption Standard*)

Teknik Informatika S-1,
Institut Teknologi Nasional Malang
e-mail : meme_ad@ymail.co.id

Dosen Pembimbing : **I. Joseph Dedy Irawan, ST, MT**
II. Michael Ardita, ST, MT

Abstraksi

Triple DES (Triple Data Encryption Standard) merupakan salah satu algoritma simetris pada kriptografi yang digunakan untuk mengamankan data dengan cara menyandikan data. Proses yang dilakukan dalam penyandian datanya, yaitu proses enkripsi dan proses dekripsi. Algoritma triple DES adalah suatu algoritma pengembangan dari algoritma DES (Data Encryption Standard). Perbedaan DES dengan triple DES terletak pada panjangnya kunci yang digunakan. Pada DES menggunakan satu kunci yang panjangnya 56-bit, sedangkan pada triple DES menggunakan 3 kunci yang panjangnya 168-bit (masing-masing panjangnya 56-bit). Pada triple DES, 3 kunci yang digunakan bisa bersifat saling bebas ($K1 \neq K2 \neq K3$) atau hanya dua buah kunci yang saling bebas dan satu kunci lainnya sama dengan kunci pertama ($K1 \neq K2$ dan $K3 = K1$). Karena tingkat kerahasiaan algoritma triple DES terletak pada panjangnya kunci yang digunakan, maka penggunaan algoritma triple DES dianggap lebih aman dibandingkan dengan algoritma DES.

Untuk memudahkan penggunaan algoritma triple DES, maka dibuat suatu program algoritma triple DES dengan alat bantu software komputer, yaitu Borland Delphi 7 yang dapat mengenkripsi dan mendekripsi file yang berekstensi bitmap(.bmp).

Kata kunci : 3DES (Triple Data Encryption Standard), DES (Data Encryption Standard), kriptografi, enkripsi, dekripsi, kunci.

KATA PENGANTAR

Puji Syukur kepada TUHAN YANG MAHA ESA atas limpahan rahmat-Nya penyusun dapat menyelesaikan laporan tugas akhir dengan judul "Aplikasi Kriptografi File Citra Digital Menggunakan Algoritma Triple DES(Triple Data Encryption Standard)"

Pembuatan tugas akhir ini disusun guna memenuhi syarat akhir kelulusan pendidikan jenjang Strata-1 di Institut Teknologi Nasional Malang. Laporan tugas akhir ini merupakan tanggung jawab tertulis atas ilmu yang didapat selama penulis mengikuti kuliah

Dengan selesainya penyusunan tugas akhir ini penyusun menyadari sepenuhnya bahwa tidak terlepas dari bantuan dan bimbingan serta arahan dari berbagai pihak secara langsung maupun tidak langsung. Untuk itu penulis berterima kasih sebesar – besarnya kepada :

- Ayah dan Mama serta Kedua Adik-adikku tercinta atas segala doa dan semangat yang diberikan.
- Bapak Ir. Soeparno Djiwo, MT selaku rektor ITN Malang.
- Bapak Ir. H. Sidik Noertjahjono, Mtselaku Dekan Fakultas Teknik Industri ITN Malang.
- Bapak Joseph Dedy Irawan, ST, MT, selaku Ketua Jurusan Teknik Informatika S-1, selaku Dosen wali dan selaku Dosen pembimbing I.
- Bapak Michael Ardita, ST, MT, selaku Dosen Pembimbing II atas bantuan ilmu, arahan, serta dukungan yang telah diberikan.
- Bapak Ir.Sentot Achmadi, M.Si selaku Sekretaris Jurusan Teknik Informatika S-1.
- Teman-teman yang telah membantu penyusunan skripsi ini.

Penyusun juga menyadari tugas akhir ini jauh dari sempurna, untuk itu berharap saran dan masukan agar tugas akhir ini dapat bermanfaat bagi kita semua khususnya teman-teman mahasiswa di Jurusan Teknik Informatika

Akhir kata, penyusun mohon maaf kepada semua pihak bilamana selama penyusunan skripsi ini penyusun membuat kesalahan secara tidak sengaja dan semoga skripsi ini dapat bermanfaat bagi kita semua.

Malang, Agustus 2012

Penyusun

DAFTAR ISI

HALAMAN JUDUL ..... **i**

LEMBAR PENGESAHAN ..... **ii**

ABSTRAKSI ..... **iii**

KATA PENGANTAR ..... **iv**

DAFTAR ISI ..... **v**

DAFTAR GAMBAR ..... **xi**

DAFTAR TABEL ..... **xiii**

BAB I PENDAHULUAN **1**

1.1 Latar Belakang 1

1.2 Rumusan Masalah 2

1.3 Tujuan Penelitian 3

1.4 Batasan Masalah 3

1.5 Manfaat Penelitian 3

1.6 Metodologi 4

1.7 Sistematika Penulisan 5

BAB II LANDASAN TEORI **6**

2.1 File 6

2.2 Citra Digital 6

2.3 Bitmap (.bmp) 7

2.4 Kriptografi 8

2.5 Pembahasan Algoritma 10

2.5.1 Algoritma Triple DES 10

2.5.2 Enkripsi 12

2.5.3 Dekripsi 14
2.5.4 Pemilihan Kunci 14
2.5.5 Proses Enkripsi dan Dekripsi 15
2.5.6 Penggunaan Triple DES 16
2.5.7 Keamanan Triple DES 17
2.5.8 Tingkat Kerahasiaan Kunci 17
2.5.9 Kekuatan Terhadap Serangan Brute Force 17
2.6 Borland Delphi 7 17
BAB III ANALISIS DAN PERANCANGAN 18
3.1 Analisi Sistem 18
3.2 Alur Sistem 18
3.2.1 Sistem Enkripsi 19
3.3 Desain Antarmuka Aplikasi 19
3.3.1 Desain Form Enkripsi 19
3.3.1 Desain Form Dekripsi 21
3.4 Perancangan Antarmuka 22
3.3.1 Mendesain Tampilan Form Utama 23
3.5 Perancangan Sistem 26
3.6 Perancangan Flowchart 26
3.6.1 Flowchart Proses Enkripsi 26
3.6.2 Flowchart Proses Dekripsi 29
BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM 32
4.1 Implementasi Sistem 33
4.1.1 Tahap Pengenkripsian 33

4.1.2 Tahap Pendekripsian 37
4.1.3 Tahap Perbandingan 41
4.1.4 Tahap Bantuan atau Help 41
4.2 Pengujian Sistem1 43
4.2.1 Pengujian Citra Digital 43
BAB V KESIMPULAN DAN SARAN 49
5.1 Kesimpulan 49
5.2 Saran 50

DAFTAR PUSTAKA 51
LAMPIRAN 53

DAFTAR GAMBAR

Gambar 2.1 Bitmap 7

Gambar 2.2 Algoritma *Triple DES* secara Detail 11

Gambar 2.3 Proses Enkripsi pada *DES* 13

Gambar 2.4 Enkripsi dan Dekripsi 15

Gambar 2.5 Diagram Enkripsi dan Dekripsi *Triple DES* dengan 3 buah Kunci 16

Gambar 3.1 Alur Sistem Kriptografi 18

Gambar 3.2 Desain Form Enkripsi 19

Gambar 3.3 Desain Form Dekripsi 21

Gambar 3.4 Desain Form *Borland Delphi 7* 23

Gambar 3.5 Desain Form Enkripsi Aplikasi Kriptografi 23

Gambar 3.6 Desain Form Dekripsi Aplikasi Kriptografi 25

Gambar 3.7 Diagram Konteks 26

Gambar 3.8 Flowchart Proses Enkripsi 27

Gambar 3.9 Flowchart Detail Enkripsi 28

Gambar 3.10 Flowchart Proses Dekripsi 30

Gambar 4.1 Implementasi Desain Kriptografi Algoritma *Triple DES* 32

Gambar 4.2 Enkripsi Citra Digital 33

Gambar 4.3 Tombol *"Load Carrier Image"* untuk Memilih File citra Digital 33

Gambar 4.4 *"Select Carrier Image"* File Citra Digital sebagai Cover Object 34

Gambar 4.5 Tombol *"Load File"* untuk Memilih File yang Dienkripsi 34

Gambar 4.6 *"Select File to Hide"* File yang Dienkripsi *Triple DES* 34

Gambar 4.7 Tombol *"Save Image As"* untuk Menyimpan File yang Baru 34

Gambar 4.8 *"Select File to Hide"* Nama Baru yang Dienkripsi *Triple DES* 35

Gambar 4.9 Melihat Citra Digital yang akan Dienkripsi 35

Gambar 4.10 untuk Atribut atau Rincian File 35
Gambar 4.11 *"Bits per Channel"* pada Enkripsi 35
Gambar 4.12 Enkripsi Kriptografi Algoritma *Triple DES* 36
Gambar 4.13 Button *"Encrypt"* 36
Gambar 4.14 *"Progress" Triple DES* 36
Gambar 4.15 *"Information"* Enkripsi *Triple DES* 36
Gambar 4.16 Kriptografi Enkripsi Citra Digital *Triple DES* 37
Gambar 4.17 *"Information Image" Triple DES* 37
Gambar 4.18 Dekripsi Citra Digital 38
Gambar 4.19 Tombol *"Load Carrier Image"* untuk Memilih File Citra Digital 38
Gambar 4.20 *"Select Carrier Image"* File Citra Digital yang Didekripsi 38
Gambar 4.21 Tombol *"Save File To"* untuk Mendekripsi File yang Terenkripsi 38
Gambar 4.22 *"Select Image to Extract the Data"* File yang Didekripsi *Triple DES* 39
Gambar 4.23 DekripsiKriptografi Algoritma *Triple DES* 39
Gambar 4.24 Button *"Decrypt"* 39
Gambar 4.25 *"Progress " Triple DES* 39
Gambar 4.26 *"Information"* Dekripsi *Triple DES* 40
Gambar 4.27 Kriptografi Enkripsi Citra Digital *Triple DES* 40
Gambar 4.28 *"Information Image" Triple DES* 40
Gambar 4.29 *"Information"* Perbandingan dari Citra Digital 41
Gambar 4.30 *"Help"* Tampilan Awal dari Bantuan 41
Gambar 4.31 *"Help"* Tampilan Enkripsi dari Bantuan 42
Gambar 4.32 *"Help"* Tampilan Dekripsi dari Bantuan 42
Gambar 4.33 Aplikasi Algoritma *Triple DES* 43
Gambar 4.34 Grafik Ukuran Kecepatan pada Bit ke-2 46

Gambar 4.35 Grafik Ukuran Kecepatan pada Bit ke-4.....47

DAFTAR TABEL

Tabel 2.1 Cara pengenkripsian dan pendekripsian 15

Tabel 3.1 Waktu dan Kecepatan Proses Enkripsi dengan Algoritma DES
Dan Algoritma Triple DES 29

Tabel 3.2 Waktu dan Kecepatan Proses Dekripsi dengan Algoritma DES
Dan Algoritma Triple DES 31

Tabel 4.1 Contoh File Citra Digital 44

Tabel 4.2 Perbandingan pada tiap-tiap Bit yang Dienkripsi 45

Tabel 4.3 Kecepatan Pada Bit Saat Enkripsi 46

Tabel 4.4 Perbandingan Data Biner dari *Pixel* (R-G-B) 48


INSTITUT TEKNOLOGI NASIONAL
ITN
MALANG

BAB I

PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi dan komunikasi menjadi salah satu media yang ada di dunia. Karena fasilitas serta kemudahan yang dimiliki teknologi dan komunikasi sudah menjadi sesuatu yang tidak asing lagi. Namun dengan perkembangan aplikasinya, maka semakin berkembang pula kriminalitas teknologi informasi dan komunikasi. Dengan berbagai macam teknik, banyak pihak yang mencoba untuk mengakses informasi yang bukan haknya. Maka dari itu sejalan dengan berkembangnya informasi ini harus juga diimbangi dengan perkembangan pengamanan.

Kriptografi telah dikenal sejak 4000 tahun yang lalu. Kriptografi dahulunya adalah usaha untuk mengubah pesan dengan menambah atau mengubah karakter tertentu. Kriptografi adalah ilmu sekaligus seni untuk menjaga keamanan pesan. Kriptografi adalah cabang dari ilmu matematika yang memiliki banyak fungsi dalam pengamanan data. Kriptografi dapat juga diartikan sebagai proses mengambil pesan dan menggunakan beberapa fungsi untuk menggenerasi materi kriptografis (pesan terenkripsi). Kriptografi adalah salah satu dari teknologi yang digunakan dalam layanan *security*, seperti *integrity*; *confidentiality*; *identity* dan *non repudiation.*[11][17]

Dalam kriptografi terdapat dua konsep utama yakni enkripsi dan dekripsi. Enkripsi adalah proses dimana informasi atau data yang hendak dikirim diubah menjadi bentuk yang hampir tidak dikenali sebagai informasi awalnya dengan menggunakan algoritma tertentu. Dekripsi adalah kebalikan dari enkripsi yaitu mengubah kembali bentuk tersamar tersebut menjadi informasi awal.[11][17]

Algoritma *DES* adalah standard yang dijadikan oleh *NBS*, biro standard nasional Amerika, sejak November 1976. *DES* menggunakan kunci berukuran 56-bit. Ukuran kunci mencerminkan kekuatan dari suatu algoritma enkripsi. Perkembangan

perangkat keras maupun perangkat lunak dan meluasnya penggunaan jaringan komputer terdistribusi menyebabkan penemuan bahwa algoritma *DES* sudah tidak aman lagi untuk digunakan, terutama dalam hal pengiriman data melalui internet. Perangkat keras yang khusus digunakan untuk mencari kunci dari *DES* dengan metode paling mendasar, *brute force attack*, dapat dibangun dalam beberapa jam saja. Hal ini disebabkan oleh ukuran kunci dari *DES* yang relatif pendek tadi, yaitu 56-bit. Pertimbangan-pertimbangan tersebut menandakan bahwa sebuah standard algoritma yang baru sangatlah diperlukan untuk tetap menjaga kerahasiaan dari suata data. Dalam hal ini, kunci yang lebih panjang juga merupakan keharusan.[8][12]

Salah satu alternatif lain adalah penggunan *triple DES* untuk menggantikan *DES*. Sebetulnya *triple DES* memiliki algoritma yang sama dengan *DES*, namun metode ini memperbesar ukuran kunci karena *triple DES* dibentuk dari algoritma *DES* dengan cara menggunakannya tiga kali. *Triple DES* mengenkripsi plainteks dengan satu kunci *DES* dan kemudian mengenkripsinya lagi dengan kunci *DES* kedua sama tidak amannya dengan enkripsi menggunakan satu kunci *DES*. Kelihatan bahwa jika kedua kunci memiliki n bit, serangan brute force untuk mencoba semua kunci memiliki yang mungkin akan membutuhkan $2^n \times 2^n \times 2^n$ kombinasi yang berbeda. Merkle dan Hellman menunjukkan bahwa plaintext yang diketahui, serangan *Man in the Middle* dapat memecahkan enkripsi ganda pada 2^{n+1} kali percobaan. Tipe serangan ini dicapai dengan mengenkripsi dari akhir, dan dekripsi dari yang lainnya, dan membandingkan hasilnya ditengah. Karena itu, *triple DES* digunakan untuk mendapatkan enkripsi yang lebih kuat. [8][12]

1.2. Rumusan Masalah

Berdasarkan latar belakang diatas, maka timbul suatu permasalahan didalam penelitian ini dicoba untuk bagaimana menerapkan sistem keamanan file tersebut diatas pada Aplikasi Kriptografi File Citra Digital, dan untuk referensinya maka penyusun akan melakukan penelitian mengenai kriptografi (metode enkripsi dan deskripsi) dengan menggunakan algoritma *triple DES* (*Triple data Encryption Standard*).

1.3. Tujuan Penelitian

Berdasarkan permasalahan yang ada, maka tujuan dari penelitian yang dilakukan adalah berikut :

1. Membangun Aplikasi Kriptografi File Citra Digital Menggunakan Algoritma *Triple DES* (*Triple data Encryption Standard*).
2. Untuk menguji tingkat keamanan data atau pesan dengan menggunakan Algoritma *Triple DES* (*Triple data Encryption Standard*).
3. Untuk melindungi file citra digital dari pihak-pihak yang tidak berhak.

1.4. Batasan Masalah

Agar implementasi lebih terarah dan memudahkan dalam pembahasan, maka perlu adanya batasan masalah pada penulisan skripsi ini yang diharapkan mampu membatasi pembahasan agar sesuai dengan tujuan penelitian itu sendiri. Adapun batasan masalah yang diajukan adalah sebagai berikut :

1. Lingkup dari pengembangan aplikasi kriptografi ini dibatasi pada penggunaan media gambar dengan format bitmap (.bmp).
2. Algoritma *Triple DES* (*Triple data Encryption Standard*) kriptografi yang digunakan konsep pada proses enkripsi dan dekripsi.
3. Ukuran citra digital bitmap atau media utama penyisipan data atau pesan adalah 1 - 1,5MB.
4. Media pesan yang disisipkan berukuran tidak lebih dari 200KB.

1.5. Manfaat Penelitian

Berikut adalah beberapa manfaat dari penelitian yang dilakukan :

1. Sebagai *security* (pengaman) data-data penting terutama file citra dari pihak yang tidak berhak.
2. Mengurangi tingkat percurian data yang sering terjadi.
3. Meminimalisir kerusakan data ketika dikirim kepada pihak lain saat pengiriman.

4. Menjadi jaminan atas identitas dan keabsahan data yang dikirim, karena menggunakan *proteksi key*.
5. Pengiriman data dapat dijadikan sebagai barang bukti yang tidak dapat disangkal pihak lain.

1.6. Metodologi

1.6.1. Metode Penelitian

Adapun metode penelitian yang digunakan adalah sebagai berikut:

1. Studi Literatur

   Pengumpulan data yang dilakukan dengan mencari bahan-bahan kepustakaan dan referensi dari berbagai sumber sebagai landasan teori yang ada hubunganya dengan permasalahan yang dijadikan objek penelitian.

2. Analisa Kebutuhan Sistem

   Data dan informasi yang telah diperoleh akan dianalisa agar didapatkan kerangka global yang bertujuan untuk mendefinisikan kebutuhan sistem baik hardware maupun software, di mana nantinya akan digunakan sebagai acuan perancangan sistem.

3. Perancangan Sistem

   Berdasarkan data dan informasi yang telah diperoleh serta analisa kebutuhan untuk membangun sistem ini, akan dibuat rancangan kerangka global yang menggambarkan mekanisme dari sistem yang akan dibuat.

4. Coding

   Tahapan ini menerjemahkan hasil perancangan spesifikasi program dari tahapan sebelumnya kedalam baris–baris kode program yang dapat dimengerti oleh komputer.

5. Eksperimen dan Evaluasi

   Pada tahap ini, sistem yang telah selesai dibuat akan diuji coba, yaitu pengujian berdasarkan fungsionalitas program, dan akan dilakukan koreksi dan penyempurnaan program jika diperlukan.

1.7. Sistematika Penulisan

Sistematika penulisan yang digunakan dalam penyusunan skripsi ini adalah sebagai berikut:

BAB I : Pendahuluan

Bab ini berisikan tentang Latar Belakang, Maksud dan Tujuan, Tempat dan Waktu Pelaksanaan, Ruang Lingkup, Metode Pengumpulan Data, dan Sistematika Penulisan.

BAB II : Landasan Teori

Berisi tentang landasan teori mengenai permasalahan yang berhubungan dengan penelitian yang dilakukan.

BAB III : Perancangan dan Analisa Sistem

Dalam bab ini berisi mengenai analisa kebutuhan sistem baik software maupun hardware yang diperlukan untuk membuat kerangka global yang menggambarkan mekanisme dari sistem yang akan dibuat.

BAB IV : Pembuatan dan Pengujian Sistem

Berisi tentang implementasi dari perancangan sistem yang telah dibuat serta pengujian terhadap sistem tersebut.

BAB V : Penutup

Merupakan bab terakhir yang memuat intisari dari hasil pembahasan yang berisikan kesimpulan dan saran yang dapat digunakan sebagai pertimbangan untuk pengembangan penulisan selanjutnya.


INSTITUT TEKNOLOGI NASIONAL
ITN
MALANG

BAB II

LANDASAN TEORI

2.1. File [18]

File atau Berkas adalah sekumpulan data (informasi) yang berhubungan yang diberi nama dan tersimpan di dalam media penyimpanan sekunder (*secondary storage*) file memiliki ekstensi. Ekstensi berkas merupakan penandaan jenis berkas lewat nama berkas. Ekstensi biasanya ditulis setelah nama berkas dipisahkan dengan sebuah tanda titik.

Pada sistem yang lama (MS-DOS) ekstensi hanya diperbolehkan maksimal tiga huruf, contohnya : exe, bat, com, dan txt. Batasan itu dihilangkan pada sistem yang lebih baru (*Windows*) contohnya : mpeg, java. Pada UNIX bahkan dikenal ada file yang memiliki lebih dari satu ekstensi, contohnya : tar.Z, tar.gz.

2.2. Citra Digital [16]

Citra digital merupakan citra hasil digitalisasi yang dapat diolah pada suatu komputer digital. Citra digital tersusun atas sejumlah elemen. Elemen-elemen yang menyusun citra digital disebut *pixel*. *Pixel* merupakan kependekan dari *picture element*, yang berarti element atau unsur penyusun citra digital. Kata *pixel* pertama kali dipublikasikan pada tahun 1965 oleh Frederic C. Bilingsley. Satu *pixel* berarti satu titik pada citra. Sebagai contoh: sebuah citra digital berdimensi 300x400 *pixel* (panjang: 300; lebar:400) tersusun atas 300x400 = 120.000 *pixel*.

Element pada citra digital terdiri dari :

1. Kecerahan (*Brightness*) merupakan intensitas cahaya.
2. Kontras (*Contrast*) merupakan sebaran terang dan gelap dalam sebuah citra.
3. Kontur (*Contour*) merupakan keadaan yang ditimbulkan oleh perubahan intensitas pada pixel-pixel yang bertetangga.
4. Warna (*Color*) merupakan persepsi yang dirasakan mata terhadap panjang gelombang cahaya λ yang dipantulkan objek.

5. Bentuk (*Shape*) yang dilihat mata 2D, object asli 3D.
6. Teksture (*Texture*) merupakan distribusi spasial dari derajat keabuan di dalam sekumpulan pixel yang bertetangga.

2.3. Bitmap (.bmp)

Bitmap adalah representasi dari citra grafis yang terdiri dari susunan titik yang tersimpan di memori komputer. Dikembangkan oleh *Microsoft* dan nilai setiap titik diawali oleh satu bit data untuk gambar hitam putih, atau lebih bagi gambar berwarna.

Ukuran sebenarnya untuk n-bit (2^n warna) bitmap dalam byte dapat dihitung:

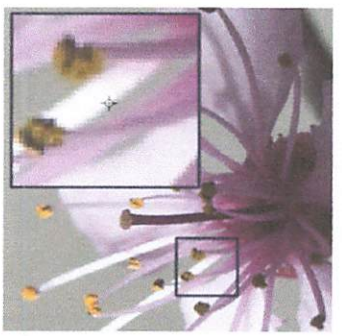
Ukuran file BMP

$$\approx 54+4\cdot 2^n+\frac{lebar\cdot tinggi\cdot n}{8}$$, dimana tinggi dan lebar dalam *pixel*.[13]

Kerapatan titik-titik tersebut dinamakan resolusi, yang menunjukkan seberapa tajam gambar ini ditampilkan, ditunjukkan dengan jumlah baris dan kolom, contohnya 1024x768. [13][18]

Untuk menampilkan citra bitmap pada monitor atau mencetaknya pada printer, komputer menterjemahkan bitmap ini menjadi *pixel* (pada layar) atau titik tinta (pada printer). Beberapa format file bitmap yang populer adalah *BMP, PCX dan TIFF*.[18]

Sebuah Bitmap adalah salah satu tipe file dari banyak tipe file untuk image yang disimpan dalam bentuk data berekstensi BMP. Komputer memakai bilangan bit 1 dan 0 ke dalam penyimpanan data. Sebuah Bitmap adalah kumpulan-kumpulan banyak bit yang membentuk sebuah gambar saat di render ke sebuah tampilan seperti monitor pada komputer.[13][18]



Gambar 2.1. Gambar Bitmap

Kelebihan Grafis Bitmap [18]

1. Dapat ditambahkan efek khusus tertentu sehingga dapat membuat objek tampil sesuai keinginan.
2. Dapat menghasilkan objek gambar bitmap darionjek gambar vektor dengan cara mudah dan cepat, mutu hasilnya pun dapat ditentukan.

Kelemahan Grafis Bitmap [18]

1. Objek gambar tersebut memiliki permasalahan ketika diubah ukurannya, khususnya ketika objek gambar diperbesar.
2. Efek yang diidapat dari objek berbasis bitmap yakni akan terlihat pecah atau berkurang detailnya saat dicetak pada resolusi yang lebih rendah.

Bitmap dikenal dengan istilah ransel, adalah susunan piksel yang kecil dan dimana setiap piksel sudah dipetakan pada posisi masing-masing dan memiliki warna warni yang dipersentasikan secara numerik. Ataupun juga dapat disebut sebuah struktur data yang mewakili susunan piksel warna yang ditampilkan pada layar monitor, kertas atau media tampilan lainnya.[18]

2.4. Kriptografi

Kriptografi adalah suatu ilmu yang mempelajari bagaimana cara menjaga agar data atau pesan tetap aman saat dikirimkan, dari pengirim ke penerima tanpa mengalami gangguan dari pihak ketiga. Menurut *Bruce Scheiner* dalam bukunya "*Applied Cryptography*", kriptografi adalah ilmu pengetahuan dan seni menjaga pesan-pesan agar tetap aman(*secure*).[11][15]

Kriptografi adalah ilmu sekaligus seni untuk menjaga keamanan pesan. Kriptografi adalah cabang dari ilmu matematika yang memiliki banyak fungsi dalam pengamanan data. Kriptografi dapat juga diartikan sebagai proses mengambil pesan dan menggunakan beberapa fungsi untuk menggenerasi materi kriptografis (pesan terenkripsi). Kriptografi adalah salah satu dari teknologi yang digunakan dalam layanan *security*, seperti *integrity*; *confidentiality*; *identity* dan *non repudiation*.[11][15]

Algoritma kriptografi yang baik tidak ditentukan oleh kerumitan dalam mengolah data atau pesan yang akan disampaikan. Ada 4 syarat yang perlu dipenuhi, yaitu:

1. Kerahasiaan.
   Pesan (*plaintext*) hanya dapat dibaca oleh pihak yang memliki kewenangan.
2. *Autentikasi.*
   Pengirim pesan harus dapat diidentifikasi dengan pasti, penyusup harus dipastikan tidak bisa berpura-pura menjadi orang lain.
3. *Integritas.*
   Penerima pesan harus dapat memastikan bahwa pesan yang dia terima tidak dimodifikasi ketika sedang dalam proses transmisi data.
4. *Non-Repudiation.*
   Pengirim pesan harus tidak bisa menyangkal pesan yang dia kirimkan.[8][12]

Algoritma sandi adalah algoritma yang berfungsi untuk melakukan tujuan kriptografis. Secara umum berdasarkan kesamaan kuncinya, algoritma sandi dibedakan menjadi:

a. Kunci simetris (*symetric key*)
   Skema algoritma sandi akan disebut kunci simetris apabila untuk setiap proses enkripsi maupun dekripsi data secara keseluruhan digunakan kumci yang sama. Contoh algoritma yang menggunakan kunci simetris: *DES, MARS, IDEA, Triple DES, AES.*
b. Kunci asimetris (*asymetric key*)
   Skema ini adalah algoritma yang menggunakan kunci yang berbeda untuk proses enkripsi dan dekripsinya. Contoh algoritma yang menggunakan kunci asimetris: *Knapsack, RSA, Diffie Hellman.*[8][10][12]

Algoritma enkripsi yang paling banyak digunakan di dunia adalah *Data Encryption Standard (DES)* yang diadopsi oleh *NIST* (*National Institute of Standards and Technology*) sebagai standar pengolah informasi Federal AS. Data dienkrip dalam blok-blok 64 bit menggunakan kunci 56 bit. *DES* mentransformasikan input 64

bit dalam beberapa tahap enkripsi ke dalam output 64 bit. Dengan demikian, *DES* termasuk keluarga *block cipher*. Dengan tahapan dan kunci yang sama, *DES* digunakan untuk membalik enkripsi. [3][8][12]

Sedangkan berdasarkan besar data yang diolah dalam satu kali proses, maka algoritma kriptografi dapat dibedakan menjadi dua jenis yaitu :

- Algoritma *block cipher*

  Informasi/data yang hendak dikirim dalam bentuk blok-blok besar (misal 64-bit) dimana blok-blok ini dioperasikan dengan fungsi enkripsi yang sama dan akan menghasilkan informasi rahasia dalam blok-blok yang berukuran sama.
- Algoritma *stream cipher*

  Informasi/data yang hendak dikirim dioperasikan dalam bentuk blok-blok yang lebih kecil (byte atau bit), biasanya satu karakter persatuan persatuan waktu proses, menggunakan tranformasi enkripsi yang berubah setiap waktu.[8][12]

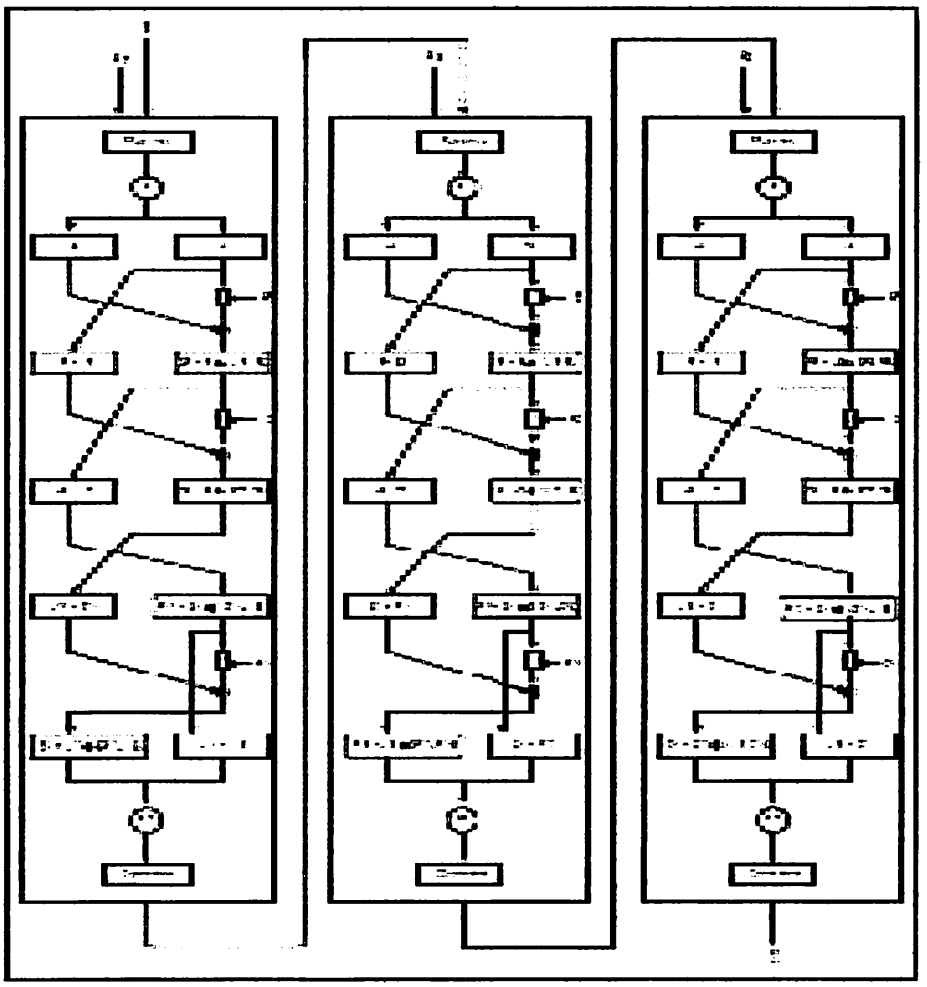
2.5. Pembahasan Algoritma

Algoritma enkripsi atau dekripsi *triple DES* seperti algoritma kriptografi lainnya yaitu memiliki algoritma umum. Pada pembahasan ini akan diberikan gambaran secara umum tentang algoritma *triple DES* dan merancang prosedur tentang pembuatan aplikasi enkripsi atau dekripsinya serta memberikan penjelasan dari prosedur-prosedur yang digunakan.[1][7][20]

2.5.1. Algoritma *Triple DES*

Triple DES (*Triple Data Encryption Standard*) merupakan suatu algoritma pengembangan dari algoritma *DES* (*Data Encryption Standard*). Pada dasarnya algoritma yang digunakan sama, hanya pada *triple DES* dikembangkan dengan melakukan enkripsi dengan implementasi algoritma *DES* sebanyak tiga kali. *Triple DES* memiliki tiga buah kunci yang berukuran 168-bit (tiga kali kunci 56 bit dari *DES*). Pada algoritma *triple DES* dibagi menjadi tiga tahap, setiap tahapnya

merupakan implementasi dari algoritma *DES*. Gambar 2.2 menjelaskan tentang tahapan algoritma *Triple DES*. [1][7][20]



Gambar 2.2. Algoritma *TRIPLE DES* secara detail

Tahap pertama, plainteks yang diinputkan dioperasikan dengan kunci eksternal pertama (K1) dan melakukan proses enkripsi dengan menggunakan algoritma *DES*. Sehingga menghasilkan pra-cipherteks pertama. Tahap kedua, pra-cipherteks pertama yang dihasilkan pada tahap pertama, kemudian dioperasikan dengan kunci eksternal kedua (K2) dan melakukan proses enkripsi atau proses dekripsi (tergantung cara pengenkripsian yang digunakan) dengan menggunakan algoritma DES. Sehingga menghasilkan pra-cipherteks kedua. Tahap terakhir, pra-cipherteks kedua yang dihasilkan pada tahap kedua, dioperasikan dengan kunci eksternal ketiga (K3) dan

melakukan proses enkripsi dengan menggunakan algoritma *DES*, sehingga menghasilkan cipherteks(C).[2][7][20]

2.5.2.Enkripsi

Proses enkripsi algoritma *triple DES* dapat dicapai dengan beberapa cara, yaitu dengan menggunakan dua buah kunci atau tiga buah kunci:

1. Dengan dua buah kunci

   Enkripsi: C = Ek1(Dk2(Ek1(P)))

   Penjelasan:

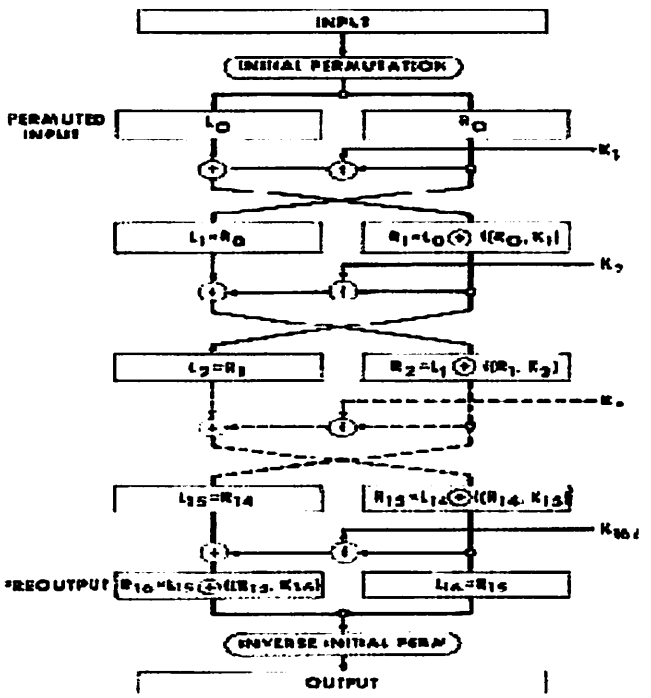
   Enkripsi pesan *P* mula-mula dengan kunci K1, lalu hasilnya didekripsi lagi dengan kunci K2 kemudian dienkripsi lagi dengan kunci K1 dan hasil enkripsi terakhir adalah cipherteks (*C*).
2. Dengan tiga buah kunci

   Enkripsi: C = Ek3(Dk2(Ek1(P)))

   Penjelasan:

   Enkripsi pesan *P* mula-mula dengan kunci K1, lalu hasilnya didekripsi lagi dengan kunci K2 kemudian dienkripsi lagi dengan kunci K3 dan hasil enkripsi terakhir adalah cipherteks (*C*). [1][3][8]

   Berikut adalah gambar dari proses enkripsi pada *DES*



Gambar 2.3 Proses Enkripsi pada DES

Plainteks yang diinputkan pertama akan disubatitusikan pada matriks permutasi awal (*initial permutation*) atau IP panjangnya 64-bit. Kemudian dibagi menjadi dua bagian, yaitu kiri (*L*) dan kanan (*R*) masing-masing panjangnya menjadi 32-bit. Kedua bagian ini masuk ke dalam 16 putaran *DES*. Satu putaran *DES* merupakan model jaringan Feistel, secara matematis jaringan Feistel dinyatakan sebagai berikut:

$$L_i = R_{i-1} \qquad ; 1 \le i \le 16 \qquad DES$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, k_i) \qquad Triple\ DES$$

Bagian R disubstituaikan pada fungsi ekspansi panjangnya menjadi 48-bit kemudian di-XOR-kan dengan kunci internal yang sudah diproses sebelumnya pada proses pembangkitan kunci (pada putaran pertama menggunakan kunci internal pertama, dan seterusnya). Hasil XOR kemudian disubstitusikan pada *S-box* yang dikelompokkan menjadi 8 kelompok, masing-masing 6-bit hasilnya menjadi 4-bit. Kelompok 6-bit pertama menggunakan *S1*, kelompok 6-bit kedua menggunakan *S2*, dan seterusnya. Setelah proses *S-box* tersebut panjangnya menjadi 32-bit.

Kemudian disubstitusikan lagi pada matriks permutasi *P-box*, kemudian di-XOR-kan dengan bagian *L*. Hasil dari XOR tersebut disimpan untuk bagian *R* selanjutnya. Sedangkan untuk bagian *L* diperoleh dari bagian *R* yang sebelumnya. Proses tersebut dilakukan 16 kali. Setelah 16 putaran selesai, bagian *L* dan *R* digabungkan dan disubstitusikan pada matriks permutasi awal balikan (*invers initial permutation*) atau IP-1, hasilnya merupakan cipherteks 64-bit.

2.5.3. Dekripsi

Proses dekripsi algoritma *triple DES* dapat dicapai dengan beberapa cara, yaitu dengan menggunakan dua buah kunci atau tiga buah kunci:

1. Dengan dua buah kunci

   Dekripsi: P = Dk1(Ek2(Dk1(C)))

   Penjelasan:

   Mula-mula kunci K1 digunakan untuk mendekripsi *C*, lalu hasilnya dienkripsi lagi dengan kunci K2 kemudian didekripsi lagi dengan kunci K1dan hasil dekripsi terakhir adalah pesan semula (*P*).
2. Dengan tiga buah kunci

   Dekripsi: P = Dk1(Ek2(Dk3(C)))

   Penjelasan:

   Mula-mula kunci K3 digunakan untuk mendekripsi *C*, lalu hasilnya dienkripsi lagi dengan kunci K2 kemudian didekripsi lagi dengan kunci K1dan hasil dekripsi terakhir adalah pesan semula (*P*).[1][8]

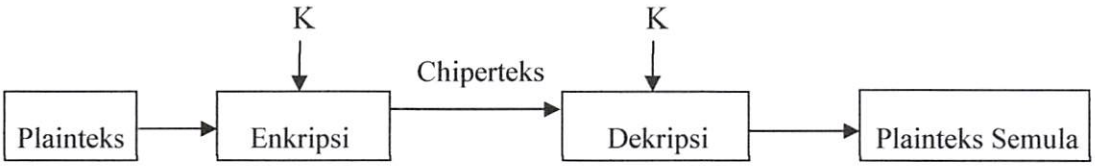
2.5.4. Pemilihan Kunci

Ada dua pilihan untuk pemilihan kunci eksternal algoritma *Triple DES*, yaitu:

a. K1, K2, dan K3 adalah kunci-kunci yang saling bebas K1 ≠ K2 ≠ K3 ≠ K1
b. K1 dan K2 adalah kunci-kunci yang saling bebas, dan K3 sama dengan K1 K1 ≠ K2 dan K3 = K1.[10][14]

2.5.5. Proses Enkripsi dan Dekripsi

Proses menyandikan plainteks menjadi cipherteks disebut enkripsi (*encryption*) atau *enciphering* (standard nama menurut ISO 7498-2). Proses mengembalikan cipherteks menjadi plainteksnya disebut dekripsi (*decryption*) atau *deciphering* (standard nama menurut ISO 7498-2).[8]



Gambar 2.4 Enkripsi dan Dekripsi

Proses enkripsi dan dekripsi algoritma *triple DES* dapat dicapai dengan beberapa cara, yaitu:

Cara	Enkripsi	Dekripsi
1	DES – EDE2 ▪ $K_1 \neq K_2$, $K_3 = K_1$ ▪ $C = E[D\{E(P, K_1), K_2\}, K_3]$	DES – DED2 ▪ $K_1 \neq K_2$, $K_3 = K_1$ ▪ $P = D[E\{D(C, K_3), K_2\}, K_1]$
2	DES – EEE2 ▪ $K_1 \neq K_2$, $K_3 = K_1$ ▪ $C = E[E\{E(P, K_1), K_2\}, K_3]$	DES – DDD2 ▪ $K_1 \neq K_2$, $K_3 = K_1$ ▪ $P = D[D\{D(C, K_3), K_2\}, K_1]$
3	DES – EDE3 ▪ $K_1 \neq K_2 \neq K_3 \neq K_1$ ▪ $C = E[D\{E(P, K_1), K_2\}, K_3]$	DES – DED3 ▪ $K_1 \neq K_2 \neq K_3 \neq K_1$ ▪ $P = D[E\{D(C, K_3), K_2\}, K_1]$
4	DES – EEE3 ▪ $K_1 \neq K_2 \neq K_3 \neq K_1$ ▪ $C = E[E\{E(P, K_1), K_2\}, K_3]$	DES – DDD3 ▪ $K_1 \neq K_2 \neq K_3 \neq K_1$ ▪ $P = D[D\{D(C, K_3), K_2\}, K_1]$

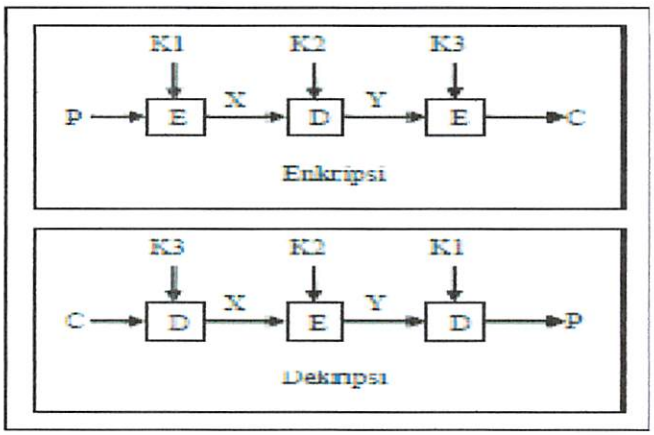
Tabel 2.1 Cara pengenkripsian dan pendekripsian

Varian ini umum dikenal sebagai mode EEE (untuk enkripsi) karena pada proses enkripsi semuanya menggunakan enkripsi. Untuk menyederhanakan interoperability antara *DES* dan *triple DES*, maka langkah ditengah (pada proses enkripsi *triple DES*) diganti dengan dekripsi (mode EDE). Dengan pengubahan ini, maka dibuat beberapa versi *triple DES*. Versi pertama *triple DES* menggunakan dua buah kunci, k1 dan k2:

Enkripsi: C = Ek1(Dk2(Ek1(P)))

Dekripsi: P = Dk1(Ek2(Dk1(C)))

Enkripsi DES tunggal dengan kunci K dapat dinyatakan sebagai *triple DES* - EDE with K1 = K2 = K. Gambar dibawah ini memperlihatkan versi *triple DES* yang menggunakan dua buah kunci. Penggunaan enkripsi pada langkah ditengah tidak mempengaruhi keamanan algoritma. Untuk lebih jelas lihat Gambar 2.4.



Gambar 2.5 Diagram enkripsi dan dekripsi *Triple DES* dengan 3 buah kunci.

Secara umum, *triple DES* dengan dua buah kunci mempunyai panjang kunci 2 x 56 = 112 bit, jauh lebih pendek daripada *triple DES* dengan tiga buah kunci yang mempunyai panjang kunci 3 x 56 = 168 bit.[1][2][8][14]

2.5.6. Penggunaan *Triple DES* [20]

Penggunaan *triple DES* semakin hari semakin menurun digantikan oleh *Advanced Encryption Standard (AES)*. Sebuah pengecualian dalam skala besar adalah dalam industri pembayaran elektronik yang masih menggunakan *Double DES* dan secara ekstensif mengembangaman standardnya. Ini menjamin bahwa TRIPLE DES akan tetap aktif di dunia kriptografi hingga masa yang belum dapat ditentukan.

Secara desain, *DES* dan juga *triple DES*, cenderung lambat pada perangkat lunak, pada prosesor modern, *AES* cenderung lebih cepat. *Triple DES* lebih cocok untuk implementasi perangkat keras, walaupun *AES* masih tetap lebih cepat.

2.5.7. Keamanan *Triple DES* [8]

Secara umum *triple DES* dengan tiga kunci berbeda memiliki kunci berukuran 168-bit (3 kali kunci 56-bit dari *DES*), namun dengan metode *meet in the middle* keamanan yang diberikan hanyalah 112-bit. Sebuah varian, *Double DES*, menggunakan kunci k1=k3, yang berarti mengecilkan ukuran kunci ke 112-bit dan ukuran *storage* menjadi 128-bit. Namun mode ini lemah terhadap beberapa serangan jenis *chose-plaintext* atau *knownplaintext*. Oleh sebab itu, mode ini biasanya hanya didesain dengan keamanan 80-bit.

2.5.8. Tingkat Kerahasiaan Kunci [8]

Semakin panjang kunci yang digunakan, semakin kuat tingkat kerahasiaannya. Algoritma *triple DES* menggunakan kunci yang panjangnya 168 bit, maka jumlah seluruh kombinasi kemungkinan kunci yang harus dicoba untuk memecahkan cipherteks adalah 2168 = 3,741x1050 kali. Karena, ada 168 posisi pengisian bit yang masing-masing mempunyai dua nilai kemungkinan, yaitu 0 dan 1.

2.5.9. Kekuatan Terhadap Serangan *Brute Force* [8]

Brute force adalah teknik mencoba satu persatu kemungkinan kunci untuk memperoleh plainteks. Waktu yang diperlukan untuk mencoba seluruh kemungkinan kunci oleh serangan *brute force* adalah seperti berikut:

$$\frac{2^{168}}{3600 \times 24 \times 366} = \frac{3.741 \times 10^{50}}{31.622.400} = 1.183 \times 10^{43}$$ tahun (*Risanto, 2006*).

2.6. Borland Delphi 7 [21]

Borland Delphi merupakan pilihan dalam pembuatan aplikasi visual karena memberikan produktivitas yang tinggi. Delphi merupakan program aplikasi database yang berbasis Object Pascal dari Borland. Delphi juga memberikan fasilitas pembuatan aplikasi visual. Di dalam delphi 7 memberikan fasilitas untuk dua *platform*, yaitu untuk *platform Windows* dan *Linux*. *Library* untuk *Windows* disebut VCL dan *Library* untuk *Linux* disebut *CLX*.


INSTITUT TEKNOLOGI NASIONAL
ITN
MALANG

BAB III

ANALISIS DAN PERANCANGAN

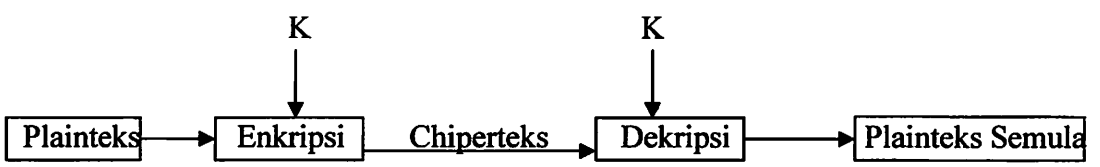
Analisis dan perancangan sistem berfungsi untuk mempermudah dalam memahami dan menyusun tahapan selanjutnya yang akan dilakukan, menguraikan dari suatu sistem yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan-permasalahan sehingga ditemukan kelemahan, kesempatan, dan hambatan yang terjadi serta kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya.

3.1. Analisis Sistem

Aplikasi yang akan dibuat ini adalah sebuah aplikasi kriptografi dengan file citra digital sebagai bahan sisipan dimana fungsi utama dari aplikasi kriptografi ini adalah agar dapat menyisipkan sebuah file data informasi, serta dapat mengdekripsi kembali data tersebut tanpa mngurangi informasi didalamnya.

3.2. Alur Sistem

Alur sistem merupakan suatu alat bantu untuk memudahkan penjelasan cara kerja algoritma dan aliran data yang akan disandikan. Objek masukan guna untuk proses kriptografi berupa file citra digital yang berformat (.bmp), pesan rahasia bisa berupa file yang berekstensi (.doc), (.xls), (.txt), dan (.jpeg). *Triple DES* sebagai kunci pengaman. Sedangkan hasil dari proses pengenkripsian ini adalah citra digital yang sudah memiliki pesan rahasia didalamnya. Sedangkan pada proses dekripsi, objek masukan berupa citra digital yang berformat (.bmp) yang memiliki pesan didalamnya dan juga algoritma *triple DES* yang digunakan pada saat proses penyisipan pesan sebelumnya.



Gambar 3.1 Alur Sistem Kriptografi

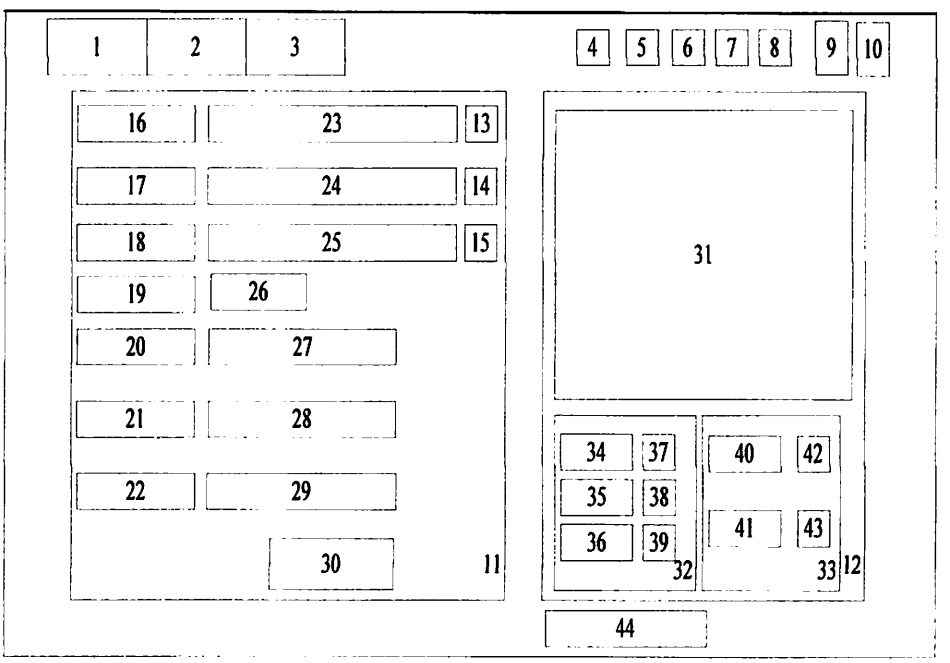
3.2.1.Sistem Enkripsi

Sistem untuk pengenkripsian pada kriptografi, membutuhkan data berupa file citra digital sebagai media penyisipan, pesan yang ingin disisipkan, serta algoritma *triple DES* sebagai pengaman. Citra digital yang digunakan sebagai media penyisipan pesan berformat (.bmp).

3.3. Desain Antarmuka Aplikasi

Dalam rancangan pembuatan perangkat lunak kriptografi berikut digunakan program *Borland Delphi 7* berikut adalah desain dari beberapa form yang dibuat dalam aplikasi algoritma kriptografi ini.

3.3.1.Desain Form Enkripsi

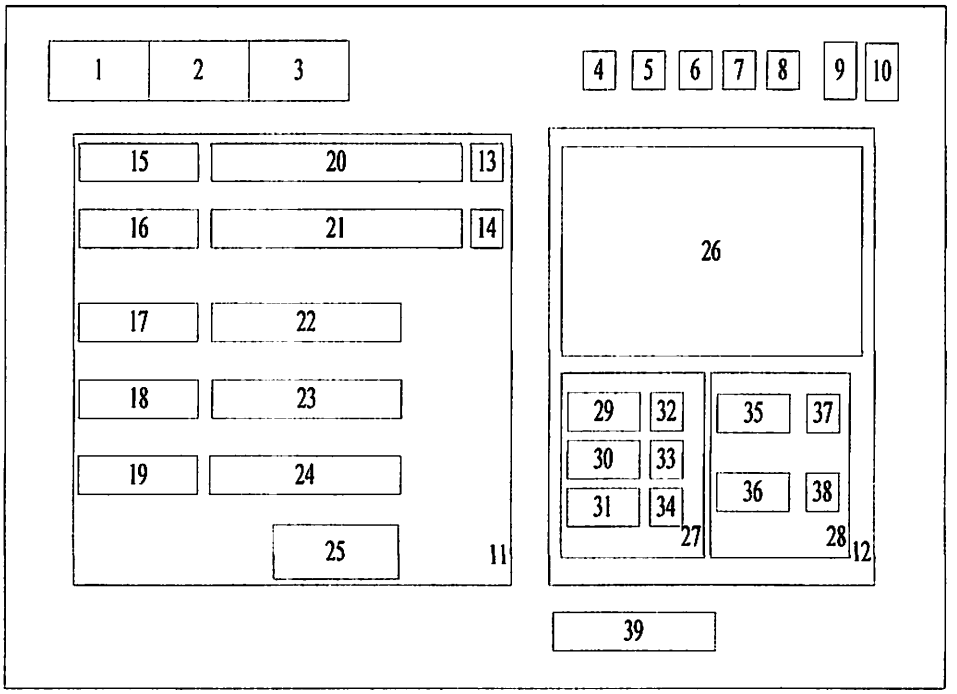


Gambar 3.2 Desain Form Enkripsi

Berikut ini adalah keterangan dari desain form enkripsi :

1. Sui Form
2. Sui Form
3. Sui Form
4. Save Dialog
5. Open Dialog
6. Open Picture Dialog
7. Save Picture Dialog
8. XP Manifest
9. Message Dialog
10. Sui Main Menu
11. Group Box
12. Group Box
13. Sui Image Button
14. Sui Image Button
15. Sui Image Button
16. Label
17. Label
18. Label
19. Label
20. Label
21. Label
22. Label
23. Sui Edit
24. Sui Edit
25. Sui Edit
26. Sui Edit and Up Down
27. Sui Edit
28. Sui Edit
29. Sui Edit
30. Sui Image Button
31. Sui Image Panel
32. Group Box
33. Group Box
34. Label
35. Label
36. Label
37. Label
38. Label
39. Label
40. Label
41. Label
42. Label
43. Label
44. Label and Progress Bar

3.3.2.Desain Form Dekripsi



Gambar 3.3 Desain Form Dekripsi

Berikut ini adalah keterangan dari desain form dekripsi :

1. Sui Form
2. Sui Form
3. Sui Form
4. Save Dialog
5. Open Dialog
6. Open Picture Dialog
7. Save Picture Dialog
8. XP Manifest
9. Message Dialog
10. Sui Main Menu
11. Group Box
12. Group Box
13. Sui Image Button
14. Sui Image Button
15. Label
16. Label
17. Label
18. Label
19. Label
20. Sui Edit
21. Sui Edit
22. Sui Edit
23. Sui Edit
24. Sui Edit
25. Sui Image Button
26. Sui Image Panel
27. Group Box
28. Group Box
29. Label
30. Label
31. Label
32. Label
33. Label
34. Label
35. Label
36. Label
37. Label
38. Label
39. Label and Progress Bar

3.4. Perancangan Antarmuka

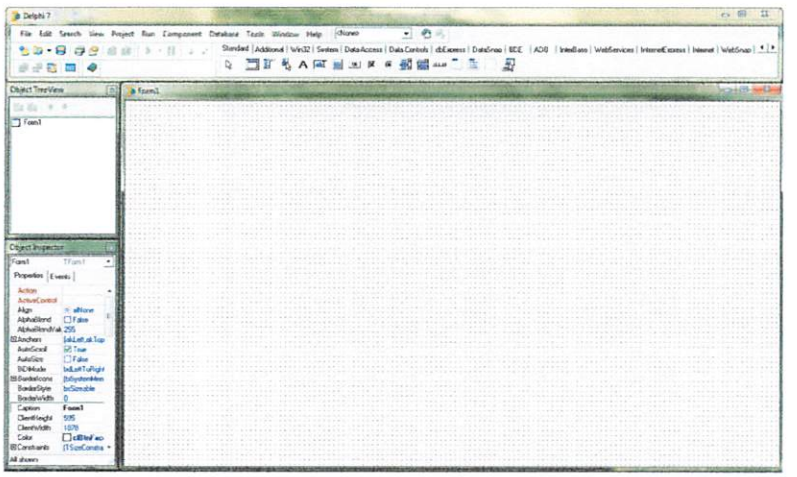
Perancangan antarmuka merupakan langkah awal dari pembuatan aplikasi ini. Bahasa pemrograman yang dipakai dalam perancangan aplikasi kriptografi ini menggunakan *Borland Delphi 7*.

Dalam perancangan ini, ada beberapa tahapan yang dilakukan. Berikut ini adalah rincian dari tahapan tersebut.

3.4.1.Mendesain Tampilan *Form* Utama

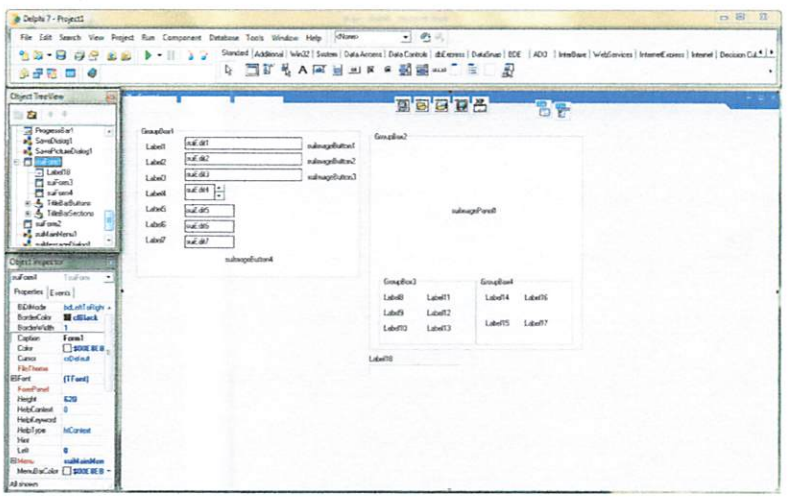
Dalam mendesain tampilan utama, digunakan komponen-komponen yang ada pada *ToolBox* dari *Borland Delphi 7*. Berikut ini adalah langkah-langkah dari pendesainan aplikasi kriptografi.

- Langkah pertama mendesain tampilan *form* utama adalah dengan membuka program *Borland Delphi 7*. Setelah itu buat *project* baru dengan mengklik *file-new-application.*



Gambar 3.4 Desain Form *Borland Delphi 7*

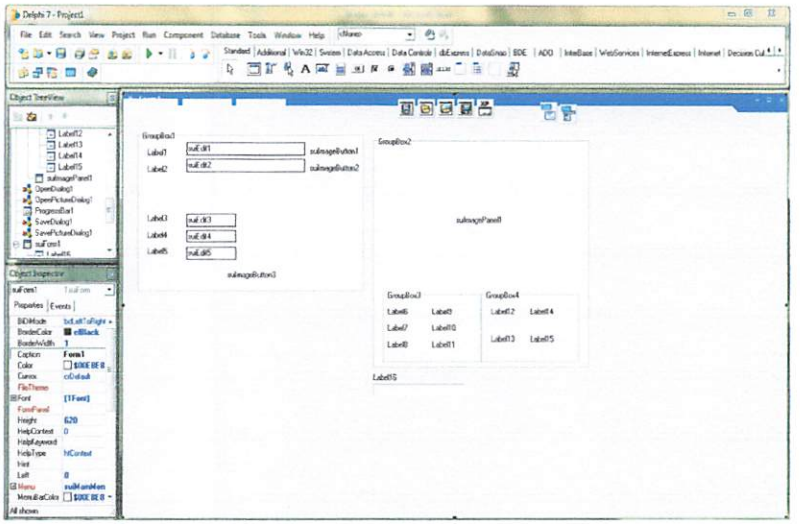
- Dilanjutkan dengan memasukkan *form* yang dibutuhkan dalam *form* enkripsi pada aplikasi kriptografi sebagai berikut:



Gambar 3.5 Desain Form Enkripsi Aplikasi Kriptografi

```
//============================== Encrypt Image ==============================//

procedure TForm_Utama.Load_ImageClick(Sender: TObject);
var
  strimage: String;
begin
  if (OpenPictureDialog.Execute) then
    begin
      strimage:= OpenPictureDialog.FileName;
      ImagePanel.Picture.LoadFromFile(strimage);
      LoadImage.Text:= strimage;
      if GetSize(OpenPictureDialog.FileName) > 4194304 then
        ShowMessage('Ukuran file lebih dari 4 MB')
      else
        imagesize.Caption:= FormatByteSize(GetSize(OpenPictureDialog.FileName));
        imageheight.Caption:= IntToStr(ImagePanel.Picture.Height) + ' Pixel';
        imagewidth.Caption:= IntToStr(ImagePanel.Picture.Width) + ' Pixel';
    end;
end;

procedure TForm_Utama.Load_FileClick(Sender: TObject);
var
  strfile: String;
begin
  if (OpenDialog.Execute) then
  begin
    strfile:= OpenDialog.FileName;
    LoadFile.Text:= strfile;
    filename.Caption:= ExtractFileName(OpenDialog.FileName);
    filesize.Caption:= FormatByteSize(GetSize(OpenDialog.FileName));
  end;
end;
```

- Dilanjutkan dengan memasukkan *form* yang dibutuhkan dalam *form* dekripsi pada aplikasi kriptografi sebagai berikut:

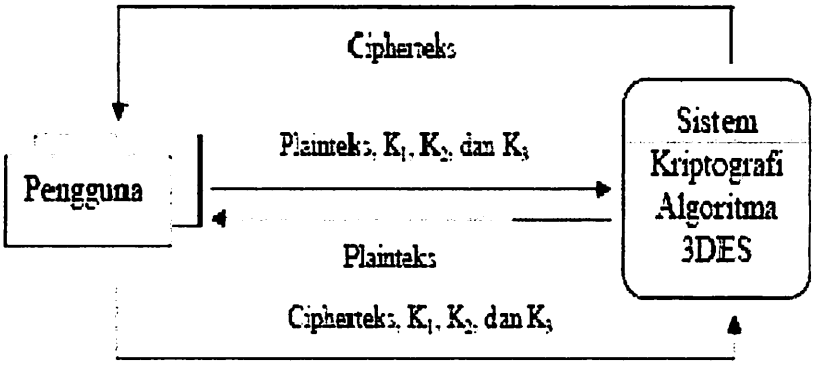


Gambar 3.6 Desain Form Dekripsi Aplikasi Kriptografi

```
//============================== Decrypt Image ==============================//

procedure TForm_Utama.Load_enc_ImageClick(Sender: TObject);
var
  strencimg: String;
begin
  if (OpenPictureDialog.Execute) then
  begin
    strencimg:= OpenPictureDialog.FileName;
    ImagePanel.Picture.LoadFromFile(strencimg);
    LoadEncImage.Text:=strencimg;

    imagesize.Caption:= FormatByteSize(GetSize(OpenPictureDialog.FileName));
    imageheight.Caption:= IntToStr(ImagePanel.Picture.Height) + ' Pixel';
    imagewidth.Caption:= IntToStr(ImagePanel.Picture.Width) + ' Pixel';
  end;
end;

procedure TForm_Utama.Save_FileClick(Sender: TObject);
var
  strencfile: String;
begin
  if (SaveDialog.Execute) then
  begin
    strencfile:= SaveDialog.FileName;
    SafeFile.Text:= strencfile;
  end;
end;

procedure TForm_Utama.btnDecryptClick(Sender: TObject);
begin
  try
```

3.5. Perancangan Sistem

Perancangan perangkat lunak disini meliputi dua bagian penting yaitu proses enkripsi dan dekripsi. Perancangan dimulai dengan pembuatan diagram konteks, berupa gambaran sistem penerapan algoritma *triple DES* secara garis besar :



Gambar 3.7 Diagram Konteks

3.6. Perancangan Flowchart

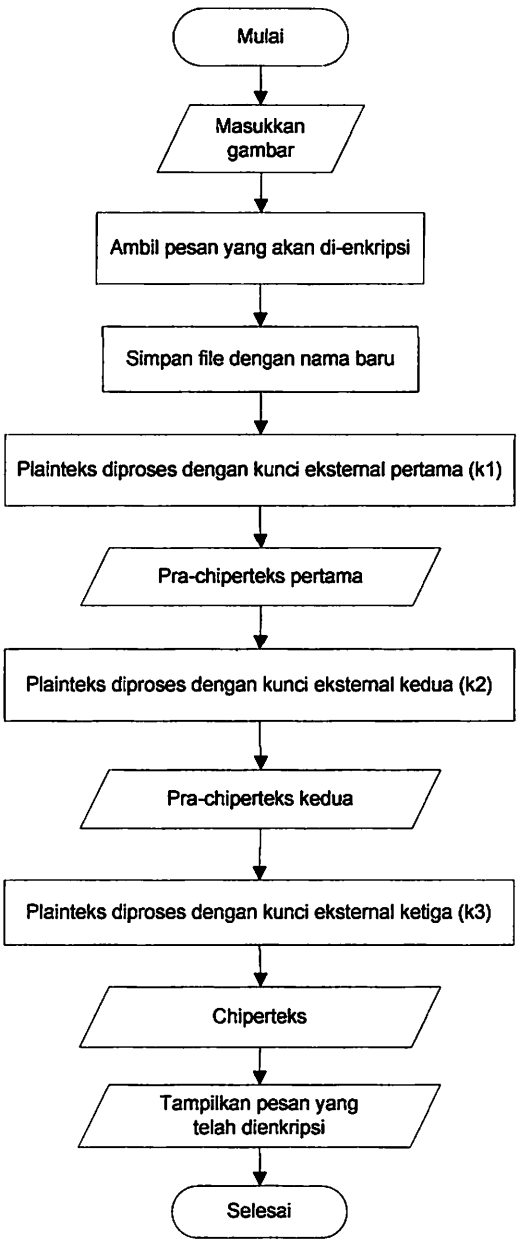
Flowchart adalah penyajian yang sistematis tentang proses dan logika dari kegiatan penanganan informasi atau penggambaran secara grafik dari langkah langkah dan urut-urutan prosedur dari suatu program. Flowchart menolong analis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif lain dalam pengoperasian.

Beberapa flowchart dibawah ini merupakan prosedur dari program enkripsi dan dekripsi *triple DES* yaitu proses enkripsi dan proses dekripsi.

3.6.1. Flowchart Proses Enkripsi

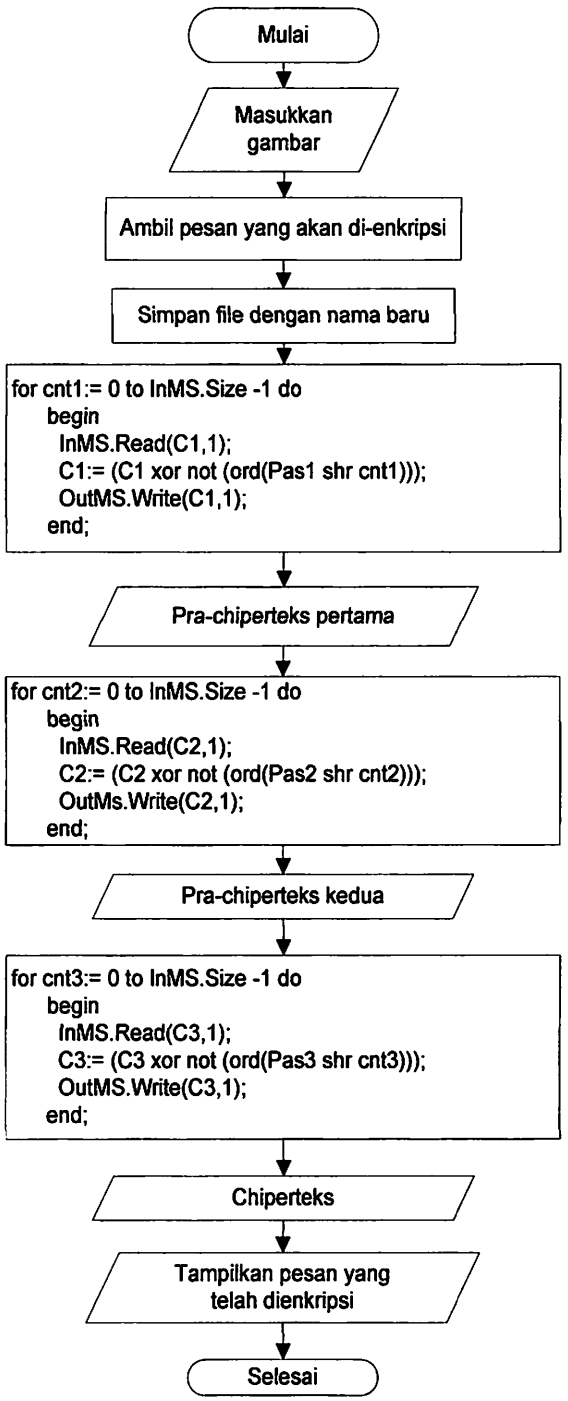
Flowchart proses enkripsi merupakan aliran proses untuk menjalankan program enkripsi yang digunakan untuk mengenkripsi file berupa teks. Pada tampilan program enkripsi ini, pengguna dapat mencari file untuk dienkripsi dengan menggunakan tombol pencari dan tombol penyimpan untuk menyimpan hasil enkripsi tersebut.

Untuk lebih detail dan jelasnya dilihat pada Gambar 3.8 berikut ini :



Gambar 3.8 Flowchart Proses Enkripsi

Berikut juga menampilkan flowchart detail enkripsi pada gambar 3.9 dibawah ini:



Gambar 3.9 Flowchart detail Enkripsi

Waktu Proses dan Kecepatannya untuk Proses Enkripsi dengan Algoritma DES dan Algoritma triple DES.

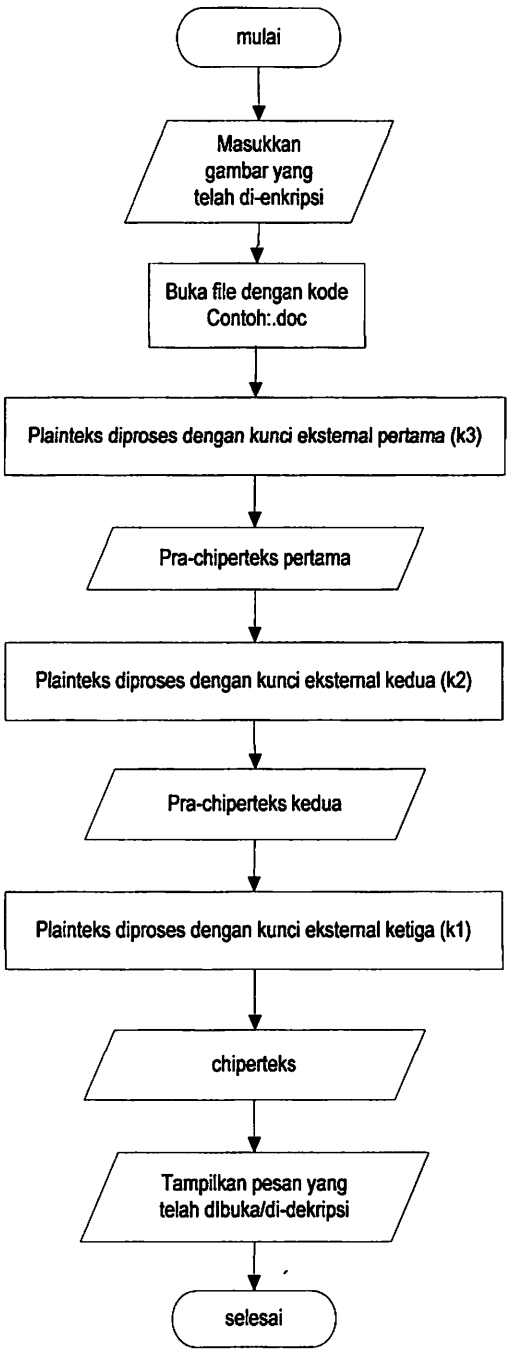
No	Nama File			Ukuran File (KB)		Waktu Proses (detik)		Kecepatan (KB/detik)	
	Input	Output							
		DES	3DES	Input	Output	DES	3DES	DES	3DES
1	P1.txt	EP1 DES.txt	EP1 3DES.txt	1	2	11.34	33.093	0.08818	0.03022
2	P2.txt	EP2 DES.txt	EP2 3DES.txt	2	4	22.658	66.197	0.08827	0.03021
3	P3.txt	EP3 DES.txt	EP3 3DES.txt	3	6	33.98	99.302	0.08829	0.03021
4	P4.txt	EP4 DES.txt	EP4 3DES.txt	4	8	45.26	132.324	0.08838	0.03023
5	P5.txt	EP5 DES.txt	EP5 3DES.txt	5	10	56.586	165.29	0.08836	0.03025
6	P6.txt	EP6 DES.txt	EP6 3DES.txt	6	12	67.924	198.463	0.08833	0.03023
7	P7.txt	EP7 DES.txt	EP7 3DES.txt	7	14	79.262	231.15	0.08831	0.03028
8	P8.txt	EP8 DES.txt	EP8 3DES.txt	8	16	90.733	264.882	0.08817	0.03020
9	P9.txt	EP9 DES.txt	EP9 3DES.txt	9	18	101.909	297.451	0.08831	0.03026
10	P10.txt	EP10 DES.txt	EP10 3DES.txt	10	20	113.342	330.389	0.08823	0.03027
Kecepatan Rata-rata								0.08828	0.03024

Tabel 3.1 Waktu dan Kecepatan Proses Enkripsi dengan Algoritma DES dan Algoritma triple DES

3.6.2. Flowchart Proses Dekripsi

Flowchart proses dekripsi merupakan aliran proses untuk menjalankan program dekripsi yang digunakan untuk mendekripsi file yang telah dienkripsi. Pada tampilan program dekripsi ini, pengguna dapat mencari file untuk didekripsi dengan menggunakan tombol pencari dan tombol penyimpan untuk menyimpan hasil dekripsi tersebut.

Untuk lebih jelasnya lihat Gambar 3.10 berikut :



Gambar 3.10 Flowchart Proses Dekripsi

Waktu Proses dan Kecepatannya untuk Proses Dekripsi dengan Algoritma DES dan Algoritma triple DES.

No	Nama File				Ukuran File (KB)		Waktu Proses (detik)		Kecepatan (Bit/detik)	
	Input		Output							
	DES	3DES	DES	3DES	Input	Output	DES	3DES	DES	3DES
1	[illegible]	[illegible]	[illegible]	[illegible]	2	1	[illegible]	[illegible]	[illegible]	[illegible]
2	[illegible]	[illegible]	[illegible]	[illegible]	4	3	[illegible]	[illegible]	[illegible]	[illegible]
3	[illegible]	[illegible]	[illegible]	[illegible]	6	5	[illegible]	[illegible]	[illegible]	[illegible]
4	[illegible]	[illegible]	[illegible]	[illegible]	8	4	[illegible]	[illegible]	[illegible]	[illegible]
5	[illegible]	[illegible]	[illegible]	[illegible]	12	5	[illegible]	[illegible]	[illegible]	[illegible]
6	[illegible]	[illegible]	[illegible]	[illegible]	12	6	[illegible]	[illegible]	[illegible]	[illegible]
7	[illegible]	[illegible]	[illegible]	[illegible]	14	7	[illegible]	[illegible]	[illegible]	[illegible]
8	[illegible]	[illegible]	[illegible]	[illegible]	16	8	[illegible]	[illegible]	[illegible]	[illegible]
9	[illegible]	[illegible]	[illegible]	[illegible]	18	9	[illegible]	[illegible]	[illegible]	[illegible]
10	[illegible]	[illegible]	[illegible]	[illegible]	22	10	[illegible]	[illegible]	[illegible]	[illegible]
Kecepatan Rata-rata									[illegible]	[illegible]

Tabel 3.2 Waktu dan Kecepatan Proses Dekripsi dengan Algoritma DES dan Algoritma triple DES

Keterangan dari gambar diatas adalah sebagai berikut:

P = pesan

EM = enkripsi pesan

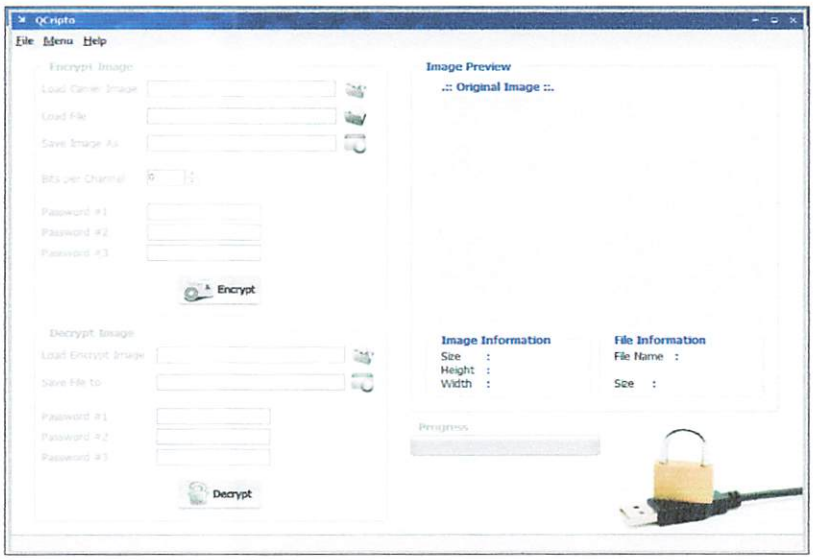
DP = dekripsi pesan.


INSTITUT TEKNOLOGI NASIONAL
ITN
MALANG

BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

Pada bab berikut ini menjelaskan tentang implementasi dan pengujian sistem dari aplikasi kriptografi yang telah dibuat. Implementasi meliputi proses-proses yang terdapat dalam aplikasi kriptografi, desain enkripsi dan dekripsi serta algoritma *triple DES*.Untuk memperjelas gambaran tentang aplikasi kriptografi yang telah dibuat berikut ini adalah gambar desain aplikasi kriptografi serta enkripsi dan dekripsi serta algoritma *triple DES*.



Gambar 4.1 Implementasi Desain Kriptografi Algoritma *Triple DES*

Pada pengujian sistem ini digunakan dua buah perangkat yaitu perangkat lunak dan juga perangkat keras. Dimana masing-masing perangkat yang digunakan akan dikelompokkan sebagai berikut:

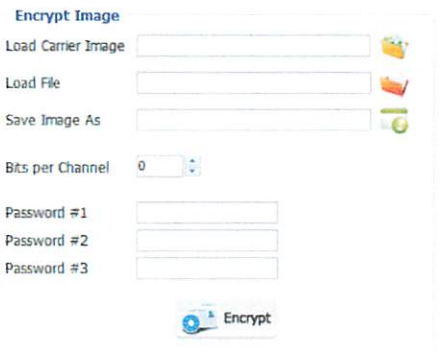
- Perangkat lunak yang digunakan untuk melakukan pengujian ini yaitu:
  1. Sistem Operasi *Windows 7 Professional.*
  2. *Borland Delphi 7* yang digunakan untuk menjalankan program kriptografi algoritma *triple DES.*
  3. Program kriptografi.
- Sedangkan perangkat keras yang digunakan yaitu:
  1. Processor Intel® Core™Duo CPU T2500 @ 2.00GHz 2.00GHz
  2. RAM 1.50 GB

3. Monitor 14"
4. System type 32-bit Operating System
5. Hard disk 112 GB

4.1 Implementasi

4.1.1 Tahap Pengenkripsian

Tahap pengenkripsian ini digunakan untuk melakukan proses penyisipan file rahasia kedalam citra digital dan juga melihat informasi tentang citra digital yang akan digunakan sebagai media untuk pengenkripsian. Berikut adalah gambar dari enkripsi citra digital.



Gambar 4.2 Enkripsi Citra Digital

Proses dimulai dengan aplikasi memilih file citra digital yang memiliki format (.bmp) sebagai *cover object* atau tempat yang ingin digunakan guna menyimpan data yang akan dienkripsi. Untuk mengambil *cover object* dapat dilakukan dengan menekan tombol button "*Load Carrier Image*" pada *Group Box Encrypt Image* seperti yang ada pada gambar 4.3 berikut.

Gambar 4.3 Tombol "*Load Carrier Image*" untuk Memilih File Citra Digital

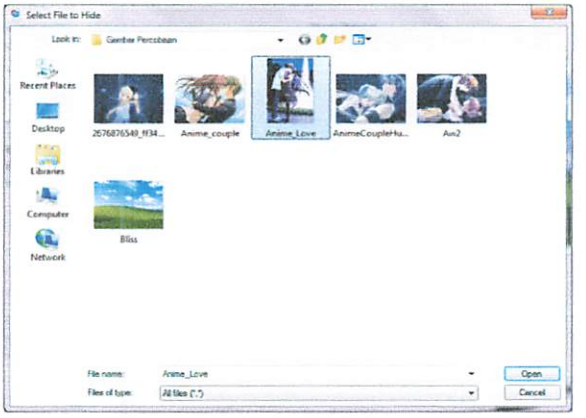
Setelah itu akan muncul *"Select Carrier Image"* yang akan digunakan untuk memilih file citra digital yang berformat (.bmp) sebagai *cover object* seperti pada gambar 4.4 berikut ini.

Gambar 4.4 *"Select Carrier Image"* File Citra Digital sebagai *cover object*

Setelah itu masukkan file yang akan dienkripsikan dengan mengklik button "*Load File*"seperti gambar 4.5 berikut ini.

Gambar 4.5 Tombol "*Load File*" untuk Memilih File yang Dienkripsi

Setelah itu akan muncul *"Select File to Hide"* yang akan digunakan untuk memilih file sebagai file yang dirahasiakan seperti pada gambar 4.6 berikut ini.



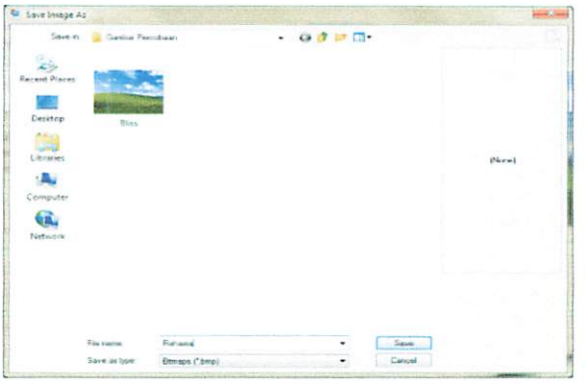
Gambar 4.6 *"Select File to Hide"* File yang Dienkripsi *Triple DES*

Kemudian yang dilanjutkan pada penyimpanan file dengan nama file citra digital yang baru, klik tombol button *"Save Image As"* pada gambar 4.7 dibawah.

Gambar 4.7 Tombol *"Save Image As"* untuk Menyimpan File yang Baru

Setelah itu akan muncul *"Save Image As"* yang akan digunakan untuk

menyimpan nama file baru sebagai file yang disisipi file lain seperti pada gambar 4.8 ini.



Gambar 4.8 *"Select File to Hide"* Nama File Baru yang Dienkripsi *Triple DES*

Setelah memilih citra digital yang diinginkan, untuk melihat citra digital yang akan dienkripsi, lihat *"Group Box Image Preview"* pada gambar 4.9 dibawah ini.

Gambar 4.9 Melihat Citra Digital yang akan Dienkripsi

Untuk atribut atau rincian file yang terdapat pada citra digital lihat pada gambar 4.10 ini.

Gambar 4.10 Untuk Atribut atau Rincian File

Kemudian dilanjutkan memasukkan bit citra digital keberapa yang akan dienkripsi pengguna. Berikut gambar dari *"Bits per Channel"* pada enkripsi.

Gambar 4.11 *"Bits per Channel"* pada Enkripsi

Selanjutnya pengguna mengisi *password* yang telah memiliki perhitungan algoritma *triple DES* . Berikut gambar dari proses enkripsi menggunakan algoritma *triple DES.*

Gambar 4.12 Enkripsi Kriptografi Algoritma *Triple DES*

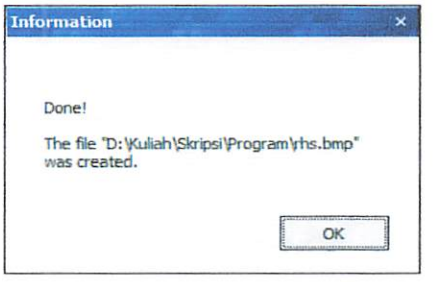
Setelah semua terisi dengan benar lanjutkan dengan mengklik button yang bertulis *"Encrypt"* seperti pada gambar 4.13 berikut.

Gambar 4.13 Button *"Encrypt"*

Setelah selesai mengenkripsi file dengan algorirma *triple DES* maka *progress* akan berjalan sesuai dengan lama data yang terenkripsi, seperti pada gambar 4.14 yang ada dibawah ini.

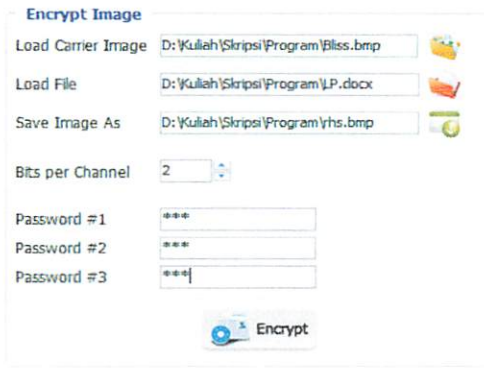
Gambar 4.14 *"progress" Triple DES*

Selanjutnya akan muncul "Information" seperti pada gambar 4.15 berikut.



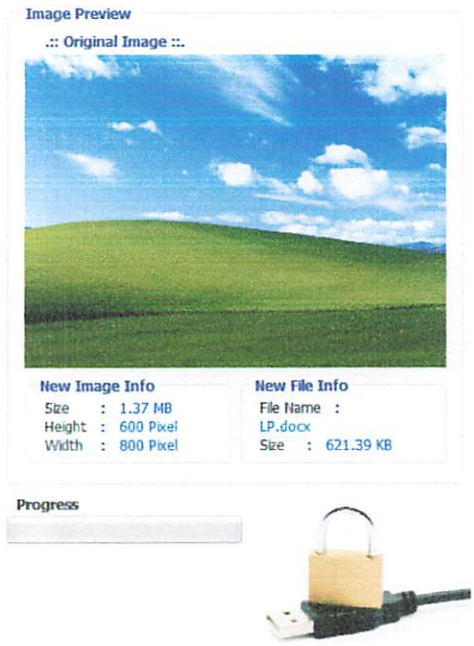
Gambar 4.15 *"Information"* Enkripsi *Triple DES*

Berikut ini adalah gambar kriptografi enkripsi citra digital algoritma *triple DES.*



Gambar 4.16 Kriptografi Enkripsi Citra Digital *Triple DES*

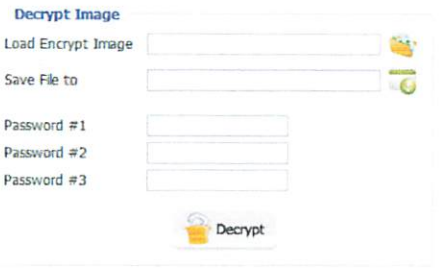
Tampilan informasi pada citra yang akan dienkripsi adalah sebagai berikut.



Gambar 4.17 "*Information Image*" *Triple DES*

4.1.2 Tahap Pendekripsian

Tahap ini adalah tahap dimana file yang dienkripsi akan dikembalikan lagi kefile semula dengan menggunakan algoritma yang sama yaitu *triple DES.* Berikut ini adalah tampilan dekripsi citra digital pada gambar 4.18 ini.

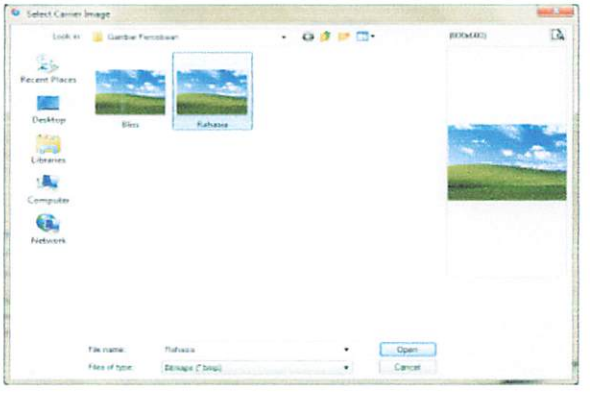


Gambar 4.18 Dekripsi Citra Digital

Proses dimulai hampir sama dengan saat pengenkripsian dengan aplikasi memilih file citra digital yang telah dienkripsi pada tahap sebelumnya. Kemudian dilanjutkan dengan tahap pendekripsian dengan menekan tombol button "*Load Carrier Image*" pada *Group Box Encrypt Image* seperti yang ada pada gambar 4.19 berikut.

Gambar 4.19 Tombol "*Load Carrier Image*" untuk Memilih File Citra Digital

Setelah itu akan muncul *"Select Carrier Image"* yang akan digunakan untuk memilih file citra digital yang berformat (.bmp) yang telah terenkripsikan pada sebelumnya seperti pada gambar 4.20 berikut ini.



Gambar 4.20 *"Select Carrier Image"* File Citra Digital yang Didekripsi

Yang kemudian dilanjutkan pada proses dekripsi file dengan nama file yang telah disembunyikan atau terenkripsi, klik tombol button *"Save File To"* pada gambar 4.21 dibawah.

Gambar 4.21 Tombol *"Save File To"* untuk Mendekripsi File yang Terenkripsi

Setelah itu akan muncul *"Select Image to Extract the Data"* yang akan digunakan untuk membuka file yang telah dienkripsi dengan nama file asli seperti pada gambar 4.22 berikut ini.

Gambar 4.22 *"Select Image to Extract the Data"* File yang Didekripsi *Triple DES*

Selanjutnya pengguna mengisi *password* yang telah memiliki perhitungan algoritma *triple DES* untuk membuka file yang terenkripsi pada sebelumnya. Berikut gambar dari proses enkripsi menggunakan algoritma *triple DES.*

Gambar 4.23 Dekripsi Kriptografi Algoritma *Triple DES*

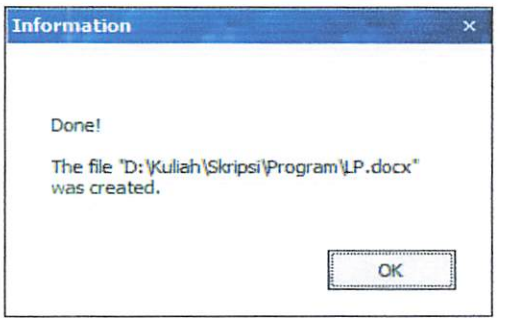
Setelah semua pasword terisi dengan benar lanjutkan dengan mengklik button yang bertulis *"Decrypt"* seperti pada gambar 4.24 berikut.

Gambar 4.24 Button *"Decrypt"*

Selesai mengdekripsi file dengan algorirma *triple DES* maka *progress* akan berjalan sesuai dengan lama data yang terdekripsi, seperti pada gambar 4.25 yang ada dibawah ini.

Gambar 4.25 *"progress" Triple DES*

Selanjutnya akan muncul *"Information"* seperti pada gambar 4.26 berikut.

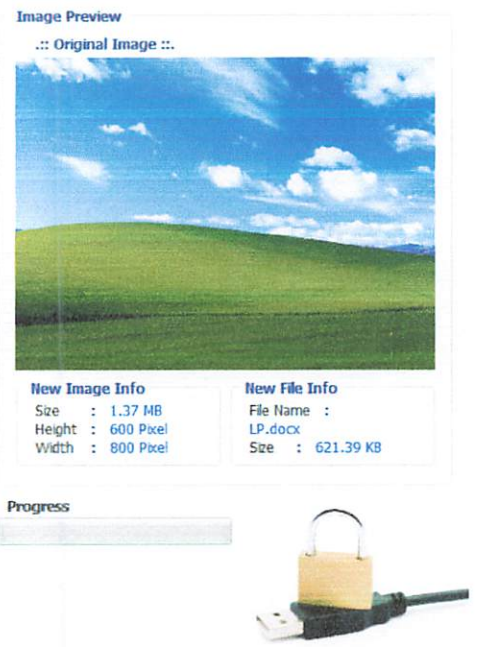


Gambar 4.26 *"Information"* dekripsi *Triple DES*

Berikut ini adalah gambar kriptografi dekripsi citra digital algoritma *triple DES.*

Gambar 4.27 Kriptografi Enkripsi Citra Digital *Triple DES*

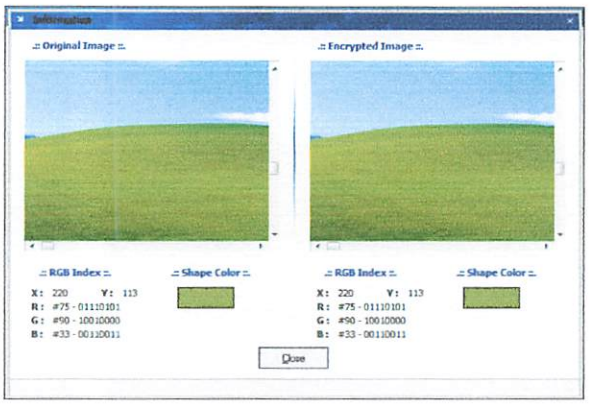
Tampilan informasi pada citra yang akan didekripsi adalah sebagai berikut.



Gambar 4.28 *"Information Image" Triple DES*

4.1.3 Tahap Perbandingan

Berikut adalah tampilan perbandingan file yang telah dienkripsi dan file asli.

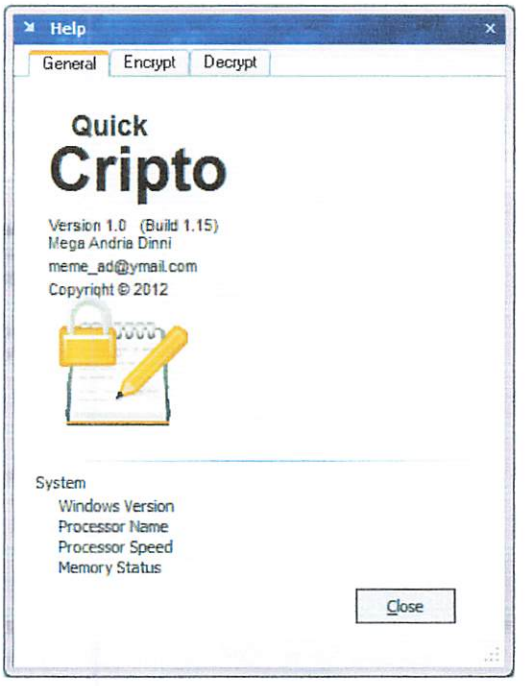


Gambar 4.29 "*Information*" Perbandingan dari Citra Digital

4.1.4 Tahap Bantuan atau *Help*

Pada tahapan ini pengguna akan diberikan bantuan mengenai tentang tahapan-tahapan yang akan dilakukan dalam pembuatan enkripsi maupun dekripsi yang dapat dilihat pada gambar-gambar yang ada dibawah berikut ini.

Berikut adalah tampilan awal dari bantuan atau *Help* seperti pada gambar 4.30.



Gambar 4.30 "*Help*" Tampilan Awal dari Bantuan

Tampilan selanjutnya ialah bantuan dalam pembuatan enkripsi, berikut

adalah gambarnya.

Gambar 4.31 "*Help*" Tampilan Enkripsi dari Bantuan

Tampilan terakhir dari *"Help"* ialah bantuan dalam pembuatan dekripsi, berikut dapat dilihat gambarnya.

Gambar 4.32 "*Help*" Tampilan Dekripsi dari Bantuan

4.2 Pengujian Sistem

Pengujian sistem merupakan proses selanjutnya setelah implementasi perangkat lunak selesai dilakukan. Pengujian ini dilakukan untuk menguji algoritma *triple DES* yang digunakan.

4.2.1 Pengujian Citra Digital

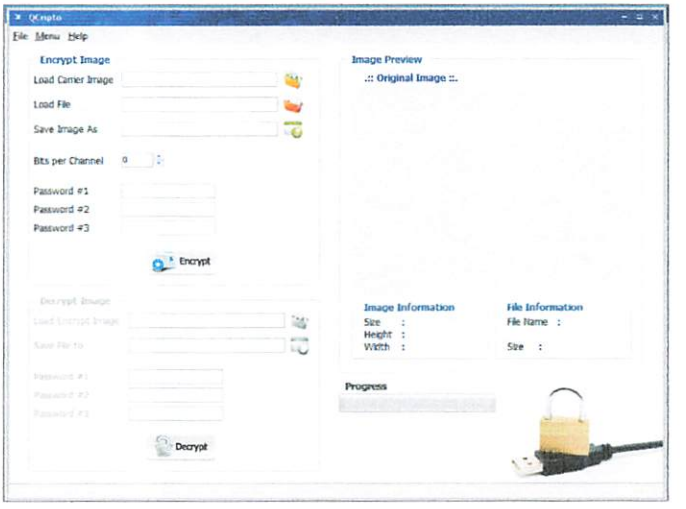
Contoh file yang akan dienkripsi dan didekripsi berikut ini diambil dari file dan kunci yang digunakan adalah saling bebas

K1, K2, dan K3 adalah kunci-kunci yang saling bebas (K1 ≠ K2 ≠ K3 ≠ K1) yaitu:

- Kunci 1: Enkripsi
- Kunci 2 : Keamanan
- Kunci 3 : Dekripsi

Cara pengenkripsian yang dipilih adalah *DES – EDE3* dan cara pendekripsian yang dipilih adalah DES – DED3.

Aplikasi yang akan ditampilkan adalah sebagai berikut:



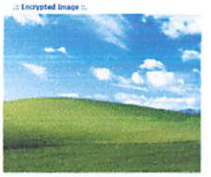
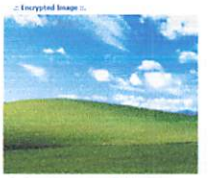
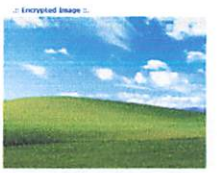
Gambar 4.33 Aplikasi Algoritma *Triple DES*

Contoh dari file citra:

File citra digital	Keterangan
	File Anime_couple berikut adalah salah satu contoh file citra digital yang dapat digunakan sebagai file sisipan. Format (.jpg) dengan besar ukuran file 116KB (24 bit).
	File 28105 berikut adalah salah satu contoh file citra digital yang dapat digunakan sebagai file sisipan. Format (.jpg) dengan besar ukuran file 32KB (24 bit).
	File anime berikut adalah salah satu contoh file citra digital yang dapat digunakan sebagai file sisipan. Format (.jpg) dengan besar ukuran file 40KB (24 bit).

Tabel 4.1 Contoh File Citra Digital

Aplikasi yang akan ditampilkan adalah perbandingan pada tiap-tiap bit yang dienkripsi sebagai berikut:

Bit / channel	Hasil (subyektif)	Keterangan
1		Hasil gambar pada bit ke-1 dapat dilihat disamping, dengan hasil gambarnya berkuwalitas bagus.
2		Hasil gambar pada bit ke-2 dapat dilihat disamping, dengan hasil gambarnya berkuwalitas masih bagus.
3		Hasil gambar pada bit ke-3 dapat dilihat disamping, dengan hasil gambarnya berkuwalitas cukup bagus.
4		Hasil gambar pada bit ke-4 dapat dilihat disamping, dengan hasil gambarnya berkuwalitas cukup bagus.
5		Hasil gambar pada bit ke-5 dapat dilihat disamping, dengan hasil gambarnya berkuwalitas kurang bagus.
6		Hasil gambar pada bit ke-6 dapat dilihat disamping, dengan hasil gambarnya berkuwalitas kurang bagus.
7		Hasil gambar pada bit ke-1 dapat dilihat disamping, dengan hasil gambarnya berkuwalitas tidak bagus.

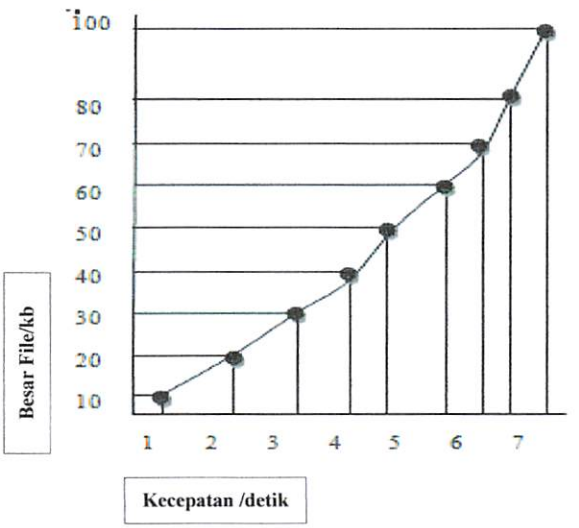
Tabel 4.2 Perbandingan Pada Tiap-tiap Bit yang Dienkripsi

Menentukan kecepatan pada bit-bit yang ada pada enkripsi, berikut adalah beberapa contoh yang ada pada tabel dibawah ini.

Besar File/kb	Bit ke-	Kecepatan /detik	Bit ke-	Kecepatan /detik
10	2	1.12	4	1.42
20	2	2.40	4	2.54
30	2	3.47	4	3.40
40	2	4.15	4	4.20
50	2	4.99	4	5.09
60	2	5.98	4	6
70	2	6.32	4	6.52
80	2	6.89	4	7.02
100	2	7.41	4	7.61

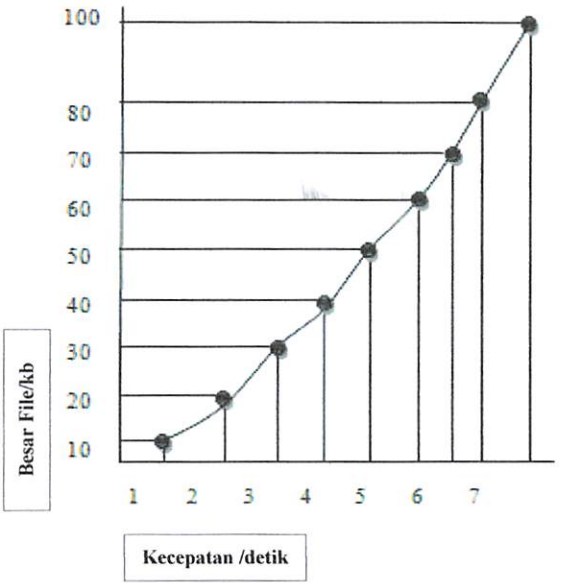
Tabel 4.3 Kecepatan Pada Bit Saat Enkripsi

Berikut juga adalah grafik ukuran kecepatan file pada bit ke-2, dapat dilihat seperti pada gambar dibawah ini:



Gambar 4.34 Grafik Ukuran Kecepatan pada Bit ke-2

Berikut adalah grafik ukuran kecepatan file pada bit ke-4:



Gambar 4.35 Grafik Ukuran Kecepatan pada Bit ke-4

Analisa perbandingan data biner dari *pixel* (R-G-B) file carrier dengan file yang tersisipi atau terenkripsi. Berikut data yang diambil pada bit ke-4 sebagai salah satu contohnya.

Citra Digital Carrier	Citra Digital Enkripsi
• Pada titik *pixel* X = 3, Y = 76 Maka data binernya: R = #51-01010001, G = #8F-10001111, B = #F6-11110110 • Pada titik *pixel* X = 72, Y = 73 Maka data binernya: R = #4E-01001110, G = #87-10000111, B = #F2-11110010 • Pada titik *pixel* X = 194, Y = 83 Maka data binernya:	• Pada titik *pixel* X = 3, Y = 76 Maka data binernya: R = #58-01011011, G = #8F-10001111, B = #F4-11110100 • Pada titik *pixel* X = 72, Y = 73 Maka data binernya: R = #4E-01001110, G = #8D-10001101, B = #F1-11110001 • Pada titik *pixel* X = 72, Y = 73 Maka data binernya:

R = #3F-00111111, G = #7A-01111010, B = #EE-11101111 • Pada titik *pixel* X = 227, Y = 83 Maka data binernya: R = #3C-00111100, G = #77-01110111, B = #EF-11101111	R = #30-00110000, G = #7B-01111011, B = #EF-11101111 • Pada titik *pixel* X = 227, Y = 83 Maka data binernya: R = #32-00110010, G = #7E-01111110, B = #ED-11101101

Tabel 4.4 Perbandingan Data Biner dari *Pixel* (R-G-B)


INSTITUT TEKNOLOGI NASIONAL
ITN
MALANG

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari pembuatan Aplikasi kriptografi file citra digital dengan menggunakan algoritma *triple DES* secara keseluruhan ini dapat diambil kesimpulan:

1. Wadah atau gambar cover harus berformat bitmap (.bmp) dan ukuran filenya tidak lebih dari 1,5MB.
2. Aplikasi menggunakan bit antara 1 – 4 bit saja, karena jika lebih maka citra cover akan mengalami kerusakan.
3. Aplikasi juga dinyatakan aman untuk dikirim dan diterima pihak yang berwenang, karena proses enkripsi dan dekripsi dengan tiga kali penguncian yang sesuai dengan algoritma.

5.2. Saran

Dalam penyelesaian "Aplikasi kriptografi file citra digital dengan menggunakan algoritma *triple DES*" memiliki banyak kekurangan.

1. Untuk perkembangan selanjutnya, menambahkan besar ukuran file sisipan agar dapat lebih fleksibel lagi.
2. Untuk menjaga dan memelihara kekuatan aplikasi terhadap serangan, maka perlu dilakukan proses pengujian serta perbandingan terhadap aplikasi lain.


INSTITUT TEKNOLOGI NASIONAL
ITN
MALANG

DAFTAR PUSTAKA

[1]. F. Antonios, P. Nikolaos, M. Panagiotis, and A. Emmanouel, "Hardware Implementation of Triple-DES Encryption/ Decryption Algorithm", International Conference on Telecommunications and Multimedia, 2006.

[2]. H. Kummert dan Nentec GmbH. *The PPP Triple-DES Encryption Protocol (3DES)*, RFC 2420, 1998.

[3]. Komputer, Wahana Semarang. 2004. *Memahami Model Enkripsi dan Security* Data. Yogyakarta: Penerbit Andi.

[4]. Menezes, A. J., P. C. v. Oorschot, et al. (1996). Handbook of Applied Cryptography. United States, CRC Press: 816.

[5]. C. Boyd. "Modern Data Encryption," Electronics & Communication Engineering Journal, October 1993, pp 271-278.

[6]. "Data encryption standard (DES) ", National Bureau of Standards (U. S.), Federal Information Processing Standards Publication 46, National Technical Information Service,Springfield, VA,Apr. 1977.

[7]. Jurnal Ilmiah Ilmu Komputer, Vol 3, No 2 (2005). *3 DES Algorithm Analysis for Message Disguising.* Sony Hartono Wijaya, Sugi Guritman, Wisnu Ananta Kusuma.

[8]. Risanto. 2006. *Keamanan Data dengan Kriptografi Kunci Simetris Algoritma DES.* Skripsi tidak diterbitkan. Bandung: Program Pascasarjana UNPAD.

[9]. Ye Guodong. (2010). "Image scrambling encryption algorithm of pixel bit based on chaos map." *Pattern Recognition Letters* 31 :347–354.

[10]. Anonymous, 1998. "Introduction to public-key cryptography." (online). *http://docs.sun.com/source/816-6154-10/*.

[11]. Felix, Fidens. 2006. *Dasar Kriptografi*, (online), http://www.ilmukomputer.com, (diakses Oktober 2011).

[12]. Hasan, Rusydi. 2003. *Mengenal Algoritma DES*, (online), http://www.ilmukomputer.com, (diakses Oktober 2011).

[13]. Martinus, Ady, H. *Cara Menghitung Bitmap*, (online), (diakses Oktober 2011).

[14]. NIST. 2004. *Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher*, (online) , http://www.csrc.nist.gov, (diakses Oktober 2011).

[15]. Schneier. 1996. *Kriptografi*, (online), http://id.wikipedia.org/wiki/Kriptografi, (diakses Oktober 2011).

[16]. Wikipedia http://id.wikipedia.org/wiki/citradigital, (diakses November 2011).

[17]. Wikipedia http://id.wikipedia.org/wiki/file, (diakses November 2011).

[18]. Wikipedia http://id.wikipedia.org/wiki/bitmap, (diakses November 2011).

[19]. Wikipedia. *Data Encryption Standard*. (November 2011), http://en.wikipedia.org/wiki/Data_Encryption_Standard.

[20]. Wikipedia. *Triple DES*. (November 2011), http://en.wikipedia.org/wiki/Triple_DES.

[21]. *36 Jam Belajar Komputer Borland Delphi 7*, (diakses Desember 2011) http://www.downloadebookgratis.com/


INSTITUT TEKNOLOGI NASIONAL
ITN
MALANG

LAMPIRAN

LEMBAR PERSEMBAHAN

Skripsi ini saya persembahkan kepada yang tercinta

Taufiq Nurdiansyah, ST

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK INFORMATIKA S-1
Jl. Karanglo Km. 2 Malang

BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

Nama : Mega Andria Dinni

NIM : 0818901

Jurusan : Teknik Informatika

Judul : Aplikasi Kriptografi File Citra Digital Menggunakan Algoritma Triple DES (Triple Data Encryption Standard)

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada:

Hari : Kamis

Tanggal : 9 Agustus 2012

Nilai : A

Panitia Ujian Skripsi :
Ketua Majelis Penguji

Joseph Dedy Irawan, ST, MT.

NIP.197404162005021002

Anggota Penguji :

Penguji Pertama	Penguji Kedua
Ali Mahmudi B.Eng,. PhD	**Ahmad Faisol, ST**
NIP.P. 1031000429	**NIP.P. 1031000431**

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK INFORMATIKA S-1
Jl. Karanglo Km. 2 Malang

FORMULIR PERBAIKAN SKRIPSI

Nama : Mega Andria Dinni

NIM : 0818901

Jurusan : Teknik Informatika S-1

Judul : Aplikasi Kriptografi File Citra Digital Menggunakan Algoritma Triple DES (Triple Data Encryption Standard)

Penguji	Perbaikan	Tanda Tangan
Penguji 1	1. Pada bagian Help tambahkan tentang kami 2. Daftar pustaka yang diurut sesuai abjad 3. Perbaiki halaman lampiran	
Penguji 2	1. Aplikasi harus bisa memfilter jika file gambar tidak sesuai titik kedalamam 2. Hilangkan kesimpulan no.1 3. Buktikan kesimpulan no.4 dengan penguji 4. Tambahkan dasar teori flowchart	

Anggota Penguji :

Penguji Pertama

Ali Mahmudi B.Eng,. PhD

NIP.P. 1031000429

Penguji Kedua

Ahmad Faisol, ST

NIP.P. 1031000431

Mengetahui,

Dosen Pembimbing I

Joseph Dedy Irawan, ST, MT

NIP.197404162005021002

Dosen Pembimbing II

Michael Ardita,ST, MT

NIP. 103 1000 434

FORMULIR BIMBINGAN SKRIPSI

Nama : Mega Andria Dinni

NIM : 0818901

Judul Skripsi : Aplikasi Kriptografi File Citra Digital Menggunakan Algoritma Triple DES (Triple Data Encryption Standard)

No.	Tanggal	Uraian	Paraf Pembimbing
1	23-04-2012	Tambahkan sumber daftar pustaka pada bab-2	
2	14-05-2012	Flowchart sesuaikan dengan program pada bab-3	
3	24-05-2012	Tambahkan gambar proses enkripsi triple des pada bab-2	
4	29-06-2012	Tambahkan desain form pembuatan pada bab-3	
5	09-07-2012	Tambahkan beberapa script pada bab-3	
6	11-07-2012	Tambahkan flowchart enkripsi detail	
7	13-07-2012	Tambahkan desain user interface pada bab-3	
8	13-07-2012	Perbaiki kesimpulan dan sesuaikan dengan program	
9	14-07-2012	Tambahkan daftar pustaka	

Malang, 2012

Dosen Pembimbing

Joseph Dedy Irawan, ST, MT

NIP.197404162005021002

FORMULIR BIMBINGAN SKRIPSI

Nama : Mega Andria Dinni

NIM : 0818901

Judul Skripsi : Aplikasi Kriptografi File Citra Digital Menggunakan Algoritma Triple DES (Triple Data Encryption Standard)

No	Tanggal	Uraian	Paraf Pembimbing
1	19-04-2012	Perbaiki manfaat pada bab-1	
2	21-04-2012	Tinjauan pustaka diganti dengan landasan teori pada bab-1	
3	23-04-2012	Bab-1 dan seterusnya "Pendahuluan", "Landasan Teori" dicenterkan ketengah	
4	14-05-2012	Tambahkan keterangan pada tiap-tiap sumber pada daftar pustaka atau keterangan tambahkan[...]	
5	24-05-2012	Sumber yang didapat dari buku,jurnal atau sumber lainnya harus jelas	
6	06-06-2012	Algoritma triple des (gambarnya) diberikan yang lebih detai lagi	
7	05-07-2012	Perbaiki flowchart sesuaikan dengan program	
8	06-07-2012	Pada bab-3 3.3 dan seterusnya dipindahakan ke bab-2	
9	09-07-2012	Kesimpulan dan saran rubah semua sesuaikan kembali	
10	11-07-2012	Daftar pustaka sesuaikan kembali dan tambah sumbernya	

Malang, 2012

Dosen Pembimbing

Michael Ardita,ST, MT

NIP. 103 1000 434

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK
PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG
Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Nomor : ITN-62/T.Inf/TA/2012 — 11 April 2012
Lampiran : -
Perihal : Bimbingan Skripsi

Kepada : Yth. Sdr. Michael Ardita, ST, MT
Dosen Pembimbing Program Studi Teknik Informatika S1
Institut Teknologi Nasional
M a l a n g

Dengan hormat
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa :

Nama	:	Mega Andria Dini
Nim	:	0818901
Prodi	:	Teknik Informatika S1
Fakultas	:	Teknologi Industri

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu 6 (enam) bulan, terhitung mulai tanggal ;

11 April 2012 s/d 11 Oktober 2012

Sebagai satu syarat untuk menempuh Ujian Sarjana Teknik, Program Studi Teknik Informatika S1.
Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui
Program Studi Teknik Informatika S1
K e t u a

Joseph Dedy Irawan, ST, MT
NIP : 197404162005021002

Form S-4a

LAMPIRAN

"Quick_Cripto"

```
unit Quick_Cripto;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, SUIForm, SUISkinControl, SUIImagePanel, Menus,
  SUIMainMenu, RzPanel, Mask, RzEdit, StdCtrls, SUIEdit, SUIButton, XPMan,
  ExtDlgs, ComCtrls, RzPrgres, SUIProgressBar, SUIDlg;
type
  TForm_Utama = class(TForm)
    suiForm_Utama: TsuiForm;
    EncryptBox: TRzGroupBox;
    LoadImage: TsuiEdit;
    LCI: TLabel;
    LF: TLabel;
    encPas1: TsuiEdit;
    encPas2: TsuiEdit;
    encPas3: TsuiEdit;
    EP1: TLabel;
    EP2: TLabel;
    EP3: TLabel;
    Load_Image: TsuiImageButton;
    Load_File: TsuiImageButton;
    btnEncrypt: TsuiImageButton;
    DecryptBox: TRzGroupBox;
    LoadEncImage: TsuiEdit;
    SafeFile: TsuiEdit;
    LEI: TLabel;
    SFT: TLabel;
    Load_enc_Image: TsuiImageButton;
    Save_File: TsuiImageButton;
    MainStatusBar: TRzStatusBar;
    suiMainMenu1: TsuiMainMenu;
    File1: TMenuItem;
    Menu1: TMenuItem;
    Help1: TMenuItem;
    NewEncrypt: TMenuItem;
    NewDecrypt: TMenuItem;
    CloseApp: TMenuItem;
    ResetEncrypt: TMenuItem;
    ResetDecript: TMenuItem;
```

F1About1: TMenuItem;
btnDecrypt: TsuiImageButton;
InformationBox: TRzGroupBox;
ImagePanel: TsuiImagePanel;
SI: TLabel;
HI: TLabel;
WI: TLabel;
II: TRzGroupBox;
Label11: TLabel;
Label12: TLabel;
Label13: TLabel;
imagesize: TLabel;
imageheight: TLabel;
imagewidth: TLabel;
SaveDialog: TSaveDialog;
OpenDialog: TOpenDialog;
OpenPictureDialog: TOpenPictureDialog;
SavePictureDialog: TSavePictureDialog;
XPManifest: TXPManifest;
edtBPC: TsuiEdit;
BPC: TLabel;
udBPC: TUpDown;
decPas1: TsuiEdit;
decPas2: TsuiEdit;
decPas3: TsuiEdit;
DP1: TLabel;
DP2: TLabel;
DP3: TLabel;
FI: TRzGroupBox;
Progress: TLabel;
FN: TLabel;
FS: TLabel;
Label3: TLabel;
Label4: TLabel;
filename: TLabel;
filesize: TLabel;
SaveImage: TsuiEdit;
Save_Image: TsuiImageButton;
Label1: TLabel;
simbol: TsuiImagePanel;
ProBar: TProgressBar;
LoadFile: TsuiEdit;
MessageDialog: TsuiMessageDialog;
LabelImage: TLabel;
ImageComparation: TMenuItem;

```
    procedure Load_ImageClick(Sender: TObject);
    procedure Load_FileClick(Sender: TObject);
    procedure Save_ImageClick(Sender: TObject);
    procedure btnEncryptClick(Sender: TObject);
    procedure Load_enc_ImageClick(Sender: TObject);
    procedure Save_FileClick(Sender: TObject);
    procedure btnDecryptClick(Sender: TObject);
    procedure Refresh;
    procedure NewEncryptClick(Sender: TObject);
    procedure NewDecryptClick(Sender: TObject);
    procedure CloseAppClick(Sender: TObject);
    procedure ResetEncryptClick(Sender: TObject);
    procedure ResetDecriptClick(Sender: TObject);
    procedure F1About1Click(Sender: TObject);
    procedure encPas1KeyPress(Sender: TObject; var Key: Char);
    procedure encPas2KeyPress(Sender: TObject; var Key: Char);
    procedure encPas3KeyPress(Sender: TObject; var Key: Char);
    procedure decPas1KeyPress(Sender: TObject; var Key: Char);
    procedure decPas2KeyPress(Sender: TObject; var Key: Char);
    procedure decPas3KeyPress(Sender: TObject; var Key: Char);
    procedure ImageComparationClick(Sender: TObject);
  private
    { Private declarations }
    Procedure EnDecryptFile (pathin, pathout: String; Pas1, Pas2, Pas3: Word);
    function GetSize(const AFile:String):Int64;
    function FormatByteSize(const bytes: Longint): string;
  public
    { Public declarations }
  end;
var
  Form_Utama: TForm_Utama;
implementation
{$R *.dfm}
uses Q_Process, ActiveX, Help, ImageInformation;
//=========================Encrypt Image=========================//
procedure TForm_Utama.Load_ImageClick(Sender: TObject);
var
  strimage: String;
begin
  if (OpenPictureDialog.Execute) then
    begin
      strimage:= OpenPictureDialog.FileName;
      ImagePanel.Picture.LoadFromFile(strimage);
      LoadImage.Text:= strimage;
      if GetSize(OpenPictureDialog.FileName) > 4194304 then
```

```
      ShowMessage('Ukuran file lebih dari 4 MB')
    else
      imagesize.Caption:= FormatByteSize(GetSize(OpenPictureDialog.FileName));
      imageheight.Caption:= IntToStr(ImagePanel.Picture.Height) + ' Pixel';
      imagewidth.Caption:= IntToStr(ImagePanel.Picture.Width) + ' Pixel';
  end;
end;
procedure TForm_Utama.Load_FileClick(Sender: TObject);
var
 strfile: String;
begin
 if (OpenDialog.Execute) then
 begin
  strfile:= OpenDialog.FileName;
  LoadFile.Text:= strfile;
  filename.Caption:= ExtractFileName(OpenDialog.FileName);
  filesize.Caption:= FormatByteSize(GetSize(OpenDialog.FileName));
 end;
end;
procedure TForm_Utama.Save_ImageClick(Sender: TObject);
var
 strnew: String;
begin
 if (SavePictureDialog.Execute)then
 begin
  strnew:= SavePictureDialog.FileName;
  SaveImage.Text:= strnew;
 end;
end;
procedure TForm_Utama.btnEncryptClick(Sender: TObject);
begin
 try
  try
    btnEncrypt.Enabled:= False;
    edtBPC.Enabled:= False;
    udBPC.Enabled:= False;
    EnDecryptFile(OpenDialog.FileName,                    OpenDialog.FileName,
StrToInt(encPas1.Text), StrToInt(encPas2.Text), StrToInt(encPas3.Text));
    Encrypt(OpenDialog.FileName,                    OpenPictureDialog.FileName,
SavePictureDialog.FileName, udBPC.Position, ProBar);
    {MessageBox(Application.Handle, PChar('Done!' + #13#10#13#10 + 'The file "'
+
          SavePictureDialog.FileName + '" was created.'),
          'Information', MB_OK + MB_ICONINFORMATION);}
```

```
    MessageDialog.Text:= 'Done!' + #13#10#13#10 + 'The file "' +
SavePictureDialog.FileName + '" was created.' + #13#10;
    MessageDialog.ShowModal;
   finally
    ImageComparation.Enabled:= True;
    udBPC.Enabled:= True;
    edtBPC.Enabled:= True;
    btnEncrypt.Enabled:= True;
    ProBar.Position:= 0;
    LabelImage.Caption:= '.:: Encrypted Image ::.';
    ImagePanel.Picture.LoadFromFile(SavePictureDialog.FileName);
    Form_Image.Original.Picture.LoadFromFile(OpenPictureDialog.FileName);
    Form_Image.EncryptImage.Picture.LoadFromFile(SavePictureDialog.FileName);
    II.Caption:= ' New Image Info ';
    imagesize.Caption:= FormatByteSize(GetSize(SavePictureDialog.FileName));
    imageheight.Caption:= IntToStr(ImagePanel.Picture.Height) + ' Pixel';
    imagewidth.Caption:= IntToStr(ImagePanel.Picture.Width) + ' Pixel';
    OpenPictureDialog.NewInstance;
    OpenDialog.NewInstance;
    SavePictureDialog.NewInstance;
    LoadImage.Clear;
    LoadFile.Clear;
    encPas1.Clear;
    encPas2.Clear;
    encPas3.Clear;
   end;
  except
   on e: Exception do
    {MessageBox(Handle, PChar('An error has occured.' + e.Message), 'Information',
MB_OK + MB_ICONSTOP);}
    MessageDialog.Text:= 'An error has occured' + #13#10#13#10 + e.Message;
  end;
end;
//========================= Decrypt Image ========================//
procedure TForm_Utama.Load_enc_ImageClick(Sender: TObject);
var
  strencimg: String;
begin
  if (OpenPictureDialog.Execute) then
  begin
   strencimg:= OpenPictureDialog.FileName;
   ImagePanel.Picture.LoadFromFile(strencimg);
   LoadEncImage.Text:=strencimg;
   imagesize.Caption:= FormatByteSize(GetSize(OpenPictureDialog.FileName));
   imageheight.Caption:= IntToStr(ImagePanel.Picture.Height) + ' Pixel';
```

```
    imagewidth.Caption:= IntToStr(ImagePanel.Picture.Width) + ' Pixel';
  end;
end;
procedure TForm_Utama.Save_FileClick(Sender: TObject);
var
  strencfile: String;
begin
  if (SaveDialog.Execute) then
  begin
    strencfile:= SaveDialog.FileName;
    SafeFile.Text:= strencfile;
  end;
end;
procedure TForm_Utama.btnDecryptClick(Sender: TObject);
begin
  try
    try
      btnDecrypt.Enabled:= False;
      Decrypt(OpenPictureDialog.FileName, SaveDialog.FileName, ProBar);
      EnDecryptFile(SaveDialog.FileName,                    SaveDialog.FileName,
StrToInt(decPas1.Text), StrToInt(decPas2.Text), StrToInt(decPas3.Text));
      {MessageBox(Application.Handle, PChar ('Done!' + #13#10#13#10 + 'The file '"
+
              SaveDialog.FileName + '" was created.'),
              'Information', MB_OK + MB_ICONINFORMATION);}
      MessageDialog.Text:=  'Done!'  +  #13#10#13#10  +  'The  file  '"  +
SaveDialog.FileName + '" was created.' + #13#10;
      MessageDialog.ShowModal;
    finally
      btnDecrypt.Enabled:= True;
      ProBar.Position:= 0;
      FI.Caption:= ' New File Info ';
      filename.Caption:= ExtractFileName(SaveDialog.FileName);
      filesize.Caption:= FormatByteSize(GetSize(SaveDialog.FileName));
      OpenPictureDialog.NewInstance;
      SaveDialog.NewInstance;
      LoadEncImage.Clear;
      decPas1.Clear;
      decPas2.Clear;
      decPas3.Clear;
    end;
  except
    on e: Exception do
      {MessageBox(Handle, PChar ('An error has occurred.' + #13#10#13#10 +
e.Message),
```

```
            'Information', MB_OK + MB_ICONSTOP);}
    MessageDialog.Text:= 'An error has occured' + #13#10#13#10 + e.Message;
  end;
end;
//======================== Encryption File ========================//
Procedure TForm_Utama.EnDecryptFile (pathin, pathout: String; Pas1, Pas2, Pas3:
Word);
var
  InMS, OutMS: TMemoryStream;
  cnt1, cnt2, cnt3: Integer;
  C1, C2, C3: byte;
begin
  InMS:= TMemoryStream.Create;
  OutMS:= TMemoryStream.Create;
  try
    InMS.LoadFromFile(pathin);
    InMS.Position:=0;
    for cnt1:= 0 to InMS.Size -1 do
      begin
        InMS.Read(C1,1);
        C1:= (C1 xor not (ord(Pas1 shr cnt1)));
        OutMS.Write(C1,1);
      end;
    for cnt2:= 0 to InMS.Size -1 do
      begin
        InMS.Read(C2,1);
        C2:= (C2 xor not (ord(Pas2 shr cnt2)));
        OutMs.Write(C2,1);
      end;
    for cnt3:= 0 to InMS.Size -1 do
      begin
        InMS.Read(C3,1);
        C3:= (C3 xor not (ord(Pas3 shr cnt3)));
        OutMS.Write(C3,1);
      end;
      OutMS.SaveToFile(pathout);
    finally
    InMS.Free;
    OutMS.Free;
    end;
end;
//======================== Size File Information ========================//
function TForm_Utama.GetSize(const AFile:String):Int64;
begin
  Result := 0;
```

```
  with TFileStream.Create(AFile, fmOpenRead) do
  begin
   try
    Result := Size;
   finally
    Free;
   end;
  end;
end;
function TForm_Utama.FormatByteSize(const bytes: Longint): string;
const
  B = 1; //byte
  KB = 1024 * B; //kilobyte
  MB = 1024 * KB; //megabyte
  GB = 1024 * MB; //gigabyte
begin
  if bytes > GB then
   result := FormatFloat('#.## GB', bytes / GB)
  else
   if bytes > MB then
    result := FormatFloat('#.## MB', bytes / MB)
   else
    if bytes > KB then
     result := FormatFloat('#.## KB', bytes / KB)
    else
     result := FormatFloat('#.## bytes', bytes) ;
end;
//============================ Menu Control ============================//
procedure TForm_Utama.Refresh;
begin
 EncryptBox.Enabled:= False;
 DecryptBox.Enabled:= False;
 InformationBox.Enabled:= False;
 Progress.Enabled:= False;
 ProBar.Enabled:= False;
//===== Clear Encrypt Image =====//
 OpenPictureDialog.NewInstance;
 OpenDialog.NewInstance;
 SavePictureDialog.NewInstance;
 LoadImage.Clear;
 LoadFile.Clear;
 SaveImage.Clear;
 udBPC.Position:=0;
 encPas1.Clear;
 encPas2.Clear;
```

```
  encPas3.Clear;
//===== Clear Decrypt Image =====//
  OpenPictureDialog.NewInstance;
  SaveDialog.NewInstance;
  LoadEncImage.Clear;
  SafeFile.Clear;
  decPas1.Clear;
  decPas2.Clear;
  decPas3.Clear;
//===== Clear Information Box =====//
  LabelImage.Caption:= '.:: Original Image ::.';
  ImagePanel.Picture:= nil;
  imagesize.Caption:= '';
  imageheight.Caption:= '';
  imagewidth.Caption:= '';
  filename.Caption:= '';
  filesize.Caption:= '';
end;
procedure TForm_Utama.NewEncryptClick(Sender: TObject);
begin
  Refresh;
  ResetEncrypt.Enabled:= True;
  ResetDecript.Enabled:= False;
  NewEncrypt.Enabled:= False;
  NewDecrypt.Enabled:= True;
  EncryptBox.Enabled:= True;
  InformationBox.Enabled:= True;
  Progress.Enabled:= True;
  ProBar.Enabled:= True;
end;
procedure TForm_Utama.NewDecryptClick(Sender: TObject);
begin
  Refresh;
  ResetDecript.Enabled:= True;
  ResetEncrypt.Enabled:= False;
  NewDecrypt.Enabled:= False;
  NewEncrypt.Enabled:= True;
  DecryptBox.Enabled:= True;
  InformationBox.Enabled:= True;
  Progress.Enabled:= True;
  ProBar.Enabled:= True;
end;
procedure TForm_Utama.CloseAppClick(Sender: TObject);
begin
  Close;
```

```
end;
procedure TForm_Utama.ResetEncryptClick(Sender: TObject);
begin
  NewEncryptClick(Sender);
end;
procedure TForm_Utama.ResetDecriptClick(Sender: TObject);
begin
  NewDecryptClick(Sender);
end;
procedure TForm_Utama.F1About1Click(Sender: TObject);
begin
  Form_Help.Show;
end;
procedure TForm_Utama.ImageComparationClick(Sender: TObject);
begin
  Form_Image.Show;
end;
procedure TForm_Utama.encPas1KeyPress(Sender: TObject; var Key: Char);
begin
  if not (Key in [#8, '0'..'9']) then
  begin
    Key:= #0;
  end;
end;
procedure TForm_Utama.encPas2KeyPress(Sender: TObject; var Key: Char);
begin
  if not (Key in [#8, '0'..'9']) then
  begin
    Key:= #0;
  end;
end;
procedure TForm_Utama.encPas3KeyPress(Sender: TObject; var Key: Char);
begin
  if not (Key in [#8, '0'..'9']) then
  begin
    Key:= #0;
  end;
end;
procedure TForm_Utama.decPas1KeyPress(Sender: TObject; var Key: Char);
begin
  if not (Key in [#8, '0'..'9']) then
  begin
    Key:= #0;
  end;
end;
```

```
procedure TForm_Utama.decPas2KeyPress(Sender: TObject; var Key: Char);
begin
  if not (Key in [#8, '0'..'9']) then
  begin
    Key:= #0;
  end;
end;
procedure TForm_Utama.decPas3KeyPress(Sender: TObject; var Key: Char);
begin
  if not (Key in [#8, '0'..'9']) then
  begin
    Key:= #0;
  end;
end;
initialization
  OleInitialize(nil);
finalization
  OleUninitialize;
end.
```

"Q_Bits"

```
unit Q_Bits;
interface
Procedure SetBitAt (var Variable: LongInt; Position: Byte; Value: Boolean);
overload;
Procedure SetBitAt (var Variable: Byte; Position: Byte; Value: Boolean); overload;
Function GetBitAt (Variable: LongInt; Position: Byte): Boolean;
implementation
Procedure SetBitAt (var Variable: LongInt; Position: Byte; Value: Boolean);
begin
 if Value then
   Variable:= Variable or (1 shl Position)
 else
   Variable:= Variable and ((1 shl Position) xor $FFFFFFFF);
end;
Procedure SetBitAt (var Variable: Byte; Position: Byte; Value: Boolean);
begin
 if Value then
   Variable:= Variable or (1 shl Position)
 else
   Variable:= Variable and ((1 shl Position) xor $FF);
end;
Function GetBitAt (Variable: LongInt; Position: Byte): Boolean;
begin
 if Variable and (1 shl Position) <> 0 then
   Result:= True
 else
   Result:= False;
end;
end.
```

"ImageInformation"

```
unit ImageInformation;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, SUIForm, RzPanel, SUIButton, RzBckgnd, SUIImagePanel,
  StdCtrls, SUIMemo, RzEdit;

type
  TForm_Image = class(TForm)
    suiForm_Information: TsuiForm;
    RzStatusBar1: TRzStatusBar;
    Information_Close: TsuiButton;
    Original: TsuiImagePanel;
    RzSeparator1: TRzSeparator;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    ShapeO: TShape;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Label10: TLabel;
    Label11: TLabel;
    Label12: TLabel;
    ShapeE: TShape;
    ValueRO: TLabel;
    ValueGO: TLabel;
    ValueBO: TLabel;
    ValueRE: TLabel;
    ValueGE: TLabel;
    ValueBE: TLabel;
    Label13: TLabel;
    ValueXO: TLabel;
    Label14: TLabel;
    ValueYO: TLabel;
    Label15: TLabel;
    Label16: TLabel;
    ValueXE: TLabel;
```

```
    ValueYE: TLabel;
    ScrollBarOH: TScrollBar;
    ScrollBarOV: TScrollBar;
    ScrollBarEH: TScrollBar;
    ScrollBarEV: TScrollBar;
    EncryptImage: TsuiImagePanel;
    procedure Information_CloseClick(Sender: TObject);
    procedure OriginalMouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure ScrollBarOVChange(Sender: TObject);
    //procedure FormCreate(Sender: TObject);
    procedure ScrollBarEVChange(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }

  end;

var
  Form_Image: TForm_Image;
  MyBitmap1, MyBitmap2: TBitmap;

implementation

{$R *.dfm}

uses Quick_Cripto, Types;

procedure TForm_Image.Information_CloseClick(Sender: TObject);
begin
  Close;
end;

function CharToBin(x:Char):ShortString;
begin
 if UpCase(X)='0' then result:='0000';
 if UpCase(X)='1' then result:='0001';
 if UpCase(X)='2' then result:='0010';
 if UpCase(X)='3' then result:='0011';
 if UpCase(X)='4' then result:='0100';
 if UpCase(X)='5' then result:='0101';
 if UpCase(X)='6' then result:='0110';
 if UpCase(X)='7' then result:='0111';
```

```
 if UpCase(X)='8' then result:='1000';
 if UpCase(X)='9' then result:='1001';
 if UpCase(X)='A' then result:='1010';
 if UpCase(X)='B' then result:='1011';
 if UpCase(X)='C' then result:='1100';
 if UpCase(X)='D' then result:='1101';
 if UpCase(X)='E' then result:='1110';
 if UpCase(X)='F' then result:='1111';
end;

procedure TForm_Image.OriginalMouseMove(Sender: TObject;
  Shift: TShiftState; X, Y: Integer);
var
 Data: TColor;
 R,G,B: Byte;
 sR,sG,sB: String[2];
begin

  //====================== Original ======================//
  Data:= Original.Picture.Bitmap.Canvas.Pixels[X,Y];

  R:= Data shr 00;
  G:= Data shr 08;
  B:= Data shr 16;

  ShapeO.Brush.Color:= Data;

  sR:= IntToHex(R,2);
  sG:= IntToHex(G,2);
  sB:= IntToHex(B,2);

  ValueXO.Caption:= IntToStr(X);
  ValueYO.Caption:= IntToStr(Y);

  ValueRO.Caption:= '#' + sR + ' - ' + CharToBin(sR[1]) + CharToBin(sR[2]);
  ValueGO.Caption:= '#' + sG + ' - ' + CharToBin(sG[1]) + CharToBin(sG[2]);
  ValueBO.Caption:= '#' + sB + ' - ' + CharToBin(sB[1]) + CharToBin(sB[2]);

  //====================== Original ======================//
  Data:= EncryptImage.Picture.Bitmap.Canvas.Pixels[X,Y];

  R:= Data shr 00;
  G:= Data shr 08;
  B:= Data shr 16;
```

```
  ShapeE.Brush.Color:= Data;

  sR:= IntToHex(R,2);
  sG:= IntToHex(G,2);
  sB:= IntToHex(B,2);

  ValueXE.Caption:= IntToStr(X);
  ValueYE.Caption:= IntToStr(Y);

  ValueRE.Caption:= '#' + sR + ' - ' + CharToBin(sR[1]) + CharToBin(sR[2]);
  ValueGE.Caption:= '#' + sG + ' - ' + CharToBin(sG[1]) + CharToBin(sG[2]);
  ValueBE.Caption:= '#' + sB + ' - ' + CharToBin(sB[1]) + CharToBin(sB[2]);
end;

procedure TForm_Image.ScrollBarOVChange(Sender: TObject);
var
  RectDest, RectSource: TRect;
begin
  //MyBitmap1.Assign(Original.Picture.Graphic);
  MyBitmap1.LoadFromFile(Form_Utama.OpenPictureDialog.FileName);
  Original.Picture.Bitmap.Assign(MyBitmap1);
  ScrollBarOH.Max:= MyBitmap1.Width - 1 - Original.Width;
  ScrollBarOV.Max:= MyBitmap1.Height - 1 - Original.Height;

  ScrollBarEH.Position:= ScrollBarOH.Position;
  ScrollBarEV.Position:= ScrollBarOV.Position;

  RectDest:= Rect(0, 0, Original.Width, Original.Height);
  RectSource:=          Rect(ScrollBarOH.Position,          ScrollBarOV.Position,
ScrollBarOH.Position + Original.Width, ScrollBarOV.Position + Original.Height);
  Original.Picture.Bitmap.Canvas.CopyRect(RectDest,          MyBitmap1.Canvas,
RectSource);
end;

procedure TForm_Image.ScrollBarEVChange(Sender: TObject);
var
  RectDest, RectSource: TRect;
begin
  //MyBitmap2.Assign(EncryptImage.Picture.Graphic);
  MyBitmap2.LoadFromFile(Form_Utama.SavePictureDialog.FileName);
  EncryptImage.Picture.Bitmap.Assign(MyBitmap2);
  ScrollBarEH.Max:= MyBitmap2.Width - 1 - EncryptImage.Width;
  ScrollBarEV.Max:= MyBitmap2.Height - 1 - EncryptImage.Height;

  ScrollBarOH.Position:= ScrollBarEH.Position;
```

```
  ScrollBarOV.Position:= ScrollBarEV.Position;

  RectDest:= Rect(0, 0, EncryptImage.Width, EncryptImage.Height);
  RectSource:= Rect(ScrollBarEH.Position, ScrollBarEV.Position,
ScrollBarEH.Position + EncryptImage.Width, ScrollBarEV.Position +
EncryptImage.Height);
  EncryptImage.Picture.Bitmap.Canvas.CopyRect(RectDest, MyBitmap2.Canvas,
RectSource);
end;

procedure TForm_Image.FormCreate(Sender: TObject);
begin
  MyBitmap1:= TBitmap.Create;
  MyBitmap2:= TBitmap.Create;
end;

end.
```