PENYANDIAN FILE GAMBAR DENGAN METODE SUBSTITUSI DAN TRANSPOSISI SERTA IMPLEMENTASINYA MENGGUNAKAN BAHASA PEMROGRAMAN BORLAND DELPHI 7.0

Oleh :

Romi Asfanul Khaqim

J2A 605 098

Disusun Sebagai Salah Satu Syarat untuk Memperoleh Gelar Sarjana

Program Strata-1 pada Jurusan Matematika

JURUSAN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS DIPONEGORO

SEMARANG

2011

HALAMAN PENGESAHAN I

Judul : **Penyandian File Gambar Dengan Metode Substitusi dan Transposisi serta Implementasinya Menggunakan Bahasa Pemrograman Delphi 7.0**

Nama Mahasiswa : **Romi Asfanul Khaqim**

NIM : **J2A 605 098**

Telah diujikan pada Ujian Sarjana tanggal 27 Juli 2011 dan dinyatakan lulus pada tanggal Agustus 2011.

Semarang, Agustus 2011

Panitia Penguji Ujian Sarjana

Program Studi Matematika Jurusan Matematika

Ketua,

Drs Kushartantya MIkomp
NIP. 1950 03 01 1979 03 1 003

Mengetahui, Ketua Jurusan Matematika FMIPA UNDIP	Mengetahui, A/n Ketua Program Studi Matematika Sekretaris
Dr. Widowati, S.Si, M.Si NIP. 1969 02 14 1994 03 2 002	Suryoto, S.Si, M.Si NIP. 1968 07 14 1994 03 1 004

HALAMAN PENGESAHAN II

Judul Skripsi : **Penyandian File Gambar Dengan Metode Substitusi dan Transposisi serta Implementasinya Menggunakan Bahasa Pemrograman Delphi 7.0**

Nama Mahasiswa : **Romi Asfanul Khaqim**

NIM : **J2A 605 098**

Telah diajukan pada sidang Tugas Akhir tanggal 27 Juli 2011

Semarang, Agustus 2011

Pembimbing I	Pembimbing II
Drs Eko Adi Sarwoko, M. Kom	Nurdin Bahtiar, S.Si, M. T
NIP. 1965 11 07 1992 03 1 003	NIP. 1979 07 20 2003 12 1 002

KATA PENGANTAR

Puji Syukur penulis panjatkan kehadirat Allah SWT, karena atas segala limpahan rahmat dan hidayah-Nya, penulis dapat menyelesaikan Laporan Tugas Akhir yang berjudul **"Penyandian File Gambar dengan Metode Substitusi dan Transposisi Serta Implementasinya Menggunakan Bahasa Pemrograman Borland Delphi 7.0"**. Penulisan laporan Tugas Akhir ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu (S1) pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Diponegoro di Semarang.

Selama pelaksanaan penyusunan Laporan Tugas Akhir ini, penulis menyadari bahwa penyusunan laporan ini tidak akan berjalan baik tanpa adanya dukungan, bimbingan, arahan dan bantuan dari berbagai pihak yang sangat mendukung. Oleh karena itu dengan segala kerendahan hati, penulis ingin mengucapkan terima kasih dengan tulus kepada :

1. Bapak Dr. Muhammad Nur, DEA selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Diponegoro.
2. Ibu Dr. Widowati, S.Si, M.Si selaku Ketua Jurusan Matematika Fakultas MIPA Universitas Diponegoro.
3. Bapak Bambang Irawanto, S.Si, M.Si selaku Ketua Program Studi Matematika Fakultas MIPA Universitas Diponegoro.
4. Bapak Drs. Eko Adi Sarwoko, M.kom selaku Dosen Pembimbing I yang telah memberikan memberi petunjuk, nasehat, pengarahan serta saran dan bimbingan.

5. Bapak Nurdin Bahtiar, S.Si, M.T selaku dosen pembimbing II yang telah memberi petunjuk, nasehat, pengarahan serta saran dan bimbingan dalam menyelesaikan Laporan Tugas Akhir ini.
6. Ibu Triastuti Wuryandari, S.Si, M.Si Selaku Dosen Wali
7. Bapak dan Ibu dosen Jurusan Matematika atas semua ilmu yang telah diberikan.
8. Ibu dan (*alm*) Bapak tercinta saya yang telah mendoakan dan memberikan semua fasilitas serta semua saudara yang senantiasa selalu memberikan suport dan semangat dalam penyusunan Tugas Akhir ini.
9. Saudara – Saudaraku yang selalu mendukung, menasehati dan memberi semangat dalam pembuatan tugas akhir ini.
10. Semua pihak yang telah membantu dalam penulisan selama ini, yang tidak mungkin disebutkan satu persatu.

Penulis menyadari bahwa Tugas Akhir ini masih banyak kekurangannya. Untuk itu penulis mengharapkan kritik dan saran dari pembaca akan menjadi masukan yang sangat berharga. Semoga Tugas Akhir ini dapat membawa manfaat bagi penulis sendiri khususnya dan bagi para pembaca pada umumnya.

Semarang, Juli 2011

Penulis

ABSTRAK

Ilmu yang mempelajari bagaimana sebuah pesan kita aman sehingga tidak dapat dibaca oleh pihak yang tidak berhak adalah kriptografi. Suatu pesan atau informasi yang merupakan salah satu hal penting dalam berkomunikasi yang perlu untuk dijaga kerahasiaannya. Untuk itu perlu dibuat sebuah aplikasi yang mampu mengamankan informasi pada umumnya dan file gambar pada khususnya. Kriptografi memiliki dua algoritma yaitu enkripsi dan dekripsi yang memungkinkan pesan hanya dapat dibuat dan dibaca oleh yang berhak. Metode substitusi dan transposisi merupakan teknik enkripsi konvensional yang dapat digunakan untuk mengamankan informasi file gambar. Dalam aplikasi digunakan bahasa pemrograman Borland Delphi untuk mengimplementasikan algoritma enkripsi dan dekripsi dari substitusi dan transposisi, sehingga menghasilkan bentuk file gambar yang tidak dapat dibaca oleh pihak lain tanpa melakukan proses dekripsi terlebih dahulu.

Kata kunci : Kriptografi, File Gambar, Enkripsi, Dekripsi, Substitusi, Transposisi.

ABSTRACT

The study of how a message if safe so we can’t be read by unauthorized parties is cryptography. A message or information which is one important thing in communicating that needs to be kept confidential. For that need to be made an application that is able to secure information in general and particular image file. Cryptography has two of the encryption and decryption algorithms that allow massages can only be cre and read be the right. Method of the substitution and transposition is a conventional encryption techniques that can be used to source the information image file. In applications Borland Delphi programming language used to implement algorithms for encryption and decryption of substitution and transposition, resulting in the form of image files that cant not be read other without the decryption process first.

Keywords: Cryptography, Image File, Encryption, Decryption, Substitution, Transposition.

DAFTAR ISI

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR TABEL

Halaman

BAB I

PENDAHULUAN

1.1 Latar Belakang

Saat ini, kemajuan teknologi informasi sedang berkembang dengan pesat yang memungkinkan semua orang dapat berkomunikasi dari satu tempat ke tempat lain yang berjarak ribuan kilometer dengan berbagai media dan berbagai macam bentuk data. Data yang dikirim itu menggunakan jalur transmisi telekomunikasi yang belum tentu terjamin keamanannya. Dengan demikian setiap orang yang bermaksud menyimpan sesuatu secara pribadi dan rahasia akan melakukan segala cara untuk menyembunyikannya sehingga orang lain tidak tahu.

Dengan internet orang bisa *browsing, download, chatting, facebook,* dan sebagainya. Tidak menutup kemungkinan orang akan meng-*upload* ataupun men-*download* gambar dan menyimpannya secara rahasia agar tidak diketahui orang lain. Contoh yang sederhana, ketika mengirim surat kepada seseorang akan membungkus surat tersebut dengan amplop agar tidak terbaca oleh orang lain. Untuk menambah kerahasiaan surat tersebut agar tetap tidak terbaca orang lain dengan mudah apabila amplop dibuka, sehingga diperlukan suatu mekanisme untuk membuat isi surat tidak mudah dipahami.

Untuk mengatasi masalah tersebut diperlukan metode penyandian data yang dikenal dengan ilmu kriptografi, adalah ilmu yang mempelajari bagaimana supaya pesan atau dokumen itu aman, tidak bisa dibaca oleh pihak yang tidak berhak *(anauthorized persons)*. Masalah kerahasiaan ini memang sudah ada jauh sebelum adanya komputer. Julius Caesar, yang khawatir

jangan sampai pesan untuk para jenderalnya jatuh ke tangan musuh, sehingga ia menggunakan metode enkripsi sederhana dengan menggeser huruf abjad dengan nilai tertentu. Munir,2006.[4]

Algoritma kriptografi selalu terdiri dari dua macam yaitu enkripsi dan dekripsi. Teknik untuk menyandikan *plaintext* menjadi *ciphertext* disebut enkripsi, sedangkan proses mengembalikan *ciphertext* menjadi *plaintext* seperti semula dinamakan dekripsi. Metode enkripsi substitusi dan metode enkripsi transposisi merupakan salah satu teknik enkripsi konvensional (simetri) yang digunakan orang sejak berabad-abad lalu untuk mengamankan pesan yang dikirimkan kepada orang lain. Munir,2006.[4]

Sehubungan dengan latar belakang diperlukan pengamanan file untuk di simpan sendiri atau untuk dikirimkan ke pihak lain yang tidak sekedar proteksi *disk* atau pengamanan secara *hardware* saja namun diperlukan salah satu teknik lain untuk pengamanan file. Sehingga penulis bermaksud membahas pembuatan suatu aplikasi yang mampu mengacak posisi piksel pada gambar dengan bahasa pemrograman Borland Delphi 7.0.

1.2 Perumusan Masalah

Keamanan merupakan salah satu faktor penting dalam jaringan komputer, guna menyimpan suatu file atau gambar yang bersifat rahasia dan pribadi agar tidak mudah dipahami oleh orang lain. Sehingga diperlukan beberapa enkripsi guna membuat pesan, file, atau informasi gambar agar tidak dapat dibaca atau dimengerti sembarangan orang, kecuali untuk penerima yang berhak.

Sebagai tolak ukur permasalahan yang dihadapi adalah bagaimana membuat aplikasi yang mampu mengacak piksel file gambar dengan memadukan metode substitusi dan metode transposisi (enkripsi) dan mampu mengembalikan piksel gambar pada posisi semula (dekripsi) dengan mengunakan bahasa pemrograman Borland Delphi 7.0.

1.3 Pembatasan Masalah

Berdasarkan perumusan masalah di atas agar lebih jelas, terdapat batasan – batasan masalah yang akan dibahas lebih khusus difokuskan pada :

1 Dalam pembuatan aplikasi ini hanya akan membahas mengenai penyandian pada file gambar.

2 Gambar yang dapat digunakan hanya berformat BMP *true color* 32 bit, dan ukuran gambar maksimal yang dapat digunakan 3264 x 2448 dengan ukuran panjang lebih besar dari pada tinggi.

3 Pembahasan yang dituliskan adalah terbatas pada proses pengacakan piksel gambar, dengan menggabungkan metode substitusi kaisar dan metode transposisi, kemudian terbatas pada proses pengembalian posisi piksel ke posisi semula.

4 Pengidentifikasian keaslian objek gambar, tidak dibahas dalam penulisan skripsi ini.

5 Aplikasi dibuat dengan menggunakan bahasa pemrograman Borland Delphi 7.0.

1.4 Tujuan dan Manfaat

Tujuan yang ingin dilakukan dalam penulisan tugas akhir ini adalah pembuatan program aplikasi penyandian file gambar dengan mengabungkan

metode substitusi dan transposisi dengan menggunakan bahasa pemprograman Borland Delphi 7.

Sedangkan manfaat yang diharapkan adalah program aplikasi yang mampu menjaga kerahasiaan file gambar agar tidak dapat diketahui oleh pihak ketiga, serta memudahkan pengguna untuk menyimpan file gambar secara rahasia dan pribadi. Selain itu juga bisa sebagai acuan untuk pembuatan aplikasi keamanan yang serupa.

1.5 Metodologi

Metode yang digunakan untuk penyusunan tugas akhir ini adalah sebagai berikut:

1. Jenis Data yang Dibutuhkan

   Jenis data yang dibutuhkan dalam penyusunan tugas akhir ini adalah data sekunder, yaitu data yang diperoleh dari buku-buku, serta literatur lain yang mendukung penyusunan tugas akhir ini.

2. Metode Pengumpulan Data

   Metode pengumpulan data yang digunakan dalam penyusunan tugas akhir ini adalah metode kepustakaan, yaitu metode pengumpulan data dengan cara mengutip serta mempelajari *literature* yang ada hubungannya dengan tugas akhir ini.

3. Metode Pengembangan Sistem

   a. Perencanaan sistem

Dalam tahap ini diuraikan definisi tujuan pembuatan program. Fasilitas yang dibutuhkan program. Fasilitas yang diperlukan agar sistem yang dibangun dapat berjalan dengan baik dan teruji kemampuannya.

b. Analisa sistem

Pada tahap ini dilakukan pemahaman struktur perangkat lunak, menguraikan bahasa pemrograman yang digunakan untuk memudahkan mencari solusi hipotesa maupun algoritma yang akan digunakan dan cara penulisan yang nantinya akan dipakai. Analisa yang dibutuhkan meliputi analisa kebutuhan perangkat lunak dan perangkat keras, analisa kebutuhan sistem, dan analisa kebutuhan proses.

c. Perancangan sistem

Perancangan sistem pada tugas akhir ini menggunakan perangkat pemodelan logik seperti membuat diagram konteks yang menggambarkan hubungan sistem dengan lingkungan, membuat diagram alir yang merupakan bentuk lebih detil dari diagram konteks, membuat desain pembuka, main menu, input output dan berdasarkan algoritma dan pemrograman dengan *flowchart* yang sudah dibuat.

d. Implementasi sistem

Implementasi sistem merupakan penerapan rancangan sistem yang telah dibuat ke dalam bahasa pemrograman Borland Delphi 7.0 yang akan digunakan.

e. Pengujian

Pengujian ini akan menguji sistem secara keseluruhan apakah aplikasi yang dibuat telah dapat berjalan dengan benar dan sesuai dengan tujuan yang ingin dicapai.

1.6 SISTEMATIKA PENULISAN

Penulisan tugas akhir ini dengan judul "Penyandian File Gambar dengan Metode Substitusi dan Metode Transposisi serta Implementasinya Menggunakan Bahasa Pemrograman Borland Delphi 7.0" memiliki alur penyusunan sebagai berikut :

BAB I. PENDAHULUAN

Bagian pendahuluan ini berisi latar belakang pembuatan, permasalahan, pembatasan masalah, tujuan dan manfaat, metodologi, serta sistematika penulisan tugas akhir.

BAB II. DASAR TEORI

Bagian ini berisi tentang pembahasan mengenai pengolahan citra digital pengertian penyandian file gambar, kriptografi, teknik kriptografi klasik, pemrograman Borland Delphi 7.0

BAB III. ANALISA DAN PERANCANGAN SISTEM

Bagian ini berisi tentang analisa-analisa pendukung pembuatan sistem, rancangan-rancangan untuk membuat aplikasi ini termasuk diagram konteks dan flowchart program, serta perancangan antar muka program.

BAB IV. PEMBAHASAN DAN IMPLEMENTASI SISTEM

Bagian ini berisi penerapan atau implementasi rancangan yang dibuat, pembahasan aplikasi program, dan pembuatan dialog menu utama. Hasil pembuatan aplikasi diimplementasikan pada proses enkripsi dan diskripsi gambar. Setelah diuji kemudian dibahas juga tentang analisa hasil program.

BAB V. PENUTUP

Bagian ini berisi kesimpulan dari pembahasan yang dilakukan, yang tidak terlepas dari tujuan pembuatan aplikasi dan saran-saran untuk perbaikan aplikasi ini.

BAB II

LANDASAN TEORI

2.1. Citra Digital

Sebuah citra digital adalah kumpulan piksel-piksel yang disusun dalam larik dua dimensi yang dapat diobservasi oleh sistem visual manusia. Ditinjau dari sudut pandangan matematis, citra merupakan fungsi menerus *(continue)* dari intensitas cahaya pada bidang dwimitra. Sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya tersebut. Pantulan cahaya ini ditangkap oleh alat-alat optik, misalnya mata manusia, kamera digital, dan sebagainya, sehingga banyak objek citra tersebut terekam. Munir, 2004.[3]

Citra terbagi menjadi dua jenis citra yaitu citra diam dan citra bergerak. Citra diam adalah citra tunggal yang bergerak *(moving images)* adalah rangkaian citra diam yang ditampilkan secara berurutan (sekuensial) sehingga memberi kesan pada mata kita sebagai gambar yang bergerak. Sedangkan citra diam adalah citra yang tidak bergerak. Indek baris dan kolom *(x,y)* dari sebuah piksel dinyatakan dalam bilangan bulat. Piksel (0,0) terletak pada sudut kiri atas pada citra, indek *x* bergerak ke kanan dan indek *y* bergerak ke bawah. Ahmand, 2005.[11]

Agar dapat diolah dengan komputer digital, suatu citra harus direpresentasikan secara numerik dengan nilai-nilai diskrit. Representasi citra dari fungsi kontinu menjadi nilai-nilai diskrit disebut pencitraan (*imaging*) atau digitalisasi. Citra yang dihasilkan inilah yang disebut citra digital (*Digital Image*), dinyatakan sebagai kumpulan piksel dalam matrik dua dimensi. Pada umumnya

citra digital berbentuk empat persegi panjang dan dimensi ukurannya dinyatakan tinggi dikalikan dengan lebar atau lebar dikalikan dengan panjang.

Citra digital yang berukuran MxN lazim dinyatakan dengan matriks yang berukuran M baris dan N kolom seperti pada gambar 2.1 :

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

Gambar 2.1: Representasi citra digital dalam matriks N x M Gonzalez, 1987.[8].

Setiap elemen pada citra digital (berarti elemen matriks) disebut sebagai *picture element* atau piksel (*pixel*). Jadi citra yang berukuran M x N mempunyai MN buah piksel. Misalkan sebuah citra *digital* berukuran 256 x 256 piksel dengan derajat keabuan 256 level dan dipresentasikan secara numerik dengan matriks terdiri dari 256 baris (di indek dari 0 sampai 255) dan 256 kolom seperti contoh pada gambar 2.2 :

(0,0)

$$\begin{bmatrix} 0 & 134 & 145 & \cdots & 201 \\ 10 & 110 & 145 & \cdots & 212 \\ 90 & 78 & 152 & \cdots & 199 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 132 & 154 & 128 & \cdots & 222 \end{bmatrix}$$

(255,255)

(0,0)

0	134	145	...	210
10	110	145	...	212
90	78	152	...	199
⋮	⋮	⋮	⋮	⋮
132	154	128	...	222

(255,255)

Gambar 2.2: Contoh Representasi citra dalam matriks N x M. Gonzalez, 1987.[8]

Piksel pertama pada koordinat (0,0) mempunyai intensitas 0 yang berarti warna piksel tersebut hitam dan intensitas 255 tampak sebagai titik putih, piksel kedua pada koordinat (0,1) mempunyai intensitas 134 yang berarti warna antara hitam dan putih. Munir, 2004.[3]

Citra monokrom atau citra hitam putih merupakan citra satu kanal dimana citra f(x,y) merupkan fungsi tingkat keabuan dari hitam keputih, x menyatakan variable baris atau garis jelajah dan y menyatakan variable kolom atau posisi piksel garis jelajah. Sebaliknya citra berwarna dikenal juga citra multi-spektral, dimana warna citra biasanya dinyatakan dalam tiga komponen warna: merah, hijau, biru (RGB), citra berwarna $\{f_{merah}$ (x,y), f_{hijau} (x,y), f_{biru} (x,y)$\}$ merupakan fungsi harga vektor tingkat keabuan merah hijau dan biru. Murni, 1992. [1].

2.2. Format File Bitmap (BMP)

Format citra yang baku di lingkungan sistem operasi Microsoft Windows adalah file bitmap (BMP). Pada saat ini format BMP kurang begitu populer dan mulai jarang digunakan dibanding format JPG atau GIF, karena file BMP pada umumnya tidak dimampatkan, sehingga ukuran relatif lebih besar dari pada file JPG atau GIF.

Terjemahan bebas bitmap adalah pemetaan bit. Artinya nilai intensitas piksel di dalam citra dipetakan ke sejumlah bit tertentu. Peta bit umumnya adalah 8, yang berarti setiap piksel panjangnya 8 bit. Delapan bit ini mempresentasikan nilai intensitas piksel. Dengan demikian ada sebanyak 2^8 =256 derajat keabuan, mulai dari 0 (00000000) sampai 255 (11111111).

Terdapat tiga macam citra dalam format BMP, yaitu citra biner, citra berwarna dan citra hitam-puth (*grayscale*). Citra biner hanya memiliki dua nilai keabuan 0 dan 1. Oleh kerena itu 1 bit telah cukup untuk mempresentasikan nilai piksel. Citra berwarna adalah citra yang lebih umum. Warna yang terlihat di dalam citra bitmap merupakan kombinasi dari tiga komponen warna, yaitu :

R (Red), G (Green) dan B (Blue). Kombinasi dari tiga warna RGB tersebut menghasilkan warna yang khas untuk piksel yang bersangkutan. Pada citra 256 warna, setiap piksel memiliki panjang 8-bit, akan tetapi komponen RGBnya disimpan dalam tabel RGB yang disebut *palet.* Berikut ini akan memperlihatkan panjang informasi palet untuk detiap versi *bitmap*, masing-masing untuk citra 16 warna, 256 warna dan 16,7 juta warna. Berkas citra 24-bit tidak mempunyai palet RGB, karena langsung di uraikan ke dalam data *bitmap*. Lihat tabel 2.1

Tabel 2.1 : Panjang informasi *palet bitmap* berwarna

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

Informasi *palet* warna terletak sesudah *header bitmap*. Informasi *palet* warna dinyatakan dalam satu tabel RGB. Setiap *entry* pada tabel terdiri atas tiga buah *field* yaitu, R (*Red*), G (*Green*), dan B (*Blue*). Data bitmap diletakan sesudah informasi *palet*. Munir, 2004. [3]

Format citra 8-*bit* dapat dilihat pada gambar 2.4. format citra 4-*bit* (16 warna), hampir sama dengan format citra 8-*bit.* Pada citra 4-*bit* dan citra 8-*bit*, warna suatu piksel di acu dari tabel informasi palet *entry* ke-k (*k* merupakan nilai rentang 0-15 untuk citra 16 warna dan 0-155 untuk citra 256 warna). Sebagai contoh pada gambar 2.4, piksel pertama bernilai 2, warna piksel pertama ini ditentukan oleh komponen RGB pada *palet* warna *entry* ke-2, yaitu R=14, G=14 dan B=16.piksel kedua serupa dengan piksel pertama. Piksel ketiga bernilai 1, warna ditentukan oleh komponen RGB pada *palet* warna *entry* ke-1, yaitu R=20, G=45 dan B=24. Demikian seterusnya untuk piksel-piksel lainnya. Khusus untuk citra hitam-putih 8-*bit*, komponen R,G dan B suatu piksel bernilai sama dengan

data bitmap piksel tersebut. Jadi piksel dengan nilai data bitmap 129, memiliki nilai R=129, G=129 dan B=129. ,lihat gamabr 2.3 Munir, 2004. [3]

<header berkas>
<header bitmap>
<palet warna>

	R	G	B
1	20	45	24
2	14	13	16
3	12	17	15
…	…	…	…
255	46	78	25

<data bitmap>
2 2 1 1 1 3 5…

Gambar 2.3: Format citra 8-*bit* (256 warna) (Munir, 2004) [3]

Citra yang lebih kaya warna adalah citra 24-*bit*. Setiap piksel panjangnya 24-*bit*, karena setiap bit langsung menyatakan komponen warna merah (8-bit), komponen warna hijau (8-*bit*) dan komponen warna biru (8-*bit*). Citra 24-*bit* juga disebut citra 16 juta warna karena mampu menghasilkan 2^{24} =16.777.216 kombinasi warna. Contohnya seperti pada Gambar 2.4 berikut ini, dimana piksel pertama memiliki nilai R=20, G=19 dan B=21. Piksel kedua memiliki nilai R=24, G=24 dan B=23 dan demikian seterusnya. Munir, 2004. [3]

<header berkas>
<header bitmap>

<data bitmap>
20 19 21 24 24 23 24

Gambar 2.4 : Format citra 24-*bit* (16,7 juta warna) Munir, 2004. [3]

2.3. Pengertian Penyandian File Gambar

Kata penyandian file gambar terdiri dari tiga buah kata yaitu pertama adalah penyandian, mempunyai kata dasar “sandi” yang menurut kamus besar Bahasa Indonesia berarti kode, sedangkan penyandian adalah sebuah bentuk kata

kerja yang berarti suatu kegiatan menyandikan atau mengkodekan dengan tujuan tertentu. Kedua adalah file yaitu sebutan sekumpulan *byte* atau deretan karakter atau kode-kode yang membentuk sebuah dokumen yang memiliki nama yang unik, sedangkan yang ketiga adalah kata gambar yang dalam kamus besar bahasa Indonesia adalah "citra", citra adalah objek elemen-elemenya dinyatakan dengan suatu besaran numerik yang membentuk (*array*). Murni, 1992. [1]. Sehingga penyandian file gambar dapat diartikan kegiatan menyandikan atau mengkodekan sekumpulan elemen penyusun gambar (piksel) dengan tujuan mengamankan informasi dari pihak yang tidak berhak.

2.4. Kriptografi

Penyandian merupakan salah satu alternatif atau cara untuk mengamankan atau menjaga suatu kerahasiaan data atau gambar. Seni dan ilmu untuk menyandikan atau menjaga keamanan atau serta kerahasian pesan disebut kriptografi. Kurniawan, 2004. [5]

Kriptografi juga dapat diartikan sebagai ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti keabsahan, integritas data, serta autentikasi data, dengan kata lain kriptografi digunakan untuk menjamin keleluasaan pribadi dan pembuktian keaslian pesan dalam berkomunikasi.

Kriptografi sendiri berasal dari bahasa yunani yaitu *kryptos* yang artinya "*secret*" (rahasia) dan *graphein* yang artinya *"writing"* (tulisan), jadi kriptografi adalah "*secret writing*" (tulisan rahasia). Munir,2006. [4]. Informasi atau pesan adalah salah satu hal penting yang harus disampaikan dalam berkomunikasi. Pesan yang disampaikan dari satu pihak ke pihak yang lain dapat berupa file teks, file suara, maupun pesan yang berupa file gambar. Dalam menyampaikan sebuah

informasi atau pesan ke pihak lain, kerahasian dan keaslian pesan perlu dijaga. Sehingga pesan perlu disandikan sebelum dilakukan pengiriman.

Pada dasarnya kriptografi terdiri dua algoritma yaitu, algoritma enkripsi (E) dan algoritma dekripsi (D). Dasar matematis yang mendasari proses enkripsi dan dekripsi adalah relasi antara dua himpunan yaitu himpunan yang berisi elemen *plaintext* dan himpunan yang berisi himpunan *ciphertext*. Pesan atau informasi yang dapat dibaca disebut sebagai *plaintext*, sedangkan teknik untuk membuat pesan tidak dapat terbaca disebut enkripsi. Pesan yang sudah melewati tahap enkripsi disebut *ciphertext*, sedangkan dekripsi adalah teknik untuk mengubah *ciphertext* menjadi *plaintext*. Kurniawan,2004. [5]. sedangkan kunci atau *key* adalah parameter yang digunakan untuk transformasi enkripsi dan dekripsi. Kunci biasanya berupa string atau deret bilanngan. Munir, 2006. [4].

Dalam menyandikan pesan atau mengenkripsi pesan, terdapat dua jenis algoritma yang berdasar jenis kuncinya, yaitu :

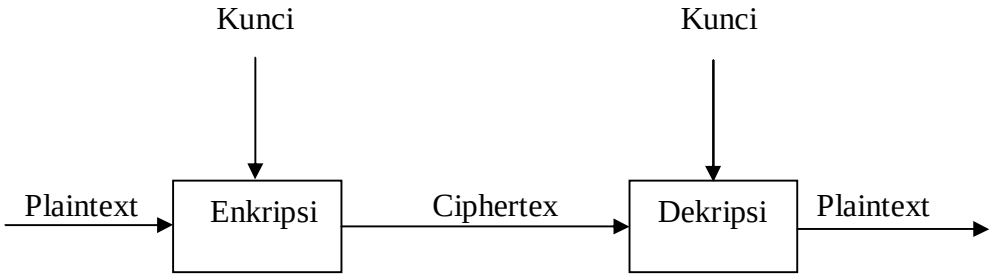
1. Algoritma Simetri (menggunakan satu kunci untuk enkripsi dan dekripsinya).
2. Algoritma Asimetri (menggunakan kata kunci yang berbeda untuk enkripsi dan dekripsinya).

Algoritma simetri disebut juga sebagai algoritma konvensional adalah algoritma untuk membuat pesan yang disandikan menggunakan satu kunci untuk enkripsi dan dekripsinya. kurniawan, 2004. [5].

Disebut konvensional karena algoritma yang biasa digunakan orang sejak berabad-abad yang lalu adalah algoritma jenis ini. Algoritma simetrik sering juga disebut sebagai algoritma kunci rahasia, algoritma kunci tunggal, atau algoritma satu kunci, dan mengharuskan pengirim dan penerima menyetujui suatu kunci

tertentu sebelum mereka dapat berkomunikasi dengan aman. Keamanan algoritma simetri tergantung pada kunci, agar komunikasi tetap aman kunci harus tetap dirahasiakan. kurniawan, 2004. [5].

Proses enkripsi-dekripsi algoritma kriptografi kunci simetris dapat dilihat pada gambar 2.5 :



Gambar 2.5**.** Proses enkripsi-dekripsi kunci simetris Munir, 2006. [4].

Algoritma kunci simetri mengacu pada metode enkripsi yang dalam hal ini baik penggirim maupun penerima memiliki kunci yang sama. Agoritma kunci simetri modern beroperasi pada bit dan dapat dikelompokkan menjadi dua. Munir, 2006. [4].

1. Algoritma aliran (*Stream Ciphers*) algoritma kriptografi beroperasi pada plainteks atau cipherteks dalam bentuk bit tunggal, yang dalam hal ini rangkaina bit dienkripsikan atau didekripsikan bit per bit. Algoritma aliran mengenkripsi satu bit setiap kali.
2. Algoritma blok (*Block Ciphers*) algoritma kriptografi beroperasi pada plainteks atau cipherteks dalam bentuk blok bityang panjangnya sudah ditentukan sebelumnya. Misalnya panjang blok adalah 64 bit, maka itu berarti algoritma enkripsi melakukan 8 karakter setiap kali enkripsi (1 karakter = 8 bit dalam pengkodean *ASII*).

Adapun yang akan digunakan dalam penulisan tugas akhir kali ini adalah algoritma simetri atau algoritma klasik (konvensional), karena memakai kunci

yang sama untuk kegiatan enkripsi dan dekripsinya. Keamanan dari pesan yang menggunakan algoritma ini tergantung pada kunci, jika kunci tersebut diketahui oleh orang lain, maka orang tersebut bisa melakukan enkripsi dan dekripsi terhadap pesan tersebut.

2.5. Algoritma Kriptografi Klasik

Pada algoritma kriptografi klasik (simetri), merupakan algoritma kriptografi yang biasa digunakan orang sejak berabad-abad yang lalu dengan berbasis karakter, yaitu enkripsi dan dekripsi dilakukan pada setiap karakter pesan. Munir,2006.[4]. Dan pada dasarnya, algoritma kriptografi klasik dapat dikelompokkan kedalam dua metode dasar yang biasa digunakan, yaitu:

a. Metode substitusi
b. Metode transposisi

Tiga alasan dasar menggunakan algoritma kriptografi klasik adalah :

1. Memahami konsep dasar kriptografi.
2. Dasar dari algoritma kriptografi modern.
3. Untuk memahami kelemahan sistem kode.

2.5.1. Metode Penyandian Substitusi

Substitusi adalah penggantian setiap karakter *plaintext* dengan karakter lain. kurniawan,2004. [5]. Dengan kata lain teknik substitusi adalah suatu teknik enkripsi simetri dimana dilakukan penggantian setiap objek *plaintext* dengan obyek lain, teknik ini menerapkan konsep korespondensi satu-satu untuk tiap-tiap objek *plaintext* yang akan disandikan. Kemudian dalam perkembangannya, dalam metode penyandian substitusi modern, digunakan sebuah program aplikasi tertentu dimana teks asli yang berbentuk kumpulan karakter dalam sebuah file

digital diganti dengan kumpulan karakter lain secara digital sehingga menghasilkan file sandi yang siap dikomunikasikan.

Terdapat empat istilah substitusi kode, Aryus, 2008. [2]. antara lain :

a. *Monoalphabetic* : setiap karakter teks-kode mengganti salah satu karakter teks-asli.
b. *polyalphabetic* : Setiap karakter teks-kode dapat menggantikan lebih dati satu macam karakter teks-asli.
c. *Monograf* : satu enkripsi dilakukan terhadap satu karakter teks-asli.
d. *Polygraph* : satu enkripsi dilakukan terhadap lebih dari satu karakter teks asli.

2.5.1.1. Subtitusi Kode kaisar

Subtitusi kode yang pertama dalam dunia penyandian terjadi pada pemerintahan Yulius Caesar yang dikenal dengan kode kaisar, yaitu dengan mengganti posisi huruf awal alphabet. Ariyus, 2008. [2]

Salah satu contoh cara substitusi kaisar adalah dengan dengan pergeseran huruf, urutan abjad ABCD…….Z bisa digeser sebanyak 1 huruf sehingga huruf A akan menjadi B, B akan menjadi C dan seterusnya. Pergeserannya bisa dibuat lebih banyak yaitu 2 huruf sehingga huruf A akan menjadi C, B akan menjadi D dan seterusnya. Pergeseran bisa lebih banyak lagi tergantung bagaimana kita merumuskannya. Cara pergeseran ini termasuk *monoalphabetic* di mana satu huruf pasti akan berubah menjadi huruf tertentu yang lain. Karena relasi antara huruf plaintext dan huruf ciphertext satu-satu, yang artinya suatu huruf plaintext pasti menjadi suatu huruf ciphertext tertentu. kurniawan, 2004. [5]. Pergeseran kunci tergantung dari keingginan pengguna metode ini. Lihat gambar 2.6.

Plaintext	Ciphertext
ABC	DEF
Hello	Khoor
Attack	Dwwdfn

Secara lebih detail dapat diperhatikan contoh berikut :

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Menjadi :

D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	0	1	2

Gambar 2.6 . Contoh Substitusi kode kaisar Ariyus, 2008. [2].

Contoh algoritma kode kaisar ialah untuk teks asli diberikan simbol "P" dan teks kodenya "C" dan kunci "K". Sehingga dapat dibuat rumusnya sebagai berikut : (Ariyus, 2008). [2].

Proses enkripsi : C = E(P) = (P+K) mod (26)

Proses Dekripsi : P = D(C) = (C-K) mod (26)

Pada contoh di atas dapat dimasukkan kunci dengan nilai tiga sehingga manjadi :

C=E(P) = (P+3) mod (26)

untuk proses enkripsinya dan untuk dekripsinya di dapat sebagai berikut :

P=D(C) = (C-3) mod (26)

2.5.1.2. Substitusi Deret Campuran Kata Kunci

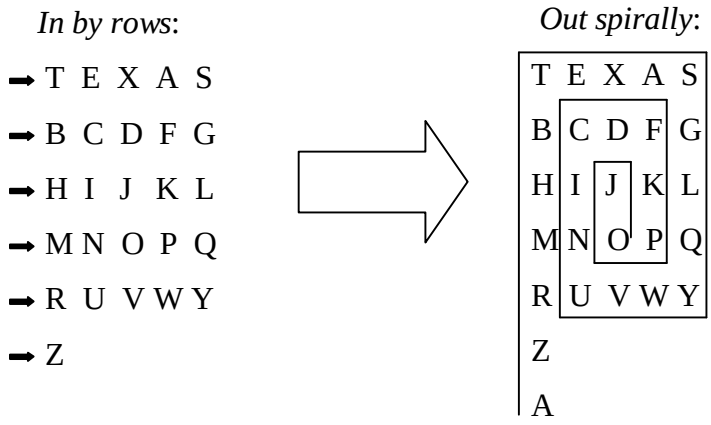
Substitusi deret campuran kata kunci adalah substitusi yang kata kuncinya didapat dari mengumpulkan karakter yang sama dari sebuah plaintext dan pada ciphertextnya ditambahkan semua sisa karakter dalam abjad. Sebagai contoh kata kunci : ***MILTARY INTELLIGENCE***, untuk menjadikan sebagai kunci yang dapat digunakan, maka kata kunci

tersebut harus disederhanakan dengan cara huruf yang sama dianggap satu huruf. Jadi huruf I yang ada pada kata kunci yang terdiri dari 4 huruf dianggap 1 huruf, L yang berjumlah 3 huruf dianggap 1 dan seterusnya. Sehingga kunci tersebut menjadi : ***MILTARYNEGC***

Kemudian plaintext dengan abjad (ABCD…Z) sisanya disusun di belakang kata kunci tersebut, Untuk mensubstitusi plaintext (abcd…z) dengan kata kunci ***MILTARY INTELLIGENCE*** maka diperlukan plaintext "a" lalu disubstitusi M, "b" disubstitusi I dan seterusnya, sehingga menjadi MILTARYNEGC BDFHJKOPQSUVWXZ. kurniawan, 2004. [5].

Selain contoh di atas, substitusi deret campuran kata kunci juga bisa dibentuk dengan menggunakan bentuk spiral.

Contoh substitusi deret campuran kata kunci dalam bentuk sepiral dengan kunci : *TEXAS* dapat dilihat pada gambar 2.7.



Gambar 2.7 : Substitusi deret campuran kata kunci bentuk spiral. kurniawan, 2004. [5]

Dari contoh di atas menunjukkan teknik substitusi deret campuran kata kunci dengan kunci TEXAS dengan mengikuti arah jarum jam, sehingga diperoleh ciphertext : ZRMHBTEXASGLQYWVUNICDFKPOJ

2.5.2. Metode Penyandian Transposisi

Teknik transposisi pada dasarnya adalah membuat *ciphertext* dengan menggantikan posisi objek-objek *plaintext* tanpa menggantikan objek *plaintext* tersebut, jadi pada teknik transposisi ini tidak diperlukan karakter lain. Pada teknik transposisi ini pembuatan *ciphertext* dilakukan dengan pembacaan nilai matrix pada kolom per kolom sesuai dengan kunci yang digunakan. kurniawan, 2004. [5]

Teknik ini menggunakan permutasian karakter sebagai contoh cipher dari plaintext “saya sedang belajar kriptografi” lihat gambar 2.8.

kunci	**4**	**3**	**1**	**5**	**2**	**6**
Plaintext	**s**	**a**	**Y**	**a**	**s**	**e**
	d	**a**	**N**	**g**	**b**	**e**
	l	**a**	**J**	**a**	**r**	**k**
	r	**i**	**P**	**t**	**o**	**g**
	r	**a**	**F**	**i**	**y**	**Z**

Gambar 2.8 : Contoh Metode Transposisi kurniawan, 2004. [5]

Plaintext disusun ke kanan kemudian ke bawah. kuncinya adalah 4 3 1 5 2 6, sehingga keluaran cipher mengikuti kunci menurun ke bawah ; ynjpf sbroy aaaia sdirr agati eekgz. Karakter y dan z ditambahkan untuk menutupi jejak bahwa jumlah karakter yang sebenarnya hanya sebanyak 4 kolom sehingga lebih mempersulit analisis cipher.

Kunci dapat diperoleh dari kata yang mudah dibaca dan kemudian dikodekan menjadi bilangan. Sistem ini dinamakan algoritma transposisi kolom dengan kunci *numeric*. kurniawan, 2004. [5]. Misalnya :

Kata Sandi	S	a	n	t	a	n
Bilangan	5	1	3	6	2	4

Di sini huruf a yang didobel diberi nomer 1 dan 2, kemudian huruf yang dekat dengan a yaitu n, diberi nomer 3 dan 4 karena dobel, sedangkan huruf berikutnya diberi angka 5 dan 6. Bilangan 513624 dapat digunakan untuk menjadi kunci pada transposisi sebelumnya menggantikan kunci 435126 yang dipilih secar acak. Pemilihan secara acak lebih aman, namun juga lebih sukar untuk diingat. Cipher transposisi yang seperti ini mudah karena memiliki frekuensi kemunculan huruf yang sama seperti plaintext asalnya. Pemecahan kode dilakukan dengan cara mencoba-coba plaintext disusun menurut baris dari kolom digraph dan trigraph juga akan sangat membantu. kurniawan, 2004. [5].

2.6. Diagram Arus Data *(Data Flow Diagram)*

Sebelum mengimplementasi program, maka dilskukan pembutan DFD atau *Data Flow Diagram* (DFD) atau dalam bahasa indonesia menjadi diagram alir data (DAD) adalah sebuah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*). Rosa, salahuddin, 2011.[10].

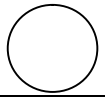
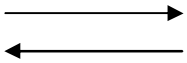
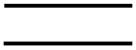
DFD dapat digunakan untuk merepresentasikan sebuah sistem atau perangkat lunak pada beberapa level abstraksi. DFD dapat dibagi menjadi beberapa level yang lebih detail untuk merepresentasikan aliran informasi atau fungsi yang lebih detail. DFD menyediakan mekanisme untuk pemodelan

fungsional ataupun pemodelan informasi. Oleh karena itu, DFD lebih sesuai digunakan untuk memodelkan fungsi-fungsi perangkat lunak yang akan diimplementasikan menggunakan pemrograman terstuktur. Rosa, Salahuddin, 2011. [10].

DFD menggambarkan penyimpanan data dan proses yang mentransformasikan data. DFD menunjukkan hubungan antara data pada sistem dan proses pada sistem.

Ada beberapa simbol DFD, salah satu diantaranya menurut Yourdon/ De Marco (Lihat tabel 2.2).

Tabel 2.2 Simbol – Simbol DFD menurut Yourdon/ De Marco

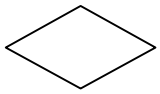
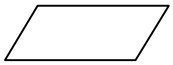
Simbol	Nama Simbol	Fungsi / Keterangan
	Proses	Tempat terjadinya kegiatan pengolahan/proses
	Terminator	Entitas luar yang terlibat langsung dengan sistem
	Flow	Menunjukan arah aliran dari dan kemana
	Storage	Sebagai alat penyimpan

2.7. Diagram Alir (*Flowchart*)

Flowchart adalah bagan yang memperlihatkan urutan prosedur dan proses dari beberapa file didalam media tertentu. Melalui *flowchart* dapat terlihat jenis media penyimpanan yang dipakai dalam pengolahan data. Selain itu juga menggambarkan file yang dipakai sebagai *input* maupun *output*. Tosin, 1997. [9]

Flowchart disusun dengan simbol. Simbol – simbol dapat dipakai sebagai alat bantu menggambarkan proses di dalam program. Simbol tersebut dapat dilihat dalam tabel 2.3.

Tabel 2.3 Simbol-Simbol *Flowchart*

Simbol	Nama Simbol	Fungsi / Keterangan
	Proses	Menunjukan kegiatan proses dari operasi program komputer
	Terminal	Menunjukan awal dan akhir dari suatu proses
	Keputusan	Digunakan untuk suatu penyelesaian kondisi dalam program
	Input / Output	Digunakan untuk mewakili data input/output
	Aliran Data	Menunjukkan petunjuk dari aliran fisik pada program

2.8. Pemrograman Borland Delphi 7

Delphi berasal dari bahasa pemrograman yang cukup terkenal, yaitu bahasa *pascal*. Bahasa pascal diciptakan pada tahun 1971 oleh ilmuwan dari swiss, yaitu Niklaus Wirth. Nama *pascal* diambil dari ahli matematika dan filsafat Perancis, yaitu *Blasie Pascal* (1623-1622).

Karena pemrograman *windows* dengan *turbo pascal* masih dirasa cukup sulit, maka sejak tahun 1993 *Borland International* mengembangkan bahasa pascal yang bersifat visual. Hasil dari pengembangan ini adalah dirilisnya Delphi 1 pada tahun 1995. Perkembangan Delphi tidak berhenti sampai di situ. Pada tahun berikutnya 1996, *Borland International* merilis Delphi 2 untuk *windows* 95/NT. Kemudian dalam tahun-tahun berikutnya, *Borland International* merilis

beberapa versi pengembangan Delphi yang memiliki tambahan fitur baru dibandingkan dengan versi sebelumnya. malik.2006. [7].

2.8.1 Bagian Utama Borland Delphi 7

Pada dasarnya IDE Delphi dibagi menjadi tujuh bagian utama yaitu *Menu*, *Speed Bar*, *Componen Palette*, *Form Designer*, *Code Explorer*, *Object Tree View* dan *Object Inspector*.

2.8.1.1 Menu

Menu pada Delphi memiliki kegunaan seperti pada aplikasi windows lainnya. Dari menu ini, bisa memanggil atau menyimpan program, menjalankan program, dan lain sebagainya. Sesuatu yang berhubungan dengan IDE Delphi dapat dilakukan dari menu.

2.8.1.2 *Speed Bar*

Speed bar atau sering juga disebut *toolbar* berisi kumpulan tombol yang tidak lain adalah pengganti dari beberapa item menu yang sering digunakan. Dengan kata lain, setiap tombol pada speed bar menggantikan salah satu item menu.

2.8.1.3 *Component Palette*

Component palette berisi kumpulan icon yang melambangkan komponen-komponen pada VCL (*Visual Component Library*) atau CLX (*Component Library for Cross Platform*). Pada komponen palette terdapat beberapa tab yaitu : *standard*, *additional*, *data access*, dan tab yang lainnya. *Icon* yang ditampilkan pada *Component Palette* tidak memiliki keterangan yang menyatakan nama komponen. Lihat gambar 2.9. malik, 2005. [6].

Gambar 2.9 : *Component Palette* Delphi

2.8.1.4 *Form Designer*

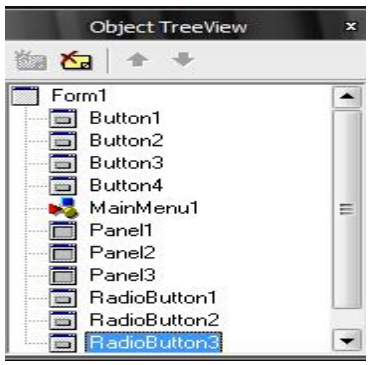
Sesuai dengan namanya, *form designer* merupakan tempat untuk merancang jendela aplikasi. Perancangan *form* dilakukan dengan meletakkan komponen-komponen yang diambil dari *component palette*.

2.8.1.5 *Code Explorer*

Code explorer adalah tempat dimana akan menuliskan program. Tempat meletakan pernyataan-pernyataan dalam bahasa *object pascal*. Yang perlu diperhatikan dalam *code explorer* adalah tidak perlu menuliskan semua kode sumber. IDE Delphi telah menuliskan semacan kerangka program untuk kita. malik, 2005. [6]

2.8.1.6 *Object Treeview*

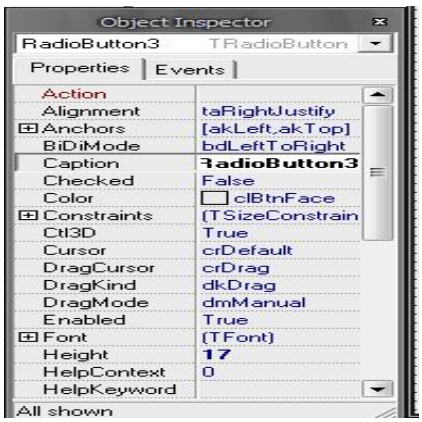
Object treeview berisi daftar komponen yang telah kita letakan pada form *designer*. Lihat gambar 2.10.



Gambar 2.10 : *Object Treeview* Delphi

2.8.1.7 *Object Inspector*

Object Inspector digunakan untuk mengubah karakteristik sebuah komponen, pada *object inspector* kita dapat melihat dua tag yaitu *properties* dan *events*. Kita dapat mengaktifkan salah satu tag dengan mengklik *properties* atau *events*. Pada tag *properties* kita bisa mengubah properti dari komponen. Pada tag *events* kita dapat menyisipkan kode untuk menangani kejadian tertentu. Lihat gambar 2.11.



Gambar 2.11 : *Object Inspector* Delphi

2.7.2 *Variable*

Dalam dunia pemrograman, *variable* digunakan untuk menyimpan data. Pada Delphi, pendeklarasian *variable* mengikuti sintaks berikut :

Var nama_variabel1:tipe_variabel;

Variable pada Delphi harus mengikuti beberapa aturan sebagai berikut :

- Nama *variable* maksimum terdiri dari 63 karakter.
- Nama *variable* hanya boleh mengandung huruf, angka garis bawah (_) dan tidak boleh diawali dengan angka.
- Kita tidak bisa menggunakan kata kunci milik Delphi. Sebagai contoh *variable* dengan nama *if, else for* tidak di perbolehkan.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini akan dijelaskan mengenai analisis pembuatan sistem serta perancangan desain sistem.

3. 1 Analisis Pembuatan Sistem

Untuk mencapai tujuan dari pembuatan sistem diperlukan tahapan analisis – analisis pendukung sistem. Dengan menganalisis secara teliti diharapkan tidak menemui hambatan – hambatan yang berarti dalam mengembangkan sistem ini.

3.1.1. Analisis Kebutuhan Perangkat Lunak dan Perangkat Keras

1. Kebutuhan perangkat lunak untuk mengimplentasikan pembuatan aplikasi adalah sebagai berikut :
   - ✓ Windows 98/2000/ Windows XP/ Windows Vista/ Windows 7
   - ✓ Software Borland Delphi 7.0
   - ✓ File Gambar untuk proses kriptografi, antara lain: BMP
2. Kebutuhan perangkat keras untuk mengimplementasikan pembuatan aplikasi adalah sebagai berikut :
   - ✓ Sistem komputer processor 1 Ghz ke atas.
   - ✓ VGA mampu untuk menampilkan 32 bit atau resolusi gambar 1024 x 768.
   - ✓ Memori RAM minimal 256 Mbyte
   - ✓ Free space harddisk minimum 500 Mbyte
   - ✓ Keyboard dan mouse.

3.1.2 Analisis Kebutuhan Sistem

Aplikasi perangkat lunak merupakan suatu sistem yang digunakan untuk mengaplikasikan proses enkripsi dan proses dekripsi dengan menggunakan teknik enkripsi konvensional.

Pada sistem ini terdapat batasan yang jelas sebagai tujuan utamanya. Hal ini berguna agar perangkat lunak tersebut tidak keluar dari jalur atau rencana yang telah ditetapkan. Diberikan kebutuhan sistem yang dibangun diantaranya adalah :

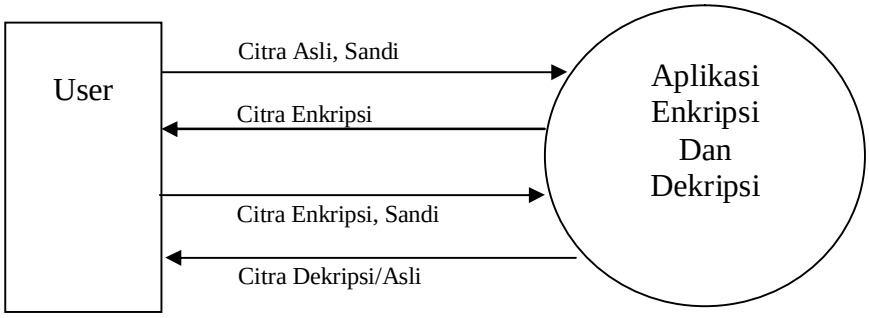
1. Mempunyai kemampuan untuk memasukkan gambar asli yang selanjutnya dilakukan proses enkripsi.
2. Mempunyai kemampuan untuk melakukan enkripsi dengan substitusi yang selanjutnya dilakukan transposisi kemudian dilakukan proses.
3. Mampu melakukan proses dekripsi, yaitu *invers* transposisi dan *invers* substitusi.
4. Mampu menampilkan waktu yang di butuhkan untuk proses enkripsi maupun dekripsi.
5. Mampu menyimpan citra ter-enkripsi.

3. 2 Pemodelan Fungsional

Pemodelan fungsional ini berguna untuk memodelkan fungsi – fungsi yang digunakan dalam perangkat lunak. Dalam implementasinya adalah *Data Flow Diagram* (DFD). Untuk membangun sistem ini *dibreakdown* menjadi 2 (DFD Level 0 sampai DFD Level 1).

3.2.1 DFD Level 0 Aplikasi Perangkat Lunak Secara Umum

Dalam DFD level 0 berguna untuk menggambarkan secara umum perangkat lunak berjalan. Pendefinisian dengan menggunakan DFD level 0 memberikan gambaran data yang mengalir antara sistem dengan *user* yang digambarkan secara global. Rosa & Salahudin, 2011. [10]. Pada proses DFD level 0 diperlihatkan pada gambar 3.1.

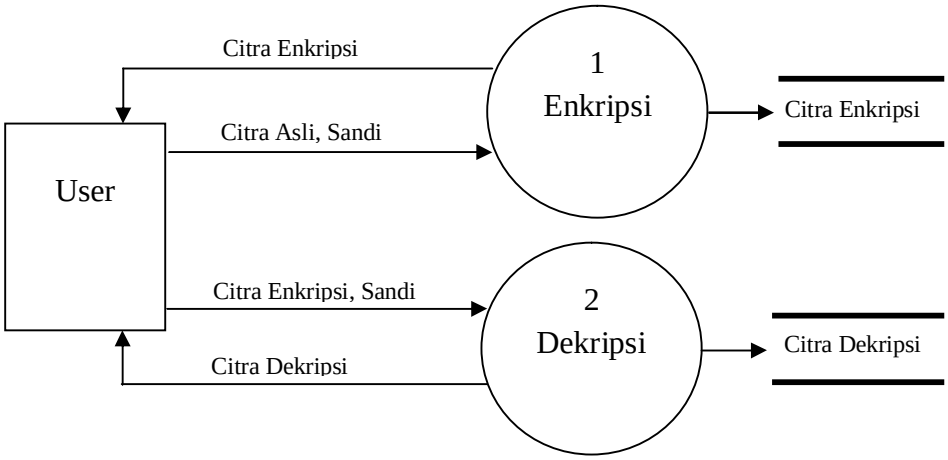


Gambar 3.1 DFD Level 0 Aplikasi Perangkat Lunak Secara Umum

Pada gambar 3.1 menjelaskan bahwa user dapat memberikan *input*-an kepada sistem berupa citra asli, *Kata Sandi*, citra ter-Enkripsi pada proses Enkripsi dan dekripsi.

3.2.2 DFD Level 1 Aplikasi Enkripsi dan Dekripsi

Dari DFD level 0 pada gambar 3.1 dapat *dibreakdown* menjadi DFD level 1 yang merupakan penjelasan lebih rinci dari DFD level 0. DFD Level 1 dapat dilihat pada gambar 3.2.



Gambar 3.2 DFD Level 1 Aplikasi Enkripsi dan Dekripsi

Pada DFD level 1 pada gambar 3.2, dipecah menjadi 2 proses yaitu Aplikasi Enkripsi dan Aplikasi Dekripsi yang berguna untuk menjelaskan fungsi – fungsi dan arus data yang mengalir pada sistem.

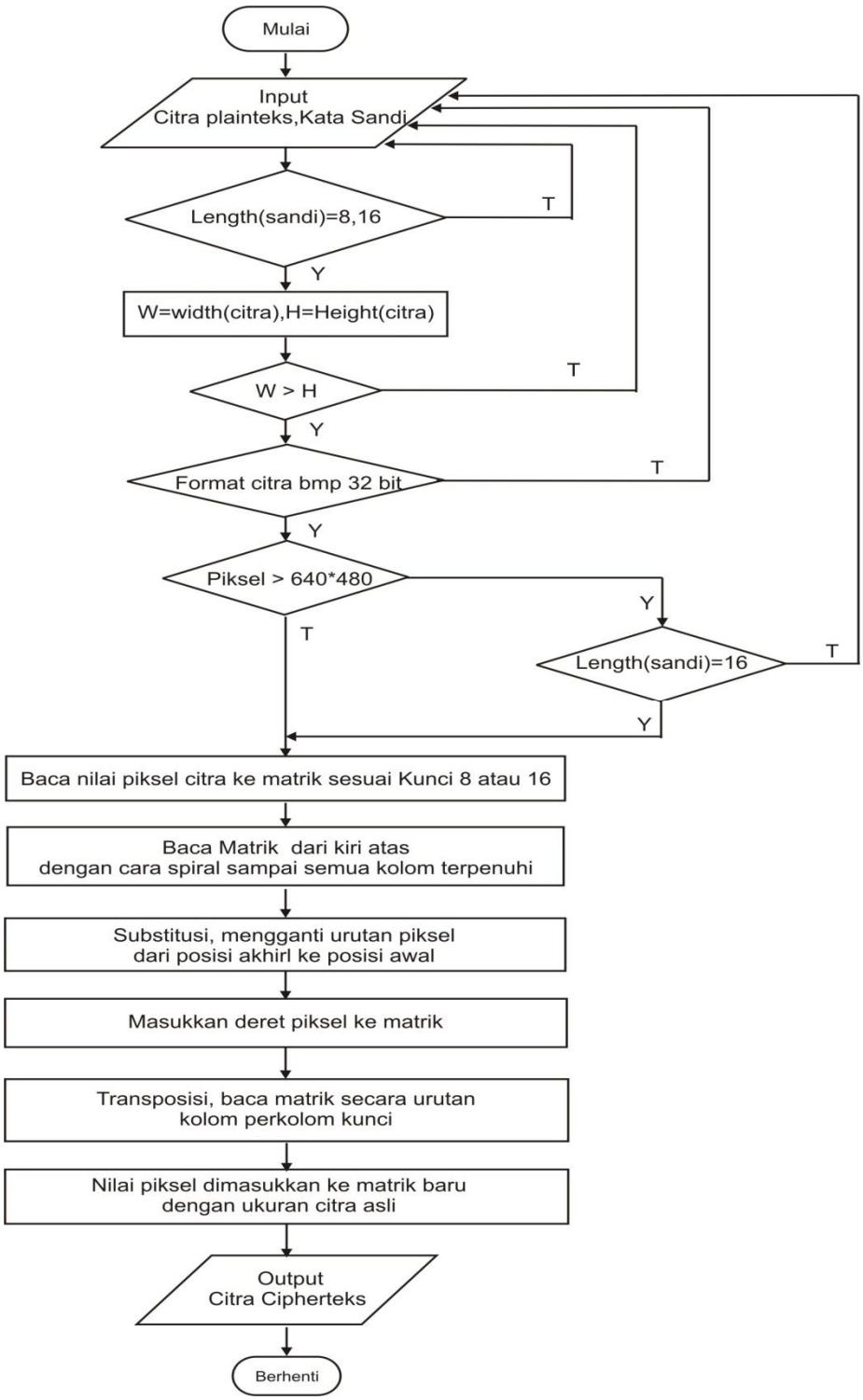
3.3 Perancangan Program

Pada perancangan program ini yang pertama harus diperhatikan ialah algoritma program dan diagram alir atau *Flowchart* adalah bagan yang memperlihatkan urutan prosedur dan proses dari beberapa file di dalam media tertentu. Melalui *flowchart* dapat terlihat jenis media penyimpanan yang dipakai dalam pengolahan data. Tosin, 1994. [9]. Sebagai langkah awal untuk memulai pembuatan program. Proses perancangan program ini merupakan kebutuhan yang direpresentasikan kedalam perangkat lunak sebelum dimulai pembuatan *code* / *coding*. Meliputi diagram alir (*flowchart*) dari proses enkripsi dan diagram alir proses dekripsi .

3.3.1 Perancangan Proses Enkripsi

Proses Enkripsi merupakan proses kriptografi/ penyandian dengan kunci tertentu ke dalam citra asli. Adapun algoritma pada proses enkripsi adalah dengan teknik enkripsi konvensional.

Secara runtun pada langkah – langkah di atas diperlihatkan diagram alir pada gambar 3.3.



Gambar 3.3 Diagram Alir Proses Enkripsi

Pada teknik enkripsi konvensional (simetri) ini, terdapat dua metode dasar yang digunakan, yaitu:

1. Metode substitusi
2. Metode transposisi

3.3.1.1 Metode Substitusi

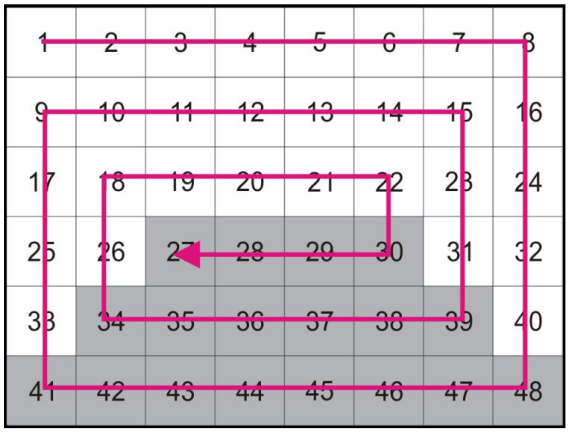
Objek yang akan di substitusikan dalam pembuatan skripsi ini adalah piksel. Adapun langkah – langkahnya dapat diilustrasikan sebagai berikut :

1. Nilai piksel-piksel dari gambar dimasukkan kedalam sebuah matrik dengan ordo yang sama dengan ukuran gambar, dapat dilihat pada gambar 3.4 :

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48

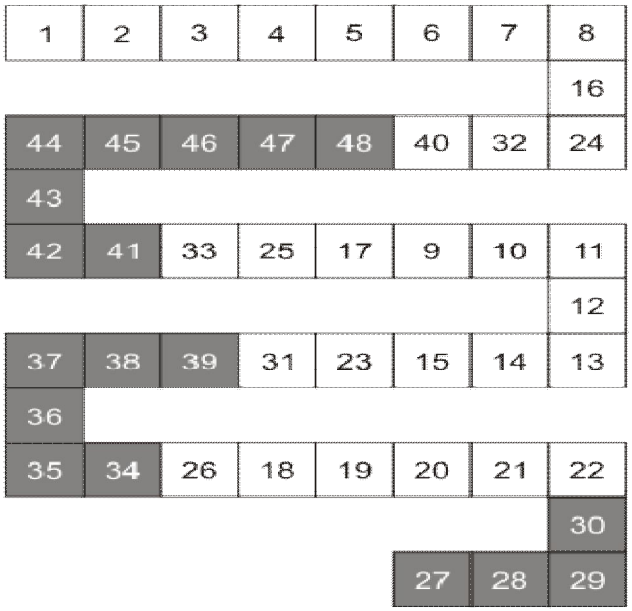
Gambar 3.4 Nilai piksel-piksel dalam Matrik

2. Dari matrik tersebut dapat dilakukan pambacaan piksel gambar dengan aturan sepiral dimulai dari pojok kiri atas. Untuk proses pembacaan secara spiral dapat dilihat pada gambar 3.5 :



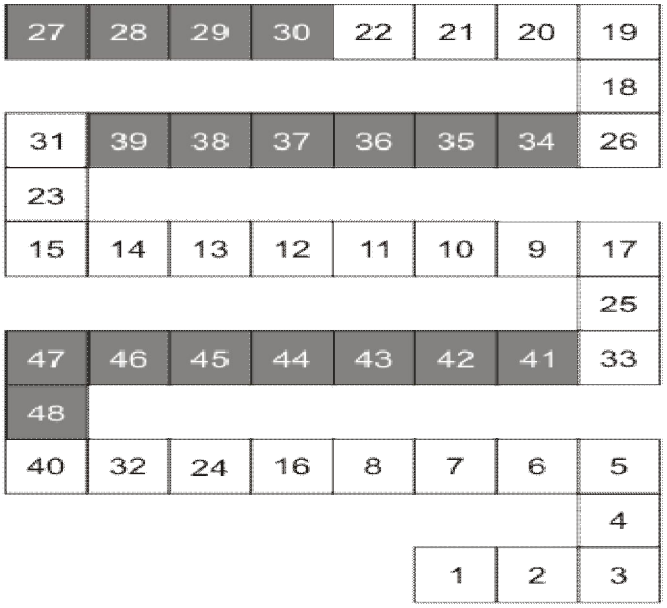
Gambar 3.5 Pembacaan piksel dengan Spiral

3. Hasil pembacaan piksel matrik di atas dibuat menjadi deretan nilai piksel dalam bentuk garis lurus, dapat dilihat pada gambar 3.6 :



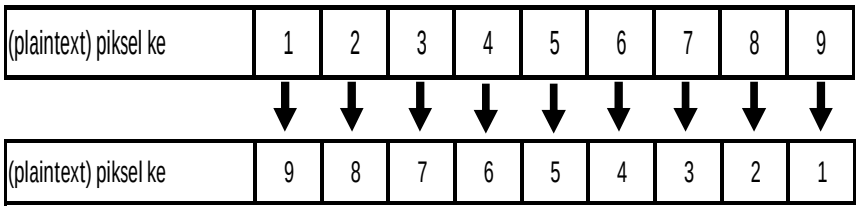
Gambar 3.6 Deretan Nilai Pikes dalam Garis Lurus

4. Dari hasil pembacaan yang terbentuk, lakukan posisi piksel secara urut. Posisi piksel akhir digantikan oleh piksel pertama, sehingga didapat dalam bentuk urutan nilai piksel baru sebagai ciphertext pertama (C1). Lihat pada gambar 3.7 :



Gambar 3.7 Substitusi dengan urutan piksel baru

Jadi hubungan antara piksel ciphertext dan piksel penggantinya merupakan hasil substitusi korespondensi satu-satu. Dapat diterangkan lebih jelas pada gambar 3.8 ;



Gambar 3.8 Hasil substitusi Korespondensi satu-satu

3.3.1.2 Metode Transposisi

Teknik transposisi pada dasarnya adalah membuat ciphertext dengan menggantikan posisi objek – objek plaintext tanpa mengantikan objek plaintext tersebut. Pada metode transposisi ini pembuatan ciphertext dilakukan dengan pembacaan nilai matrik pada kolom perkolom sesuai kunci yang digunakan. Adapun langkah – langkahnya dapat diilustrasikan sebagai berikut :

1. Susunan piksel dari hasil substitusi adalah plaintext keadaan awal dari transposisi, lihat gambar 3.9 :

27	28	29	30	22	21	20	19
18	26	34	35	36	37	38	39
31	23	15	14	13	12	11	10
9	17	25	33	41	42	43	44
45	46	47	48	40	32	24	16
8	7	6	5	4	3	2	1

Gambar 3.9 Susunan piksel Awal / hasil substitusi

2. Untai nilai dari piksel – piksel substitusi dimasukkan kedalam matrik dengan ukuran ordo n (n adalah sesuai dengan panjang kunci yang digunakan) dikali x (jumlah piksel / panjang kunci). Lihat pada gambar 3.10 :

E	N	K	R	I	P	S	I
1	5	4	7	2	6	8	3
27	28	29	30	22	21	20	19
18	26	34	35	36	37	38	39
31	23	15	14	13	12	11	10
9	17	25	33	41	42	43	44
45	46	47	48	40	32	24	16
8	7	6	5	4	3	2	1

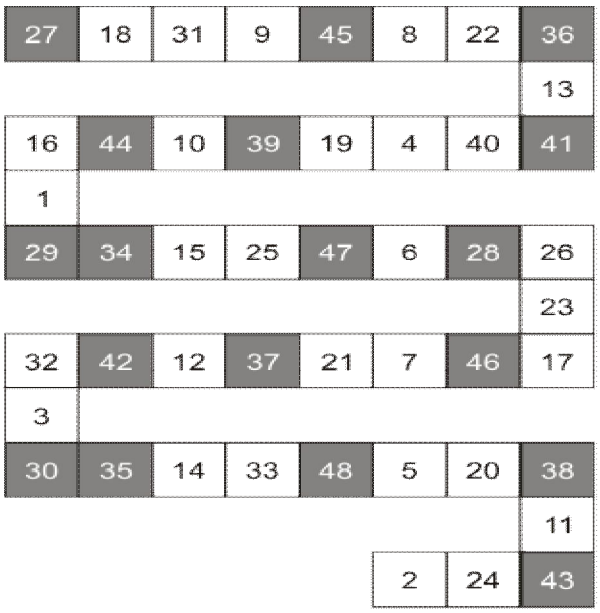
Gambar 3.10 Nilai piksel dalam matrik dengan ukuran ordo N

3. Dari matrik awal, dilakukan pembuatan ciphertext dengan pembacaan matrik transposisi secara kolom per kolom sesuai dengan urutan abjad kunci yang telah diurutkan dan dibuat sebuah untaian nilai piksel yang baru lihat gambar 3.11 :

E	N	K	R	I	P	S	I
1	5	4	7	2	6	8	3
27	28	29	30	22	21	20	19
18	26	34	35	36	37	38	39
31	23	15	14	13	12	11	10
9	17	25	33	41	42	43	44
45	46	47	48	40	32	24	16
8	7	6	5	4	3	2	1

Gambar 3. 11 Pembacaan matrik secara kolom perkolom

4. Pembacaan kolom dimulai dari huruf “e” pada urutan ke 1, huruf “I” yang kedua pada urutan ke 3 dan seterusnya hingga huruf “s” urutan ke 8, maka didapat untaian pada gambar 3.12 :



Gambar 3.12 Pembacaan Kolom

5. Nilai piksel di dalam untaian yang terbentuk dari pembacaan matrik transposisi dimasukkan ke dalam matrik baru dengan ordo sesuai dengan ukuran gambar ukuran semula sebagai hasil enkripsi transposisi (ciphertext kedua (C2)). Lihat pada gambar 3.13 :

27	18	31	9	45	8	22	36
13	41	40	4	19	39	10	44
16	1	29	34	15	25	47	6
28	26	23	17	46	7	21	37
12	42	32	3	30	35	14	33
48	5	20	38	11	43	24	2

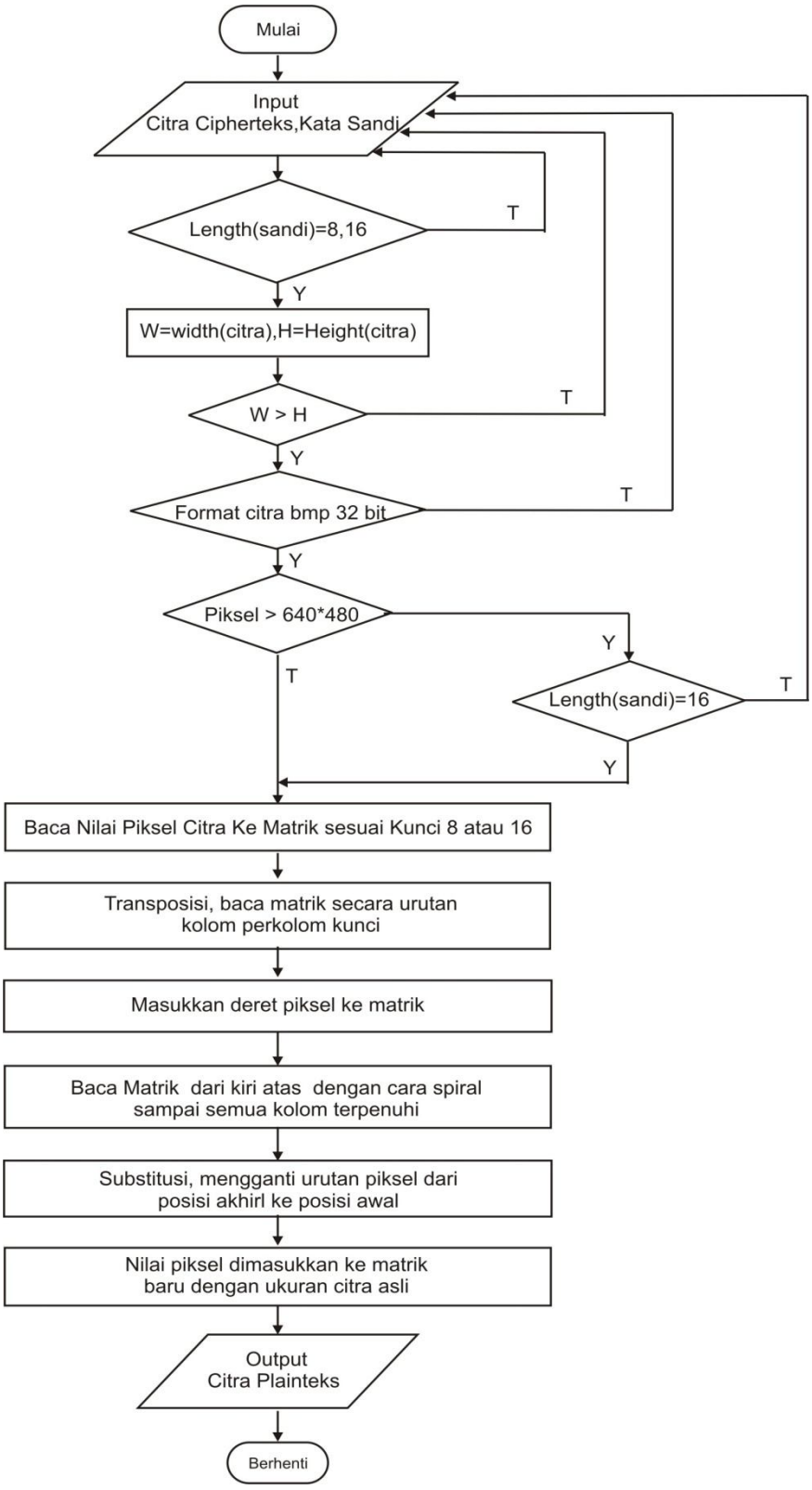
Gambar 3.13 Pembacaan Ulang

Secara spesifik beberapa hal yang dipaparkan dalam analisis sistem pada penyandian file gambar antara lain

1) Penyandian file gambar menerima input gambar dengan ukuran maksimal 2448x1836 pixel.
2) Dalam pemberian kata sandi dengan ketentuan 8 karakter atau 16 karakter, tetapi ketika ukuran piksel gambar lebih dari 640x480 piksel, kata kunci yang harus digunakan adalah 16 karakter. Karena kalau menggunakan kunci 8 gambar akan mudah untuk dibaca.
3) Gambar input selanjutnya dibaca secara spiral dalam kolom matrik dengan dimulai dari pojok kiri atas, kemudian hasil pembacaan tersebut disubstitusi dengan cara mengganti urutan piksel, setelah itu gambar dalam kolom matrik ditransposisikan sesuai kata kunci untuk mendapatkan gambar ciphertext.
4) Setelah sistem mampu menunjukkan langkah-langkah penyelesaian masalah penyandian dalam bentuk gambar plaintext menjadi gambar ciphertext, dengan hasil piksel gambar yang sudah teracak.

3.3.2 Perancangan Proses Dekripsi

Proses Dekripsi adalah pengembalian citra asli di dalam citra yang terenkripsi dengan kunci yang sama waktu proses enkripsi. Adapun algoritma untuk proses dekripsi dilihat pada diagram alir pada gambar 3.14.



Gambar 3.14 Diagram Alir Proses Dekripsi

Pada proses dekripsi ini dapat dipahami secara sederhana sebagai berikut :

1. Penyandian file gambar menerima input gambar *ciphertext* dengan ukuran maksimal 2448x1836 pixel dengan format bitmap 32 bit.
2. Piksel gambar input selanjutnya dibaca secara baris perbaris dimulai dari pojok kiri atas, kemudian hasil pembacaan tersebut ditransposisi dengan cara mengganti urutan piksel sesuai kata kunci, setelah itu hasil transposisi disubstitusikan kemudian piksel disusun secara spiral dari pojok kiri atas sehingga menghasilkan gambar plaintext.
3. Sistem mampu menunjukkan langkah-langkah penyelesaian masalah penyandian dalam bentuk gambar ciphertext menjadi gambar plaintext, dengan hasil piksel gambar yang utuh.

Pada proses pengembalian gambar (dekripsi) dapat diilustrasikan seperti pada proses enkripsi, dapat dilihat pada gambar-gambar sebagai berikut :

1. Nilai piksel yang terbentuk dari proses enkripsi diperoleh suatu gambar yang tidak dapat terbaca, selanjutnya gambar tersebut dimasukkan ke dalam matrik baru dengan ordo sesuai dengan ukuran gambar ukuran semula sebagai hasil enkripsi. Lihat pada gambar 3.15 :

27	18	31	9	45	8	22	36
13	41	40	4	19	39	10	44
16	1	29	34	15	25	47	6
28	26	23	17	46	7	21	37
12	42	32	3	30	35	14	33
48	5	20	38	11	43	24	2

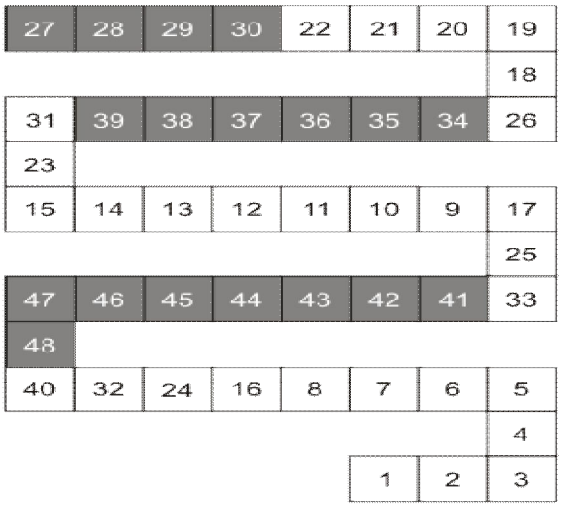
Gambar 3.15 hasil enkripsi

2. Dari pembacaan matrik dilakukan proses pengembalian cipherteks ke plainteks, dengan memasukkan dan membaca matrik secara kolom per kolom sesuai dengan urutan abjad kunci yang telah diurutkan dan dibuat sebuah untaian nilai piksel lihat gambar 3.16 :

E	N	K	R	I	P	S	I
1	5	4	7	2	6	8	3
27	28	29	30	22	21	20	19
18	26	34	35	36	37	38	39
31	23	15	14	13	12	11	10
9	17	25	33	41	42	43	44
45	46	47	48	40	32	24	16
8	7	6	5	4	3	2	1

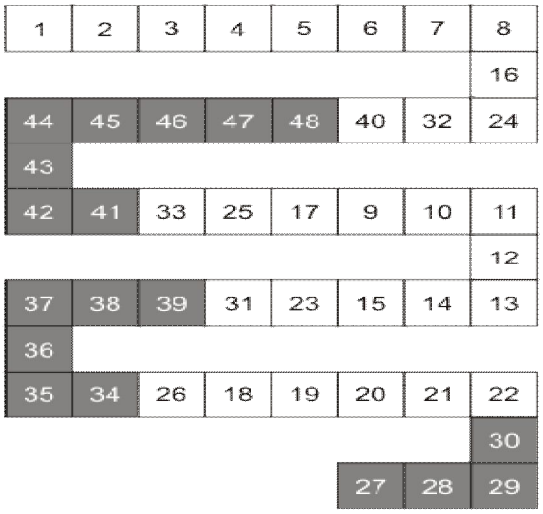
Gambar 3.16 Pembacaan matrik sesuai urutan kunci

3. Dari hasil pembacaan sesuai urutan kunci yang telah ditentukan, selanjutnya dari matrik tersebut dijadikan deret dapat dilihat pada gambar 3.17:



Gambar 3.17 Pembacaan deret

4. Dari hasil pembacaan yang terbentuk, dilakukan pembacaan posisi piksel secara urut. Kemudian Posisi piksel akhir digantikan oleh piksel pertama, sehingga didapat dalam bentuk urutan nilai piksel baru sebagai plainteks pertama dapat dilihat pada gambar 3.18 :



Gambar 3.18 Proses Pembacaan penggantian posisi piksel

5. Dari proses substitusi di atas kemudian dimasukkan matrik dengan pembacaan perbaris sesuai kunci tersebut. Sehingga gambar dapt kembali seperti semula.dapat dilihat pada gambar 3.19 :

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48

Gambar 3.19 Hasil dekripsi (*plaintext*)

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1. Implementasi

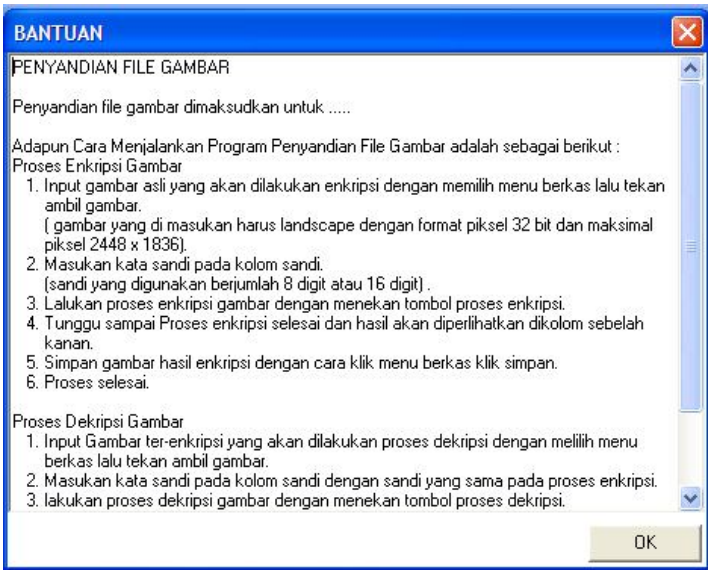
Aplikasi penyandian file gambar pada citra digital dibuat dengan program bantu Borland Delphi 7. Pada aplikasi ini terdapat beberapa fungsi yaitu : fungsi yang dapat membuka citra asli yang dapat dienkripsi, fungsi yang dapat membuka citra hasil enkripsi, fungsi yang dapat melakukan proses enkripsi, fungsi yang dapat melakukan proses dekripsi, serta terdapat petunjuk tentang penggunaan aplikasi penyandian file gambar. Berikut adalah tampilan dari form awal ketika pertama kali program dijalankan seperti pada gambar 4.1.

Gambar 4.1 *Form* Awal

Setelah form awal dijalankan maka secara otomatis akan membuka form Penyandian gambar dimana pada form ini dapat dilakukan proses enkripsi. Tampilan form enkripsi seperi pada gambar 4.3.

Pada tampilan *form* penyandian terdapat tiga menu yaitu menu berkas, penyandian gambar dan menu bantuan. Menu berkas dimaksudkan adalah pengguna dapat melakukan pengambilan gambar, penyimpanan gambar dan

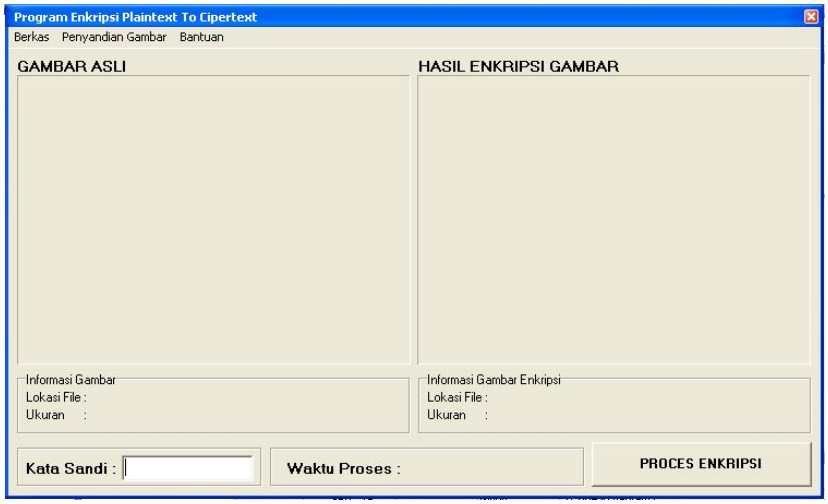
keluar dari sistem. Menu penyandian gambar dimaksudkan seorang pengguna dapat beralih ke proses enkripsi maupun dekripsi. Menu bantuan dimaksudkan agar seorang Pengguna dapat melihat tutorial dalam penggunaan aplikasi enkripsi dan dekripsi. Form bantuan dapat dilihat seperti pada gambar 4.2.



Gambar 4.2 *form* Bantuan

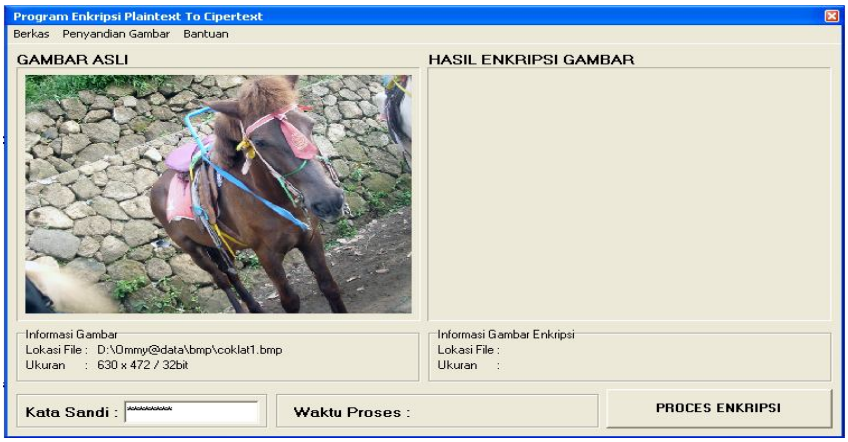
4.1.1. *Form* Proses Enkripsi

Pada proses enkripsi ini ditampilkan *form* enkripsi seperti pada gambar 4.3.



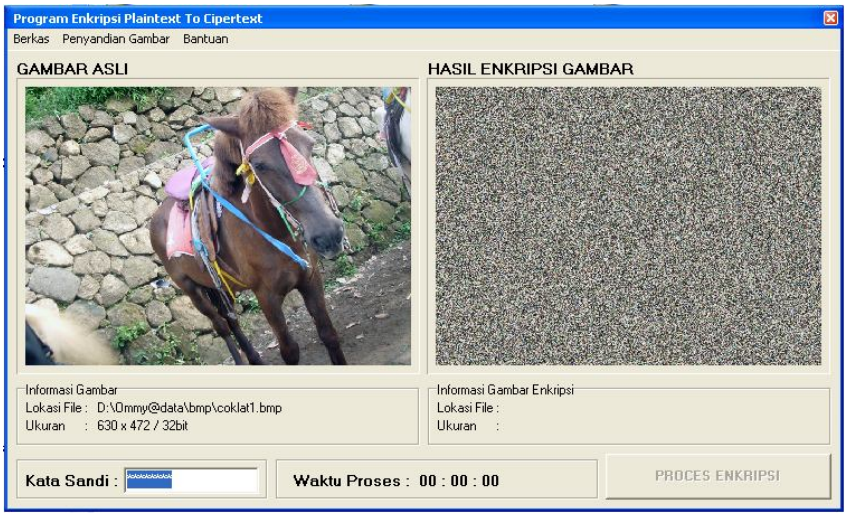
Gambar 4.3 *Form* Proses Enkripsi

Pada *form* enkripsi di atas dilakukan proses enkripsi dengan citra berwarna “Coklat1.Bmp” dengan ukuran 630 x 472 pixel dan penyandian dengan kata sandi “asdfghjk” dengan menekan menu Berkas kemudian klik ambil gambar, kemudian memasukkan kata sandi pada kolom sandi. Berikut adalah tampilan inputan citra asli dan inputan kata sandi pada gambar 4.4.



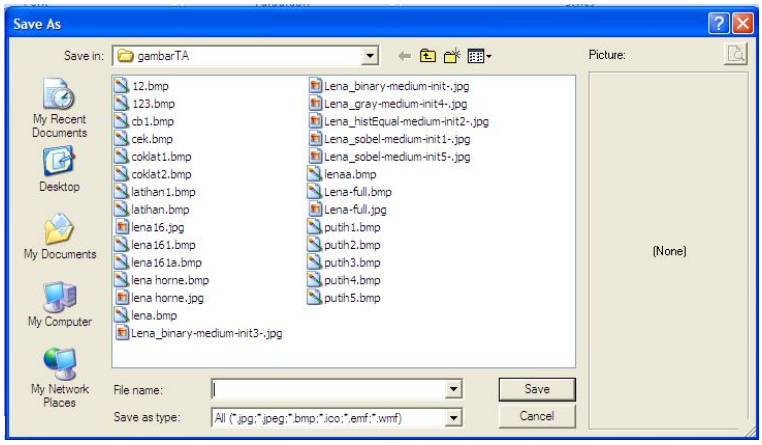
Gambar 4.4 Tampilan Inputan Citra Asli dan Kata Sandi.

Setelah dilakukan inputan citra asli dan Kata Sandi kemudian dilakukan proses enkripsi dengan menekan tombol “PROSES ENKRIPSI” untuk mendapatkan citra ter-Enkripsi. Berikut adalah tampilan hasil akhir proses enkripsi pada gambar 4.5.



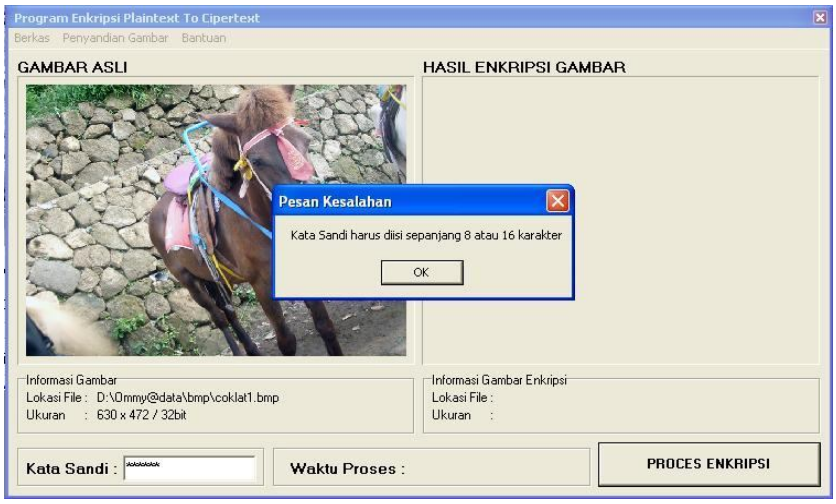
Gambar 4.5 Tampilan Hasil Akhir Proses Enkripsi

Pada proses enkripsi telah didapat citra ter-Enkripsi dan waktu proses enkripsi yaitu 2 detik, dapat disimpulkan bahwa proses enkripsi berhasil. Setelah didapatkan citra yang terenkripsi dilakukan proses simpan untuk menyimpan citra yang terenkripsi. Berikut tampilan untuk proses penyimpanan pada gambar 4.6.



Gambar 4.6 Tampilan Proses Simpan

Dalam pemberian kata sandi dengan ketentuan 8 karakter atau 16 karakter, Oleh karena itu pemberian kata sandi yang bukan 8 karakter atau 16 karakter maka program tidak melakukan proses enkripsi. Untuk tampilan program dapat dilihat pada gambar 4.7



Gambar 4.7 Pesan kesalahan jika sandi selain 8 atau 16 karakter

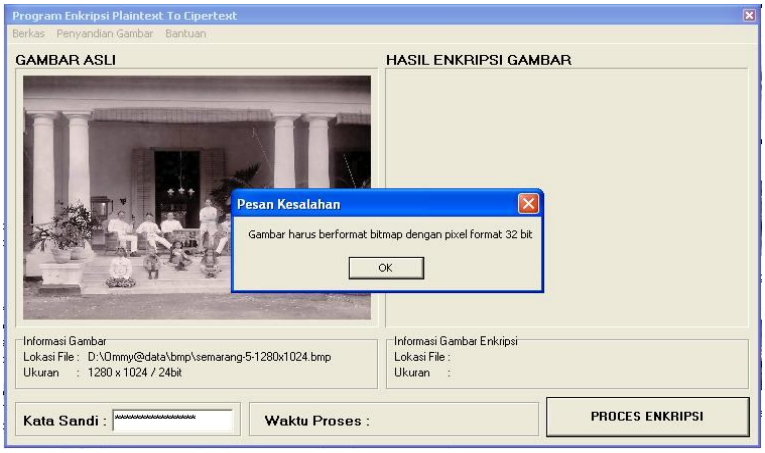
Apabila citra yang digunakan sebagai plainteksnya ukuran pikselnya di atas 640x480 piksel, maka kata sandi yang digunakan harus 16 karakter. Misalkan digunakan citra Ruang.bmp dengan ukuran 900x750 piksel apabila digunakan kata sandi 8 karakter maka program muncul peringatan agar menggunakan 16 karakter. Untuk tampilan programnya dapat dilihat pada gambar 4.8.

Gambar 4.8 Konfirmasi penggunaan sandi 16 karakter

Pada proses enkripsi ini citra yang dapat digunakan ukuran Panjang x lebar dan panjang harus lecih besar dari pada lebar. Apabila digunakan gambar dengan ukuran lebar lebih besar dari pada panjang, maka program tidak dapat memproses. misalkan pada citra merah putih.bmp dengan ukuran 462x800 piksel, maka program akan memberi konfirmasi agar citra berorientasi mendatar. Tampilan program dapat dilihat pada gambar 4.9

Gambar 4.9 Konfirmasi gambar harus panjang > tinggi

Selain itu proses enkripsi juaga harus menggunakan citra berformat 32 bit. Apabila digunakan citra yang bukan 32 bit misalkan semarang3.bmp maka proses enkripsi juga tidak bisa dilakukan. Tampilan peringatan program dapat dilihat pada gambar 4.10



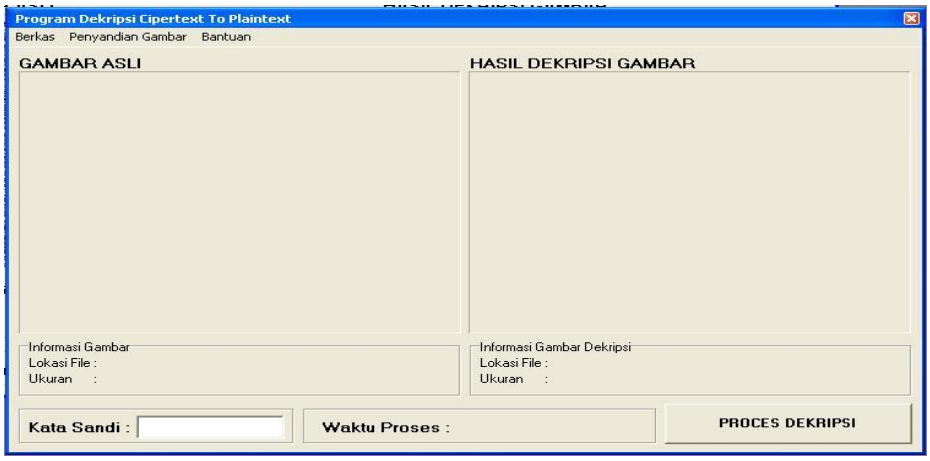
Gambar 4.10 konfirmasi gambar dengan 32 bit

4.1.2. *Form* Proses Dekripsi

Setelah selesai dalam proses Enkripsi dilakukan proses dekripsi. Untuk masuk dalam *form* dekripsi dapat dilihat dari *form* enkripsi pada menu penyandian gambar dan selanjutnya pilih proses dekripsi, untuk lebih jelasnya dapat dilihat pada gambar 4.11.

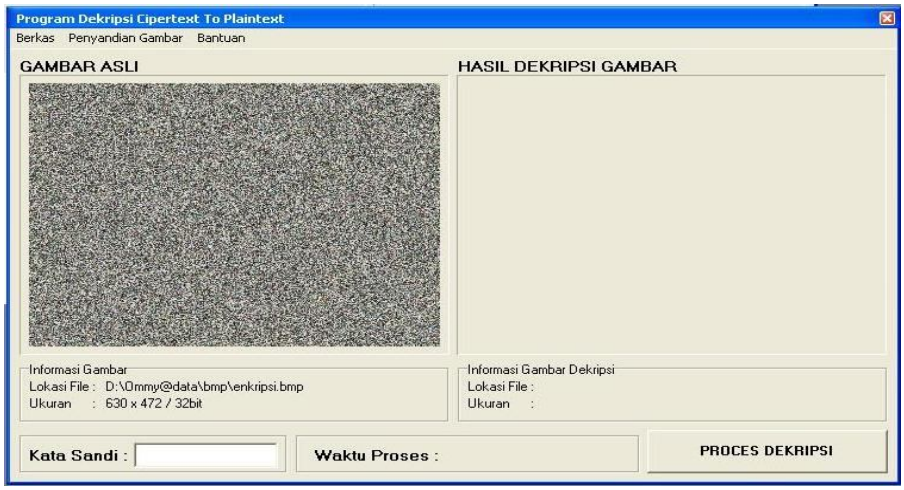
Gambar 4.11 Tampilan Untuk Masuk Proses Dekripsi

Setelah dilakukan pilihan dekripsi akan tampil *form* porses dekripsi seperti pada gambar 4.12.



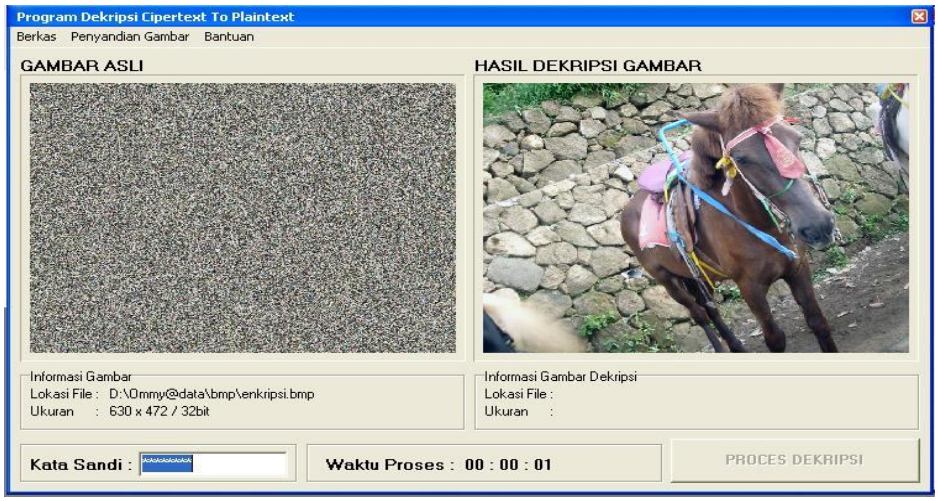
Gambar 4.12 *Form* Proses Dekripsi

Pada *form* Proses dekripsi di atas dilakukan proses dekripsi terhadap citra ter-Enkripsi pada citra coklat1 yang telah disimpan dengan nama “Coklat1ter-Enkripsi.bmp”. Dengan mengitputkan citra pada menu file Berkas selanjutnya dilakukan input citra pada sub menu ‘Ambil Gambar’. Setelah diinputkan citra yang terenkripsi dilakukan pemasukan kata sandi yang sebelumnya digunakan dalam proses enkripsi yaitu “asdfghjk”. Untuk inputan citra terenkripsi dan kata sandi dapat dilihat pada gambar 4.13.



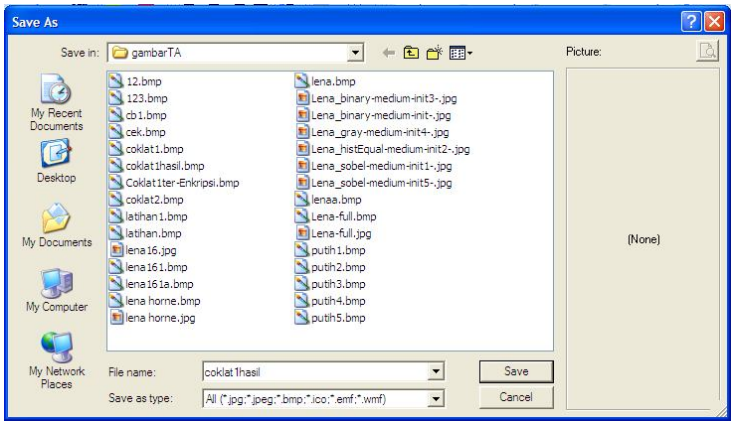
Gambar 4.13 Inputan citra ter-Enkripsi dengan kata sandi

Kemudian setelah dilakukan inputan citra ter-Enkripsi dan kata sandi dilakukan proses dekripsi dengan menekan tombol “PROSES DEKRIPSI” Berikut adalah tampilan hasil akhir proses dekripsi pada gambar 4.14



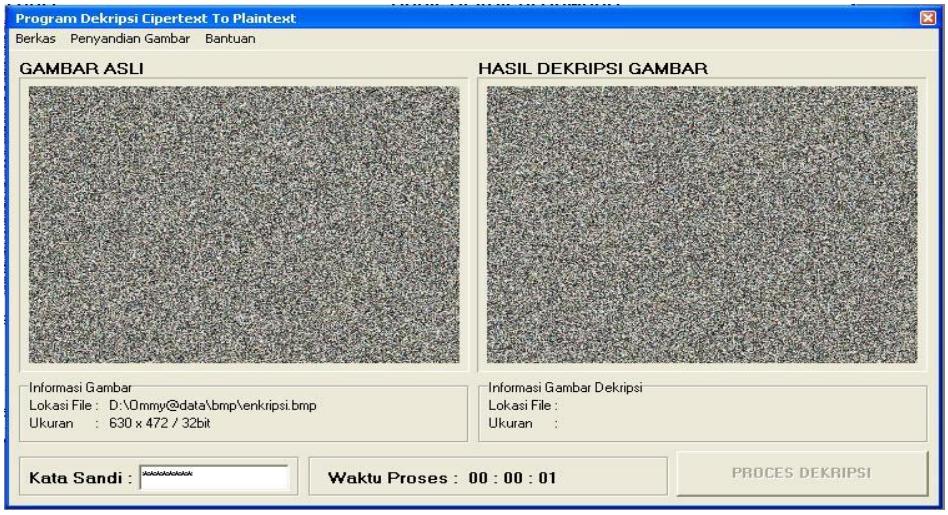
Gambar 4.14 Hasil proses dekripsi

Pada gambar 4.14 didapat citra hasil dekripsi, dari citra hasil dekripsi terbukti bahwa proses dekripsi berhasil dilakukan karena citra hasil dekripsi hampir sama dengan citra asli. Selain itu juga dapat diketahui waktu proses dekripsi yaitu 2 detik. Setelah didapatkan citra hasil dekripsi dilakukan proses simpan untuk menyimpan citra yang ter-enkripsi. Berikut tampilan untuk proses penyimpanan pada gambar 4.15.



Gambar 4.15 Proses simpan citra hasil dekripsi

Apabila dalam memasukkan kata sandi tidak sama dengan kata sandi pada waktu proses enkripsi maka gambar hasil dari proses dekripsi tidak akan berhasil. Hasil dari proses dekripsi dengan kata sandi yang berbeda dapat dilihat pada gambar 4.16



Gambar 4.16 Hasil proses dekripsi dengan sandi yang salah

4.2. Implementasi Program

Pada sub bab ini dimaksudkan untuk mengimplementasikan dari rancangan dan analisa yang telah dibuat pada bab sebelumnya.

4.2.1 Prosedur Proses Enkripsi Terhadap Citra Asli

Pada proses enkripsi terhadap citra asli pertama memasukan dulu citra yang di gunakan. Citra yang dipilih adalah citra bitmap dengan 32 bit, kemudian memasukan kata sandi 8 atau 16 karakter. Dalam pemberian kata sandi dengan ketentuan 8 karakter atau 16 karakter, ini dikarenakan dalam penggunaan format citra bitmap 32 bit, yang berarti setiap piksel panjangnya 32 bit. Sehingga untuk pemakaian kata sandi harus merupakan kelipatan dari 4, bila tidak maka pada saat penyimpanan akan ditambahkan beberapa bit pada gambar hingga merupakan

kelipatan dari 4. Oleh karena itu apabila terjadi penambahan dari bit maka gambar gambar tidak dapat kembali seperti semula. Selain itu penggunaan format gambar 32 bit dikarenakan komputer sekarang sudah banyak yang memakai memori di atas 32 bit. Setelah inputan citra asli dan kata sandi dilakukanlah proses enkripsi dengan perintah sebagai berikut:

Deklarasi Program :

```
if  keylength=8 then
  begin
    //waktu awal
    a:=time;
    //baca pixel plaintext dan dimasukkan ke array deret1
    maxderet:=1;
    for g:=0 to tg-1 do
    begin
      for h:=0 to lb-1 do
      begin
deret1[maxderet]:=Image1.Picture.Bitmap.Canvas.Pixels[h,g];
        inc(maxderet);
      end;
    end;
    //memasuk deret1 kedalam matrik1 dimulai dari kiri baris atas
    maxderet:=1;
    for g:=1 to tg do
    begin
      for h:=1 to lb do
      begin
        matrik1[h,g]:=deret1[maxderet];
        inc(maxderet);
      end;
    end;
    //membaca matrik secara spiral
    kolomawal:=1;
    kolomakhir:=lb;
    barisawal:=1;
    barisakhir:=tg;
    maxderet:=1;
    while maxderet <=(lb*tg) do
    begin
      // baca kekanan
      for g:=kolomawal to kolomakhir do
      begin
        deret2[maxderet]:=matrik1[g,barisawal];
        inc(maxderet);
      end;
      //baca ke bawah
      barisawal:=barisawal+1;
      for g:=barisawal to barisakhir do
      begin
        deret2[maxderet]:=matrik1[kolomakhir,g];
        inc(maxderet);
      end;
```

```
  //baca ke kiri
  kolomakhir:=kolomakhir-1;
  for g:=kolomakhir downto kolomawal do
  begin
    deret2[maxderet]:=matrik1[g,barisakhir];
    inc(maxderet);
  end;
  //baca ke atas
  barisakhir:=barisakhir-1;
  for g:=barisakhir downto barisawal do
  begin
    deret2[maxderet]:=matrik1[kolomawal,g];
    inc(maxderet);
  end;
  kolomawal:=kolomawal+1;
end;
//Proses Subtitusi atau pembalikan array deret2
g:=maxderet;
h:=maxderet;
for maxderet:=1 to g do
begin
  deret3[maxderet]:=deret2[h-1];
  dec(h);
end;
//memasukkan hasil subtitusi kematrik2 perbaris
tgmatrik1:=((tg*lb)div 8);
maxderet:=1;
for g:=1 to tgmatrik1 do
begin
  for h:=1 to 8 do
  begin
    matrik2[h,g]:=deret3[maxderet];
    inc(maxderet);
  end;
end;
//baca matrik sesuai urutan kolom kunci1
maxderet:=1;
urutankolom:= kunci1;
for i:=ord('1') to ord('8') do
begin
  for g:=1 to length(urutankolom) do
  begin
    if i=ord(urutankolom[g]) then
    begin
      for h:=1 to tgmatrik1 do
      begin
        deret4[maxderet]:=matrik2[g,h];
        inc(maxderet);
      end;
    end;
  end;
end;
//Proses Subtitusi atau pembalikan array deret4
g:=maxderet;
h:=maxderet;
for maxderet:=1 to g do
begin
  deret5[maxderet]:=deret4[h-1];
  dec(h);
```

```
 end;
 //memasukkan hasil subtitusi ke dalam matrik3
 maxderet:=1;
 for g:=1 to tg do
 begin
   for h:=1 to lb do
   begin
     matrik3[h,g]:=deret5[maxderet];
     inc(maxderet);
   end;
 end;
//membaca matrik secara spiral
 kolomawal:=1;
 kolomakhir:=lb;
 barisawal:=1;
 barisakhir:=tg;
 maxderet:=1;
 while maxderet <=(lb*tg) do
 begin
   // baca kekanan
   for g:=kolomawal to kolomakhir do
   begin
     deret6[maxderet]:=matrik3[g,barisawal];
     inc(maxderet);
   end;
   //baca ke bawah
   barisawal:=barisawal+1;
   for g:=barisawal to barisakhir do
   begin
     deret6[maxderet]:=matrik3[kolomakhir,g];
     inc(maxderet);
   end;
   //baca ke kiri
   kolomakhir:=kolomakhir-1;
   for g:=kolomakhir downto kolomawal do
   begin
     deret6[maxderet]:=matrik3[g,barisakhir];
     inc(maxderet);
   end;
   //baca ke atas
   barisakhir:=barisakhir-1;
   for g:=barisakhir downto barisawal do
   begin
     deret6[maxderet]:=matrik3[kolomawal,g];
     inc(maxderet);
   end;
   kolomawal:=kolomawal+1;
 end;
 //memasukkan hasil sepiral kematrik4 perbaris
 tgmatrik1:=((tg*lb)div 8);
 maxderet:=1;
 for g:=1 to tgmatrik1 do
 begin
   for h:=1 to 8 do
   begin
     matrik4[h,g]:=deret6[maxderet];
     inc(maxderet);
   end;
 end;
```

```
//baca matrik sesuai urutan kolom kunci2
maxderet:=1;
urutankolom:= kunci2;
for i:=ord('1') to ord('8') do
begin
  for g:=1 to length(urutankolom) do
  begin
    if i=ord(urutankolom[g]) then
    begin
      for h:=1 to tgmatrik1 do
      begin
        deret7[maxderet]:=matrik4[g,h];
        inc(maxderet);
      end;
    end;
  end;
end;
//menempatkan deret7 ke matrik1 secara spiral
maxderet:=1;
kolomawal:=1;
kolomakhir:=lb;
barisawal:=1;
barisakhir:=tg;
while maxderet <= (tg*lb) do
begin
  //tulis ke kanan
  for g:=kolomawal to kolomakhir do
  begin
    matrik1[g,barisawal]:=deret7[maxderet];
    inc(maxderet);
  end;
  barisawal:=barisawal+1;
  //tulis ke bawah
  for g:=barisawal to barisakhir do
  begin
    matrik1[kolomakhir,g]:=deret7[maxderet];
    inc(maxderet);
  end;
  kolomakhir:=kolomakhir-1;
  //tulis ke kiri
  for g:=kolomakhir downto kolomawal do
  begin
    matrik1[g,barisakhir]:=deret7[maxderet];
    inc(maxderet);
  end;
  barisakhir:=barisakhir-1;
  //tulis ke atas
  for g:=barisakhir downto barisawal do
  begin
    matrik1[kolomawal,g]:=deret7[maxderet];
    inc(maxderet);
  end;
  kolomawal:=kolomawal+1;
end;
//membaca matrik1 dimasukkan ke deret1 perbaris
maxderet:=1;
for g:=1 to tg do
begin
  for h:=1 to lb do
```

```
  begin
    deret1[maxderet]:=matrik1[h,g];
    inc(maxderet);
  end;
end;
//memasukkan deret1 kedalam kematrik2 perbaris
tgmatrik1:=((tg*lb)div 8);
maxderet:=1;
for g:=1 to tgmatrik1 do
begin
  for h:=1 to 8 do
  begin
    matrik2[h,g]:=deret1[maxderet];
    inc(maxderet);
  end;
end;
//baca matrik sesuai urutan kolom kunci3
maxderet:=1;
urutankolom:= kunci3;
for i:=ord('1') to ord('8') do
begin
  for g:=1 to length(urutankolom) do
  begin
    if i=ord(urutankolom[g]) then
    begin
      for h:=1 to tgmatrik1 do
      begin
        deret2[maxderet]:=matrik2[g,h];
        inc(maxderet);
      end;
    end;
  end;
end;
//Proses Subtitusi atau pembalikan array deret2
g:=maxderet;
h:=maxderet;
for maxderet:=1 to g do
begin
  deret3[maxderet]:=deret2[h-1];
  dec(h);
end;
//memasukkan hasil subtitusi ke dalam matrik3
maxderet:=1;
for g:=1 to tg do
begin
  for h:=1 to lb do
  begin
    matrik3[h,g]:=deret3[maxderet];
    inc(maxderet);
  end;
end;
//membaca matrik secara spiral
kolomawal:=1;
kolomakhir:=lb;
barisawal:=1;
barisakhir:=tg;
maxderet:=1;
while maxderet <=(lb*tg) do
begin
```

```
  // baca kekanan
  for g:=kolomawal to kolomakhir do
  begin
    deret4[maxderet]:=matrik3[g,barisawal];
    inc(maxderet);
  end;
  //baca ke bawah
  barisawal:=barisawal+1;
  for g:=barisawal to barisakhir do
  begin
    deret4[maxderet]:=matrik3[kolomakhir,g];
    inc(maxderet);
  end;
  //baca ke kiri
  kolomakhir:=kolomakhir-1;
  for g:=kolomakhir downto kolomawal do
  begin
    deret4[maxderet]:=matrik3[g,barisakhir];
    inc(maxderet);
  end;
  //baca ke atas
  barisakhir:=barisakhir-1;
  for g:=barisakhir downto barisawal do
  begin
    deret4[maxderet]:=matrik3[kolomawal,g];
    inc(maxderet);
  end;
  kolomawal:=kolomawal+1;
end;
//memasukkan hasil sepiral kematrik4 perbaris
tgmatrik1:=((tg*lb)div 8);
maxderet:=1;
for g:=1 to tgmatrik1 do
begin
  for h:=1 to 8 do
  begin
    matrik4[h,g]:=deret4[maxderet];
    inc(maxderet);
  end;
end;
//baca matrik sesuai urutan kolom kunci4
maxderet:=1;
urutankolom:= kunci4;
for i:=ord('1') to ord('8') do
begin
  for g:=1 to length(urutankolom) do
  begin
    if i=ord(urutankolom[g]) then
    begin
      for h:=1 to tgmatrik1 do
      begin
        deret5[maxderet]:=matrik4[g,h];
        inc(maxderet);
      end;
    end;
  end;
end;
//memasukkan hasil urutan kunci4 ke dalam matrik1 perbaris
maxderet:=1;
```

```
for g:=1 to tg do
begin
  for h:=1 to lb do
  begin
    matrik1[h,g]:=deret5[maxderet];
    inc(maxderet);
  end;
end;
//baca matrik perkolom
maxderet:=1;
for g:=1 to lb do
begin
  for h:=1 to tg do
  begin
    deret6[maxderet]:=matrik1[g,h];
    inc(maxderet);
  end;
end;

//memasukkan deret6 kematrik2 perbaris
tgmatrik1:=((tg*lb)div 8);
maxderet:=1;
for g:=1 to tgmatrik1 do
begin
  for h:=1 to 8 do
  begin
    matrik2[h,g]:=deret6[maxderet];
    inc(maxderet);
  end;
end;
//baca matrik sesuai urutan kolom kunci5
maxderet:=1;
urutankolom:= kunci5;
for i:=ord('1') to ord('8') do
begin
  for g:=1 to length(urutankolom) do
  begin
    if i=ord(urutankolom[g]) then
    begin
      for h:=1 to tgmatrik1 do
      begin
        deret7[maxderet]:=matrik2[g,h];
        inc(maxderet);
      end;
    end;
  end;
end;
// menggambar ke image2
maxderet:=1;
for g:=0 to tg-1 do
begin
  for h:=0 to lb-1 do
  begin
    form2.Image3.Canvas.Pixels[h,g]:=deret7[maxderet];
    inc(maxderet);
  end;
end;
image2.Picture:=form2.Image3.Picture;
```

Pada proses enkripsi ini Penyandian file gambar menerima input gambar dengan ukuran maksimal 3264 x 2448 pixel. Gambar masukkan selanjutnya dibaca secara sepiral dalam kolom matrik dengan dimulai dari pojok kiri atas, kemudian hasil pembacaan tersebut disubstitusi dengan cara mengganti urutan piksel, setelah itu gambar dalam kolom matrik di transposisikan sesuai kata kunci untuk mendapatkan gambar ciphertext. Setelah sistem mampu menunjukkan langkah-langkah penyelesaian masalah penyandian dalam bentuk gambar plaintext menjadi gambar ciphertext, dengan hasil piksel gambar yang sudah teracak.

4.2.2 Prosedur Proses Dekripsi Terhadap Gambar Ter-Enkripsi

Pada prosedur dekripsi terhadap citra yang sudah terenkripsi dimaksudkan untuk mendapat citra yang sesuai dengan citra asli. Untuk mendapatkan citra yang sesuai dengan citra asli yaitu dengan memasukkan citra yang telah ter-enkripsi yang selanjutnya dilakukan transformasi proses dekripsi berdasarkan kata sandi yang sesuai dengan kata sandi waktu proses enkripsi. Prosedur yang digunakan seperti deklarasi program sebagai berikut :

Deklarasi Program:

```
if  keylength=8 then
  begin
    //waktu awal
    a:=time;
    //membaca pixel gambar dari image1 ke dalam deret1
    maxderet:=1;
    for g:=0 to tg-1 do
    begin
      for h:=0 to lb-1 do
      begin
deret1[maxderet]:=Image1.Picture.Bitmap.Canvas.Pixels[h,g];
        inc(maxderet);
      end;
    end;
    //mencari tinggi matrik2
    tgmatrik1:=((tg*lb) div 8);
    //memasukkan nilai ke matrik urut kolom sesuai kunci5
```

```
 maxderet:=1;
 urutankolom:=kunci5;
 for i:=ord('1') to ord('8') do
 begin
   for g:=1 to length(urutankolom) do
   begin
     if i = ord(urutankolom[g]) then
     begin
       for h:=1 to tgmatrik1 do
       begin
         matrik2[g,h]:=deret1[maxderet];
         inc(maxderet);
       end;
     end;
   end;
 end;
 //membaca matrik per baris
 maxderet:=1;
 for g:=1 to tgmatrik1 do
 begin
   for h:=1 to 8 do
   begin
     deret2[maxderet]:=matrik2[h,g];
     inc(maxderet);
   end;
 end;
//memasuk deret2 kedalam matrik1 perkolom
 maxderet:=1;
 for g:=1 to lb do
 begin
   for h:=1 to tg do
   begin
     matrik1[g,h]:=deret2[maxderet];
     inc(maxderet);
   end;
 end;
 //baca matrik perbaris
 maxderet:=1;
 for g:=1 to tg do
 begin
   for h:=1 to lb do
   begin
     deret3[maxderet]:=matrik1[h,g];
     inc(maxderet);
   end;
 end;
//mencari tinggi matrik2
 tgmatrik1:=((tg*lb) div 8);
 //memasukkan nilai ke matrik urut kolom sesuai kunci4
 maxderet:=1;
 urutankolom:=kunci4;
 for i:=ord('1') to ord('8') do
 begin
   for g:=1 to length(urutankolom) do
   begin
     if i = ord(urutankolom[g]) then
     begin
       for h:=1 to tgmatrik1 do
       begin
```

```
        matrik4[g,h]:=deret3[maxderet];
        inc(maxderet);
      end;
    end;
  end;
end;
//membaca matrik per baris
maxderet:=1;
for g:=1 to tgmatrik1 do
begin
  for h:=1 to 8 do
  begin
    deret4[maxderet]:=matrik4[h,g];
    inc(maxderet);
  end;
end;
//memasukkan deret kedalam matrik secara spiral
maxderet:=1;
kolomawal:=1;
kolomakhir:=lb;
barisawal:=1;
barisakhir:=tg;
while maxderet <= (tg*lb) do
begin
  //tulis ke kanan
  for g:=kolomawal to kolomakhir do
  begin
    matrik3[g,barisawal]:=deret4[maxderet];
    inc(maxderet);
  end;
  barisawal:=barisawal+1;
  //tulis ke bawah
  for g:=barisawal to barisakhir do
  begin
    matrik3[kolomakhir,g]:=deret4[maxderet];
    inc(maxderet);
  end;
  kolomakhir:=kolomakhir-1;
  //tulis ke kiri
  for g:=kolomakhir downto kolomawal do
  begin
    matrik3[g,barisakhir]:=deret4[maxderet];
    inc(maxderet);
  end;
  barisakhir:=barisakhir-1;
  //tulis ke atas
  for g:=barisakhir downto barisawal do
  begin
    matrik3[kolomawal,g]:=deret4[maxderet];
    inc(maxderet);
  end;
  kolomawal:=kolomawal+1;
end;
//baca matrik perbaris
maxderet:=1;
for g:=1 to tg do
begin
  for h:=1 to lb do
  begin
```

```
    deret5[maxderet]:=matrik3[h,g];
    inc(maxderet);
  end;
end;
//proses substitusi pixel
g:=maxderet;
h:=maxderet;
for maxderet:=1 to g do
begin
  deret6[maxderet]:=deret5[h-1];
  dec(h);
end;
//mencari tinggi matrik2
tgmatrik1:=((tg*lb) div 8);
//memasukkan nilai ke matrik urut kolom sesuai kunci3
maxderet:=1;
urutankolom:=kunci3;
for i:=ord('1') to ord('8') do
begin
  for g:=1 to length(urutankolom) do
  begin
    if i = ord(urutankolom[g]) then
    begin
      for h:=1 to tgmatrik1 do
      begin
        matrik2[g,h]:=deret6[maxderet];
        inc(maxderet);
      end;
    end;
  end;
end;
//membaca matrik per baris
maxderet:=1;
for g:=1 to tgmatrik1 do
begin
  for h:=1 to 8 do
  begin
    deret7[maxderet]:=matrik2[h,g];
    inc(maxderet);
  end;
end;
//memasuk deret1 kedalam matrik3 dimulai dari kiri baris atas
maxderet:=1;
for g:=1 to tg do
begin
  for h:=1 to lb do
  begin
    matrik1[h,g]:=deret7[maxderet];
    inc(maxderet);
  end;
end;
//membaca matrik secara spiral
kolomawal:=1;
kolomakhir:=lb;
barisawal:=1;
barisakhir:=tg;
maxderet:=1;
while maxderet <=(lb*tg) do
begin
```

```
  // baca kekanan
  for g:=kolomawal to kolomakhir do
  begin
    deret1[maxderet]:=matrik1[g,barisawal];
    inc(maxderet);
  end;
  //baca ke bawah
  barisawal:=barisawal+1;
  for g:=barisawal to barisakhir do
  begin
    deret1[maxderet]:=matrik1[kolomakhir,g];
    inc(maxderet);
  end;
  //baca ke kiri
  kolomakhir:=kolomakhir-1;
  for g:=kolomakhir downto kolomawal do
  begin
    deret1[maxderet]:=matrik1[g,barisakhir];
    inc(maxderet);
  end;
  //baca ke atas
  barisakhir:=barisakhir-1;
  for g:=barisakhir downto barisawal do
  begin
    deret1[maxderet]:=matrik1[kolomawal,g];
    inc(maxderet);
  end;
  kolomawal:=kolomawal+1;
end;
//mencari tinggi matrik4
tgmatrik1:=((tg*lb) div 8);
//memasukkan nilai ke matrik urut kolom sesuai kunci2
maxderet:=1;
urutankolom:=kunci2;
for i:=ord('1') to ord('8') do
begin
  for g:=1 to length(urutankolom) do
  begin
    if i = ord(urutankolom[g]) then
    begin
      for h:=1 to tgmatrik1 do
      begin
        matrik4[g,h]:=deret1[maxderet];
        inc(maxderet);
      end;
    end;
  end;
end;
//membaca matrik per baris
maxderet:=1;
for g:=1 to tgmatrik1 do
begin
  for h:=1 to 8 do
  begin
    deret2[maxderet]:=matrik4[h,g];
    inc(maxderet);
  end;
end;
//memasukkan deret kedalam matrik secara spiral
```

```
maxderet:=1;
kolomawal:=1;
kolomakhir:=lb;
barisawal:=1;
barisakhir:=tg;
while maxderet <= (tg*lb) do
begin
  //tulis ke kanan
  for g:=kolomawal to kolomakhir do
  begin
    matrik3[g,barisawal]:=deret2[maxderet];
    inc(maxderet);
  end;
  barisawal:=barisawal+1;
  //tulis ke bawah
  for g:=barisawal to barisakhir do
  begin
    matrik3[kolomakhir,g]:=deret2[maxderet];
    inc(maxderet);
  end;
  kolomakhir:=kolomakhir-1;
  //tulis ke kiri
  for g:=kolomakhir downto kolomawal do
  begin
    matrik3[g,barisakhir]:=deret2[maxderet];
    inc(maxderet);
  end;
  barisakhir:=barisakhir-1;
  //tulis ke atas
  for g:=barisakhir downto barisawal do
  begin
    matrik3[kolomawal,g]:=deret2[maxderet];
    inc(maxderet);
  end;
  kolomawal:=kolomawal+1;
end;
//baca matrik perbaris
maxderet:=1;
for g:=1 to tg do
begin
  for h:=1 to lb do
  begin
    deret3[maxderet]:=matrik3[h,g];
    inc(maxderet);
  end;
end;
//proses substitusi pixel
g:=maxderet;
h:=maxderet;
for maxderet:=1 to g do
begin
  deret4[maxderet]:=deret3[h-1];
  dec(h);
end;
//mencari tinggi matrik2
tgmatrik1:=((tg*lb) div 8);
//memasukkan nilai ke matrik urut kolom sesuai kunci1
maxderet:=1;
urutankolom:=kunci1;
```

```
for i:=ord('1') to ord('8') do
begin
  for g:=1 to length(urutankolom) do
  begin
    if i = ord(urutankolom[g]) then
    begin
      for h:=1 to tgmatrik1 do
      begin
        matrik2[g,h]:=deret4[maxderet];
        inc(maxderet);
      end;
    end;
  end;
end;
//membaca matrik per baris
maxderet:=1;
for g:=1 to tgmatrik1 do
begin
  for h:=1 to 8 do
  begin
    deret5[maxderet]:=matrik2[h,g];
    inc(maxderet);
  end;
end;
//proses substitusi pixel
g:=maxderet;
h:=maxderet;
for maxderet:=1 to g do
begin
  deret6[maxderet]:=deret5[h-1];
  dec(h);
end;
//memasukkan deret kedalam matrik secara spiral
maxderet:=1;
kolomawal:=1;
kolomakhir:=lb;
barisawal:=1;
barisakhir:=tg;
while maxderet <= (tg*lb) do
begin
  //tulis ke kanan
  for g:=kolomawal to kolomakhir do
  begin
    matrik1[g,barisawal]:=deret6[maxderet];
    inc(maxderet);
  end;
  barisawal:=barisawal+1;
  //tulis ke bawah
  for g:=barisawal to barisakhir do
  begin
    matrik1[kolomakhir,g]:=deret6[maxderet];
    inc(maxderet);
  end;
  kolomakhir:=kolomakhir-1;
  //tulis ke kiri
  for g:=kolomakhir downto kolomawal do
  begin
    matrik1[g,barisakhir]:=deret6[maxderet];
    inc(maxderet);
```

```
  end;
  barisakhir:=barisakhir-1;
  //tulis ke atas
  for g:=barisakhir downto barisawal do
  begin
    matrik1[kolomawal,g]:=deret6[maxderet];
    inc(maxderet);
  end;
  kolomawal:=kolomawal+1;
end;
//baca matrik perbaris
maxderet:=1;
for g:=1 to tg do
begin
  for h:=1 to lb do
  begin
    deret7[maxderet]:=matrik1[h,g];
    inc(maxderet);
  end;
end;
//menggambarkan ke image2
maxderet:=1;
for g:=0 to tg-1 do
begin
  for h:=0 to lb-1 do
  begin
    image3.Canvas.Pixels[h,g]:=deret7[maxderet];
    inc(maxderet);
  end;
end;
image2.Picture:=image3.Picture;
```

Pada prosedur dekripsi ini dapat dijelaskan bahwa penyandian file gambar menerima masukkan gambar *ciphertext* dengan ukuran maksimal 3264 x 2448 pixel. Piksel gambar input selanjutnya dibaca secara baris perbaris dimulai dari pojok kiri atas, kemudian hasil pembacaan tersebut ditransposisi dengan cara mengganti urutan piksel sesuai kata kunci, setelah itu hasil transposisi di substitusikan, kemudian piksel disusun secara spiral dari pojok kiri atas sehingga menghasilkan gambar plaintext. Kemudian pembentukan dalam bentuk gambar ciphertext menjadi gambar plaintext, dengan hasil piksel gambar yang utuh.

4.3. Pengujian

Pada pembahasan ini akan dilakukan pengujian tingkat keberhasilan pada proses enkripsi dan dekripsi yaitu setiap citra sebagai media penyandiannya dienkripsi dengan beberapa kata sandi. Pada proses dekripsinya citra yang terenkripsi di lakukan dekripsi dengan sandi yang sama maupun berbeda. Citra yang menjadi media pengujiannya antara lain **Kuda.Bmp, Alam.bmp, Merahputih.bmp, Ruang.bmp, Tangga.bmp, Tentara.bmp, Semarang.bmp** Citra-citra tersebut dapat dilihat pada gambar 4.17

4.17a Alam.bmp

4.17b Ruang.bmp

4.17c Tentara.bmp

4.17d Semarang.jpg

4.17e Kuda.bmp

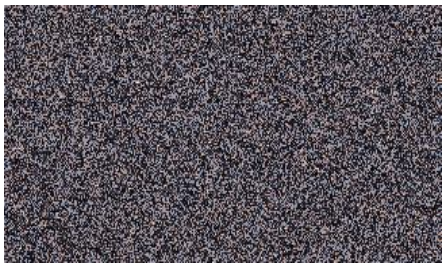
4.17f Tangga.bmp

4.17g Merahputih.bmp

Gambar 4.17 Citra Yang Akan Dilakukan Pengujian

4.3.1 Pengujian pada Proses Enkripsi

Pengujian pada proses enkripsi ini, citra sebagai medianya diuji dengan beberapa kata sandi. Sebagai salah satu pengujiannya pada citra **alam.bmp** di enkripsi dengan kata sandi “1a2s3d4f”, maka akan dihasilkan cipherteks dengan ukuran piksel tetap sama dengan ukuran piksel plainteksnya. Hasil pengujian dengan citra Alam.bmp dapat dilihat pada gambar 4.18:



plaintext (alam.bmp 324x186 piksel) *ciphertext* (alam 324x168 piksel)

Gambar 4.18 hasil pengujian proses enkripsi citra alam.bmp

Hasil pengujian diatas dapat dilihat bahwa setelah dilakukan enkripsi dengan kata sandi, plainteks maupun cipherteksnya tetap memiliki ukuran piksel yang sama. Untuk hasil pengujian proses enkripsi pada citra yang lainnya dapat dilihat pada table 4.1

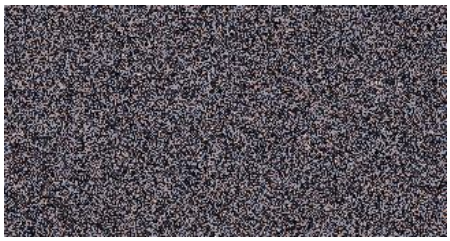
Tabel 4.1 Hasil Percobaan Proses Enkripsi

No	Nama Gambar	Ukuran Plainteks (piksel)	Kata sandi	Ukuran Cipherteks (piksel)	Waktu (detik)	Keterangan
1	Kuda. bmp	630x472	namamama	630 x 472	02	Piksel tetap sama
			12345678asdfghjk		02	Piksel tetap sama
			namalama	-	-	Tidak dapat diproses
2	Merah Putih.bmp	462x800	-	-	-	Citra Berorientasi portrait
3	Semarang. jpg	1280x1024	-	-	-	Format Citra tidak 32 bit
4	Alam.bmp	324x168	1a2s3d4f	324x168	00	Piksel tetap sama
			Asdfghjk12345678		00	Piksel tetap sama
			Asdfg		-	Kata sandi 5 char
5	Ruang.bmp	600x350	zxcvbnma	600x350	01	Piksel tetap sama
			qwertyuioasdfghj		01	Piksel tetap sama
6	Tangga.bmp	800x600	12345678qwertyui	800x600	03	Piksel tetap sama
			12345678	-	-	Kata sandi harus 16 karakter
7	Tentara.bmp	820x540	qawsedrftgyhujik	820x540	02	Piksel tidak berubah
			87654321	-	-	Kata sandi harus 16 karakter

Dari hasil percobaan tersebut ada yang berhasil sehingga diperoleh cipherteksnya, ada juga yang tidak berhasil dilakukan proses enkripsi. Percobaan yang tidak berhasil dikarenakan pemberian kata sandi tidak memenuhi ketentuan yaitu harus dengan 8 karakter atau 16 karakter, selain itu gambar yang digunakan adalah berorientasi portrait, dan juga gambar yang digunakan bukan 32 bit sehingga proses enkripsi tidak berhasil. Jadi dalam proses enkripsi ini citra yang harus digunakan adalah format 32 bit dan berorientasi lanscap, sedangkan kata sandi yang digunakan adalah 8 karakter atau 16 karakter.

4.3.2 Pengujian pada Proses Dekripsi

Pengujian pada proses dekripsi ini, citra yang sudah di enkripsi dilakukan proses dekripsi dengan beberapa sandi untuk mendapatkan citra awal. Sebagai salah satu pengujian dekripsi digunakan cipherteks dari citra alam.bmp yang sudah di enkripsi dengan kata sandi “1a2s3d4f”, maka pada proses dekripsi ini dengan kata sandi yang sama untuk mendapatkan citra seperti plainteks sebelumnya. Dapat dilihat pada gambar 4.19



ciphertext (alam.bmp 324 x186 piksel)

plaintext (alam 324x168 piksel)

Gambar 4.19. Hasil pengujian proses dekripsi

Dari pengujian dekripsi diatas ternyata diperoleh citra plainteks tetap sama dengan citra sebelum dilakukan enkripsi dengan ukuran piksel juga tetap sama. Hasil pengujian pada proses dekripsi pada cipherteks yang lainnya dapat dilihat pada tabel 4.2

Tabel 4.2 Hasil Percoabaan dekripsi

Nama gambar	Ukuran Plainteks *(Width x height)* piksel	Kata Sandi	Ukuran Cipherteks *(Width x height)* piksel	Waktu (detik)	Ket
Kuda.bmp	630x472	asdfghjk	630x472	02	Baik
		12345678asdfghjk		02	Tidak
		12345678		-	rusak
Alam. Bmp	324x168	1a2s3d4f	324x168	00	Baik
		Asdfghjk12345678		00	Baik
		asdfghjk		-	rusak
Ruang.bmp	600x350	zxcvbnma	600x350	01	Baik
		qwertyuioasdfghj		01	Baik
		12345678			rusak
Tangga.bmp	800x600	12345678qwertyui	800x600	03	Baik
		12345678		-	rusak
Tentara.bmp	820x540	qawsedrftgyhujik	820x540	02	Baik
		87654321		-	rusak

Pada pengujian proses dekripsi ini didapat citra dengan kualitas baik karena sesuai denga citra asli sebelum dilakukan proses enkripsi, ini karena dalam proses dekripsi sandi yang digunakan sesuai dengan kata sandi yang digunakan pada proses enkripsi. Akan tetapi apabila pemberian kata sandi tidak sama dengan sebelumnya maka citra hasil proses dekripsi akan rusak atau tidak dapat dihasilkan citra seperti citra aslinya.

BAB V
PENUTUP

5.1 Kesimpulan

Setelah menyelesaikan penulisan tentang penyandian file gambar dengan metode substitusi dan tranposisi, dan dilakukan pengujian dari program yang telah dibuat pada bab-bab sebelumnya dapat diambil kesimpulan :

1. Aplikasi penyandian pada file gambar dengan metode substitusi dan transposisi dapat menghasilkan suatu gambar yang tidak dapat dikenali seperti gambar semula.
2. Teknik penyandian dengan substitusi dan transposisi yang diterapkan dalam mengakses suatu bit-bit dari gambar berhasil memanipulasi posisi dan mengacak susunan piksel pada gambar.
3. Gambar yang sudah melewati proses penyandian dan tidak dapat dikenali dapat dikembalikan lagi oleh program aplikasi yang telah dibuat, sehingga gambar dapat kembali lagi dan dapat dikenali.
4. Gambar yang memiliki ukuran piksel yang kecil menghasilkan waktu proses yang singkat, sebaliknya jika ukuran piksel gambar besar, maka waktu yang diperoleh akan semakin lama.
5. Ketika dilakukan pengujian dalam membandingkan gambar asli sebelum gambar dilakukan penyandian dengan gambar setelah disandikan dan gambar dekripsi tidak terdapat perbedaan dalam ukuran piksel gambar.

5.2 Saran

Sebagai pengembangan lebih lanjut dari program aplikasi penyandian file gambar dengan metode substitusi dan transposisi, penulis memberikan saran yang perlu diperhatikan :

1. Aplikasi program ini akan jauh lebih baik, kalau program ini dapat membedakan antara gambar plainteks dan gambar cipherteks.
2. Untuk lebih baiknya dalam pengembangan aplikasi ini tidak hanya terbatas pada satu format gambar, melainkan sumua format gambar dapat digunakan .

DAFTAR PUSTAKA

[1] Aniati Murti dan Suryana Setiawan, 1992. *Pengantar Penggolahan Citra,* PT Elex Media Komputindo. Jakarta.

[2] Dony Aryus, 2008. *Pengantar Ilmu kriptografi keamanan Teori analisis dan Implemtasi.* Andi. Yogyakarta.

[3] Ir. Rinaldi Munir, M.T, 2004, *Pengolahan Citra Dengan Pendekatan Algoritmik*. Informatika, Bandung.

[4] Ir. Rinaldi Munir, M.T, 2006, *Kriptografi*. Informatika. Bandung.

[5] Ir. Yusuf Kurniawan, M.T, 2004, *Kriptografi Keamanan Internet dan Jaringan Komunikasi*. Informatika. Bandung.

[6] Jaja Jamaludin Malik, 2005, *Tip & Trik Unik Delphi,* Andy. Yogyakarta.

[7] Jaja Jamaludin Malik, 2006, *Kumpulan Latihan Pemrograman Delphi*, Andy. Yogyakarta.

[8] Rafael C. Gonsalez / paul Wintz, 1987, " *Digital Image Processing*",Wesly publishing Company Inc.

[9] Rijanto TOSIN, 1997, *Flowchart untuk Siswa dan Mahasiswa,* Dinastindo. Jakarta.

[10] Rosa A.S-M.Shalahuddin, 2011, *Rekayasa Perangkat Lunak*, Modula. Bandung

[11] Usman Ahman, 2005. *Pengolahan Citra Digital dan Teknik Pemrogramannya.* Graha Ilmu. Yogyakarta.