SKRIPSI

RANCANG BANGUN APLIKASI ENKRIPSI DAN DEKRIPSI *AUDIO VIDEO* MENGGUNAKAN ALGORITMA *BLOWFISH* BERBASIS *VISUAL BASIC*

Disusun Oleh

BENNY GRES SANUTRA

07. 12. 590

JURUSAN TEKNIK ELEKTRO S-1

KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

INSTITUT TEKNOLOGI NASIONAL MALANG

2012

[illegible]

LEMBAR PERSETUJUAN

RANCANG BANGUN APLIKASI ENKRIPSI DAN DEKRIPSI *AUDIO VIDEO* MEGGUNAKAN ALGORITMA *BLOWFISH* BERBASIS *VISUAL BASIC*

SKRIPSI

Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Teknik Komputer dan Informatika Strata Satu (S-1)

Disusun oleh :

BENNY GRES SANUTRA

07. 12. 590

Mengetahui,

Ketua Jurusan Teknik Elektro S-1

Ir. Yusuf Ismail Nakhoda, MT
NIP.Y.1018800189

Diperiksa dan Disetujui

Dosen Pembimbing I	Dosen Pembimbing II
Sotyohadi, ST	**Michael Ardita, ST, MT**
NIP.Y.103.970.0309	**NIP.P.103.100.0434**

JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2012

RANCANG BANGUN APLIKASI ENKRIPSI DAN DEKRIPSI AUDIO VIDEO MEGGUNAKAN ALGORITMA BLOWFISH BERBASIS VISUAL BASIC

Benny Gres Sanutra
(07.12.590)

Dosen Pembimbing:
Sotyohadi, ST
Michael Ardita, ST, MT

Konsentrasi Komputer dan Informatika, Jurusan Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Nasional Malang
Jln. Raya Karanglo Km 2 Malang
Email: b3_gr3ss@yahoo.com

Abstrak

Untuk menjaga keamanan data ataupun informasi yang tersimpan dalam bentuk file, salah satu caranya dengan menggunakan metode kriptografi untuk mengenkripsi data file tersebut sehingga tidak dilihat oleh pihak yang tidak berhak. Salah satu algoritma kriptografi adalah algoritma Blowfish yang merupakan algoritma kriptografi modern kunci simetris berbentuk cipher block. Aplikasi yang dibangun ini dapat mengenkripsi file audio video. Enkripsi dilakukan dengan menggunakan kunci tertentu, sehingga menghasilkan chipertext (file yang sudah dienkrip atau disandikan) yang tidak dapat dibaca ataupun dimengerti. Ciphertext tersebut dapat dikembalikan seperti semula jika didekripsi menggunakan kunci yang sama sewaktu mengenkripsi file tersebut. Kunci yang digunakan maksimum 56 karakter(448 bit). Perangkat lunak yang digunakan untuk User-System Interface-nya adalah Visual Basic 6.0.

Kata Kunci: kriptografi, kunci simetrik, algoritma blowfish, enkripsi, dekripsi, audio video, visual basic 6.0

SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini :

Nama : Benny Gres Sanutra

NIM : 07.12.590

Program Studi : Teknik Elektro S-1

Konsentrasi : Teknik Komputer dan Informatika

Dengan ini menyatakan bahwa Skripsi yang saya buat adalah hasil karya sendiri, tidak merupakan plagiasi dari karya orang lain. Dalam Skripsi ini tidak memuat karya orang lain, kecuali dicantumkan sumbernya sesuai dengan ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat, dan apabila di kemudian hari ada pelanggaran atas surat pernyataan ini, saya bersedia menerima sangsinya.

Malang, 22 Maret 2012

Yang membuat Pernyataan,

Benny Gres Sanutra
NIM : 07.12.590

SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini

Nama : Henry Tyres Samosir

NIM : 07.12.508

Program Studi : Teknik Elektro S-1

Konsentrasi : Teknik Komputer dan Informatika

Dengan ini menyatakan bahwa Skripsi yang saya buat adalah hasil karya sendiri, tidak merupakan plagiasi dari karya orang lain. Dalam Skripsi ini tidak memuat karya orang lain, kecuali dicantumkan sumbernya sesuai dengan ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat, dan apabila di kemudian hari ada pelanggaran atas surat pernyataan ini, saya bersedia menerima sanksinya.

Malang, 22 Maret 2012

Yang membuat Pernyataan,

Henry Tyres Samosir

NIM : 07.12.508

KATA PENGANTAR

Puji syukur kehadirat Tuhan Yang Maha Esa, yang telah memberikan berkat-Nya, sehingga penulis dapat menyelesaikan laporan Skripsi ini dengan baik dan lancar.

Laporan Skripsi ini merupakan salah satu persyaratan akademik dalam menyelesaikan program Strata 1 Jurusan Teknik Elektro, Konsentrasi Komputer & Informatika, Institut Teknologi Nasional Malang. Adapun judul laporan Skripsi ini adalah:

RANCANG BANGUN APLIKASI ENKRIPSI DAN DEKRIPSI *AUDIO VIDEO* MEGGUNAKAN ALGORITMA *BLOWFISH* BERBASIS *VISUAL BASIC*

Selanjutnya pada kesempatan ini penulis juga menyampaikan rasa terimakasih yang sebesar-besarnya kepada pihak–pihak yang telah banyak membantu penulis selama penyusunan tugas akhir, diantaranya :

1. Bapak Ir. Yusuf Ismail Nahkoda, MT selaku Ketua Jurusan Teknik Elektro S-1 ITN Malang.
2. Bapak Dr. Aryuanto Soetedjo, ST, MT selaku Sekertaris Jurusan Teknik Elektro S-1 ITN Malang dan pengusul serta penyedia ruang Skripsi.
3. Bapak Sotyohadi, ST selaku Dosen Pembimbing I
4. Bapak Michael Ardita, ST, MT selaku Dosen Pembimbing II
5. Kedua orangtua dan kakakku yang telah memberikan dukungan untuk selalu berdoa, berusaha dan nasehat yang telah diberikan sampai saat ini.
6. Seluruh dosen dan pegawai ITN Kampus 2 Malang.

7. Semua teman-teman mahasiswa ITN Malang yang tidak mungkin saya sebutkan satu-persatu, Anak-anak Budi jaya kos.
8. Semua pihak yang telah membantu penulis dalam menyelesaikan skripsi ini yang tidak bisa penulis sebutkan satu persatu.

Penulis berharap agar buku laporan Skripsi ini dapat memberikan banyak manfaat bagi semua pihak yang membutuhkan, khususnya bagi rekan-rekan mahasiswa. Penulis menyadari bahwa dalam penyusunan laporan ini masih banyak kekurangan, oleh karena itu mohon maaf apabila dalam buku ini terdapat hal-hal yang kurang berkenan dihati para pembaca.

Penulis juga mengharap koreksi, kritik serta saran-saran yang bermanfaat demi kesempurnaan buku Laporan Skripsi ini.

Malang, Februari 2012

Benny Gres Sanutra

DAFTAR ISI

DAFTAR TABEL

BAB III PEMBAHASAN DAN PERANCANGAN

BAB IV ALGORITMA DAN IMPLEMENTASI

DAFTAR GAMBAR

BAB II LANDASAN TEORI

BAB III PEMBAHASAN DAN PERANCANGAN

BAB IV ALGORITMA DAN IMPLEMENTASI

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kemudahan akses media komunikasi membawa pengaruh terhadap keamanan informasi yang menggunakan media komunikasi sebagai media penyampaian. Informasi menjadi sangat rentan untuk diketahui, diambil atau bahkan dimanipulasi dan disalahgunakan oleh pihak lain yang tidak berhak. Selama pengiriman dan ketika sampai di tujuan, informasi tersebut harus tetap rahasia dan terjaga keasliannya atau tidak dimodifikasi. Penerima informasi harus yakin bahwa informasi tersebut memang benar berasal dari pengirim yang tepat, begitu juga sebaliknya, pengirim yakin bahwa penerima informasi adalah orang yang sesungguhnya. Selain itu penerima tidak ingin pengirim membantah pernah mengirim informasi tersebut, dan jika hal tersebut terjadi penerima perlu membuktikan ketidakbenaran penyangkalan tersebut. Untuk permasalahan-permasalahan keamanan tersebut diperlukan suatu metode untuk menjaga keamanan informasi. Salah satu metodenya adalah kriptografi.

Kriptografi akan merahasiakan informasi dengan menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Saat ini banyak bermunculan algoritma kriptografi yang terus dianalisis, dicoba dan disempurnakan untuk mencari algoritma yang dianggap memenuhi standar keamanan. Beberapa algoritma kriptografi yang dikenal antara lain *DES, Rijndael, Blowfish, RC4, Vigenere Cipher, Enigma, IDEA* dan lainnya.

Blowfish merupakan salah satu algoritma yang tidak dipatenkan dan cukup kuat karena memiliki ruang kunci yang besar dan panjangnya bisa beragam, sehingga tidak mudah diserang pada bagian kuncinya. Suatu sistem kriptografi yang baik terletak pada kerahasiaan kunci dan bukan pada kerahasiaan algoritma yang digunakan. *Blowfish* pada strategi implementasi yang tepat akan lebih optimal, dapat berjalan pada memori kurang dari 5 KB dan kesederhanaan

pada algoritmanya. Untuk itu dibangun sebuah aplikasi yang dapat digunakan untuk mengamankan data atau informasi berupa file audio video dengan menggunakan metode *Blowfish* ini.

Atas dasar uraian di atas, maka dibuat suatu "RANCANG BANGUN APLIKASI ENKRIPSI DAN DEKRIPSI AUDIO VIDEO MEGGUNAKAN ALGORITMA BLOWFISH BERBASIS VISUAL BASIC".

1.2 RumusanMasalah

Dalam skripsi ini, yang menjadi permasalahan adalah keamanan data audio video sangat penting, maka pada skripsi ini dapat dirumuskan sebagai berikut :

1. Bagaimana membuat rancang bangun aplikasi enkripsi dan dekripsi pada audio video?
2. Bagaimana mengaplikasikan algoritma *blowfish* supaya bisa mengenkripsi ataupun mendeskripsikan data audio video?

1.3Tujuan

Adapun tujuan penulisan skripsi ini adalah :

1. Membuat rancang bangun aplikasi enkripsi dan dekripsi audio video menggunakan algoritma *blowfish* .
2. Dengan penerapan algoritma *blowfish* yang terdapat pada aplikasi ini diharapkan dapat mengamankan data audio video lebih baik lagi.

1.4 BatasanMasalah

Dalam Pembuatan aplikasi ini, penulis membuat ruang lingkup pembahasan masalah atau batasan masalah sebagai berikut :

1. Rancang bangun aplikasi yang dihasilkan hanya untuk enkripsi pada file *audio video* yang ber*extensi (mp3 ,wav, flac, aac ,ogg, amr, avi, swf, flv, mkv, mp4, mpg, dat, wmv, 3gp)* dan dekripsi file ber*extensi (blw)*.
2. Algoritma *Blowfish* hanya menggunakan operasi – operasi seperti : penambahan, XOR, dan lookup table pada operan32-bit.
3. Panjang kunci yang digunakan maksimal 448-bit.
4. Rancang bangun aplikasi ini hanya bisa dioperasikan pada *windows XP*.

1.5 Metolodogi Penelitian

Adapunmetode penelitian yang digunakan adalah sebagai berikut:

1. Studi literatur

   Pengumpulan data yang dilakukan dengan mencari bahan-bahan kepustakaan dan referensi dari berbagai sumber sebagai landasan teori yang ada hubungannya dengan permasalahan yang dijadikan objek penelitian.
2. Analisa Kebutuhan Aplikasi

   Data dan informasi yang telah diperoleh akan dianalisa agar dihasilkan kerangka global yang bertujuan untuk mendefinisikan kebutuihan aplikasi dimana natinya akan digunakan sebagai acuan perancangan aplikasi.
3. Perancangan dan Implementasi

   Bersdasarkan data dan informasi yang telah diperoleh serta analisa kebutuhan untuk membangun aplikasi ini, akan dibuat rancangan kerangka global yang menggambatkan mekanisme dari aplikasi yang akan dibuat dan diimplemenatasikan ke dalam aplikasi
4. Coding dan Penjelasan Listing Program

   Pada tahap ini akan menjelaskan fungsi-fungsi pada listing program yang digunkan untuk proses ekspansi kunci, enkripsi dan dekripsi.
5. Eksperimen dan Evaluasi

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

1.6 Sistematika Penulisan

Untuk mempermudah dan memahami pembahasan penulisan skripsi ini, maka sistematika penulisan disusun sebagai berikut :

Bab I : Pendahuluan

Berisi Latar Belakang, Rumusan Masalah, Tujuan Penelitian, Pembatasan Permasalahan, Metode Penelitian dan Sistematika Penulisan.

Bab II : Tinjauan Pustaka

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

Bab III : Perancangan dan Analisa Sistem

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

Bab IV : Pembuatan dan Pengujian Sistem

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

Bab V : Penutup

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

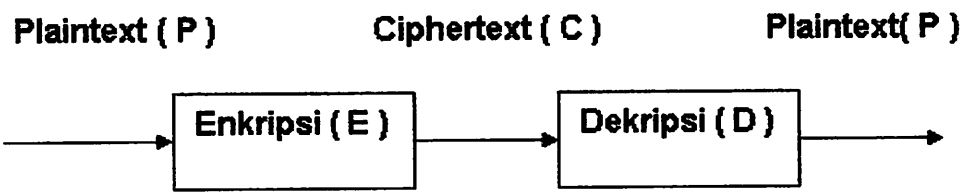
BAB II
LANDASAN TEORI

Dalam pembuatan aplikasi enkripsi dan dekripsi audio video ini, mengacu pada beberapa dasar teori yang mendukung sistem kerja dari aplikasi. Adapun dasar teori yang digunakan dalam perancangan aplikasi ini adalah sebagai berikut.

2.1 Enkripsi dan Dekripsi [1]

2.1.1 Defínsi Umum Enkripsi dan Dekripsi

Enkripsi yaitu suatu proses pengaman suatu data yang disembunyikan atau proses konversi data (plaintext) menjadi bentuk yang tidak dapat dibaca atau dimengerti. Sedangkan Dekripsi yaitu kebalikan dari proses enkripsi yaitu proses konversi data yang sudah dienkripsi (ciphertext) kembali menjadi data aslinya (Original Plaintext) sehingga dapat dibaca/ dimengerti kembali. Pesan yang akan dienkripsi disebut plaintext yang dimisalkan plaintext (P), proses enkripsi dimisalkan enkripsi (E), proses dekripsi dimisalkan dekripsi (D), dan pesan yang sudah dienkripsi disebut ciphertext yang dimisalkan ciphertext (C) maka dapat digambarkan pada gambar berikut ini :



Proses Enkripsi dan Dekripsi

Data atau informasi yang akan dienkripsi (plaintext) diacak oleh suatu kunci yang telah ditentukan kemudian output dari proses enkripsi(cipher text) dikembalikan kebentuk aslinya oleh sebuah kunci yang sama.

Metoda-metoda kriptografi yang sering digunakan biasanya berdasarkan pada konsep matematika dimana konsep ini berfungsi dalam membentuk logika

dan algoritma disamping itu ada juga bentuk kriptografi yang tidak memakai suatu konsep tertentu sehingga sulit sekali untuk dikenali pengacakan datanya. Algoritma kriptografi dengan menggunakan kunci dapat dikelompokkan menjadi 2 yaitu :

- **Kunci Simetris / Symetric key** : Kunci simetris bisa disebut juga conventional key, single key, one key atau secret key. Algoritma simetris pada proses enkripsi dan dekripsinya menggunakan satu kunci, sehingga pengirim dan penerima terlebih dahulu harus memiliki kunci yang sama yang telah disepakati untuk digunakan sehingga pengirim dan penerima dapat melakukan komunikasi. Plaintext akan melewati proses enkripsi dan menghasilkan ciphertext, kemudian disandikan kembali dengan menggunakan kunci yang sama sehingga menjadi bentuk aslinya. Metoda kunci simetris/ symetric key lebih sesuai digunakan dalam satu area gedung karena pengiriman pesannya tidak menggunakan penyimpanan pesan, sehingga keamanan algoritma simetris ini terletak pada keamanan pengiriman kunci dan pada panjangnya kunci yang dipergunakan. Kelemahan algoritma dengan menggunakan kunci simetris ini adalah kunci harus didistribusikan dengan aman, karena kunci ini mempunyai derajat kerahasiaan yang sama dengan data yang dikirim, selain itu juga kunci tidak boleh terungkap sedikitpun.

- **Kunci Asimetris / Asymetric key** : Pada kunci asimetris/ asymetric key, penggunaan kunci untuk proses enkripsi data berbeda dengan kunci untuk proses dekripsinya sehingga metoda enkripsi dengan menggunakan kunci asimetris/ asymetric key berbeda bila dibandingkan dengan penggunaan metoda kunci simetris/ simetric key.

2.1.2 Algoritma *Blowfish* [2]

Blowfish merupakan enkripsi yang termasuk dalam golongan *Symmetric Cryptosystem* , metode enkripsinya mirip dengan DES (DES-like Cipher) diciptakan oleh seorang *Cryptanalyst* bernama Bruce Schneier Presiden perusahaan Counterpane Internet Security, Inc (Perusahaan konsultan tentang kriptografi dan keamanan Komputer) dan dipublikasikan tahun 1994. Dibuat untuk digunakan pada komputer yang mempunyai microposesor besar (32-bit keatas dengan *cache* data yang besar).

Blowfish dikembangkan untuk memenuhi kriteria desain yang cepat dalam implementasinya dimana pada keadaan optimal dapat mencapai 26 *clock cycle per byte*, kompak dimana dapat berjalan pada memori kurang dari 5 KB, sederhana dalam algoritmanya sehingga mudah diketahui kesalahannya, dan keamanan yang variabel dimana panjang kunci bervariasi (minimum 32 bit, maksimum 448 bit, *Multiple* 8 bit, *default* 128 bit).Blowfish dioptimasikan untuk berbagai aplikasi dimana kunci tidak sering berubah, seperti pada jaringan komunikasi atau enkripsi file secara otomatis. Dalam pengimplementasiannya dalam komputer bermicroprosesor 32-bit dengan *cache* data yang besar (Pentium dan Power PC) Blowfish terbukti jauh lebih cepat dari DES.

Dalam penerapannya sering kali algortima ini menjadi tidak optimal. Karena strategi implementasi yang tidak tepat. Algoritma Blowfish akan lebih optimal jika digunakan untuk aplikasi yang tidak sering berganti kunci, seperti jaringan komunikasi atau enkripsi file otomatis. Selain itu, karena algoritma ini membutuhkan memori yang besar, maka algoritma ini tidak dapat diterapkan untuk aplikasi yang memiliki memori kecil seperti smart card. Panjang kunci yang digunakan, juga mempengaruhi keamanan penerapan algoritma ini. Algoritma Blowfish terdiri atas dua bagian, yaitu ekspansi kunci dan enkripsi data.

a. Ekspansi kunci (Key-expansion) [6]

Berfungsi merubah kunci (minimum 32-bit, maksimum 448-bit) menjadi beberapa array subkunci (subkey) dengan total 4168 byte

(18x32-bit untuk P-array dan 4x256x32-bit untuk S-box sehingga totalnya 33344 bit atau 4168 byte). Kunci disimpan dalam K-array:

K1, K2, ... Kj 1 _ j _ 14

Kunci-kunci ini yang dibangkitkan (generate) dengan menggunakan subkunci yang harus dihitung terlebih dahulu sebelum enkripsi atau dekripsi data. Sub-sub kunci yang digunakan terdiri dari :

P-array yang terdiri dari 18 buah 32-bit subkunci,

P1, P2, ..., P18

S-box yang terdiri dari 4 buah 32-bit, masing-masing memiliki 256 entri :

S1,0, S1,1, ..., S1,255

S2,0, S2,1, ..., S2,255

S3,0, S3,1, ..., S3,255

S4,0, S4,1, ..., S4,255

Langkah-langkah perhitungan atau pembangkitan subkunci tersebut adalah sebagai berikut:

1. Inisialisasi P-array yang pertama dan juga empat S-box, berurutan, dengan string yang telah pasti. String tersebut terdiri dari digit-digit heksadesimal dari phi, tidak termasuk angka tiga di awal.

   Contoh :

   P1= 0x243f6a88

   P2= 0x85a308d3

   P3= 0x13198a2e

   P4= 0x03707344

   dan seterusnya sampai S-box yang terakhir (daftar heksadesimal digit dari phi untuk P-array dan Sbox bisa lihat Lampiran).
2. XOR-kan P1 dengan 32-bit awal kunci, XOR-kan P2 dengan 32-bit berikutnya dari kunci, dan seterusnya untuk semua bit kunci. Ulangi siklus seluruh bit kunci secara berurutan sampai seluruh P-array ter-XOR-kan dengan bit-bit kunci. Atau jika disimbolkan :

P1 = P1⊕K1, P2 = P2⊕K2, P3 = P3⊕K3, . . . P14 = P14⊕K14, P15 = P15⊕K1, . . . P18 = P18⊕K4.

Keterangan : ⊕ adalah simbol untuk XOR.

3. Enkripsikan string yang seluruhnya nol (all-zero string) dengan algoritma Blowfish, menggunakan subkunci yang telah dideskripsikan pada langkah 1 dan 2.
4. Gantikan P1 dan P2 dengan keluaran dari langkah 3.
5. Enkripsikan keluaran langkah pada nomer 3 dengan langkah pada nomer 2 menggunakan algoritma Blowfish dengan subkunci yang telah dimodifikasi (sebab langkah pada nomer 2 menghasilkan kunci baru).
6. Gantikan P3 dan P4 dengan keluaran dari langkah 5.
7. Lanjutkan langkah-langkah di atas, gantikan seluruh elemen P-array dan

   kemudian keempat S-box secara berurutan, dengan hasil keluaran algoritma Blowfish yang terus-menerus berubah.

Total keseluruhan, terdapat 521 iterasi untuk menghasilkan subkunci-subkunci dan membutuhkan memori sebesar 4KB.

b. Enkripsi Data [8]

Terdiri dari iterasi fungsi sederhana (Feistel Network) sebanyak 16 kali putaran (iterasi), masukannya adalah 64-bit elemen data X. Setiap putaran terdiri dari permutasi kunci-dependent dan substitusi kunci dan data dependent. Semua operasi adalah penambahan (*addition*) dan XOR pada variabel 32-bit. Operasi tambahan lainnya hanyalah empat penelusuran tabel array berindeks untuk setiap putaran. Langkahnya adalah seperti berikut :

1. Bagi X menjadi dua bagian yang masing-masing terdiri dari 32-bit: XL, XR.
2. Lakukan langkah berikut :

   For i = 1 to 16:

   XL = XL _ Pi

XR = F(XL) _ XR

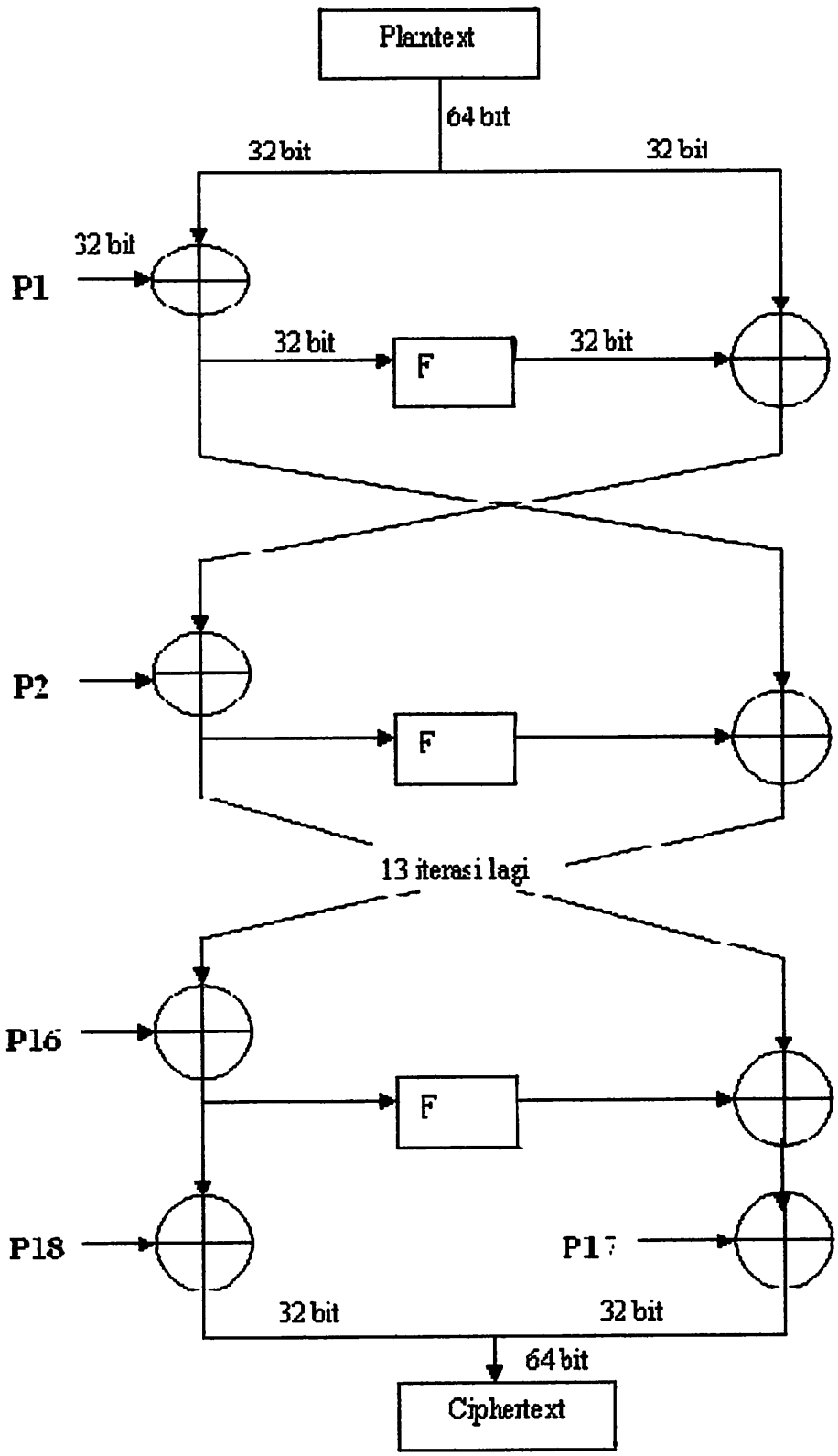
Tukar XL dan XR

3. Setelah iterasi ke-16, tukar XL dan XR lagi untuk melakukan membatalkan pertukaran terakhir.
4. Lalu lakukan

   XR = XR _ P17

   XL = XL _ P18

Terakhir, gabungkan kembali XL dan XR untuk mendapatkan cipherteks. Untuk lebih jelasnya, gambaran tahapan pada jaringan feistel yang digunakan Blowfish adalah seperti pada Gambar 1.

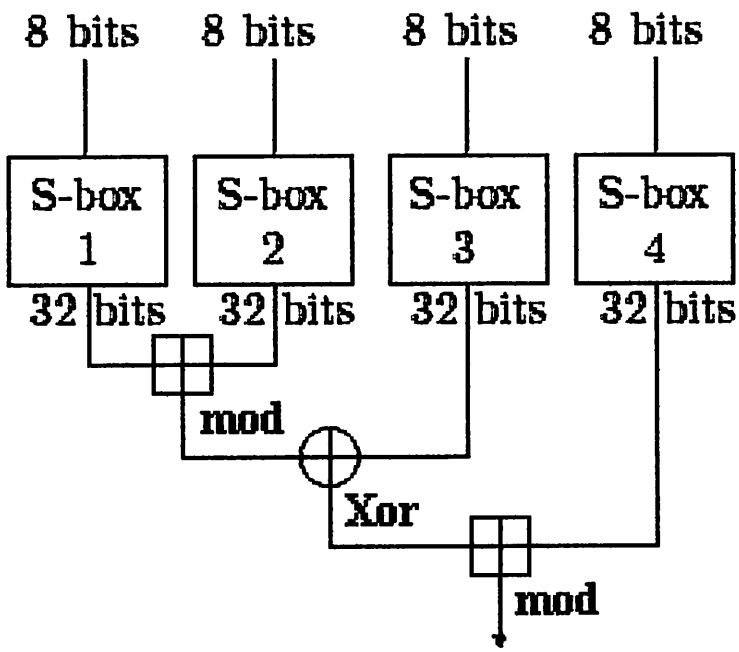


Gambar 2.1. Blok Diagram Algoritma Enkripsi Blowfish

Pada langkah kedua, telah dituliskan mengenai penggunaan fungsi F. Fungsi F adalah: bagi XL menjadi empat bagian 8-bit: a,b,c dan d.

$F(XL) = ((S1,a + S2,b \bmod 2^{32}) \text{ XOR } S3,c) + S4,d \bmod 2^{32}$

Agar dapat lebih memahami fungsi F, tahapannya dapat dilihat pada Gambar 2.



Gambar 2.2. Fungsi F dalam Blowfish

Keterangan fungsi F pada *blowfis*:

4 x 8 bit di atas merupakan inputan sub kunci dan dan inputan file yang akan proses pada s-box, jadi s-box tersebut akan melakukan permutasi pada sub knci dan data yang telah dienkripsi dengan P-array.

Untuk proses dekripsi sama persis dengan enkripsi, kecuali bahwa P1, P2,..., P18 digunakan pada urutan yang berbalik (reverse). Algoritmanya dapat dinyatakan sebagai berikut (Schneier, 1996) :

for i = 1 to 16 do

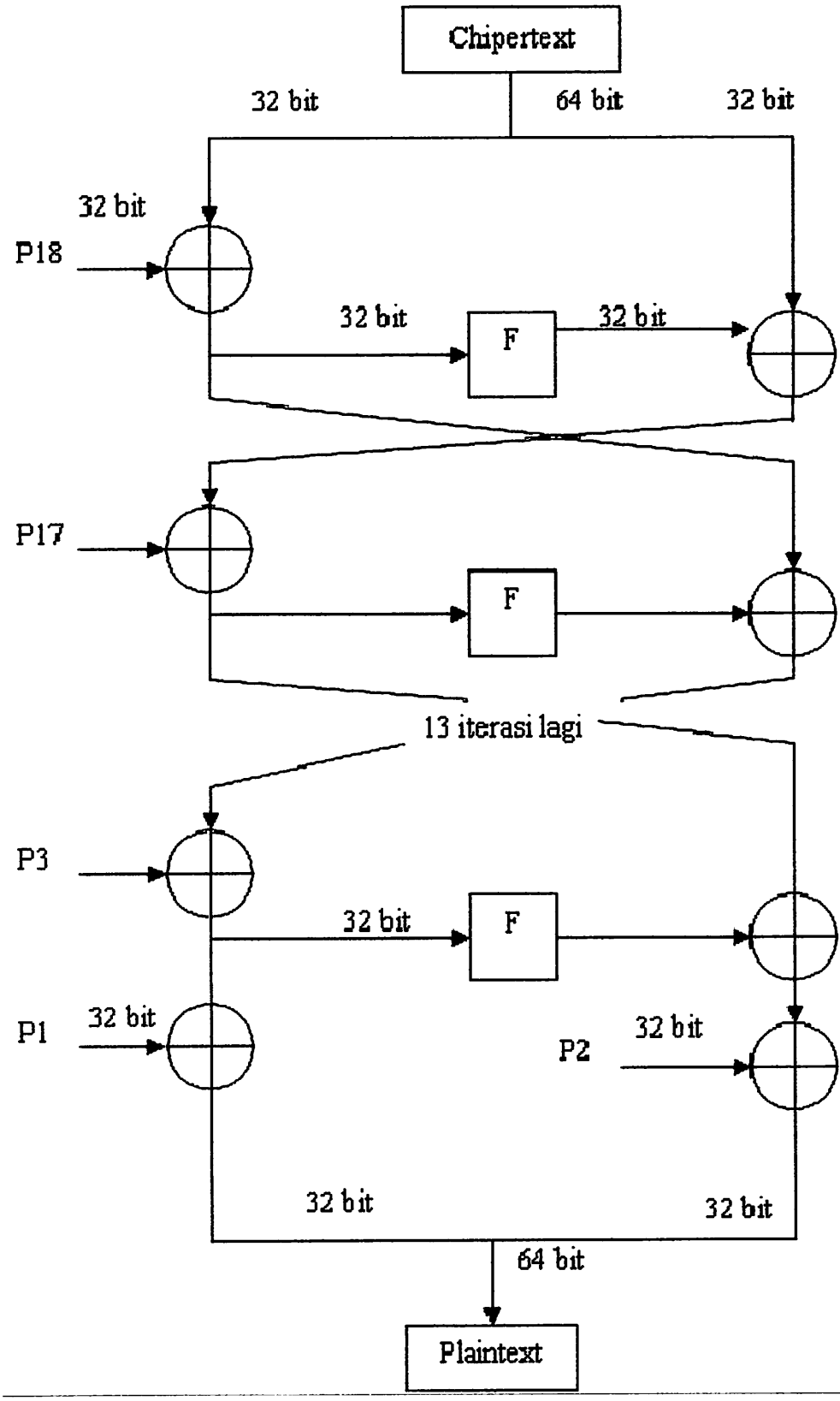
XRi = XLi-1 _ P19-i;

XLi = F[XRi] _ XRi-1;

XL17 = XR16 _ P1;

XR17 = XL16 _ P2;

Blok diagram dekripsi seperti pada Gambar 2.3



Gambar 2.3. Blok Diagram Dekripsi *Blowfish*

2.1.3 Audio Video (Multimedia) [13]

Multimedia adalah penggunaan komputer untuk menyajikan dan menggabungkan teks, suara, gambar, animasi dan video dengan alat bantu (*tool*) dan koneksi (*link*) sehingga pengguna dapat melakukan navigasi, berinteraksi, berkarya dan berkomunikasi. *Multimedia* sering digunakan dalam dunia hiburan. Selain dari dunia hiburan, *Multimedia* juga diadopsi oleh dunia game.

Multimedia dimanfaatkan juga dalam dunia pendidikan dan bisnis. Di dunia pendidikan, multimedia digunakan sebagai media pengajaran, baik dalam kelas maupun secara sendiri-sendiri. Di dunia bisnis, multimedia digunakan sebagai media profil perusahaan, profil produk, bahkan sebagai media kios informasi dan pelatihan dalam sistem e-learning.

Jenis *Multimedia* :

Pada perkembangannya *Multimedia* dibagi atas dua jenis yaitu "*Multimedia Linier*" dan "*Multimedia Interaktif. Multimedia Liner* adalah jenis *multimedia* yang berjalan lurus. *Multimedia* jenis ini bisa diliat pada semua jenis film, Tutorial Video, dll. sedangkan *Multimedia Interaktif* adalah jenis *multimedia interaksi*, artinya ada interaksi antara media dengan pengguna media melalui bantuan komputer, mouse dan keyboard.

Multimedia telah diaplikasikan dalam berbagai bentuk dan tujuan sesuai fungsi pengaplikasiannya, seperti:

1. Media Pembelajaran
2. Game: virtual game, virtual driving
3. Film, Animasi
4. Medis: animasi petunjuk medis
5. Militer
6. Bisnis
7. Desain Arsitektur
8. Olahraga dan Hobi
9. Iklan dan Promosi

2.1.4 Visual Basic [5]

Visual Basic sering disingkat sebagai VB, merupakan sebuah bahasa pemrograman yang menawarkan Integrated Development Environment (IDE) visual untuk membuat program perangkat lunak berbasis sistem operasi Microsoft Windows dengan menggunakan model pemrograman (COM). Visual Basic merupakan turunan bahasa pemrograman BASIC dan menawarkan pengembangan perangkat lunak komputer berbasis grafik dengan cepat. Beberapa bahasa skrip seperti Visual Basic for Applications (VBA) dan Visual Basic Scripting Edition (VBScript), mirip seperti halnya Visual Basic, tetapi cara kerjanya yang berbeda.

Para programmer dapat membangun aplikasi dengan menggunakan komponen-komponen yang disediakan oleh Microsoft Visual Basic Program-program yang ditulis dengan Visual Basic juga dapat menggunakan Windows API, tapi membutuhkan deklarasi fungsi luar tambahan.

Dalam pemrograman untuk bisnis, Visual Basic memiliki pangsa pasar yang sangat luas. Sebuah survey yang dilakukan pada tahun 2005 menunjukkan bahwa 62% pengembang perangkat lunak dilaporkan menggunakan berbagai bentuk Visual Basic, yang diikuti oleh C++, JavaScript, C#, dan Java.

Visual Basic adalah pengembangan dari bahasa komputer BASIC (*Beginner's All-purpose Symbolic Instruction Code*). Bahasa BASIC diciptakan oleh Professor John Kemeny dan Thomas Eugene Kurtz dari Perguruan Tinggi Dartmouth pada pertengahan tahun 1960-an. Bahasa program tersebut tersusun mirip dengan bahasa Inggris yang biasa digunakan oleh para programer untuk menulis program-program komputer sederhana yang berfungsi sebagai pembelajaran bagi konsep dasar pemrograman komputer.

Sejak saat itu, banyak versi BASIC yang dikembangkan untuk digunakan pada berbagai platform komputer, seperti Microsoft QBASIC, QUICKBASIC, GWBASIC, IBM BASICA, Apple BASIC dan lain-lain.

Apple BASIC dikembangkan oleh Steve Wozniak, mantan karyawan Hewlett Packard dan teman dekat Steve Jobs (pendiri Apple Inc.). Steve Jobs pernah bekerja dengan Wozniak sebelumnya (mereka membuat game arcade

"Breakout" untuk Atari). Mereka mengumpulkan uang dan bersama-sama merakit PC, dan pada tanggal 1 April 1976 mereka secara resmi mendirikan perusahaan komputer Apple. Popularitas dan pemakaian BASIC yang luas dengan berbagai jenis komputer turut berperan dalam mengembangkan dan memperbaiki bahasa itu sendiri, dan akhirnya berujung pada lahirnya Visual Basic yang berbasis GUI (*Graphic User Interface*) bersamaan dengan Microsoft Windows. Pemrograman Visual Basic begitu mudah bagi pemula dan programer musiman karena ia menghemat waktu pemrograman dengan tersedianya komponen-komponen siap pakai.

BAB III

PEMBAHASAN DAN PERANCANGAN

3.1 Pembahasan

Aplikasi yang akan dibuat pada tugas akhir ini adalah sebuah aplikasi enkripsi dan dekripsi *audio video* menggunakan algoritma *Blowfish*, dimana fungsi utama dari aplikasi ini mengenkripsi data audio video supaya tidak bisa dibaca *(chiperteks)* oleh pihak yang tidak berwenang, dan dapat didekripsi kembali menjadi file aslinya *(plainteks)*.

3.1.1 Persyaratan Perangkat Lunak

Perangkat lunak enkripsi dan dekripsi audio video menggunakan algoritma Blowfish ini memiliki persyaratan sebagai berikut :

1. Perangkat lunak ini terdiri dari iterasi fungsi sederhana (Feistel Network) sebanyak 16 kali putaran (iterasi), masukannya adalah 64-bit elemen data X.
2. Algoritma Blowfish hanya menggunakan operasi – operasi seperti : penambahan, XOR, dan lookup table.
3. Perangkat lunak hanya bisa menerima inputan kunci sepanjang 56 karakter (448 bit).

3.1.2 Pembahasan Proses *Blowfish*

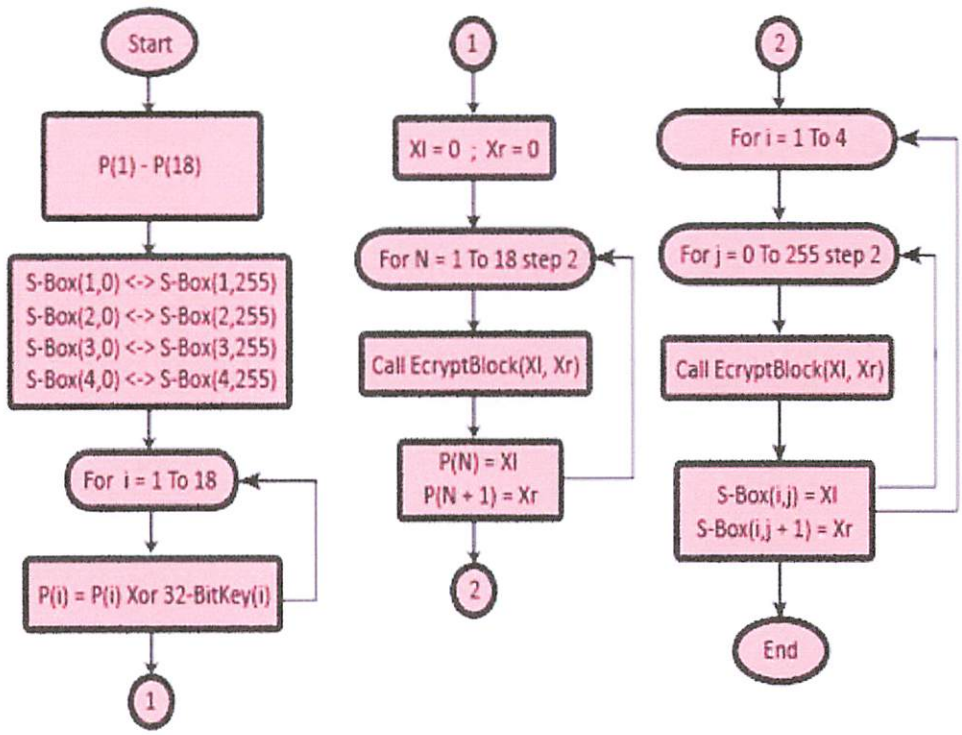
Proses penyelesaian metoda kriptografi *Blowfish* ini dapat dibagi menjadi 3 tahapan yaitu :

1. Proses Ekspansi Kunci.
2. Proses Enkripsi.
3. Proses Dekripsi.

3.1.2.1 Proses Ekspansi Kunci

Pada Algoritma Blowfish digunakan banyak Sub- Kunci. Kunci-kunci ini harus dibangkitkan terlebih dahulu sebelum dilakukan enkripsi atau dekripsi data. Berikut adalah gambaran proses pembangkitan Sub-Kunci.

Proses Ekspansi kuncinya adalah sebagai berikut :



Gambar 3.1: Proses Ekspansi Sub-Kunci

Contoh :

Misalkan *input key* = 'ALGORITMABLOWFISH', maka proses pembentukan tabel

Tabel 3.1 : Tabel perubahan karakter ke hexa

Char	A	L	G	O	R	I	T	M	A	B	L	O	W	F	I	S	H
Hex	4	4	4	4	5	4	5	4	4	4	4	4	5	4	4	5	4
a	1	C	7	F	2	9	4	D	1	2	C	F	7	6	9	8	8

Tabel 3.2 : P-Array

No	P-Array
1	243F6A88
2	85A308D3
3	13198A2E
4	03707344
5	A4093822
6	299F31D0
7	082EFA98
8	EC4E6C89
9	452821E6
10	38D01377
11	BE5466CF
12	34E90C6C
13	C0AC29B7
14	C97C50DD
15	3F84D5B5
16	B5470917
17	9216D5D9
18	8979FB1B

Tabel 3.3 : hasil XOR P1⊕32 bit awal kunci sampai semua ter-XOR-kan

No	XOR	HASIL XOR
1	243F6A88 ⊕ 414C474F	65732DC7
2	85A308D3 ⊕ 5249544D	D7EA5C9E
3	13198A2E ⊕ 41424C4F	525BC661
4	3707344 ⊕ 57464958	54363A1C
5	A4093822 ⊕ 48	A409386A
6	299F31D0 ⊕ 414C474F	68D3769F
7	82EFA98 ⊕ 5249544D	5A67AED5

8	EC4E6C89 ⊕ 41424C4F	AD0C20C6
9	452821E6 ⊕ 57464958	126E68BE
10	38D01377 ⊕ 48	38D0133F
11	BE5466CF ⊕ 414C474F	FF182180
12	34E90C6C ⊕ 5249544D	66A05821
13	C0AC29B7 ⊕ 41424C4F	81EE65F8
14	C97C50DD ⊕ 57464958	9E3A1985
15	3F84D5B5 ⊕ 48	3F84D5FD
16	B5470917 ⊕ 414C474F	F40B4E58
17	9216D5D9 ⊕ 5249544D	C05F8194
18	8979FB1B ⊕ 41424C4F	C83BB754

Setelah semua P (P1-P18) ter-XOR-kan, enkripsikan inputan string yang seluruhnya nol (all-zero string) dengan P1-P18 hasil XOR pada tabel 3.3 mengunakan algoritma *Blowfish*

Tabel 3.4 : Enkripsikan string 0 (*all-zero string*)

PUTARAN	PROSES ENKRIPSI STRING 0
P1	XL = XL ⊕ P1 XL = 0 ⊕ 65732DC7 = 65732DC7 F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((65 + 73 mod 2^{32}) XOR 32) + C7 mod 2^{32} F(XL) = 1B1 XR = F(XL) ⊕ XR XR = 1B1 ⊕ 0 = 1B1
P2	XL = XR(P1) ⊕ P2 XL = 1B1 ⊕ D7EA5C9E = D7EA5D2F F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((D7 + EA mod 2^{32}) XOR 5D) + 2F mod 2^{32} F(XL) = 21D

	XR = F(XL) ⊕ XL(P1) XR = 21D ⊕ 65732DC7 = 65732FDA
P3	XL = XR(P2) ⊕ P3 XL = 65732FDA ⊕ 525BC661 = 3728E9BB F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((37 + 28 mod 2^{32}) XOR E9) + BB mod 2^{32} F(XL) = 171 XR = F(XL) ⊕ XL(P2) XR = 171 ⊕ D7EA5D2F = D7EA5C5E
P4	XL = XR(P3)⊕ P4 XL = D7EA5C5E ⊕ 54363A1C = 83DC6642 F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((83 + DC mod 2^{32}) XOR 66) + 42 mod 2^{32} F(XL) = 77B XR = F(XL) ⊕ XL(P3) XR = 77B ⊕ 3728E9BB = 3728E9B8
P5	XL = XR(P4) ⊕ P5 XL = 3728E9B8 ⊕ A409386A = 9321D1D2 F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((93 + 21 mod 2^{32}) XOR D1) + D2 mod 2^{32} F(XL) = 137 XR = F(XL) ⊕ XL(P4) XR = 137 ⊕ 83DC6642 = 83DC6775
P6	XL = XR(P5) ⊕ P6 XL= XL = 83DC6775 ⊕ 68D3769F = EB0F11EA F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((EB + 0F mod 2^{32}) XOR 11) + EA mod 2^{32} F(XL) = 1D5 XR = F(XL) ⊕ XL(P5) XR = 1D5 ⊕ 9321D1D2 = 9321D007
P7	XL = XR(P6) ⊕ P7

	XL= 9321D007 ⊕ 5A67AED5 = C9467ED2 F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((C9 + 46 mod 2^{32}) XOR 7E) + D2 mod 2^{32} F(XL) = 243 XR = F(XL) ⊕ XL(P6) XR = 243 ⊕ EB0F11EA = EB0F13A9
P8	XL = XR(P7) ⊕ P8 XL= EB0F13A9 ⊕ AD0C20C6 = 4603336F F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((46 + 03 mod 2^{32}) XOR 33) + 6F mod 2^{32} F(XL) = E9 XR = F(XL) ⊕ XL(P7) XR = E9 ⊕ C9467ED2 = C9467E3B
P9	XL = XR(P8) ⊕ P9 XL= C9467E3B ⊕ 126E68BE = DB281685 F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((DB + 28 mod 2^{32}) XOR 16) + 85 mod 2^{32} F(XL) = 19A XR = F(XL) ⊕ XL(P8) XR = 19A ⊕ 4603336F = 460332F5
P10	XL = XR(P9) ⊕ P10 XL= 460332F5 ⊕ 38D0133F = 7ED321CA F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((7E + D3 mod 2^{32}) XOR 21) + CA mod 2^{32} F(XL) = 23A XR = F(XL) ⊕ XL(P9) XR = 23A ⊕ DB281685 = DB2814BF
P11	XL = XR(P10) ⊕ P11 XL= DB2814BF ⊕ FF182180 = 2430353F F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((24 + 30 mod 2^{32}) XOR 35) + 3F mod 2^{32}

	F(XL) = A0 XR = F(XL) ⊕ XL(P10) XR = A0 ⊕ 7ED321CA = 7ED3216A
P12	XL = XR(P11) ⊕ P12 XL= 7ED3216A ⊕ 66A05821 = 1873794B F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((18 + 73 mod 2^{32}) XOR 79) + 4B mod 2^{32} F(XL) = 10A XR = F(XL) ⊕ XL(P11) XR = 10A ⊕ 16 = 11C
P13	XL = XR(P12) ⊕ P13 XL= 11C ⊕ 81EE65F8 = 81EE64E4 F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((81 + EE mod 2^{32}) XOR 64) + E4 mod 2^{32} F(XL) = D0 XR = F(XL) ⊕ XL(P12) XR = D0 ⊕ 10A = 1DA
P14	XL = XR(P13) ⊕ P14 XL= 1DA ⊕ 9E3A1985 = 9E3A185F F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((9E + 3A mod 2^{32}) XOR 18) + 5F mod 2^{32} F(XL) = 11F XR = F(XL) ⊕ XL(P13) XR = 11F ⊕ 81EE64E4 = 81EE65FB
P15	XL = XR(P14) ⊕ P16 XL= 81EE65FB ⊕ 3F84D5FD = BE6AB006 F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((BE + 6A mod 2^{32}) XOR B0) + 06 mod 2^{32} F(XL) = 19E XR = F(XL) ⊕ XL(P14) XR = 19E ⊕ 9E3A185F = 9E3A19C1

P16	XL = XR(P15) ⊕ P16 XL= 9E3A19C1 ⊕ F40B4E58 = 6A315799 F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((6A + 31 mod 2^{32}) XOR 57) + 99 mod 2^{32} F(XL) = 165 XR = F(XL) ⊕ XL(P15) XR = 165 ⊕ BE6AB006 = BE6AB163
P17	P17 = XR(P16) ⊕ P17 P17 = BE6AB163 ⊕ C05F8194 = 7E3530F7
P18	P18 = XL(P16) ⊕ P18 P18 = 6A315799 ⊕ C83BB754 = A20AE0CD
Gabungkan kembali P17dan P18 = A20AE0CD 7E3530F7	

Keterangan :

XL = Inputan X pada Left variabel.

XR = Inputan X pada right variabel.

P(1-18) = Merupakan P- array hasil XOR pada tabel.

F(XL) = Merupakan fungsi Feistel yang didalamnya terdapat 4 S-BOX.

⊕ = XOR

((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} = Rumus dari fungsi Feistel.

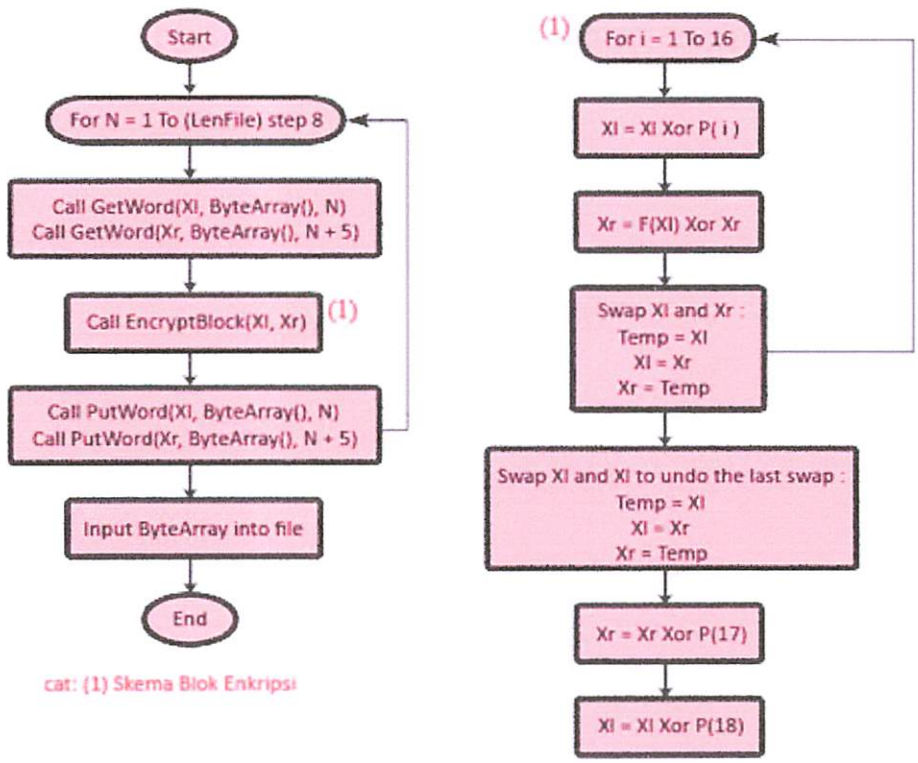
Lakukan langkah berikut supaya setiap P mendapatkan sub kunci :

Gantikan P1 dan P2 dengan keluaran dari P18 dan P17 untuk mendapatkan sub kunci.

Lakukan ekspansi kunci seperti pada tabel 3.3 dan tabel 3.4 berikutnya supaya P4-P18 mendapatkan subkunci.

3.1.2.2 Proses Enkripsi

Algoritma Blowfish memiliki 16 iterasi, masukannya adalah 64-bit elemen data atau sebut saja "X". Bagi 2 elemen X menjadi 2, yaitu XL=32 bit dan XR=32 bit. Langkah-langkah untuk melakukan proses enkripsi dapat dilihat pada gambar skema di bawah ini :



cat: (1) Skema Blok Enkripsi

Gambar 3.2: Skema Proses Enkripsi Blowfish

Contoh :

Misalkan 64-bit elemen data X = " **! ß ƒ É À i ï h** ", untuk proses enkripsi akan dirubah ke bentuk heksa seperti pada tabel 3.5 :

Tabel 3.5 : Tabel perubahan karakter ke hexa

Elemen X	**!**	**ß**	***f***	**É**	**â**	**i**	**ï**	**h**
Heksa	**21**	**E1**	**9F**	**90**	**83**	**6A**	**8B**	**68**

Kunci yang sudah diEkspansi dengan P-Array pada P1-P18 dapat di asumsikan pada tabel 3.6 :

Tabel 3.6 : Sub Kunci

Putaran	Sub Kunci
P1	243F6A88
P2	243F6A88
P3	85A308D3
P4	85A308D3
P5	13198A2E
P6	13198A2E
P7	3707344
P8	3707344
P9	A4093822
P10	A4093822
P11	299F31D0
P12	299F31D0
P13	082EFA98
P14	082EFA98
P15	EC4E6C89
P16	EC4E6C89
P17	452821E6
P18	452821E6

Tabel 3.7 : Proses Enkripsi

PUTARAN	PROSES ENKRIPSI STRING
P1	XL = XL ⊕ P1 XL = 21E19F90 ⊕ 243F6A88 =5DEF518 F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((5 + DE mod 2^{32}) XOR F5) + 18 mod 2^{32} F(XL) = 2E XR = F(XL) ⊕ XR XR = 2E ⊕ 836A8B68 = 836A8B46
P2	XL = XR(P1) ⊕ P2 XL = 836A8B46 ⊕ 243F6A88 = A755E1CE F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((A7 + 55 mod 2^{32}) XOR E1) + CE mod 2^{32} F(XL) = EB XR = F(XL) ⊕ XL(P1)

	XR = EB ⊕ 5DEF518 = 5DEF5F3
P3	XL = XR(P2) ⊕ P3 XL = 5DEF5F3 ⊕ 85A308D3= 807DFD20 F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((80 + 7D mod 2^{32}) XOR FD) + 20 mod 2^{32} F(XL) = 20 XR = F(XL) ⊕ XL(P2) XR = 20 ⊕ A755E1CE= A755E1EE
P4	XL = XR(P3) ⊕ P4 XL= A755E1EE ⊕ 85A308D3= 22F6E93D F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((22 + F6 mod 2^{32}) XOR E9) + 3D mod 2^{32} F(XL) = 22E XR = F(XL) ⊕ XL(P3) XR = 22E ⊕ 807DFD20 = 807DFF0E
P5	XL = XR (P4) ⊕ P5 XL= 807DFF0E ⊕ 13198A2E = 93647520 F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((93 + 64 mod 2^{32}) XOR 75) + 20 mod 2^{32} F(XL) = A2 XR = F(XL) ⊕ XL(P4) XR = A2 ⊕ 22F6E93D = 22F6E99F
P6	XL = XR(P5) ⊕ P6 XL= 22F6E99F ⊕ 13198A2E = 31EF63B1 F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((31 + EF mod 2^{32}) XOR 63) + B1 mod 2^{32} F(XL) = 1F4 XR = F(XL) ⊕ XL(P5) XR = 1F4 ⊕ 93647520 = 936474D4
P7	XL = XR(P6) ⊕ P7 XL= 936474D4⊕ 3707344= 90140790

	F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((90+ 14mod 2^{32}) XOR 07) + 90 mod 2^{32} F(XL) = 133 XR = F(XL) ⊕ XL(P6) XR = 133⊕ 31EF63B1 = 31EF6282
P8	XL = XR(P7) ⊕ P8 XL= 31EF6282 ⊕ 3707344= 329F11C6 F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((32+ 9F mod 2^{32}) XOR 11) + C6 mod 2^{32} F(XL) =186 XR = F(XL) ⊕ XL(P7) XR = 186 ⊕ 90140790= 90140616
P9	XL = XR (P8) ⊕ P9 XL= 90140616 ⊕ A4093822 = 341D3E34 F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((34 + 1D mod 2^{32}) XOR 3E) + 34 mod 2^{32} F(XL) = A3 XR = F(XL) ⊕ XL(P8) XR = A3 ⊕ 329F11C6= 329F1165
P10	XL = XR (P9) ⊕ P10 XL= 329F1165 ⊕ A4093822 = 96962947 F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((96 + 96mod 2^{32}) XOR 29) + 47 mod 2^{32} F(XL) = 14C XR = F(XL) ⊕ XL(P9) XR = 14C⊕ 341D3E34= 341D3F78
P11	XL = XR (P10) ⊕ P11 XL= 341D3F78 ⊕ 299F31D0= 1D820EA8 F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((1D + 82 mod 2^{32}) XOR 0E) + A8 mod 2^{32} F(XL) = 139

	XR = F(XL) ⊕ XL(P10) XR = 139 ⊕ 96962947 = 9696287E
P12	XL = XR (P11) ⊕ P12 XL= 9696287E ⊕ 299F31D0 = BF0919AE F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((BF + 09 mod 2^{32}) XOR 19) + AE mod 2^{32} F(XL) = 17F XR = F(XL) ⊕ XL(P11) XR = 17F ⊕ 1D820EA8 = 1D820FD7
P13	XL = XR(P12) ⊕ P13 XL= 1D820FD7 ⊕ 082EFA98= 15ACF54F F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((15 + AC mod 2^{32}) XOR F5) + 4F mod 2^{32} F(XL) = 83 XR = F(XL) ⊕ XL(P12) XR = 83 ⊕ BF0919AE = BF09192D
P14	XL = XR (P13)⊕ P14 XL= BF09192D ⊕ 082EFA98= B727E3B5 F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((B7 + 27 mod 2^{32}) XOR E3) + B5 mod 2^{32} F(XL) = F2 XR = F(XL) ⊕ XL(P13) XR = F2 ⊕ 15ACF54F = 15ACF5BD
P15	XL = XR(P14) ⊕ P15 XL = 15ACF5BD ⊕ EC4E6C89 = F9E29934 F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((F9 + E2 mod 2^{32}) XOR 99) + 34 mod 2^{32} F(XL) = 176 XR = F(XL) ⊕ XL(P14)

	XR = 176 ⊕ B727E3B5 = B727E2C3
P16	XL = XR(15) ⊕ P16 XL= B727E2C3 ⊕ EC4E6C89= 5B698E4A F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((5B + 69 mod 2^{32}) XOR 8E) + 4A mod 2^{32} F(XL) = 94 XR = F(XL) ⊕ XL(P15) XR = 94 ⊕ F9E29934 = F9E299A0
P17	P17 = XR (P16) ⊕P17 F9E299A0 ⊕ 452821E6 = BCCAB846
P18	P18 = XL (P16) ⊕P18 5B698E4A ⊕ 452821E6 = 1E41AFAC
Gabungkan kembali P17 dan P18 = 1E41AFAC BCCAB846	

Keterangan :

XL = Inputan X pada Left variabel.

XR = Inputan X pada right variabel.

P(1-18) = Merupakan P- array hasil XOR pada tabel.

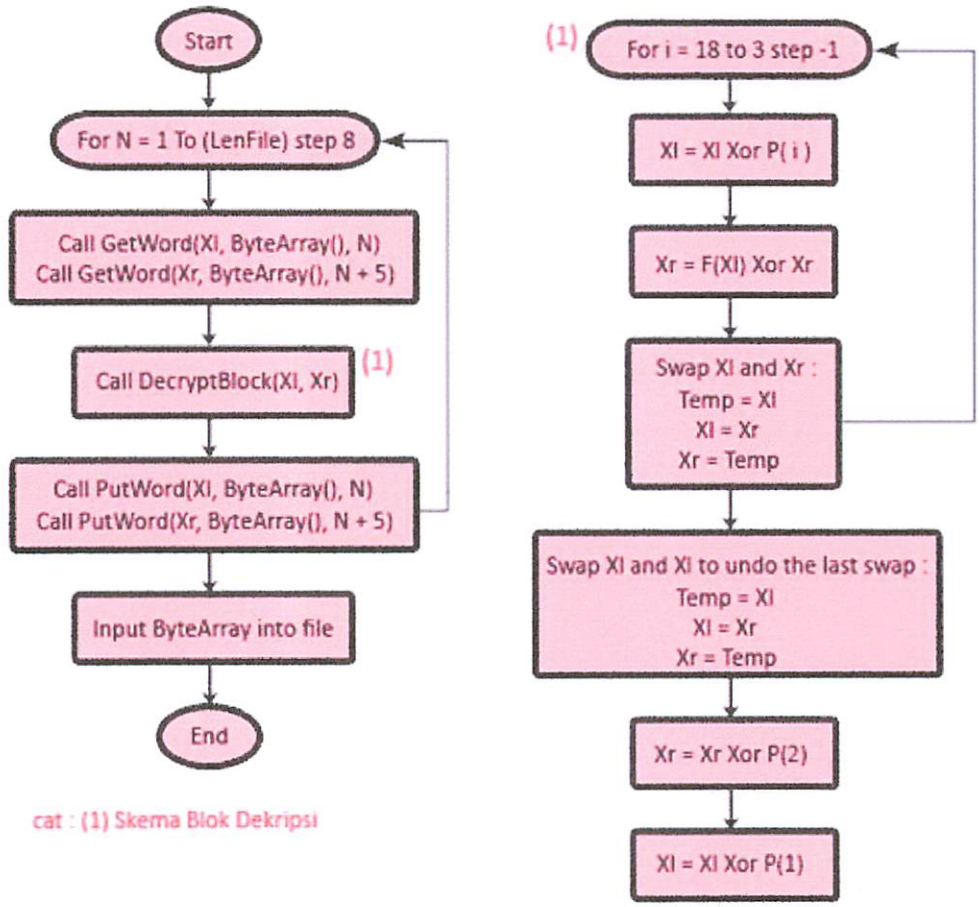
F(XL) = Merupakan fungsi Feistel yang didalamnya terdapat 4 S-BOX.

⊕ = XOR

((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} = Rumus dari fungsi Feistel.

3.1.2.3 Proses Dekripsi

Proses dekripsi pada gambar 6, hampir sama dengan proses enkripsi hanya saja Sub-Kunci P(1) sampai P(18) digunakan dalam urutan terbalik yaitu P(1) menjadi P(18), P(2) menjadi P(17) dan seterusnya. Didalam proses dekripsi cipherteks diubah kembali kedalam bentuk plainteks atau kondisi semula sebelum dienkripsi.



cat : (1) Skema Blok Dekripsi

Gambar 3.3: Skema Proses Dekripsi Blowfish

Contoh :

Proses dekripsi menggunakan data hasil enkripsi yaitu 1E41AFAC BCCAB846 (*Chiperteks*), dan kunci yang telah dibangkitkan pada tabel berikut :

Tabel 3.8 : Kunci dengan pembangkitan P-Array

Putaran	Key String
P18	452821E6
P17	452821E6
P16	EC4E6C89
P15	EC4E6C89
P14	082EFA98
P13	082EFA98
P12	299F31D0
P11	299F31D0
P10	A4093822
P9	A4093822
P8	3707344
P7	3707344
P6	13198A2E
P5	13198A2E
P4	85A308D3
P3	85A308D3
P2	243F6A88
P1	243F6A88

Tabel 3.9 : Proses Dekripsi

PUTARAN	PROSES ENKRIPSI STRING
P18	XL = XL ⊕ P18 XL = 1E41AFAC ⊕ 452821E6 = 5B698E4A F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((5B + 69 mod 2^{32}) XOR 8E) + 4A mod 2^{32} F(XL) = 94 XR = F(XL) ⊕ XR XR = 94 ⊕ BCCAB846 = BCCAB8D2
P17	XL = XR(P18) ⊕ P17 XL = BCCAB8D2 ⊕ 452821E6 = F9E29934 F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((F9 + E2 mod 2^{32}) XOR 99) + 34 mod 2^{32} F(XL) = 176 XR = F(XL) ⊕ XL(P18)

	XR = 176 ⊕ 5B698E4A = 5B698F3C
P16	XL = XR(P17) ⊕ P16 XL = 5B698F3C ⊕ EC4E6C89= B727E3B5 F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((B7 + 27 mod 2^{32}) XOR E3) + B5 mod 2^{32} F(XL) = F2 XR = F(XL) ⊕ XL(P17) XR = F2 ⊕ F9E29934 = F9E299C6
P15	XL = XR(P16) ⊕ P15 XL = F9E299C6 ⊕ EC4E6C89= 15ACF54F F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((15 + AC mod 2^{32}) XOR F5) + 4F mod 2^{32} F(XL) = 83 XR = F(XL) ⊕ XL(P16) XR = 83 ⊕ B727E3B5 = B727E336
P14	XL = XR(P15) ⊕ P14 XL = B727E336 ⊕ 082EFA98= BF0919AE F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((BF + 09 mod 2^{32}) XOR 19) + AE mod 2^{32} F(XL) = 17F XR = F(XL) ⊕ XL(P15) XR = 17F ⊕ 15ACF54F = 15ACF430
P13	XL = XR(P14) ⊕ P13 XL = 15ACF430 ⊕ 082EFA98 = 1D820EA8 F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((1D + 82 mod 2^{32}) XOR 0E + A8 mod 2^{32} F(XL) = 139 XR = F(XL) ⊕ XL(P14) XR = 139 ⊕ BF0919AE = BF091897
P12	XL = XR(P13) ⊕ P12 XL = BF091897 ⊕ 299F31D0 = 96962947

	$F(XL) = ((S1,a + S2,b \bmod 2^{32}) \text{ XOR } S3,c) + S4,d \bmod 2^{32}$ $F(XL) = ((96 + 96 \bmod 2^{32}) \text{ XOR } 29) + 47 \bmod 2^{32}$ $F(XL) = 14C$ $XR = F(XL) \oplus XL(P13)$ $XR = 14C \oplus 1D820EA8 = 1D820FE4$
P11	$XL = XR(P12) \oplus P11$ $XL = 1D820FE4 \oplus 299F31D0 = 341D3E34$ $F(XL) = ((S1,a + S2,b \bmod 2^{32}) \text{ XOR } S3,c) + S4,d \bmod 2^{32}$ $F(XL) = ((34 + 1D \bmod 2^{32}) \text{ XOR } 3E) + 34 \bmod 2^{32}$ $F(XL) = A3$ $XR = F(XL) \oplus XL(P12)$ $XR = A3 \oplus 96962947 = 969629E4$
P10	$XL = XR(P11) \oplus P10$ $XL = 969629E4 \oplus A4093822 = 329F11C6$ $F(XL) = ((S1,a + S2,b \bmod 2^{32}) \text{ XOR } S3,c) + S4,d \bmod 2^{32}$ $F(XL) = ((32 + 9F \bmod 2^{32}) \text{ XOR } 11) + C6 \bmod 2^{32}$ $F(XL) = 186$ $XR = F(XL) \oplus XL(P11)$ $XR = 186 \oplus 341D3E34 = 341D3FB2$
P9	$XL = XR(P10) \oplus P9$ $XL = 341D3FB2 \oplus A4093822 = 90140790$ $F(XL) = ((S1,a + S2,b \bmod 2^{32}) \text{ XOR } S3,c) + S4,d \bmod 2^{32}$ $F(XL) = ((90 + 14 \bmod 2^{32}) \text{ XOR } 07) + 90 \bmod 2^{32}$ $F(XL) = 133$ $XR = F(XL) \oplus XL(P10)$ $XR = 133 \oplus 329F11C6 = 329F10F5$
P8	$XL = XR(P9) \oplus P8$ $XL = 329F10F5 \oplus 3707344 = 31EF63B1$ $F(XL) = ((S1,a + S2,b \bmod 2^{32}) \text{ XOR } S3,c) + S4,d \bmod 2^{32}$ $F(XL) = ((31 + EF \bmod 2^{32}) \text{ XOR } 63) + B1 \bmod 2^{32}$ $F(XL) = 1F4$

	XR = F(XL) ⊕ XL(P9) XR = 1F4 ⊕ 90140790 = 90140664
P7	XL = XR(P8) ⊕ P7 XL = 90140664 ⊕ 3707344= 93647520 F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((93 + 64 mod 2^{32}) XOR 75) + 20 mod 2^{32} F(XL) = A2 XR = F(XL) ⊕ XL(P8) XR = A2 ⊕ 31EF63B1 = 31EF6313
P6	XL = XR(P7) ⊕ P6 XL = 31EF6313 ⊕ 13198A2E = 13198A2E F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((13 + 19 mod 2^{32}) XOR 8A) + 2E mod 2^{32} F(XL) = D4 XR = F(XL) ⊕ XL(P7) XR = D4 ⊕ 93647520 = 936475F4
P5	XL = XR(P6) ⊕ P6 XL = 936475F4 ⊕ 13198A2E = 807DFFDA F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((80 + 7D mod 2^{32}) XOR FF) + DA mod 2^{32} F(XL) = DC XR = F(XL) ⊕ XL(P6) XR = DC ⊕ 13198A2E = 13198AF2
P4	XL = XR(P5) ⊕ P4 XL = 13198AF2 ⊕ 85A308D3= 96BA8221 F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((96 + BA mod 2^{32}) XOR 82) + 21 mod 2^{32} F(XL) = 1F3 XR = F(XL) ⊕ XL(P5) XR = 1F3 ⊕ 807DFFDA = 807DFE29

P3	XL = XR(P4) ⊕ P3 XL = 807DFE29 ⊕ 85A308D3 = 5DEF6FA F(XL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} F(XL) = ((05 + DE mod 2^{32}) XOR F6) + FA mod 2^{32} F(XL) = 10F XR = F(XL) ⊕ XL (P4) XR = 10F ⊕ 96BA8221 = 96BA832E
P2	XL = XR (P3)⊕ P2 XL = 96BA832E ⊕ 243F6A88 = 836A8AF6
P1	XL = XL(P3) ⊕ P1 XL = 5DEF6FA ⊕ 243F6A88= 21E19F44
Gabungkan P1 dan P2 = 21E19F44 836A8AF6	

Keterangan :

XL = Inputan X pada Left variabel.

XR = Inputan X pada right variabel.

P(1-18) = Merupakan P- array hasil XOR pada tabel.

F(XL) = Merupakan fungsi Feistel yang didalamnya terdapat 4 S-BOX.

⊕ = XOR

((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32} = Rumus dari fungsi Feistel.

3.2 Perancangan

Perangkat lunak aplikasi ekripsi dan dekripsi audio video ini dirancang dengan menggunakan bahasa pemrograman *Microsoft Visual Basic 6.0* dengan beberapa komponen *standard* seperti:

1. Command Button, sebagai tombol.
2. Common dialog, digunakan untuk menampilkan dialog-dialog seperti Open Dialod, Save Dialog, Font Dialog dan sebagainya.
3. Text Box, sebagai tempat *input* dan lokasi penyimpanan.
4. Label, untuk menampilkan tulisan.
5. Image atau Picture Box, sebagai komponen untuk menampilkan gambar.

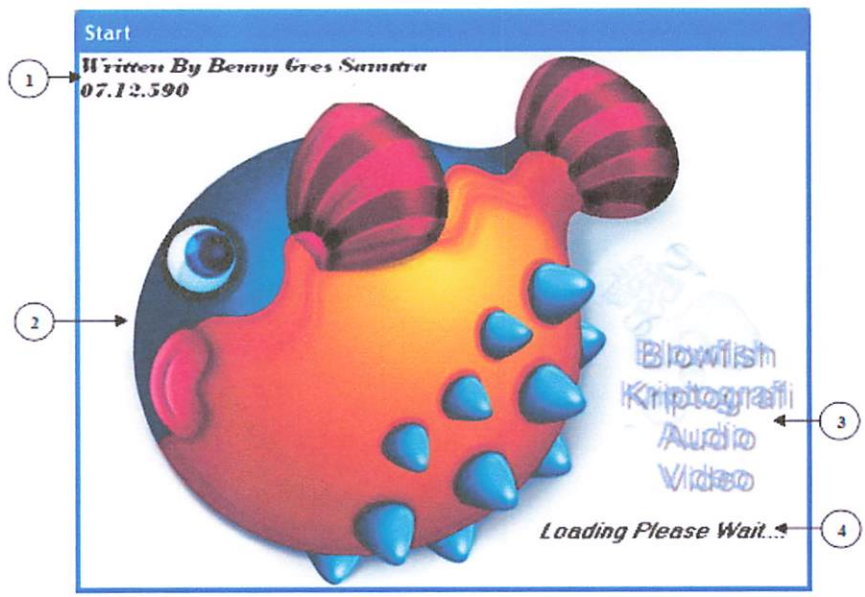
6. Timer, untuk melakukan proses start splash.
7. ImageList, image list berfungsi untuk menentukan ukuran tombol toolbar selain itu image list juga dapat memberikan gambar pada tombol toolbar.
8. Option Button, sebagai komponen untuk memilih.
9. CheckBox, Merupakan tool untuk membuat kontrol pilihan, dimana kontrol ini terpilih jika di klik user.

Perangkat lunak ini memiliki beberapa buah *form*, yaitu:

1. *Form* Start (Splash Screen).
2. *Form* Main.
3. *Form* Enkripsi.
4. *Form* Dekripsi.
5. *Form* Password.
6. *Form* About.

3.2.1 Form Splash Screen

Form Splash Screen, berisi nama perangkat lunak dan nama serta NIM mahasiswa yang membuat perangkat lunak. Rancangan *form* Splash Screen dapat dilihat pada gambar 3.4 :



Gambar 3.4 Rancangan *Form* Start

Keterangan :

1 : *Label* atau nama dan nim pembuat program.

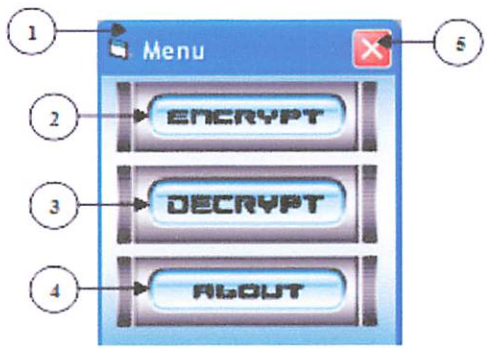
2 : image baground form.

3 : Nama program.

4 : *Label* Proses Loading.

3.2.2 Form Utama

Form Menu menampilkan button enkripsi, dekripsi dan about. Rancangan *form* Menu dapat dilihat pada gambar 3.5 :



Gambar 3.5 Rancangan *Form* Menu

Keterangan :

1 : title bar.

2 : tombol 'Encrypt', untuk menampilkan *form enkripsi*.

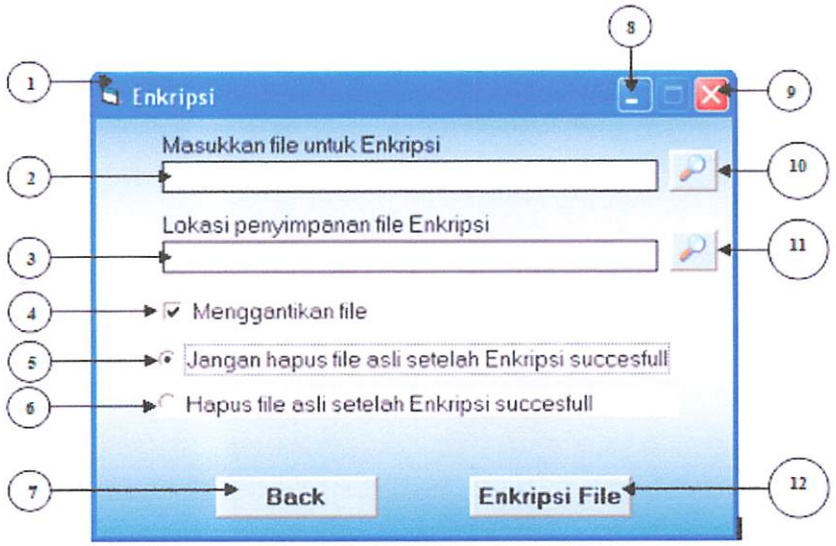
3 : tombol 'decrypt', untuk menampilkan *form dekripsi*.

4 : tombol 'about', untuk menampilkan *form about*.

5 : tombol 'Close', untuk menutup *form*.

3.2.3 Form Enkripsi

Form Enkripsi berfungsi untuk mengenkripsi file audio video. Rancangan *form* Enkripsi dapat dilihat pada gambar 3.6 :



Gambar 3.6 Rancangan *Form* Enkripsi

Keterangan:

1 : title bar.

2 : *textbox,* untuk menampilkan file yang akan dienkripsi.

3 : *textbox,* untuk menampilkan penyimpanan file yang akan dienkripsi.

4 : *checkbox*, untuk menggantikan file yang sudah ada.

5 : *optionbutton*, sebagai komponen untuk memilih.

6 : *optionbutton,* sebagai komponen untuk memilih.

7 : tombol 'back', untuk kembali ke form menu.

8 : tombol 'hide', untuk menyembumyikan *form*.

9 : tombol 'Close', untuk menutup *form*.

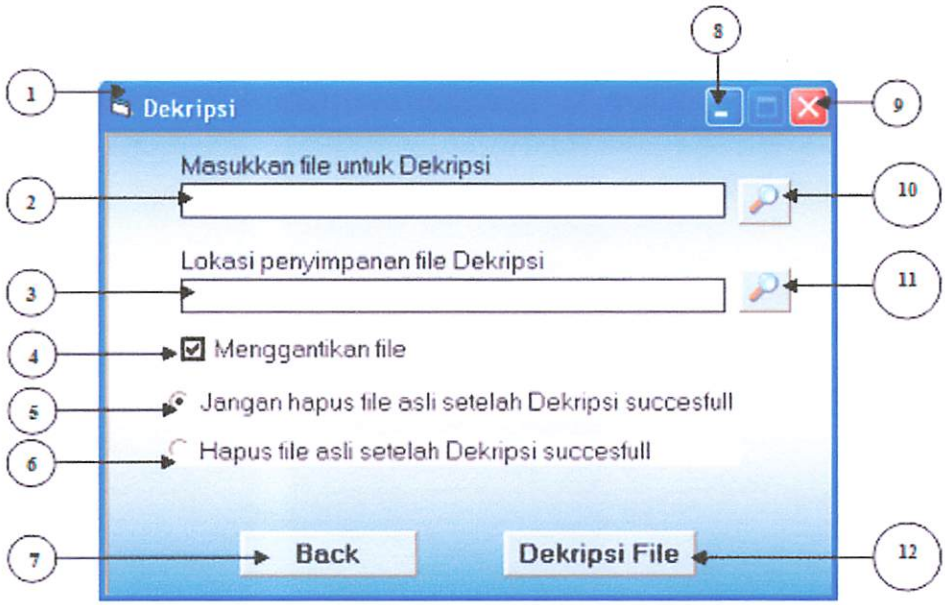
10 : tombol 'button', untuk memasukkan file yang akan dienkripsi.

11 : tombol 'button', untuk menentukan penyimpanan file yang akan dienkripsi.

12 : tombol 'enkripsi file', untuk menampilkan form password dan proses enkripsi.

3.2.4 Form Dekripsi

Form Dekripsi berfungsi sebagai *form* untuk melakukan proses dekripsi. Rancangan *form* dapat dilihat pada gambar 3.7 :



Gambar 3.7 Rancangan *Form* Dekripsi

Keterangan:

1 : title bar.

2 : *textbox,* untuk menampilkan file yang akan didekripsi.

3 : *textbox,* untuk menampilkan penyimpanan file yang akan didekripsi.

4 : *checkbox*, untuk menggantikan file yang sudah ada.

5 : *optionbutton*, sebagai komponen untuk memilih.

6 : *optionbutton,* sebagai komponen untuk memilih.

7 : tombol 'back', untuk kembali ke form menu.

8 : tombol 'hide', untuk menyembumyikan *form*.

9 : tombol 'Close', untuk menutup *form*.

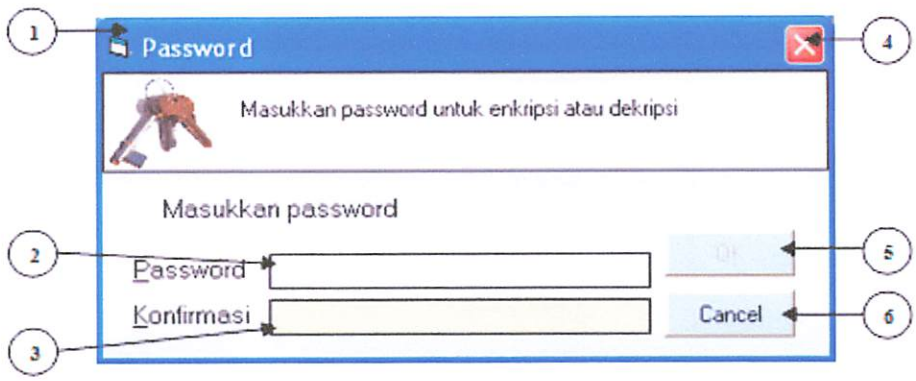
10 : tombol 'button', untuk memasukkan file yang akan didekripsi.

11 : tombol 'button', untuk menentukan penyimpanan file yang akan didekripsi.

12 : tombol 'Dekripsi file', untuk menampilkan form password dan proses Dekripsi.

3.2.5 Form Password

Form Password berfungsi untuk memasukkan password dan konfirmasi yang diinginkan. Rancangan *form* dapat dilihat pada gambar 3.8 :



Gambar 3.8 Rancangan *Form Password*

Keterangan:

1 : title bar.

2 : textbox, untuk memasukkan kunci.

3 : textbox, untuk konfirmasi kunci.

4 : tombol 'Close', untuk menutup *form*.

5 : tombol 'Ok', untuk proses enkripsi.

6 : tombol 'cancel', untuk kembali ke form enkripsi atau dekripsi.

3.2.6 Form About

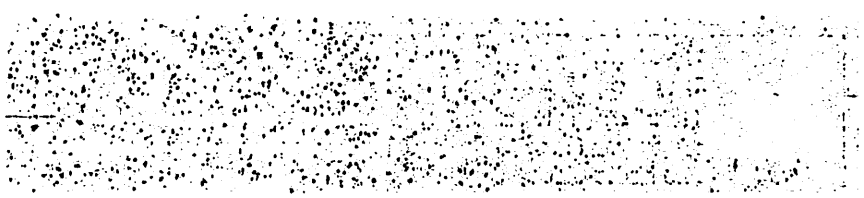
Form About berisi nama pembuat, nim, alamat email, dan sekilas penjelasan tentang software ini. Rancangan *form* About dapat dilihat pada gambar 3.9 :

Gambar 3.9 Rancangan *Form About*

Keterangan:

1 : Image, untuk menampilkan gambar dan untuk tombol close.

2 : Label, untuk menulisskan keterangan pada form about.



3.2.7 Status Enkripsi atau Dekripsi

Status enkripsi atau dekripsi berfungsi untuk pemberitahuan proses enkripsi atau dekripsi selesai di eksekusi.

Gambar 3.10 *Status Enkripsi success full*

Keterangan:

1 : Tombol OK, untuk kembali ke *form* enkripsi.

Gambar 3.11 *Status Dekripsi success full*

Keterangan:

1 : Tombol OK, untuk kembali ke *form* dekripsi.

Gambar 3.12 *Status Password salah!!!!*

Keterangan:

1 : Tombol OK, untuk kembali ke *form Password.*

BAB IV
ALGORITMA DAN IMPLEMENTASI

4.1 Algoritma

Algoritma perancangan perangkat lunak enkripsi dan dekripsi audio video menggunakan algoritma *Blowfih* dibagi menjadi 3 bagian yaitu,

1. Algoritma Proses Ekspansi Kunci.
2. Algoritma Proses Enkripsi.
3. Algoritma Proses Dekripsi.

4.1.1 Algoritma Proses Ekspansi Kunci

Proses ekspansi kunci pada Algoritma *Blowfish* dengan cara Inisialisasi P-array yang pertama dan juga empat S-box, berurutan, dengan string yang telah pasti. String tersebut terdiri dari digit-digit heksadesimal dari phi. Hasil pembentukan kunci berbeda untuk setiap putaran.

"*Pseudocode* proses ekspansi kunci adalah sebagai berikut :

```
Dim i As Long, j As Long, K As Long, dataX As Long, datal As Long, datar As Long,
Key() As Byte, KeyLength As Long
  If (m_KeyValue = New_Value) Then Exit Property
  m_KeyValue = New_Value
  KeyLength = Len(New_Value)
  Key() = StrConv(New_Value, vbFromUnicode)
  j = 0
  For i = 0 To (ROUNDS + 1)
    dataX = 0
    For K = 0 To 3
      Call CopyMem(ByVal VarPtr(dataX) + 1, dataX, 3)
      dataX = (dataX Or Key(j))
      j = j + 1
```

```
        If (j >= KeyLength) Then j = 0
    Next
    m_pBox(i) = m_pBox(i) Xor dataX
Next

datal = 0: datar = 0
For i = 0 To (ROUNDS + 1) Step 2
    Call EncryptBlock(datal, datar)
    m_pBox(i) = datal
    m_pBox(i + 1) = datar
Next
For i = 0 To 3
    For j = 0 To 255 Step 2
        Call EncryptBlock(datal, datar)
        m_sBox(i, j) = datal
        m_sBox(i, j + 1) = datar
    Next
Next
```

4.1.3 Algoritma Proses Enkripsi

Proses enkripsi pada Algoritma *Blowfish* adalah Terdiri dari iterasi fungsi sederhana (Feistel Network) sebanyak 16 kali putaran (iterasi), masukannya adalah 64-bit elemen data X. Setiap putaran terdiri dari permutasi kunci-dependent dan substitusi kunci dan data dependent. Semua operasi adalah penambahan (*addition*) dan XOR pada variabel 32-bit. Operasi tambahan lainnya hanyalah empat penelusuran tabel array berindeks untuk setiap putaran.

Algoritma proses enkripsi adalah sebagai berikut :

"*Pseudocode* Proses Ekrip *block*:

```
j = 0
    For i = 0 To (ROUNDS \ 2 - 1)
        Xl = Xl Xor m_pBox(j)
        Xr = Xr Xor f(Xl)
```

```
    Xr = Xr Xor m_pBox(j + 1)
    Xl = Xl Xor f(Xr)
    j = j + 2
  Next
  Temp = Xr
  Xr = Xl Xor m_pBox(ROUNDS)
  Xl = Temp Xor m_pBox(ROUNDS + 1)
```

"*Pseudocode* Proses enkripsi byte :

```
If (Len(Key) > 0) Then Me.Key = Key
  OrigLen = UBound(byteArray) + 1
  CipherLen = OrigLen + 12
  If (CipherLen Mod 8 <> 0) Then CipherLen = CipherLen + 8 - (CipherLen Mod 8)
  ReDim Preserve byteArray(CipherLen - 1)
  Call CopyMem(byteArray(12), byteArray(0), OrigLen)
  Call CopyMem(byteArray(8), OrigLen, 4)
  Call Randomize
  Call CopyMem(byteArray(0), CLng(2147483647 * Rnd), 4)
  Call CopyMem(byteArray(4), CLng(2147483647 * Rnd), 4)
  For Offset = 0 To (CipherLen - 1) Step 8
    Call GetWord(LeftWord, byteArray(), Offset)
    Call GetWord(RightWord, byteArray(), Offset + 4)
    LeftWord = LeftWord Xor CipherLeft
    RightWord = RightWord Xor CipherRight
    Call EncryptBlock(LeftWord, RightWord)
    Call PutWord(LeftWord, byteArray(), Offset)
    Call PutWord(RightWord, byteArray(), Offset + 4)
    CipherLeft = LeftWord
    CipherRight = RightWord
```

"*Pseudocode* Proses enkripsi byte string :

```
Dim byteArray() As Byte
  byteArray() = StrConv(Text, vbFromUnicode)
  Call EncryptByte(byteArray(), Key)
  EncryptString = StrConv(byteArray(), vbUnicode)
  If OutputInHex = True Then EncryptString = EnHex(EncryptString)
```

"*Pseudocode* Proses enkripsi *Hexa* :

```
Dim iCount As Double, sTemp As String
  Reset
  For iCount = 1 To Len(Data)
    sTemp = Hex$(Asc(Mid$(Data, iCount, 1)))
    If Len(sTemp) < 2 Then sTemp = "0" & sTemp
    Append sTemp
```

4.1.4 Algoritma Proses Dekripsi

Proses dekripsi pada Algoritma *Blowfish* hampir sama dengan proses enkripsi hanya saja Sub-Kunci P(1) sampai P(18) digunakan dalam urutan terbalik yaitu P(1) menjadi P(18), P(2) menjadi P(17) dan seterusnya. Didalam proses dekripsi cipherteks diubah kembali kedalam bentuk plainteks atau kondisi semula sebelum dienkripsi.

"Pseudocode Proses dekrip *block* :

```
K = Xr
  Xr = Xl Xor m_pBox(ROUNDS + 1)
  Xl = K Xor m_pBox(ROUNDS)
  j = ROUNDS - 2
  For i = 0 To (ROUNDS \ 2 - 1)
    Xl = Xl Xor f(Xr)
    Xr = Xr Xor m_pBox(j + 1)
    Xr = Xr Xor f(Xl)
    Xl = Xl Xor m_pBox(j)
    j = j – 2
```

"Pseudocode Proses dekrip *byte* :

```
If (Len(Key) > 0) Then Me.Key = Key
  CipherLen = UBound(byteArray) + 1
  For Offset = 0 To (CipherLen - 1) Step 8
    Call GetWord(LeftWord, byteArray(), Offset)
    Call GetWord(RightWord, byteArray(), Offset + 4)
    Call DecryptBlock(LeftWord, RightWord)
    LeftWord = LeftWord Xor CipherLeft
    RightWord = RightWord Xor CipherRight
    Call GetWord(CipherLeft, byteArray(), Offset)
    Call GetWord(CipherRight, byteArray(), Offset + 4)
    Call PutWord(LeftWord, byteArray(), Offset)
    Call PutWord(RightWord, byteArray(), Offset + 4)
```

"Pseudocode Proses dekrip *string* :

```
Dim byteArray() As Byte
  If IsTextInHex = True Then Text = DeHex(Text)
  byteArray() = StrConv(Text, vbFromUnicode)
  Call DecryptByte(byteArray(), Key)
  DecryptString = StrConv(byteArray(), vbUnicode)
```

"Pseudocode Proses dekrip *Hexa* :

```
Dim iCount As Double
  Reset
  For iCount = 1 To Len(Data) Step 2
    Append Chr$(Val("&H" & Mid$(Data, iCount, 2)))
  Next
  DeHex = GData
  Reset
```

4.2 Implementasi Dan Pengujian sistem

4.2.1 Lingkungan Implementasi

Lingkungan implementasi meliputi proses-proses yang terdapat dalam aplikasi, bagan Input desain, proses desain, output yang sesuai dengan target yang diharapkan.

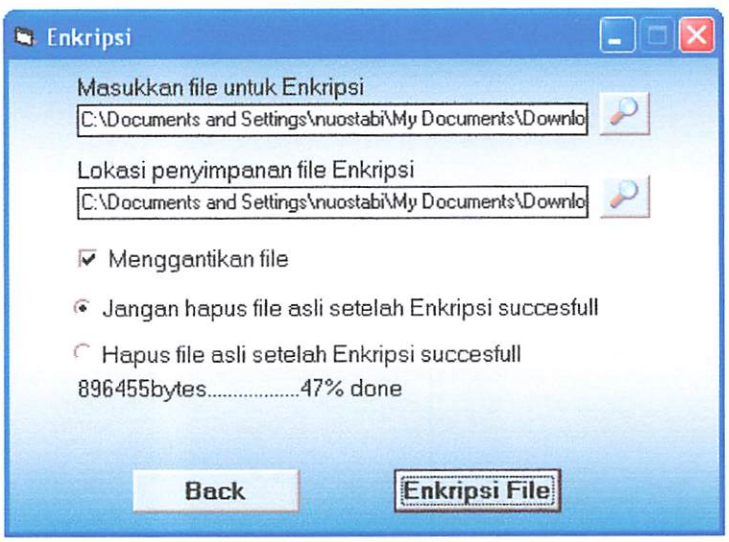
4.2.1.1 *Form* Utama

Form Menu menampilkan button enkripsi, dekripsi dan about, Dari *form* utama ini kita dapat mengacu ke *form* lainnya untuk menjalankan aplikasi ini sesuai harapan.

Gambar 4.1. *Form* Menu

4.2.1.2 *Form* Enkripsi

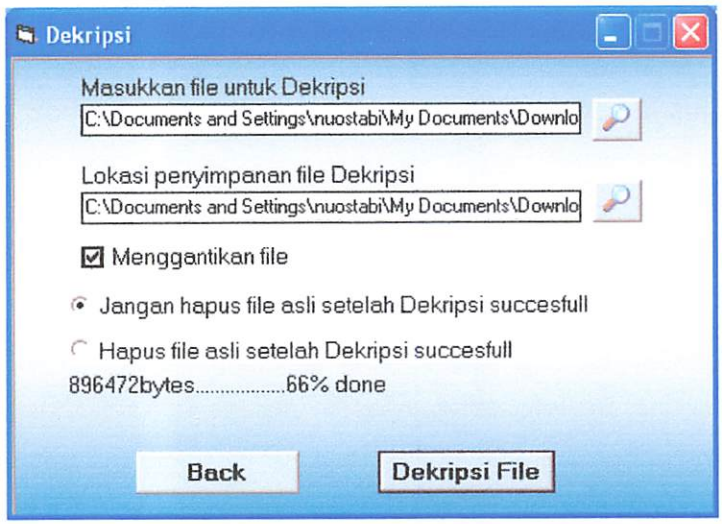
Form Enkripsi ini digunakan untuk memasukkan file enkripsi, menentukan tempat penyimpanan data enkripsi dan memproses enkripsi.



Gambar 4.2. *Form* Enkripsi

4.2.1.3 *Form* Dekripsi

Form Dekripsi ini digunakan untuk memasukkan file dekripsi, menentukan tempat penyimpanan data dekripsi dan memproses dekripsi.



Gambar 4.3. *Form* Dekripsi

4.2.1.4 *Form Password*

Form Password berfungsi untuk memasukkan password dan konfirmasi yang diinginkan.

Gambar 4.4. *Form Password*

4.2.1.5 *Status* Enkripsi atau Dekripsi

Status enkripsi atau dekripsi berfungsi untuk pemberitahuan proses enkripsi atau dekripsi selesai di eksekusi.

Gambar 4.5. *Statutus Enkripsi success full*

Gambar 4.6. *Statutus Dekripsi success full*

Gambar 4.7. *Status Password salah!!!!*

Keterangan:

1 : Tombol OK, untuk kembali ke *form Password.*

4.3 Pengujian Sistem

Untuk melakukan pengujian aplikasi ini menggunakan Laptop dengan spesifikasi :

1. Processor: Intel(R) Core(TM) i7 CPU Q 740 @ 1.73GHz (8 CPUs)
2. Memori : 1974 MB RAM
3. Hardisk : 500 GB
4. Sistem Operasi : Windows XP Profesional SP2
5. Microsoft Visual Basic 6.0

4.3.1 Pengujian Aplikasi Program

Pada bagian ini dilakukan pengujian aplikasi untuk mengenkripsi file dan setelah proses enkripsi selesai dilakukan akan dilihat hasilnya kemudian dilakukan pengujian apakah file tersebut bisa dikembalikan seperti semula. Pengujian dilakukan pada beberapa file dengan ekstensi berbeda, yaitu ekstensi-ekstensi file yang didukung oleh aplikasi ini kemudian akan dilihat perubahan dari setiap ekstensi serta hasil enkripsi dari masingmasing file. Pengujian dilakukan pada ukuran file yang berbeda dan membandingkan waktu proses untuk masing-masing proses enkripsi/dekripsi. Ekstensi file yang dapat dienkripsi dan perubahan ekstensinya dapat dilihat pada Tabel 1.

Tabel 4.1 Ekstensi File yang didukung oleh Aplikasi enkripsi dan dekripsi audio video dan Perubahan Ekstensi setelah Proses Enkripsi

Jenis file	Ekstensi	Ekstensi hasil enkripsi
	mp3	blw
	wav	blw
Suara/audio	flac	blw
	aac	blw
	ogg	blw
	amr	blw
	avi	blw
	swf	blw
	flv	blw
	mkv	blw
Video	mp4	blw
	mpg	blw
	3gp	blw
	dat	blw
	wmv	blw

4.3.2 Pengujian terhadap Ukuran File dan Waktu Proses

Hasil pengujian beberpa file pada tabel 4.1 dapat dirangkum dalam tabel berikut yang menunjukkan ukuran dan waktu proses untuk masing-masing file pada tiap proses enkripsi dan dekripsi.

Table 4.2 Hasil Proses Enkripsi beberapa File yang berekstensi berbeda

No	Nama File Plainteks	Ukuran file plainteks (byte)	Ukuran file cipherteks (byte)	Waktu proses enkripsi (detik)	Kecepatan Rata-rata, Ukuran file/detik
1	Bruno mars.mp3	4315878	4315958	1.13	3819361.062
2	horor.wav	1367144	1367222	0.65	2103298.462
3	Jeff Buckley.flac	36870606	36870686	6.28	5871115.605
4	Alexandre Pires.aac	1393458	1393534	0.78	1786484.615
5	Frank Aguiar.ogg	6243596	6243670	1.53	4080781.699
6	punk rock jawa.amr	958950	959030	0.54	1775833.333
7	freestyler1.avi	77240274	77240350	9.51	8122005.678
8	videoplayback.swf	65920457	65920534	8.13	8108297.294
9	Travis Barker vs Adrian Joung.flv	15995593	15995670	3.17	5045928.391
10	nndeserterdvr.mkv	366207454	366207534	62.32	5876242.843
11	Avril Lavigne.mpg	40230764	40,230,838	7.59	5300495.916
12	Super Junior.3gp	19373831	19373910	3.66	5293396.448
13	C.Ronaldo-Vs-KaKa-Vs-Messi.wmv	3447517	3447598	0.99	3482340.404
14	A.dat	579956204	579956278	114.47	5066447.139
15	THE LAZY SONG.mp4	21603944	21604022	4.14	5218343.961
	Kecepatan rata-rata eksekusi				4730024.857

Table 4.3 Hasil Proses Dekripsi beberapa File yang berekstensi *blw*

No	Nama File Chiperteks	Ukuran file cipherteks (byte)	Ukuran file plainteks (byte)	Waktu proses dekripsi (detik)	Kecepatan Rata-rata, Ukuran file/detik
1	Bruno mars.mp3.blw	4315958	4315878	1.53	2820887.582
2	horor.wav.blw	1367222	1367144	1.18	1158662.712
3	Jeff Buckley.flac.blw	36870686	36870606	8.02	4597342.394
4	Alexandre Pires.aac.blw	1393534	1393458	1.04	133996.5385
5	Frank Aguiar.ogg.blw	6243670	6243596	1.77	3527497.175
6	punk rock jawa.amr.blw	959030	958950	0.8	1198787.5
7	freestyler1.avi.blw	77240350	77240274	14.11	5474156.627
8	videoplayback.swf.blw	65920534	65920457	14.49	8240066.75
9	Travis Barker vs Adrian Joung.flv.blw	15995670	15995593	4.12	3882444.175
10	nndeserterdvr.mkv.blw	366207534	366207454	67.76	5404479.545
11	Avril Lavigne.mpg.blw	40,230,838	40230764	8.64	4656346.991
12	Super Junior.3gp	19373910	19373831	4.15	4668412.048
13	C.Ronaldo-Vs-KaKa-Vs-Messi.wmv.blw	3447598	3447517	1.37	2516494.891
14	A.dat.blw	579956278	579956204	130.24	4452981.25
15	THE LAZY SONG.mp4.blw	21604022	21603944	4.74	4557810.549
	Kecepatan rata-rata eksekusi				3819357.782

Dari hasil Tabel 4.3 dapat dilihat bahwa besarnya ukuran file mempengaruhi waktu atau lamanya proses enkripsi atau dekripsi. Dari kelimabelas ekstensi yang dapat diproses oleh aplikasi ini dan dilakukan percobaan pada sebuah file yang mewakili masing-masing ekstensi, terlihat bahwa semakin besar ukuran file, maka semakin banyak waktu yang diperlukan untuk proses tersebut. Waktu proses untuk enkripsi dan dekripsi untuk masing-masing file sedikit berbeda, diakibatkan ukuran antara file plainteks dan file *cipherteks*nya sedikit berbeda. Ukuran file *cipherteks* sedikit lebih besar dibandingkan *plainteks*nya. Perbedaan ukuran antara plainteks dan cipherteks ini mengakibatkan waktu proses yang diperlukan untuk dekripsi sedikit lebih besar dibandingkan untuk proses enkripsi. Penambahan jumlah byte dalam file *cipherteks* diakibatkan penambahan beberapa byte untuk mode enkripsi CBC (*Cipher Block Chaining*) pada proses enkripsi. Tetapi ketika cipherteks didekripsi kembali, ukuran file kembali seperti plainteksnya.

4.3.3 Pengujian Terhadap Ukuran File dan Maksimal File

Untuk pengujian ukuran file dan batas maksimal file dapat dilihat pada tabel dibawah ini :

Tabel 4.4. Pengujian Terhadap Ukuran File dan Maksimal File

No	File Plainteks	Ukuran File Plainteks (byte)	File Chperteks	Ukuran File Chiperteks (byte)	Hasil Dekripsi	Ukuran File Dekripsi (byte)
1	A.dat	579956204	A.dat.blw	579956278	A.dat	579956204
2	Selendang Rocker.mp4	608242274	Selendang Rocker.mp4.blw	608242349	Selendang Rocker.mp4	608242274
3	pocong jl blora.avi	675138822	pocong jl blora.avi.blw	0	pocong jl blora.avi.blw	0
4	Fast52011.mkv	734050045	Fast52011.mkv.blw	0	Fast52011.mkv.blw	0

Pada tabel 4.4 terdapat file chiperteks dan hasil dekripsi 0 (nol), itu dikarenakan aplikasi ini mempunyai keterbatasan untuk file yang akan didekripsi, file yang bisa didekripsi kurang lebih sekitar 600 Mb, jadi file yag dienkripsi melebihi dari 600 Mb maka proses ekripsi akan gagal atau eror. dan pesan erornya seperti pada gambar 4.8 :

Gambar 4.8. *Statutus proses enkripsi error*

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah menyelesaikan Rancang Bangun Aplikasi Enkripsi dan Dekripsi Audio Video ini, penulis menarik kesimpulan sebagai berikut :

1. Aplikasi ini telah berhasil dibangun dan dapat berfungsi sesuai tujuan, yaitu mengamankan data ataupun informasi yang berupa file (*plainteks*) dengan mengacak file tersebut sehingga tidak dapat dibaca atau dimengerti.
2. Aplikasi ini juga telah berhasil mengembalikan file yang telah diacak tersebut (*cipherteks*) seperti semula dengan menggunakan kunci yang sama sewaktu enkripsi.
3. Perangkat lunak ini dapat mengamankan data atau file audio video yang hanya bisa mengenkripsi file audio video dengan ukuran maksimal 600 MB.
4. Kecepatan rata-rata proses enkripsi pada aplikasi ini mencapai 4730024.857 byte per detik, dan pada dekripsi mencapai 3819357.782 byte per detik.
5. Terjadi penambahan byte pada file hasil enkripsi yang mengakibatkan ukuran file *chiperteks* dan file *plainteks* sedikit berbeda, tetapi ketika file enkripsi dikembalikan (didekripsi) ukuran file akan kembali seperti ukuran file plainteksnya.

5.2 Saran

Algoritma *Blowfish* ini bisa dikembangkan untuk keamanan data dan jaringan. Dalam pengembangannya dapat menggunakan metode penyandian atau komponen dan bahasa pemrograman lain dalam menjaga keamanan file audio video yang lebih optimal.

DAFTAR PUSTAKA

Buku, E-Book, Paper, Tesis :

[1] Ariyus, Dony, (2006). *Kriptografi Keamanan Data Dan Komunikasi.* STIMIK AMIKOM YOGYAKATA.

[2] Ariyus, Dony, (2008). *Penghantar Ilmu Kriptografi.* STIMIK AMIKOM YOGYAKATA.

[3] Kromodimoeljo, Sentot, (2010), Teori dan Aplikasi Kriptografi, SPK IT Consulting.

[4] Munir, Rinaldi, (2007). *Bahan Kuliah IF5054 Kriptografi.* Program Studi Teknik Informatika, Institut Teknologi Bandung.

[5] Sadeli, Muhammad. (2010), *Kumpulan Proyek Visual Basic 6.0 maxikom.*

[6] Schneier, Bruce, (1996), *Applied Cryptography, Second Edition*, John Wiley & Son, New York.

[7] Warso, Hendro. (2008). Dasar Pemrograman *Visual Basic 6.0.*

URL :

[8] *http://www.schneier.com/blog/archives/2005/04/blowfish_on_24.html* - 2 Februari 2012

[9] *http://www.i-bego.com/visual-basic/algoritma-blowfish-cbc-t2755.html* - 8 Februari 2012

[10] *http://zeromin0.blogspot.com/2011/07/implementasi-algoritma-blowfish-dengan.html#axzz1lu1Rrehv* – 17 Februari 2012

[11] *http://ritasari-algoritma-ritasari.blogspot.com/2009/01/algoritma-blowfish.html* - 20 Januari 2012

[12] *http://amirmahmud2008.blogspot.com/2008/12/tips-and-triks-visual-basic-60.html - 17 Januari 2012*

[13] *http://dunovteck.wordpress.com/2011/06/09/pengertian-audio-dan-video-codec - 9 Januari 2012*

Lampiran-Lampiran

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

BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

NAMA : BENNY GRES SANUTRA
NIM : 07.12.590
JURUSAN : Teknik Elektro S-1
KONSENTRASI : Teknik Komputer dan Informatika
MASA BIMBINGAN : 6 Desember 2011 s/d 6 Juni 2012
JUDUL : **RANCANG BANGUN APLIKASI ENKRIPSI DAN DEKRIPSI AUDIO VIDEO MENGGUNAKAN ALGORITMA BLOWFISH BERBASIS VISUAL BASIC**

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

Hari : Rabu
Tanggal : 22 Februari 2012
Dengan Nilai : 84,35 (A)

PANITIA UJIAN SKRIPSI

Ketua Majelis Penguji,

Ir. Yusuf Ismail Nakhoda, MT
NIP.Y.1018800189

Sekretaris Majelis Penguji,

Dr. Eng. Aryuanto S, ST, MT
NIP.Y.1030800417

ANGGOTA PENGUJI

Dosen Penguji I

M. Ibrahim Ashari, ST, MT
NIP.P.1030100358

Dosen Penguji II

Sandy Nataly Mantja, S.Kom
NIP.P.1030800418

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

FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata Satu (S-1) Jurusan Teknik Elektro Konsentrasi Teknik Komputer dan Informatika, maka perlu adanya perbaikan skripsi untuk mahasiswa:

NAMA : BENNY GRES SANUTRA
NIM : 07.12.590
JURUSAN : Teknik Elektro S-1
KONSENTRASI : Teknik Komputer dan Informatika
MASA BIMBINGAN : 6 Desember 2011 s/d 6 Juni 2012
JUDUL : **RANCANG BANGUN APLIKASI ENKRIPSI DAN DEKRIPSI AUDIO VIDEO MENGGUNAKAN ALGORITMA BLOWFISH BERBASIS VISUAL BASIC**

No	Tanggal	Uraian	Paraf
1	Penguji I 27 - 02 - 2012	Tambahkan keterangan pada gambar dan tabel.	
2	Penguji II 27 - 02 - 2012	Latar Belakang diberi spasi, rumusan masalah, tujuan dan *metodologi penelitian.*	
		Catatan kaki pada bab 2 mengacu pada daftar pustaka	
		Saran	
		Daftar pustaka	

Disetujui,

Dosen Penguji I

M. Ibrahim Ashari, ST, MT
NIP.P.1030100358

Dosen Penguji II

Sandy Nataly Mantja, S.Kom
NIP.P.1030800418

Mengetahui,

Dosen Pembimbing I

Sotyohadi, ST
NIP.Y.1039700309

Dosen Pembimbing II

Michael Ardita, ST, MT
NIP.P.1031000434

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

FORMULIR BIMBINGAN SKRIPSI

Nama : BENNY GRES SANUTRA
Nim : 07.12.590
Masa Bimbingan : 6 DESEMBER 2011 s/d 6 JUNI 2012
Judul Skripsi : **RANCANG BANGUN APLIKASI ENKRIPSI DAN DEKRIPSI AUDIO VIDEO MENGGUNAKAN ALGORITMA *BLOWFISH* BERBASIS *VISUAL BASIC***

No.	Tanggal	Uraian	Paraf Pembimbing
1.	2 - 2 - 2012	Bimbingan Laporan Skripsi, Bab : I, 2, 3	
2.	8 - 2 - 2012	Revisi Laporan skripsi Bab 1, 2, 3	
3.	9 - 2 - 2012	Bimbingan Laporan skripsi Bab 4	
4.	17 - 2 - 2012	Revisi Laporan Skripsi Bab 4 . Demo Program	
5.	18 - 2 - 2012	Bimbingan Proposal Seminar Hasil	
6.			
7.			
8.			
9.			
10.			

Malang, 20 Februari 2012
Dosen Pembimbing I

Sotyohadi ,ST
NIP.Y.103.970.0309

Form S-4b

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

FORMULIR BIMBINGAN SKRIPSI

Nama : BENNY GRES SANUTRA
Nim : 07.12.590
Masa Bimbingan : 6 DESEMBER 2011 s/d 6 JUNI 2012
Judul Skripsi : **RANCANG BANGUN APLIKASI ENKRIPSI DAN DEKRIPSI AUDIO VIDEO MENGGUNAKAN ALGORITMA *BLOWFISH* BERBASIS *VISUAL BASIC***

No.	Tanggal	Uraian	Paraf Pembimbing
1.	2-2-2012	Demo program + Laporan skripsi Bab:1,2,3	
2.	4-2-2012	Revisi program Revisi Laporan skripsi Bab : 1,2,3	
3.	17-2-2012	Acc Proposal Seminar hasil	
4.	18-2-2012	Acc Laporan Skripsi Bab : 4	
5.	19-2-2012	Acc semua Laporan skripsi BAB 1,2,3,4,5	
6.			
7.			
8.			
9.			
10.			

Malang, 20 Februari 2012
Dosen Pembimbing II

Michael Adita ,ST,MT
NIP.P.103.100.0434

Form S-4b

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi anjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika / T.Infokom, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : Bunyi gress S.
NIM : 0715590
Perbaikan meliputi :

tambahkan halaman pengantar pd gbr dan ta

Malang,

(M. Ibrahim Ashari, ST, MT)

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan U[illegible]n Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elekt[illegible]nika / T.Infokom, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : BENNY GRES SANTOSA
NIM : 07 12 590
Perbaikan meliputi :

1. LATAR BELAKANG : SPASINYA, RUMUSAN MASALAH, TUJUAN, METODOLOGI PENELITIAN
2. SPASI
3. CATAN KAKI PADA BAB II UTK MEMBACA KE DAFTAR PUSTAKA MOHON BERAPA
4. ~~[illegible]~~ DAFTAR PUSTAKA
5.

Malang, 22.2.12

(SANDY NATALY)

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG

INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 6 Desember 2011

Nomor : ITN- 885/I.TA/2/11
Lampiran : -
Perihal : BIMBINGAN SKRIPSI

Kepada : Yth. Sdr/I. **SOTYOHADI, ST**
Dosen Institut Teknologi Nasional Malang

Dosen Pembimbing
Jurusan Teknik Elektro S-1
di
Malang

Dengan hormat
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi
Untuk Mahasiswa :

Nama	: BENNY GRES. S
Nim	: 0712590
Fakultas	: Teknologi Industri
Jurusan	: Teknik Elektro S-1
Konsentrasi	: Teknik **Komputer & Informatika**

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

6 Desember 2011 s/d 6 Juni 2012

Sebagai satu syarat untuk menempuh ujian Sarjana Teknik,
Jurusan Teknik Elektro S-1,
Demikian atas perhatian serta bantuannya kami sampaikan terima kasih

Ketua Jurusan
Teknik Elektro S-1

Ir. Yusuf Ismail Nakhoda, MT
Nip. Y.1018800189

Tembusan Kepada Yth :

1. Mahasiswa Yang Berangkutan
2. Arsip

Form. S 4a

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 6 Desember 2011

Nomor : ITN- 886/I.TA/2/11
Lampiran : -
Perihal : BIMBINGAN SKRIPSI

Kepada : Yth. Sdr/I. **MICHAEL ARDITA, ST, MT**
Dosen Institut Teknologi Nasional Malang

Dosen Pembimbing
Jurusan Teknik Elektro S-1
di
Malang

Dengan hormat
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi
Untuk Mahasiswa :

Nama	: BENNY GRES. S
Nim	: 0712590
Fakultas	: Teknologi Industri
Jurusan	: Teknik Elektro S-1
Konsentrasi	: Teknik **Komputer & Informatika**

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

6 Desember 2011 s/d 6 Juni 2012

Sebagai satu syarat untuk menempuh ujian Sarjana Teknik,
Jurusan Teknik Elektro S-1,
Demikian atas perhatian serta bantuannya kami sampaikan terima kasih

Ketua Jurusan
Teknik Elektro S-1

Ir. Yusuf Ismail Nakhoda, MT
Nip. Y.1018800189

Tembusan Kepada Yth :
1. Mahasiswa Yang Berangkutan
2. Arsip

Form. S 4a

Institut Teknologi Nasional Malang
Fakultas Teknologi Industri
Jurusan Teknik Elektro S-1

BERITA ACARA SEMINAR PROPOSAL SKRIPSI
JURUSAN TEKNIK ELEKTRO S-1

Konsentrasi : Teknik Energi Listrik/Teknik Elektronika/ Teknik Komputer & Informatika*)

1.	Nama Mahasiswa: Benny Gres Sanutra			Nim: 07125 90
2.	Keterangan Pelaksanaan	Tanggal: 5 - 12 - 2011	Waktu:	Tempat Ruang:

3. Spesifikasi Judul (berilah tanda silang)**)

a. Sistem Tenaga Elektrik	e. Elektronika & Komponen
b. Energi & Konversi Energi	* Elektronika Digital & Komputer
c. Tegangan Tinggi & Pengukuran	g. Elektronika Komunikasi
d. Sistem Kendali Industri	h. lainnya .....

4. Judul Proposal yang diseminarkan Mahasiswa: Rancang Bangun Aplikasi Enkripsi dan Dekripsi Audio Vidio menggunakan Algoritma Blowfish Berbasis Visual Basic

5. Perubahan Judul yang diusulkan oleh Kelompok Dosen Keahlian: .....

6. Catatan: .....

7. Catatan: .....

Persetujuan Judul Skripsi

Disetujui, Dosen Keahlian I	Disetujui, Dosen Keahlian II	Disetujui, Dosen Keahlian III
SAODAH	M Hoirul A	SONNY PRASETIO

Mengetahui, Ketua Jurusan.	Disetujui, Calon Dosen Pembimbing ybs	
	Pembimbing I	Pembimbing II
Ir. Yusuf Ismail Nakhoda,MT NIP. Y. 1018800189		

Perhatian:

1. Keterangan: *) Coret yang tidak perlu
   **) dilingkari a, b, c, ...atau g sesuai bidang keahlian

Form S-3c

INSTITUT TEKNOLOGI NASIONAL

Jln. Raya Karanglo, Km. 2 Malang

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa/i:

Nama : Benny Gres Sanutra

NIM : 07.12.590

Semester : IX (Sembilan)

Jurusan : Teknik Elektro S-1

Konsentrasi : Teknik Komputer dan Informatika

Dengan ini menyatakan bersedia/ ~~tidak bersedia~~*) Membimbing skripsi dari mahasiswa tersebut, dengan judul:

"Rancang Bangun Aplikasi Enkripsi Dan Dekripsi Audio Video Menggunakan Algoritma *Blowfish* Berbasis *Visual Basic*"

Demikian surat pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Malang, 21 November 2011

Kami yang membuat pernyataan

Sotyohadi, ST.
NIP.Y.1039700309

Catatan :

Setelah disetujui agar formulir ini

Diserahkan mahasiswa/I yang bersangkutan

Kepada Jurusan untuk diproses lebih lanjut.

*) Coret yang tidak perlu

Form S-3b

INSTITUT TEKNOLOGI NASIONAL

Jln. Raya Karanglo, Km. 2 Malang

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa/i:

Nama : Benny Gres Sanutra

NIM : 07.12.590

Semester : IX (Sembilan)

Jurusan : Teknik Elektro S-1

Konsentrasi : Teknik Komputer dan Informatika

Dengan ini menyatakan bersedia/~~tidak bersedia~~*) Membimbing skripsi dari mahasiswa tersebut, dengan judul:

"Rancang Bangun Aplikasi Enkripsi Dan Dekripsi Audio Video Menggunakan Algoritma *Blowfish* Berbasis *Visual Basic*"

Demikian surat pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Malang, Oktober 2011

Kami yang membuat pernyataan

Michael Adita, ST, MT.
NIP.P.1031000434

Catatan :

Setelah disetujui agar formulir ini

Diserahkan mahasiswa/I yang bersangkutan

Kepada Jurusan untuk diproses lebih lanjut.

*) Coret yang tidak perlu

Form S-3b

Institut Teknologi Nasional Malang
Fakultas Teknologi Industri
Jurusan Teknik Elektro S-1

LEMBAR PENGAJUAN JUDUL SKRIPSI
JURUSAN TEKNIK ELEKTRO S-1

Konsentrasi : Teknik Energi Listrik / Teknik Elektronika /Teknik Komputer & Informatika /Teknik Komputer / Teknik Telekomunikasi*)

1.	Nama Mahasiswa: Benny Gres Sanutra			Nim: 07.12.590
2.	Waktu Pengajuan	Tanggal:	Bulan:	Tahun:
		17	November	2011

Spesifikasi Judul (berilah tanda silang)**)**

3.
- a. Sistem Tenaga Elektrik
- b. Energi & Konversi Energi
- c. Tegangan Tinggi & Pengukuran
- d. Sistem Kendali Industri
- e. Elektronika & Komponen
- f. Elektronika Digital & Komputer
- g. Elektronika Komunikasi
- h. lainnya .....

4. Konsultasikan judul sesuai materi bidang ilmu kepada Dosen*)

Dr. Aryuanto, ST, MT

Ketua Jurusan

Ir. Yusuf Ismail Nakhoda,MT
NIP. Y. 1018800189

5. Judul yang diajukan mahasiswa: Rancang Bangun Aplikasi Enkripsi dan dekripsi Audio Video Menggunakan Algoritma Blowfish Berbasis Visual Basic

6. Perubahan judul yang disetujui Dosen sesuai materi bidang ilmu .....

7. Catatan: .....

Persetujuan Judul skripsi yang dikonsultasikan kepada Dosen materi bidang ilmu

Disetujui 16/11 2011
Dosen

.....

Perhatian:
1. Formulir pengajuan ini harap dikembalikan kepada jurusan paling lambat satu minggu setelah disetujui kelompok dosen keahlian dengan dilampirkan proposal skripsi beserta persyaratan skripsi sesuai form S-1
2. Keterangan: *) Coret yang tidak perlu
   **) dilingkari a, b, c, ...atau g sesuai bidang keahlian

Form S-2

INSTITUT TEKNOLOGI NASIONAL
Jl. Raya Karanglo, Km. 2
MALANG

PERMOHONAN PERSETUJUAN SKRIPSI

Yang betanda tangan dibawah ini :

Nama	: BENNY Gres Sanutra
NIM	: 07.12.590
Semester	: 7
Fakultas	: Teknologi Industri
Jurusan	: Teknik Elektro S-1
Konsentrasi	: **TEKNIK ELEKTRONIKA** **TEKNIK ENERGI LISTRIK** **TEKNIK KOMPUTER DAN INFORMATIKA** **TEKNIK KOMPUTER** **TEKNIK TELEKOMUNIKASI**
Alamat	: Tulungagung

Dengan ini kami mengajukan permohonan untuk mendapatkan persetujuan untuk membuat ***SKRIPSI Tingkat Sarjana.*** Untuk melengkapi permohonan tersebut, bersama kami lampirkan persyaratan-persyaratan yang harus dipenuhi.
Adapun persyaratan-persyaratan pengambilan ***SKRIPSI*** adalah sebagai berikut :

1. Telah melaksanakan semua praktikum sesuai dengan konsentrasinya (...✓..)
2. Telah lulus dan menyerahkan Laporan Praktek Kerja (...✓..)
3. Telah lulus seluruh mata kuliah keahlian (MKB) sesuai konsentrasinya (...✓..)
4. Telah menempuh mata kuliah ≥ 134 sks dengan IPK ≥ 2 dan tidak ada nilai E (...✓..)
5. Telah mengikuti secara aktif kegiatan seminar skripsi yang diadakan Jurusan (.....)
6. Memenuhi persyaratan administrasi (.....)

Demikian permohonan ini untuk mendapatkan penyelesaian lebih lanjut dan atas perhatiannya kami ucapkan terima kasih.

Telah diteliti kebenaran data tersebut diatas
Recording Teknik Elektro

(Ima Hanilayani)

Malang, 23 Mei 2011
Pemohon

(Benny Gres Sanutra)

Disetujui
Ketua Jurusan Teknik Elektro

Ir. Yusuf Ismail Nakhoda, MT
NIP. Y. 1018800189

Mengetahui
Dosen Wali

( )

Catatan :
Bagi mahasiswa yang telah memenuhi persyaratan mengambil SKRIPSI agar membuat proposal dan mendapat persetujuan dari Ketua Jurusan/Sekretaris Jurusan T. Elektro S-1

1. IPK (130/138) : 3.72
2. .....
3. - ~~Praktikum~~ lengkap.

Pseudocode Class Blowfish :

```
BEGIN
  MultiUse = -1  'True
  Persistable = 0  'NotPersistable
  DataBindingBehavior = 0  'vbNone
  DataSourceBehavior  = 0  'vbNone
  MTSTransactionMode  = 0  'NotAnMTSObject
END
Attribute VB_Name = "clsBlowfish"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = False
Attribute VB_Exposed = False
Option Explicit
Private Declare Sub CopyMem Lib "kernel32" Alias "RtlMoveMemory"
(Destination As Any, Source As Any, ByVal Length As Long)

Event Progress(Percent As Long)

Private Const ROUNDS = 16

Private m_pBox(0 To ROUNDS + 1) As Long
Private m_sBox(0 To 3, 0 To 255) As Long
Private m_KeyValue As String
Private m_RunningCompiled As Boolean
Private byteArray() As Byte
Private hiByte As Long
Private hiBound As Long

Private Sub Append(ByRef StringData As String, Optional Length As Long)
   Dim DataLength As Long
   If Length > 0 Then DataLength = Length Else DataLength = Len(StringData)
   If DataLength + hiByte > hiBound Then
      hiBound = hiBound + 1024
      ReDim Preserve byteArray(hiBound)
   End If
   CopyMem ByVal VarPtr(byteArray(hiByte)), ByVal StringData, DataLength
   hiByte = hiByte + DataLength
End Sub

Private Property Get GData() As String
   Dim StringData As String
   StringData = Space(hiByte)
   CopyMem ByVal StringData, ByVal VarPtr(byteArray(0)), hiByte
   GData = StringData
End Property

Private Sub Reset()
```

```
    hiByte = 0
    hiBound = 1024
    ReDim byteArray(hiBound)
End Sub

Private Static Sub DecryptBlock(Xl As Long, Xr As Long)
    Dim i As Long, j As Long, K As Long
    K = Xr
    Xr = Xl Xor m_pBox(ROUNDS + 1)
    Xl = K Xor m_pBox(ROUNDS)
    j = ROUNDS - 2
    For i = 0 To (ROUNDS \ 2 - 1)
        Xl = Xl Xor f(Xr)
        Xr = Xr Xor m_pBox(j + 1)
        Xr = Xr Xor f(Xl)
        Xl = Xl Xor m_pBox(j)
        j = j - 2
    Next
End Sub

Private Static Sub EncryptBlock(Xl As Long, Xr As Long)
    Dim i As Long, j As Long, Temp As Long
    j = 0
    For i = 0 To (ROUNDS \ 2 - 1)
        Xl = Xl Xor m_pBox(j)
        Xr = Xr Xor f(Xl)
        Xr = Xr Xor m_pBox(j + 1)
        Xl = Xl Xor f(Xr)
        j = j + 2
    Next
    Temp = Xr
    Xr = Xl Xor m_pBox(ROUNDS)
    Xl = Temp Xor m_pBox(ROUNDS + 1)
End Sub

Public Sub EncryptByte(byteArray() As Byte, Optional Key As String)
    Dim Offset As Long, OrigLen As Long, LeftWord As Long, RightWord As
Long, CipherLen As Long, CipherLeft As Long, CipherRight As Long,
CurrPercent As Long, NextPercent As Long
    If (Len(Key) > 0) Then Me.Key = Key
    OrigLen = UBound(byteArray) + 1
    CipherLen = OrigLen + 12
    If (CipherLen Mod 8 <> 0) Then CipherLen = CipherLen + 8 - (CipherLen
Mod 8)
    ReDim Preserve byteArray(CipherLen - 1)
    Call CopyMem(byteArray(12), byteArray(0), OrigLen)
    Call CopyMem(byteArray(8), OrigLen, 4)
    Call Randomize
```

```
    Call CopyMem(byteArray(0), CLng(2147483647 * Rnd), 4)
    Call CopyMem(byteArray(4), CLng(2147483647 * Rnd), 4)
    For Offset = 0 To (CipherLen - 1) Step 8
        Call GetWord(LeftWord, byteArray(), Offset)
        Call GetWord(RightWord, byteArray(), Offset + 4)
        LeftWord = LeftWord Xor CipherLeft
        RightWord = RightWord Xor CipherRight
        Call EncryptBlock(LeftWord, RightWord)
        Call PutWord(LeftWord, byteArray(), Offset)
        Call PutWord(RightWord, byteArray(), Offset + 4)
        CipherLeft = LeftWord
        CipherRight = RightWord
        If (Offset >= NextPercent) Then
            CurrPercent = Int((Offset / CipherLen) * 100)
            NextPercent = (CipherLen * ((CurrPercent + 1) / 100)) + 1
            RaiseEvent Progress(CurrPercent)
        End If
    Next
    If (CurrPercent <> 100) Then RaiseEvent Progress(100)
End Sub

Public Function EncryptString(Text As String, Optional Key As String, Optional
OutputInHex As Boolean) As String
    Dim byteArray() As Byte
    byteArray() = StrConv(Text, vbFromUnicode)
    Call EncryptByte(byteArray(), Key)
    EncryptString = StrConv(byteArray(), vbUnicode)
    If OutputInHex = True Then EncryptString = EnHex(EncryptString)
End Function

Public Function DecryptString(Text As String, Optional Key As String, Optional
IsTextInHex As Boolean) As String
    Dim byteArray() As Byte
    If IsTextInHex = True Then Text = DeHex(Text)
    byteArray() = StrConv(Text, vbFromUnicode)
    Call DecryptByte(byteArray(), Key)
    DecryptString = StrConv(byteArray(), vbUnicode)
End Function

Private Function EnHex(Data As String) As String
    Dim iCount As Double, sTemp As String
    Reset
    For iCount = 1 To Len(Data)
        sTemp = Hex$(Asc(Mid$(Data, iCount, 1)))
        If Len(sTemp) < 2 Then sTemp = "0" & sTemp
        Append sTemp
    Next
    EnHex = GData
```

```
    Reset
End Function

Private Function DeHex(Data As String) As String
    Dim iCount As Double
    Reset
    For iCount = 1 To Len(Data) Step 2
        Append Chr$(Val("&H" & Mid$(Data, iCount, 2)))
    Next
    DeHex = GData
    Reset
End Function

Public Sub DecryptByte(byteArray() As Byte, Optional Key As String)
    On Error GoTo errorhandler
    Dim Offset As Long, OrigLen As Long, LeftWord As Long, RightWord As
Long, CipherLen As Long, CipherLeft As Long, CipherRight As Long,
CurrPercent As Long, NextPercent As Long
    If (Len(Key) > 0) Then Me.Key = Key
    CipherLen = UBound(byteArray) + 1
    For Offset = 0 To (CipherLen - 1) Step 8
        Call GetWord(LeftWord, byteArray(), Offset)
        Call GetWord(RightWord, byteArray(), Offset + 4)
        Call DecryptBlock(LeftWord, RightWord)
        LeftWord = LeftWord Xor CipherLeft
        RightWord = RightWord Xor CipherRight
        Call GetWord(CipherLeft, byteArray(), Offset)
        Call GetWord(CipherRight, byteArray(), Offset + 4)
        Call PutWord(LeftWord, byteArray(), Offset)
        Call PutWord(RightWord, byteArray(), Offset + 4)
        If Offset >= NextPercent Then
            CurrPercent = Int((Offset / CipherLen) * 100)
            NextPercent = (CipherLen * ((CurrPercent + 1) / 100)) + 1
            RaiseEvent Progress(CurrPercent)
        End If
    Next
    Call CopyMem(OrigLen, byteArray(8), 4)
    If (CipherLen - OrigLen > 19) Or (CipherLen - OrigLen < 12) Then Call
Err.Raise(vbObjectError, , "Incorrect size descriptor in Blowfish decryption")
    Call CopyMem(byteArray(0), byteArray(12), OrigLen)
    ReDim Preserve byteArray(OrigLen - 1)
    If CurrPercent <> 100 Then RaiseEvent Progress(100)
errorhandler:
End Sub

Private Static Function f(ByVal X As Long) As Long
    Dim xb(0 To 3) As Byte
    Call CopyMem(xb(0), X, 4)
```

```
    If (m_RunningCompiled) Then f = (((m_sBox(0, xb(3)) + m_sBox(1, xb(2)))
Xor m_sBox(2, xb(1))) + m_sBox(3, xb(0))) Else f =
UnsignedAdd((UnsignedAdd(m_sBox(0, xb(3)), m_sBox(1, xb(2))) Xor
m_sBox(2, xb(1))), m_sBox(3, xb(0)))
End Function

Private Static Sub GetWord(LongValue As Long, CryptBuffer() As Byte, Offset
As Long)
    Dim bb(0 To 3) As Byte
    bb(3) = CryptBuffer(Offset)
    bb(2) = CryptBuffer(Offset + 1)
    bb(1) = CryptBuffer(Offset + 2)
    bb(0) = CryptBuffer(Offset + 3)
    Call CopyMem(LongValue, bb(0), 4)
End Sub

Private Static Sub PutWord(LongValue As Long, CryptBuffer() As Byte, Offset
As Long)
    Dim bb(0 To 3) As Byte
    Call CopyMem(bb(0), LongValue, 4)
    CryptBuffer(Offset) = bb(3)
    CryptBuffer(Offset + 1) = bb(2)
    CryptBuffer(Offset + 2) = bb(1)
    CryptBuffer(Offset + 3) = bb(0)
End Sub

Private Static Function UnsignedAdd(ByVal Data1 As Long, Data2 As Long) As
Long
    Dim x1(0 To 3) As Byte, x2(0 To 3) As Byte, xx(0 To 3) As Byte, Rest As
Long, Value As Long, a As Long
    Call CopyMem(x1(0), Data1, 4)
    Call CopyMem(x2(0), Data2, 4)
    Rest = 0
    For a = 0 To 3
        Value = CLng(x1(a)) + CLng(x2(a)) + Rest
        xx(a) = Value And 255
        Rest = Value \ 256
    Next
    Call CopyMem(UnsignedAdd, xx(0), 4)
End Function

Private Function UnsignedDel(Data1 As Long, Data2 As Long) As Long
    Dim x1(0 To 3) As Byte, x2(0 To 3) As Byte, xx(0 To 3) As Byte, Rest As
Long, Value As Long, a As Long
    Call CopyMem(x1(0), Data1, 4)
    Call CopyMem(x2(0), Data2, 4)
    Call CopyMem(xx(0), UnsignedDel, 4)
    For a = 0 To 3
```

```
        Value = CLng(x1(a)) - CLng(x2(a)) - Rest
        If (Value < 0) Then
            Value = Value + 256
            Rest = 1
        Else
            Rest = 0
        End If
        xx(a) = Value
    Next
    Call CopyMem(UnsignedDel, xx(0), 4)
End Function

Public Property Let Key(New_Value As String)
    Dim i As Long, j As Long, K As Long, dataX As Long, datal As Long, datar
As Long, Key() As Byte, KeyLength As Long
    If (m_KeyValue = New_Value) Then Exit Property
    m_KeyValue = New_Value
    KeyLength = Len(New_Value)
    Key() = StrConv(New_Value, vbFromUnicode)
    j = 0
    For i = 0 To (ROUNDS + 1)
        dataX = 0
        For K = 0 To 3
            Call CopyMem(ByVal VarPtr(dataX) + 1, dataX, 3)
            dataX = (dataX Or Key(j))
            j = j + 1
            If (j >= KeyLength) Then j = 0
        Next
        m_pBox(i) = m_pBox(i) Xor dataX
    Next

    datal = 0: datar = 0
    For i = 0 To (ROUNDS + 1) Step 2
        Call EncryptBlock(datal, datar)
        m_pBox(i) = datal
        m_pBox(i + 1) = datar
    Next
    For i = 0 To 3
        For j = 0 To 255 Step 2
            Call EncryptBlock(datal, datar)
            m_sBox(i, j) = datal
            m_sBox(i, j + 1) = datar
        Next
    Next
End Property
```

```
"Class inisialisasi P-Box dan S-Box :
Private Sub Class_Initialize()
On Local Error Resume Next
  m_RunningCompiled =
((2147483647 + 1) < 0)
  m_pBox(0) = &H243F6A88
  m_pBox(1) = &H85A308D3
  m_pBox(2) = &H13198A2E
  m_pBox(3) = &H3707344
  m_pBox(4) = &HA4093822
  m_pBox(5) = &H299F31D0
  m_pBox(6) = &H82EFA98
  m_pBox(7) = &HEC4E6C89
  m_pBox(8) = &H452821E6
  m_pBox(9) = &H38D01377
  m_pBox(10) = &HBE5466CF
  m_pBox(11) = &H34E90C6C
  m_pBox(12) = &HC0AC29B7
  m_pBox(13) = &HC97C50DD
  m_pBox(14) = &H3F84D5B5
  m_pBox(15) = &HB5470917
  m_pBox(16) = &H9216D5D9
  m_pBox(17) = &H8979FB1B

  m_sBox(0, 0) = &HD1310BA6
  m_sBox(1, 0) = &H98DFB5AC
  m_sBox(2, 0) = &H2FFD72DB
  m_sBox(3, 0) = &HD01ADFB7
  m_sBox(0, 1) = &HB8E1AFED
  m_sBox(1, 1) = &H6A267E96
  m_sBox(2, 1) = &HBA7C9045
  m_sBox(3, 1) = &HF12C7F99
  m_sBox(0, 2) = &H24A19947
  m_sBox(1, 2) = &HB3916CF7
  m_sBox(2, 2) = &H801F2E2
  m_sBox(3, 2) = &H858EFC16
  m_sBox(0, 3) = &H636920D8
  m_sBox(1, 3) = &H71574E69
  m_sBox(2, 3) = &HA458FEA3
  m_sBox(3, 3) = &HF4933D7E
  m_sBox(0, 4) = &HD95748F
  m_sBox(1, 4) = &H728EB658
  m_sBox(2, 4) = &H718BCD58
  m_sBox(3, 4) = &H82154AEE
  m_sBox(0, 5) = &H7B54A41D
  m_sBox(1, 5) = &HC25A59B5
  m_sBox(2, 5) = &H9C30D539
  m_sBox(3, 5) = &H2AF26013
  m_sBox(0, 6) = &HC5D1B023
  m_sBox(1, 6) = &H286085F0
  m_sBox(2, 6) = &HCA417918
  m_sBox(3, 6) = &HB8DB38EF
  m_sBox(0, 7) = &H8E79DCB0
  m_sBox(1, 7) = &H603A180E
  m_sBox(2, 7) = &H6C9E0E8B
  m_sBox(3, 7) = &HB01E8A3E
  m_sBox(0, 8) = &HD71577C1
  m_sBox(1, 8) = &HBD314B27
  m_sBox(2, 8) = &H78AF2FDA
  m_sBox(3, 8) = &H55605C60
  m_sBox(0, 9) = &HE65525F3
  m_sBox(1, 9) = &HAA55AB94
  m_sBox(2, 9) = &H57489862
  m_sBox(3, 9) = &H63E81440
  m_sBox(0, 10) = &H55CA396A
  m_sBox(1, 10) = &H2AAB10B6
  m_sBox(2, 10) = &HB4CC5C34
  m_sBox(3, 10) = &H1141E8CE
  m_sBox(0, 11) = &HA15486AF
  m_sBox(1, 11) = &H7C72E993
  m_sBox(2, 11) = &HB3EE1411
  m_sBox(3, 11) = &H636FBC2A
  m_sBox(0, 12) = &H2BA9C55D
  m_sBox(1, 12) = &H741831F6
  m_sBox(2, 12) = &HCE5C3E16
  m_sBox(3, 12) = &H9B87931E
  m_sBox(0, 13) = &HAFD6BA33
  m_sBox(1, 13) = &H6C24CF5C
  m_sBox(2, 13) = &H7A325381
  m_sBox(3, 13) = &H28958677
  m_sBox(0, 14) = &H3B8F4898
  m_sBox(1, 14) = &H6B4BB9AF
  m_sBox(2, 14) = &HC4BFE81B
  m_sBox(3, 14) = &H66282193
  m_sBox(0, 15) = &H61D809CC
  m_sBox(1, 15) = &HFB21A991
  m_sBox(2, 15) = &H487CAC60
  m_sBox(3, 15) = &H5DEC8032
  m_sBox(0, 16) = &HEF845D5D
  m_sBox(1, 16) = &HE98575B1
  m_sBox(2, 16) = &HDC262302
  m_sBox(3, 16) = &HEB651B88
  m_sBox(0, 17) = &H23893E81
  m_sBox(1, 17) = &HD396ACC5
  m_sBox(2, 17) = &HF6D6FF3
  m_sBox(3, 17) = &H83F44239
```

```
m_sBox(0, 18) = &H2E0B4482
m_sBox(1, 18) = &HA4842004
m_sBox(2, 18) = &H69C8F04A
m_sBox(3, 18) = &H9E1F9B5E
m_sBox(0, 19) = &H21C66842
m_sBox(1, 19) = &HF6E96C9A
m_sBox(2, 19) = &H670C9C61
m_sBox(3, 19) = &HABD388F0
m_sBox(0, 20) = &H6A51A0D2
m_sBox(1, 20) = &HD8542F68
m_sBox(2, 20) = &H960FA728
m_sBox(3, 20) = &HAB5133A3
m_sBox(0, 21) = &H6EEF0B6C
m_sBox(1, 21) = &H137A3BE4
m_sBox(2, 21) = &HBA3BF050
m_sBox(3, 21) = &H7EFB2A98
m_sBox(0, 22) = &HA1F1651D
m_sBox(1, 22) = &H39AF0176
m_sBox(2, 22) = &H66CA593E
m_sBox(3, 22) = &H82430E88
m_sBox(0, 23) = &H8CEE8619
m_sBox(1, 23) = &H456F9FB4
m_sBox(2, 23) = &H7D84A5C3
m_sBox(3, 23) = &H3B8B5EBE
m_sBox(0, 24) = &HE06F75D8
m_sBox(1, 24) = &H85C12073
m_sBox(2, 24) = &H401A449F
m_sBox(3, 24) = &H56C16AA6
m_sBox(0, 25) = &H4ED3AA62
m_sBox(1, 25) = &H363F7706
m_sBox(2, 25) = &H1BFEDF72
m_sBox(3, 25) = &H429B023D
m_sBox(0, 26) = &H37D0D724
m_sBox(1, 26) = &HD00A1248
m_sBox(2, 26) = &HDB0FEAD3
m_sBox(3, 26) = &H49F1C09B
m_sBox(0, 27) = &H75372C9
m_sBox(1, 27) = &H80991B7B
m_sBox(2, 27) = &H25D479D8
m_sBox(3, 27) = &HF6E8DEF7
m_sBox(0, 28) = &HE3FE501A
m_sBox(1, 28) = &HB6794C3B
m_sBox(2, 28) = &H976CE0BD
m_sBox(3, 28) = &H4C006BA
m_sBox(0, 29) = &HC1A94FB6
m_sBox(1, 29) = &H409F60C4
m_sBox(2, 29) = &H5E5C9EC2
m_sBox(3, 29) = &H196A2463
m_sBox(0, 30) = &H68FB6FAF
m_sBox(1, 30) = &H3E6C53B5
m_sBox(2, 30) = &H1339B2EB
m_sBox(3, 30) = &H3B52EC6F
m_sBox(0, 31) = &H6DFC511F
m_sBox(1, 31) = &H9B30952C
m_sBox(2, 31) = &HCC814544
m_sBox(3, 31) = &HAF5EBD09
m_sBox(0, 32) = &HBEE3D004
m_sBox(1, 32) = &HDE334AFD
m_sBox(2, 32) = &H660F2807
m_sBox(3, 32) = &H192E4BB3
m_sBox(0, 33) = &HC0CBA857
m_sBox(1, 33) = &H45C8740F
m_sBox(2, 33) = &HD20B5F39
m_sBox(3, 33) = &HB9D3FBDB
m_sBox(0, 34) = &H5579C0BD
m_sBox(1, 34) = &H1A60320A
m_sBox(2, 34) = &HD6A100C6
m_sBox(3, 34) = &H402C7279
m_sBox(0, 35) = &H679F25FE
m_sBox(1, 35) = &HFB1FA3CC
m_sBox(2, 35) = &H8EA5E9F8
m_sBox(3, 35) = &HDB3222F8
m_sBox(0, 36) = &H3C7516DF
m_sBox(1, 36) = &HFD616B15
m_sBox(2, 36) = &H2F501EC8
m_sBox(3, 36) = &HAD0552AB
m_sBox(0, 37) = &H323DB5FA
m_sBox(1, 37) = &HFD238760
m_sBox(2, 37) = &H53317B48
m_sBox(3, 37) = &H3E00DF82
m_sBox(0, 38) = &H9E5C57BB
m_sBox(1, 38) = &HCA6F8CA0
m_sBox(2, 38) = &H1A87562E
m_sBox(3, 38) = &HDF1769DB
m_sBox(0, 39) = &HD542A8F6
m_sBox(1, 39) = &H287EFFC3
m_sBox(2, 39) = &HAC6732C6
m_sBox(3, 39) = &H8C4F5573
m_sBox(0, 40) = &H695B27B0
m_sBox(1, 40) = &HBBCA58C8
m_sBox(2, 40) = &HE1FFA35D
m_sBox(3, 40) = &HB8F011A0
m_sBox(0, 41) = &H10FA3D98
m_sBox(1, 41) = &HFD2183B8
m_sBox(2, 41) = &H4AFCB56C
m_sBox(3, 41) = &H2DD1D35B
```

```
m_sBox(0, 42) = &H9A53E479
m_sBox(1, 42) = &HB6F84565
m_sBox(2, 42) = &HD28E49BC
m_sBox(3, 42) = &H4BFB9790
m_sBox(0, 43) = &HE1DDF2DA
m_sBox(1, 43) = &HA4CB7E33
m_sBox(2, 43) = &H62FB1341
m_sBox(3, 43) = &HCEE4C6E8
m_sBox(0, 44) = &HEF20CADA
m_sBox(1, 44) = &H36774C01
m_sBox(2, 44) = &HD07E9EFE
m_sBox(3, 44) = &H2BF11FB4
m_sBox(0, 45) = &H95DBDA4D
m_sBox(1, 45) = &HAE909198
m_sBox(2, 45) = &HEAAD8E71
m_sBox(3, 45) = &H6B93D5A0
m_sBox(0, 46) = &HD08ED1D0
m_sBox(1, 46) = &HAFC725E0
m_sBox(2, 46) = &H8E3C5B2F
m_sBox(3, 46) = &H8E7594B7
m_sBox(0, 47) = &H8FF6E2FB
m_sBox(1, 47) = &HF2122B64
m_sBox(2, 47) = &H8888B812
m_sBox(3, 47) = &H900DF01C
m_sBox(0, 48) = &H4FAD5EA0
m_sBox(1, 48) = &H688FC31C
m_sBox(2, 48) = &HD1CFF191
m_sBox(3, 48) = &HB3A8C1AD
m_sBox(0, 49) = &H2F2F2218
m_sBox(1, 49) = &HBE0E1777
m_sBox(2, 49) = &HEA752DFE
m_sBox(3, 49) = &H8B021FA1
m_sBox(0, 50) = &HE5A0CC0F
m_sBox(1, 50) = &HB56F74E8
m_sBox(2, 50) = &H18ACF3D6
m_sBox(3, 50) = &HCE89E299
m_sBox(0, 51) = &HB4A84FE0
m_sBox(1, 51) = &HFD13E0B7
m_sBox(2, 51) = &H7CC43B81
m_sBox(3, 51) = &HD2ADA8D9
m_sBox(0, 52) = &H165FA266
m_sBox(1, 52) = &H80957705
m_sBox(2, 52) = &H93CC7314
m_sBox(3, 52) = &H211A1477
m_sBox(0, 53) = &HE6AD2065
m_sBox(1, 53) = &H77B5FA86
m_sBox(2, 53) = &HC75442F5
m_sBox(3, 53) = &HFB9D35CF
m_sBox(0, 54) = &HEBCDAF0C
m_sBox(1, 54) = &H7B3E89A0
m_sBox(2, 54) = &HD6411BD3
m_sBox(3, 54) = &HAE1E7E49
m_sBox(0, 55) = &H250E2D
m_sBox(1, 55) = &H2071B35E
m_sBox(2, 55) = &H226800BB
m_sBox(3, 55) = &H57B8E0AF
m_sBox(0, 56) = &H2464369B
m_sBox(1, 56) = &HF009B91E
m_sBox(2, 56) = &H5563911D
m_sBox(3, 56) = &H59DFA6AA
m_sBox(0, 57) = &H78C14389
m_sBox(1, 57) = &HD95A537F
m_sBox(2, 57) = &H207D5BA2
m_sBox(3, 57) = &H2E5B9C5
m_sBox(0, 58) = &H83260376
m_sBox(1, 58) = &H6295CFA9
m_sBox(2, 58) = &H11C81968
m_sBox(3, 58) = &H4E734A41
m_sBox(0, 59) = &HB3472DCA
m_sBox(1, 59) = &H7B14A94A
m_sBox(2, 59) = &H1B510052
m_sBox(3, 59) = &H9A532915
m_sBox(0, 60) = &HD60F573F
m_sBox(1, 60) = &HBC9BC6E4
m_sBox(2, 60) = &H2B60A476
m_sBox(3, 60) = &H81E67400
m_sBox(0, 61) = &H8BA6FB5
m_sBox(1, 61) = &H571BE91F
m_sBox(2, 61) = &HF296EC6B
m_sBox(3, 61) = &H2A0DD915
m_sBox(0, 62) = &HB6636521
m_sBox(1, 62) = &HE7B9F9B6
m_sBox(2, 62) = &HFF34052E
m_sBox(3, 62) = &HC5855664
m_sBox(0, 63) = &H53B02D5D
m_sBox(1, 63) = &HA99F8FA1
m_sBox(2, 63) = &H8BA4799
m_sBox(3, 63) = &H6E85076A
m_sBox(0, 64) = &H4B7A70E9
m_sBox(1, 64) = &HB5B32944
m_sBox(2, 64) = &HDB75092E
m_sBox(3, 64) = &HC4192623
m_sBox(0, 65) = &HAD6EA6B0
m_sBox(1, 65) = &H49A7DF7D
m_sBox(2, 65) = &H9CEE60B8
m_sBox(3, 65) = &H8FEDB266
```

```
m_sBox(0, 66) = &HECAA8C71
m_sBox(1, 66) = &H699A17FF
m_sBox(2, 66) = &H5664526C
m_sBox(3, 66) = &HC2B19EE1
m_sBox(0, 67) = &H193602A5
m_sBox(1, 67) = &H75094C29
m_sBox(2, 67) = &HA0591340
m_sBox(3, 67) = &HE4183A3E
m_sBox(0, 68) = &H3F54989A
m_sBox(1, 68) = &H5B429D65
m_sBox(2, 68) = &H6B8FE4D6
m_sBox(3, 68) = &H99F73FD6
m_sBox(0, 69) = &HA1D29C07
m_sBox(1, 69) = &HEFE830F5
m_sBox(2, 69) = &H4D2D38E6
m_sBox(3, 69) = &HF0255DC1
m_sBox(0, 70) = &H4CDD2086
m_sBox(1, 70) = &H8470EB26
m_sBox(2, 70) = &H6382E9C6
m_sBox(3, 70) = &H21ECC5E
m_sBox(0, 71) = &H9686B3F
m_sBox(1, 71) = &H3EBAEFC9
m_sBox(2, 71) = &H3C971814
m_sBox(3, 71) = &H6B6A70A1
m_sBox(0, 72) = &H687F3584
m_sBox(1, 72) = &H52A0E286
m_sBox(2, 72) = &HB79C5305
m_sBox(3, 72) = &HAA500737
m_sBox(0, 73) = &H3E07841C
m_sBox(1, 73) = &H7FDEAE5C
m_sBox(2, 73) = &H8E7D44EC
m_sBox(3, 73) = &H5716F2B8
m_sBox(0, 74) = &HB03ADA37
m_sBox(1, 74) = &HF0500C0D
m_sBox(2, 74) = &HF01C1F04
m_sBox(3, 74) = &H200B3FF
m_sBox(0, 75) = &HAE0CF51A
m_sBox(1, 75) = &H3CB574B2
m_sBox(2, 75) = &H25837A58
m_sBox(3, 75) = &HDC0921BD
m_sBox(0, 76) = &HD19113F9
m_sBox(1, 76) = &H7CA92FF6
m_sBox(2, 76) = &H94324773
m_sBox(3, 76) = &H22F54701
m_sBox(0, 77) = &H3AE5E581
m_sBox(1, 77) = &H37C2DADC
m_sBox(2, 77) = &HC8B57634
m_sBox(3, 77) = &H9AF3DDA7
m_sBox(0, 78) = &HA9446146
m_sBox(1, 78) = &HFD0030E
m_sBox(2, 78) = &HECC8C73E
m_sBox(3, 78) = &HA4751E41
m_sBox(0, 79) = &HE238CD99
m_sBox(1, 79) = &H3BEA0E2F
m_sBox(2, 79) = &H3280BBA1
m_sBox(3, 79) = &H183EB331
m_sBox(0, 80) = &H4E548B38
m_sBox(1, 80) = &H4F6DB908
m_sBox(2, 80) = &H6F420D03
m_sBox(3, 80) = &HF60A04BF
m_sBox(0, 81) = &H2CB81290
m_sBox(1, 81) = &H24977C79
m_sBox(2, 81) = &H5679B072
m_sBox(3, 81) = &HBCAF89AF
m_sBox(0, 82) = &HDE9A771F
m_sBox(1, 82) = &HD9930810
m_sBox(2, 82) = &HB38BAE12
m_sBox(3, 82) = &HDCCF3F2E
m_sBox(0, 83) = &H5512721F
m_sBox(1, 83) = &H2E6B7124
m_sBox(2, 83) = &H501ADDE6
m_sBox(3, 83) = &H9F84CD87
m_sBox(0, 84) = &H7A584718
m_sBox(1, 84) = &H7408DA17
m_sBox(2, 84) = &HBC9F9ABC
m_sBox(3, 84) = &HE94B7D8C
m_sBox(0, 85) = &HEC7AEC3A
m_sBox(1, 85) = &HDB851DFA
m_sBox(2, 85) = &H63094366
m_sBox(3, 85) = &HC464C3D2
m_sBox(0, 86) = &HEF1C1847
m_sBox(1, 86) = &H3215D908
m_sBox(2, 86) = &HDD433B37
m_sBox(3, 86) = &H24C2BA16
m_sBox(0, 87) = &H12A14D43
m_sBox(1, 87) = &H2A65C451
m_sBox(2, 87) = &H50940002
m_sBox(3, 87) = &H133AE4DD
m_sBox(0, 88) = &H71DFF89E
m_sBox(1, 88) = &H10314E55
m_sBox(2, 88) = &H81AC77D6
m_sBox(3, 88) = &H5F11199B
m_sBox(0, 89) = &H43556F1
m_sBox(1, 89) = &HD7A3C76B
m_sBox(2, 89) = &H3C11183B
m_sBox(3, 89) = &H5924A509
```

```
m_sBox(0, 90) = &HF28FE6ED
m_sBox(1, 90) = &H97F1FBFA
m_sBox(2, 90) = &H9EBABF2C
m_sBox(3, 90) = &H1E153C6E
m_sBox(0, 91) = &H86E34570
m_sBox(1, 91) = &HEAE96FB1
m_sBox(2, 91) = &H860E5E0A
m_sBox(3, 91) = &H5A3E2AB3
m_sBox(0, 92) = &H771FE71C
m_sBox(1, 92) = &H4E3D06FA
m_sBox(2, 92) = &H2965DCB9
m_sBox(3, 92) = &H99E71D0F
m_sBox(0, 93) = &H803E89D6
m_sBox(1, 93) = &H5266C825
m_sBox(2, 93) = &H2E4CC978
m_sBox(3, 93) = &H9C10B36A
m_sBox(0, 94) = &HC6150EBA
m_sBox(1, 94) = &H94E2EA78
m_sBox(2, 94) = &HA5FC3C53
m_sBox(3, 94) = &H1E0A2DF4
m_sBox(0, 95) = &HF2F74EA7
m_sBox(1, 95) = &H361D2B3D
m_sBox(2, 95) = &H1939260F
m_sBox(3, 95) = &H19C27960
m_sBox(0, 96) = &H5223A708
m_sBox(1, 96) = &HF71312B6
m_sBox(2, 96) = &HEBADFE6E
m_sBox(3, 96) = &HEAC31F66
m_sBox(0, 97) = &HE3BC4595
m_sBox(1, 97) = &HA67BC883
m_sBox(2, 97) = &HB17F37D1
m_sBox(3, 97) = &H18CFF28
m_sBox(0, 98) = &HC332DDEF
m_sBox(1, 98) = &HBE6C5AA5
m_sBox(2, 98) = &H65582185
m_sBox(3, 98) = &H68AB9802
m_sBox(0, 99) = &HEECEA50F
m_sBox(1, 99) = &HDB2F953B
m_sBox(2, 99) = &H2AEF7DAD
m_sBox(3, 99) = &H5B6E2F84
m_sBox(0, 100) = &H1521B628
m_sBox(1, 100) = &H29076170
m_sBox(2, 100) = &HECDD4775
m_sBox(3, 100) = &H619F1510
m_sBox(0, 101) = &H13CCA830
m_sBox(1, 101) = &HEB61BD96
m_sBox(2, 101) = &H334FE1E
m_sBox(3, 101) = &HAA0363CF
m_sBox(0, 102) = &HB5735C90
m_sBox(1, 102) = &H4C70A239
m_sBox(2, 102) = &HD59E9E0B
m_sBox(3, 102) = &HCBAADE14
m_sBox(0, 103) = &HEECC86BC
m_sBox(1, 103) = &H60622CA7
m_sBox(2, 103) = &H9CAB5CAB
m_sBox(3, 103) = &HB2F3846E
m_sBox(0, 104) = &H648B1EAF
m_sBox(1, 104) = &H19BDF0CA
m_sBox(2, 104) = &HA02369B9
m_sBox(3, 104) = &H655ABB50
m_sBox(0, 105) = &H40685A32
m_sBox(1, 105) = &H3C2AB4B3
m_sBox(2, 105) = &H319EE9D5
m_sBox(3, 105) = &HC021B8F7
m_sBox(0, 106) = &H9B540B19
m_sBox(1, 106) = &H875FA099
m_sBox(2, 106) = &H95F7997E
m_sBox(3, 106) = &H623D7DA8
m_sBox(0, 107) = &HF837889A
m_sBox(1, 107) = &H97E32D77
m_sBox(2, 107) = &H11ED935F
m_sBox(3, 107) = &H16681281
m_sBox(0, 108) = &HE358829
m_sBox(1, 108) = &HC7E61FD6
m_sBox(2, 108) = &H96DEDFA1
m_sBox(3, 108) = &H7858BA99
m_sBox(0, 109) = &H57F584A5
m_sBox(1, 109) = &H1B227263
m_sBox(2, 109) = &H9B83C3FF
m_sBox(3, 109) = &H1AC24696
m_sBox(0, 110) = &HCDB30AEB
m_sBox(1, 110) = &H532E3054
m_sBox(2, 110) = &H8FD948E4
m_sBox(3, 110) = &H6DBC3128
m_sBox(0, 111) = &H58EBF2EF
m_sBox(1, 111) = &H34C6FFEA
m_sBox(2, 111) = &HFE28ED61
m_sBox(3, 111) = &HEE7C3C73
m_sBox(0, 112) = &H5D4A14D9
m_sBox(1, 112) = &HE864B7E3
m_sBox(2, 112) = &H42105D14
m_sBox(3, 112) = &H203E13E0
m_sBox(0, 113) = &H45EEE2B6
m_sBox(1, 113) = &HA3AAABEA
m_sBox(2, 113) = &HDB6C4F15
m_sBox(3, 113) = &HFACB4FD0
```

```
m_sBox(0, 114) = &HC742F442
m_sBox(1, 114) = &HEF6ABBB5
m_sBox(2, 114) = &H654F3B1D
m_sBox(3, 114) = &H41CD2105
m_sBox(0, 115) = &HD81E799E
m_sBox(1, 115) = &H86854DC7
m_sBox(2, 115) = &HE44B476A
m_sBox(3, 115) = &H3D816250
m_sBox(0, 116) = &HCF62A1F2
m_sBox(1, 116) = &H5B8D2646
m_sBox(2, 116) = &HFC8883A0
m_sBox(3, 116) = &HC1C7B6A3
m_sBox(0, 117) = &H7F1524C3
m_sBox(1, 117) = &H69CB7492
m_sBox(2, 117) = &H47848A0B
m_sBox(3, 117) = &H5692B285
m_sBox(0, 118) = &H95BBF00
m_sBox(1, 118) = &HAD19489D
m_sBox(2, 118) = &H1462B174
m_sBox(3, 118) = &H23820E00
m_sBox(0, 119) = &H58428D2A
m_sBox(1, 119) = &HC55F5EA
m_sBox(2, 119) = &H1DADF43E
m_sBox(3, 119) = &H233F7061
m_sBox(0, 120) = &H3372F092
m_sBox(1, 120) = &H8D937E41
m_sBox(2, 120) = &HD65FECF1
m_sBox(3, 120) = &H6C223BDB
m_sBox(0, 121) = &H7CDE3759
m_sBox(1, 121) = &HCBEE7460
m_sBox(2, 121) = &H4085F2A7
m_sBox(3, 121) = &HCE77326E
m_sBox(0, 122) = &HA6078084
m_sBox(1, 122) = &H19F8509E
m_sBox(2, 122) = &HE8EFD855
m_sBox(3, 122) = &H61D99735
m_sBox(0, 123) = &HA969A7AA
m_sBox(1, 123) = &HC50C06C2
m_sBox(2, 123) = &H5A04ABFC
m_sBox(3, 123) = &H800BCADC
m_sBox(0, 124) = &H9E447A2E
m_sBox(1, 124) = &HC3453484
m_sBox(2, 124) = &HFDD56705
m_sBox(3, 124) = &HE1E9EC9
m_sBox(0, 125) = &HDB73DBD3
m_sBox(1, 125) = &H105588CD
m_sBox(2, 125) = &H675FDA79
m_sBox(3, 125) = &HE3674340
m_sBox(0, 126) = &HC5C43465
m_sBox(1, 126) = &H713E38D8
m_sBox(2, 126) = &H3D28F89E
m_sBox(3, 126) = &HF16DFF20
m_sBox(0, 127) = &H153E21E7
m_sBox(1, 127) = &H8FB03D4A
m_sBox(2, 127) = &HE6E39F2B
m_sBox(3, 127) = &HDB83ADF7
m_sBox(0, 128) = &HE93D5A68
m_sBox(1, 128) = &H948140F7
m_sBox(2, 128) = &HF64C261C
m_sBox(3, 128) = &H94692934
m_sBox(0, 129) = &H411520F7
m_sBox(1, 129) = &H7602D4F7
m_sBox(2, 129) = &HBCF46B2E
m_sBox(3, 129) = &HD4A20068
m_sBox(0, 130) = &HD4082471
m_sBox(1, 130) = &H3320F46A
m_sBox(2, 130) = &H43B7D4B7
m_sBox(3, 130) = &H500061AF
m_sBox(0, 131) = &H1E39F62E
m_sBox(1, 131) = &H97244546
m_sBox(2, 131) = &H14214F74
m_sBox(3, 131) = &HBF8B8840
m_sBox(0, 132) = &H4D95FC1D
m_sBox(1, 132) = &H96B591AF
m_sBox(2, 132) = &H70F4DDD3
m_sBox(3, 132) = &H66A02F45
m_sBox(0, 133) = &HBFBC09EC
m_sBox(1, 133) = &H3BD9785
m_sBox(2, 133) = &H7FAC6DD0
m_sBox(3, 133) = &H31CB8504
m_sBox(0, 134) = &H96EB27B3
m_sBox(1, 134) = &H55FD3941
m_sBox(2, 134) = &HDA2547E6
m_sBox(3, 134) = &HABCA0A9A
m_sBox(0, 135) = &H28507825
m_sBox(1, 135) = &H530429F4
m_sBox(2, 135) = &HA2C86DA
m_sBox(3, 135) = &HE9B66DFB
m_sBox(0, 136) = &H68DC1462
m_sBox(1, 136) = &HD7486900
m_sBox(2, 136) = &H680EC0A4
m_sBox(3, 136) = &H27A18DEE
m_sBox(0, 137) = &H4F3FFEA2
m_sBox(1, 137) = &HE887AD8C
m_sBox(2, 137) = &HB58CE006
m_sBox(3, 137) = &H7AF4D6B6
```

```
m_sBox(0, 138) = &HAACE1E7C
m_sBox(1, 138) = &HD3375FEC
m_sBox(2, 138) = &HCE78A399
m_sBox(3, 138) = &H406B2A42
m_sBox(0, 139) = &H20FE9E35
m_sBox(1, 139) = &HD9F385B9
m_sBox(2, 139) = &HEE39D7AB
m_sBox(3, 139) = &H3B124E8B
m_sBox(0, 140) = &H1DC9FAF7
m_sBox(1, 140) = &H4B6D1856
m_sBox(2, 140) = &H26A36631
m_sBox(3, 140) = &HEAE397B2
m_sBox(0, 141) = &H3A6EFA74
m_sBox(1, 141) = &HDD5B4332
m_sBox(2, 141) = &H6841E7F7
m_sBox(3, 141) = &HCA7820FB
m_sBox(0, 142) = &HFB0AF54E
m_sBox(1, 142) = &HD8FEB397
m_sBox(2, 142) = &H454056AC
m_sBox(3, 142) = &HBA489527
m_sBox(0, 143) = &H55533A3A
m_sBox(1, 143) = &H20838D87
m_sBox(2, 143) = &HFE6BA9B7
m_sBox(3, 143) = &HD096954B
m_sBox(0, 144) = &H55A867BC
m_sBox(1, 144) = &HA1159A58
m_sBox(2, 144) = &HCCA92963
m_sBox(3, 144) = &H99E1DB33
m_sBox(0, 145) = &HA62A4A56
m_sBox(1, 145) = &H3F3125F9
m_sBox(2, 145) = &H5EF47E1C
m_sBox(3, 145) = &H9029317C
m_sBox(0, 146) = &HFDF8E802
m_sBox(1, 146) = &H4272F70
m_sBox(2, 146) = &H80BB155C
m_sBox(3, 146) = &H5282CE3
m_sBox(0, 147) = &H95C11548
m_sBox(1, 147) = &HE4C66D22
m_sBox(2, 147) = &H48C1133F
m_sBox(3, 147) = &HC70F86DC
m_sBox(0, 148) = &H7F9C9EE
m_sBox(1, 148) = &H41041F0F
m_sBox(2, 148) = &H404779A4
m_sBox(3, 148) = &H5D886E17
m_sBox(0, 149) = &H325F51EB
m_sBox(1, 149) = &HD59BC0D1
m_sBox(2, 149) = &HF2BCC18F
m_sBox(3, 149) = &H41113564
m_sBox(0, 150) = &H257B7834
m_sBox(1, 150) = &H602A9C60
m_sBox(2, 150) = &HDFF8E8A3
m_sBox(3, 150) = &H1F636C1B
m_sBox(0, 151) = &HE12B4C2
m_sBox(1, 151) = &H2E1329E
m_sBox(2, 151) = &HAF664FD1
m_sBox(3, 151) = &HCAD18115
m_sBox(0, 152) = &H6B2395E0
m_sBox(1, 152) = &H333E92E1
m_sBox(2, 152) = &H3B240B62
m_sBox(3, 152) = &HEEBEB922
m_sBox(0, 153) = &H85B2A20E
m_sBox(1, 153) = &HE6BA0D99
m_sBox(2, 153) = &HDE720C8C
m_sBox(3, 153) = &H2DA2F728
m_sBox(0, 154) = &HD0127845
m_sBox(1, 154) = &H95B794FD
m_sBox(2, 154) = &H647D0862
m_sBox(3, 154) = &HE7CCF5F0
m_sBox(0, 155) = &H5449A36F
m_sBox(1, 155) = &H877D48FA
m_sBox(2, 155) = &HC39DFD27
m_sBox(3, 155) = &HF33E8D1E
m_sBox(0, 156) = &HA476341
m_sBox(1, 156) = &H992EFF74
m_sBox(2, 156) = &H3A6F6EAB
m_sBox(3, 156) = &HF4F8FD37
m_sBox(0, 157) = &HA812DC60
m_sBox(1, 157) = &HA1EBDDF8
m_sBox(2, 157) = &H991BE14C
m_sBox(3, 157) = &HDB6E6B0D
m_sBox(0, 158) = &HC67B5510
m_sBox(1, 158) = &H6D672C37
m_sBox(2, 158) = &H2765D43B
m_sBox(3, 158) = &HDCD0E804
m_sBox(0, 159) = &HF1290DC7
m_sBox(1, 159) = &HCC00FFA3
m_sBox(2, 159) = &HB5390F92
m_sBox(3, 159) = &H690FED0B
m_sBox(0, 160) = &H667B9FFB
m_sBox(1, 160) = &HCEDB7D9C
m_sBox(2, 160) = &HA091CF0B
m_sBox(3, 160) = &HD9155EA3
m_sBox(0, 161) = &HBB132F88
m_sBox(1, 161) = &H515BAD24
m_sBox(2, 161) = &H7B9479BF
m_sBox(3, 161) = &H763BD6EB
```

```
m_sBox(0, 162) = &H37392EB3
m_sBox(1, 162) = &HCC115979
m_sBox(2, 162) = &H8026E297
m_sBox(3, 162) = &HF42E312D
m_sBox(0, 163) = &H6842ADA7
m_sBox(1, 163) = &HC66A2B3B
m_sBox(2, 163) = &H12754CCC
m_sBox(3, 163) = &H782EF11C
m_sBox(0, 164) = &H6A124237
m_sBox(1, 164) = &HB79251E7
m_sBox(2, 164) = &H6A1BBE6
m_sBox(3, 164) = &H4BFB6350
m_sBox(0, 165) = &H1A6B1018
m_sBox(1, 165) = &H11CAEDFA
m_sBox(2, 165) = &H3D25BDD8
m_sBox(3, 165) = &HE2E1C3C9
m_sBox(0, 166) = &H44421659
m_sBox(1, 166) = &HA121386
m_sBox(2, 166) = &HD90CEC6E
m_sBox(3, 166) = &HD5ABEA2A
m_sBox(0, 167) = &H64AF674E
m_sBox(1, 167) = &HDA86A85F
m_sBox(2, 167) = &HBEBFE988
m_sBox(3, 167) = &H64E4C3FE
m_sBox(0, 168) = &H9DBC8057
m_sBox(1, 168) = &HF0F7C086
m_sBox(2, 168) = &H60787BF8
m_sBox(3, 168) = &H6003604D
m_sBox(0, 169) = &HD1FD8346
m_sBox(1, 169) = &HF6381FB0
m_sBox(2, 169) = &H7745AE04
m_sBox(3, 169) = &HD736FCCC
m_sBox(0, 170) = &H83426B33
m_sBox(1, 170) = &HF01EAB71
m_sBox(2, 170) = &HB0804187
m_sBox(3, 170) = &H3C005E5F
m_sBox(0, 171) = &H77A057BE
m_sBox(1, 171) = &HBDE8AE24
m_sBox(2, 171) = &H55464299
m_sBox(3, 171) = &HBF582E61
m_sBox(0, 172) = &H4E58F48F
m_sBox(1, 172) = &HF2DDFDA2
m_sBox(2, 172) = &HF474EF38
m_sBox(3, 172) = &H8789BDC2
m_sBox(0, 173) = &H5366F9C3
m_sBox(1, 173) = &HC8B38E74
m_sBox(2, 173) = &HB475F255
m_sBox(3, 173) = &H46FCD9B9
m_sBox(0, 174) = &H7AEB2661
m_sBox(1, 174) = &H8B1DDF84
m_sBox(2, 174) = &H846A0E79
m_sBox(3, 174) = &H915F95E2
m_sBox(0, 175) = &H466E598E
m_sBox(1, 175) = &H20B45770
m_sBox(2, 175) = &H8CD55591
m_sBox(3, 175) = &HC902DE4C
m_sBox(0, 176) = &HB90BACE1
m_sBox(1, 176) = &HBB8205D0
m_sBox(2, 176) = &H11A86248
m_sBox(3, 176) = &H7574A99E
m_sBox(0, 177) = &HB77F19B6
m_sBox(1, 177) = &HE0A9DC09
m_sBox(2, 177) = &H662D09A1
m_sBox(3, 177) = &HC4324633
m_sBox(0, 178) = &HE85A1F02
m_sBox(1, 178) = &H9F0BE8C
m_sBox(2, 178) = &H4A99A025
m_sBox(3, 178) = &H1D6EFE10
m_sBox(0, 179) = &H1AB93D1D
m_sBox(1, 179) = &HBA5A4DF
m_sBox(2, 179) = &HA186F20F
m_sBox(3, 179) = &H2868F169
m_sBox(0, 180) = &HDCB7DA83
m_sBox(1, 180) = &H573906FE
m_sBox(2, 180) = &HA1E2CE9B
m_sBox(3, 180) = &H4FCD7F52
m_sBox(0, 181) = &H50115E01
m_sBox(1, 181) = &HA70683FA
m_sBox(2, 181) = &HA002B5C4
m_sBox(3, 181) = &HDE6D027
m_sBox(0, 182) = &H9AF88C27
m_sBox(1, 182) = &H773F8641
m_sBox(2, 182) = &HC3604C06
m_sBox(3, 182) = &H61A806B5
m_sBox(0, 183) = &HF0177A28
m_sBox(1, 183) = &HC0F586E0
m_sBox(2, 183) = &H6058AA
m_sBox(3, 183) = &H30DC7D62
m_sBox(0, 184) = &H11E69ED7
m_sBox(1, 184) = &H2338EA63
m_sBox(2, 184) = &H53C2DD94
m_sBox(3, 184) = &HC2C21634
m_sBox(0, 185) = &HBBCBEE56
m_sBox(1, 185) = &H90BCB6DE
m_sBox(2, 185) = &HEBFC7DA1
m_sBox(3, 185) = &HCE591D76
```

```
m_sBox(0, 186) = &H6F05E409
m_sBox(1, 186) = &H4B7C0188
m_sBox(2, 186) = &H39720A3D
m_sBox(3, 186) = &H7C927C24
m_sBox(0, 187) = &H86E3725F
m_sBox(1, 187) = &H724D9DB9
m_sBox(2, 187) = &H1AC15BB4
m_sBox(3, 187) = &HD39EB8FC
m_sBox(0, 188) = &HED545578
m_sBox(1, 188) = &H8FCA5B5
m_sBox(2, 188) = &HD83D7CD3
m_sBox(3, 188) = &H4DAD0FC4
m_sBox(0, 189) = &H1E50EF5E
m_sBox(1, 189) = &HB161E6F8
m_sBox(2, 189) = &HA28514D9
m_sBox(3, 189) = &H6C51133C
m_sBox(0, 190) = &H6FD5C7E7
m_sBox(1, 190) = &H56E14EC4
m_sBox(2, 190) = &H362ABFCE
m_sBox(3, 190) = &HDDC6C837
m_sBox(0, 191) = &HD79A3234
m_sBox(1, 191) = &H92638212
m_sBox(2, 191) = &H670EFA8E
m_sBox(3, 191) = &H406000E0
m_sBox(0, 192) = &H3A39CE37
m_sBox(1, 192) = &HD3FAF5CF
m_sBox(2, 192) = &HABC27737
m_sBox(3, 192) = &H5AC52D1B
m_sBox(0, 193) = &H5CB0679E
m_sBox(1, 193) = &H4FA33742
m_sBox(2, 193) = &HD3822740
m_sBox(3, 193) = &H99BC9BBE
m_sBox(0, 194) = &HD5118E9D
m_sBox(1, 194) = &HBF0F7315
m_sBox(2, 194) = &HD62D1C7E
m_sBox(3, 194) = &HC700C47B
m_sBox(0, 195) = &HB78C1B6B
m_sBox(1, 195) = &H21A19045
m_sBox(2, 195) = &HB26EB1BE
m_sBox(3, 195) = &H6A366EB4
m_sBox(0, 196) = &H5748AB2F
m_sBox(1, 196) = &HBC946E79
m_sBox(2, 196) = &HC6A376D2
m_sBox(3, 196) = &H6549C2C8
m_sBox(0, 197) = &H530FF8EE
m_sBox(1, 197) = &H468DDE7D
m_sBox(2, 197) = &HD5730A1D
m_sBox(3, 197) = &H4CD04DC6
m_sBox(0, 198) = &H2939BBDB
m_sBox(1, 198) = &HA9BA4650
m_sBox(2, 198) = &HAC9526E8
m_sBox(3, 198) = &HBE5EE304
m_sBox(0, 199) = &HA1FAD5F0
m_sBox(1, 199) = &H6A2D519A
m_sBox(2, 199) = &H63EF8CE2
m_sBox(3, 199) = &H9A86EE22
m_sBox(0, 200) = &HC089C2B8
m_sBox(1, 200) = &H43242EF6
m_sBox(2, 200) = &HA51E03AA
m_sBox(3, 200) = &H9CF2D0A4
m_sBox(0, 201) = &H83C061BA
m_sBox(1, 201) = &H9BE96A4D
m_sBox(2, 201) = &H8FE51550
m_sBox(3, 201) = &HBA645BD6
m_sBox(0, 202) = &H2826A2F9
m_sBox(1, 202) = &HA73A3AE1
m_sBox(2, 202) = &H4BA99586
m_sBox(3, 202) = &HEF5562E9
m_sBox(0, 203) = &HC72FEFD3
m_sBox(1, 203) = &HF752F7DA
m_sBox(2, 203) = &H3F046F69
m_sBox(3, 203) = &H77FA0A59
m_sBox(0, 204) = &H80E4A915
m_sBox(1, 204) = &H87B08601
m_sBox(2, 204) = &H9B09E6AD
m_sBox(3, 204) = &H3B3EE593
m_sBox(0, 205) = &HE990FD5A
m_sBox(1, 205) = &H9E34D797
m_sBox(2, 205) = &H2CF0B7D9
m_sBox(3, 205) = &H22B8B51
m_sBox(0, 206) = &H96D5AC3A
m_sBox(1, 206) = &H17DA67D
m_sBox(2, 206) = &HD1CF3ED6
m_sBox(3, 206) = &H7C7D2D28
m_sBox(0, 207) = &H1F9F25CF
m_sBox(1, 207) = &HADF2B89B
m_sBox(2, 207) = &H5AD6B472
m_sBox(3, 207) = &H5A88F54C
m_sBox(0, 208) = &HE029AC71
m_sBox(1, 208) = &HE019A5E6
m_sBox(2, 208) = &H47B0ACFD
m_sBox(3, 208) = &HED93FA9B
m_sBox(0, 209) = &HE8D3C48D
m_sBox(1, 209) = &H283B57CC
m_sBox(2, 209) = &HF8D56629
m_sBox(3, 209) = &H79132E28
```

```
m_sBox(0, 210) = &H785F0191
m_sBox(1, 210) = &HED756055
m_sBox(2, 210) = &HF7960E44
m_sBox(3, 210) = &HE3D35E8C
m_sBox(0, 211) = &H15056DD4
m_sBox(1, 211) = &H88F46DBA
m_sBox(2, 211) = &H3A16125
m_sBox(3, 211) = &H564F0BD
m_sBox(0, 212) = &HC3EB9E15
m_sBox(1, 212) = &H3C9057A2
m_sBox(2, 212) = &H97271AEC
m_sBox(3, 212) = &HA93A072A
m_sBox(0, 213) = &H1B3F6D9B
m_sBox(1, 213) = &H1E6321F5
m_sBox(2, 213) = &HF59C66FB
m_sBox(3, 213) = &H26DCF319
m_sBox(0, 214) = &H7533D928
m_sBox(1, 214) = &HB155FDF5
m_sBox(2, 214) = &H3563482
m_sBox(3, 214) = &H8ABA3CBB
m_sBox(0, 215) = &H28517711
m_sBox(1, 215) = &HC20AD9F8
m_sBox(2, 215) = &HABCC5167
m_sBox(3, 215) = &HCCAD925F
m_sBox(0, 216) = &H4DE81751
m_sBox(1, 216) = &H3830DC8E
m_sBox(2, 216) = &H379D5862
m_sBox(3, 216) = &H9320F991
m_sBox(0, 217) = &HEA7A90C2
m_sBox(1, 217) = &HFB3E7BCE
m_sBox(2, 217) = &H5121CE64
m_sBox(3, 217) = &H774FBE32
m_sBox(0, 218) = &HA8B6E37E
m_sBox(1, 218) = &HC3293D46
m_sBox(2, 218) = &H48DE5369
m_sBox(3, 218) = &H6413E680
m_sBox(0, 219) = &HA2AE0810
m_sBox(1, 219) = &HDD6DB224
m_sBox(2, 219) = &H69852DFD
m_sBox(3, 219) = &H9072166
m_sBox(0, 220) = &HB39A460A
m_sBox(1, 220) = &H6445C0DD
m_sBox(2, 220) = &H586CDECF
m_sBox(3, 220) = &H1C20C8AE
m_sBox(0, 221) = &H5BBEF7DD
m_sBox(1, 221) = &H1B588D40
m_sBox(2, 221) = &HCCD2017F
m_sBox(3, 221) = &H6BB4E3BB
m_sBox(0, 222) = &HDDA26A7E
m_sBox(1, 222) = &H3A59FF45
m_sBox(2, 222) = &H3E350A44
m_sBox(3, 222) = &HBCB4CDD5
m_sBox(0, 223) = &H72EACEA8
m_sBox(1, 223) = &HFA6484BB
m_sBox(2, 223) = &H8D6612AE
m_sBox(3, 223) = &HBF3C6F47
m_sBox(0, 224) = &HD29BE463
m_sBox(1, 224) = &H542F5D9E
m_sBox(2, 224) = &HAEC2771B
m_sBox(3, 224) = &HF64E6370
m_sBox(0, 225) = &H740E0D8D
m_sBox(1, 225) = &HE75B1357
m_sBox(2, 225) = &HF8721671
m_sBox(3, 225) = &HAF537D5D
m_sBox(0, 226) = &H4040CB08
m_sBox(1, 226) = &H4EB4E2CC
m_sBox(2, 226) = &H34D2466A
m_sBox(3, 226) = &H115AF84
m_sBox(0, 227) = &HE1B00428
m_sBox(1, 227) = &H95983A1D
m_sBox(2, 227) = &H6B89FB4
m_sBox(3, 227) = &HCE6EA048
m_sBox(0, 228) = &H6F3F3B82
m_sBox(1, 228) = &H3520AB82
m_sBox(2, 228) = &H11A1D4B
m_sBox(3, 228) = &H277227F8
m_sBox(0, 229) = &H611560B1
m_sBox(1, 229) = &HE7933FDC
m_sBox(2, 229) = &HBB3A792B
m_sBox(3, 229) = &H344525BD
m_sBox(0, 230) = &HA08839E1
m_sBox(1, 230) = &H51CE794B
m_sBox(2, 230) = &H2F32C9B7
m_sBox(3, 230) = &HA01FBAC9
m_sBox(0, 231) = &HE01CC87E
m_sBox(1, 231) = &HBCC7D1F6
m_sBox(2, 231) = &HCF0111C3
m_sBox(3, 231) = &HA1E8AAC7
m_sBox(0, 232) = &H1A908749
m_sBox(1, 232) = &HD44FBD9A
m_sBox(2, 232) = &HD0DADECB
m_sBox(3, 232) = &HD50ADA38
m_sBox(0, 233) = &H339C32A
m_sBox(1, 233) = &HC6913667
m_sBox(2, 233) = &H8DF9317C
m_sBox(3, 233) = &HE0B12B4F
```

```
m_sBox(0, 234) = &HF79E59B7
m_sBox(1, 234) = &H43F5BB3A
m_sBox(2, 234) = &HF2D519FF
m_sBox(3, 234) = &H27D9459C
m_sBox(0, 235) = &HBF97222C
m_sBox(1, 235) = &H15E6FC2A
m_sBox(2, 235) = &HF91FC71
m_sBox(3, 235) = &H9B941525
m_sBox(0, 236) = &HFAE59361
m_sBox(1, 236) = &HCEB69CEB
m_sBox(2, 236) = &HC2A86459
m_sBox(3, 236) = &H12BAA8D1
m_sBox(0, 237) = &HB6C1075E
m_sBox(1, 237) = &HE3056A0C
m_sBox(2, 237) = &H10D25065
m_sBox(3, 237) = &HCB03A442
m_sBox(0, 238) = &HE0EC6E0E
m_sBox(1, 238) = &H1698DB3B
m_sBox(2, 238) = &H4C98A0BE
m_sBox(3, 238) = &H3278E964
m_sBox(0, 239) = &H9F1F9532
m_sBox(1, 239) = &HE0D392DF
m_sBox(2, 239) = &HD3A0342B
m_sBox(3, 239) = &H8971F21E
m_sBox(0, 240) = &H1B0A7441
m_sBox(1, 240) = &H4BA3348C
m_sBox(2, 240) = &HC5BE7120
m_sBox(3, 240) = &HC37632D8
m_sBox(0, 241) = &HDF359F8D
m_sBox(1, 241) = &H9B992F2E
m_sBox(2, 241) = &HE60B6F47
m_sBox(3, 241) = &HFE3F11D
m_sBox(0, 242) = &HE54CDA54
m_sBox(1, 242) = &H1EDAD891
m_sBox(2, 242) = &HCE6279CF
m_sBox(3, 242) = &HCD3E7E6F
m_sBox(0, 243) = &H1618B166
m_sBox(1, 243) = &HFD2C1D05
m_sBox(2, 243) = &H848FD2C5
m_sBox(3, 243) = &HF6FB2299
m_sBox(0, 244) = &HF523F357
m_sBox(1, 244) = &HA6327623
m_sBox(2, 244) = &H93A83531
m_sBox(3, 244) = &H56CCCD02
m_sBox(0, 245) = &HACF08162
m_sBox(1, 245) = &H5A75EBB5
m_sBox(2, 245) = &H6E163697
m_sBox(3, 245) = &H88D273CC
m_sBox(0, 246) = &HDE966292
m_sBox(1, 246) = &H81B949D0
m_sBox(2, 246) = &H4C50901B
m_sBox(3, 246) = &H71C65614
m_sBox(0, 247) = &HE6C6C7BD
m_sBox(1, 247) = &H327A140A
m_sBox(2, 247) = &H45E1D006
m_sBox(3, 247) = &HC3F27B9A
m_sBox(0, 248) = &HC9AA53FD
m_sBox(1, 248) = &H62A80F00
m_sBox(2, 248) = &HBB25BFE2
m_sBox(3, 248) = &H35BDD2F6
m_sBox(0, 249) = &H71126905
m_sBox(1, 249) = &HB2040222
m_sBox(2, 249) = &HB6CBCF7C
m_sBox(3, 249) = &HCD769C2B
m_sBox(0, 250) = &H53113EC0
m_sBox(1, 250) = &H1640E3D3
m_sBox(2, 250) = &H38ABBD60
m_sBox(3, 250) = &H2547ADF0
m_sBox(0, 251) = &HBA38209C
m_sBox(1, 251) = &HF746CE76
m_sBox(2, 251) = &H77AFA1C5
m_sBox(3, 251) = &H20756060
m_sBox(0, 252) = &H85CBFE4E
m_sBox(1, 252) = &H8AE88DD8
m_sBox(2, 252) = &H7AAAF9B0
m_sBox(3, 252) = &H4CF9AA7E
m_sBox(0, 253) = &H1948C25C
m_sBox(1, 253) = &H2FB8A8C
m_sBox(2, 253) = &H1C36AE4
m_sBox(3, 253) = &HD6EBE1F9
m_sBox(0, 254) = &H90D4F869
m_sBox(1, 254) = &HA65CDEA0
m_sBox(2, 254) = &H3F09252D
m_sBox(3, 254) = &HC208E69F
m_sBox(0, 255) = &HB74E6132
m_sBox(1, 255) = &HCE77E25B
m_sBox(2, 255) = &H578FDFE3
m_sBox(3, 255) = &H3AC372E6
End Sub
```

```
Public Function EncryptFile(InFile As String, OutFile As String, Overwrite As
Boolean, Optional Key As String) As Boolean
   On Error GoTo errorhandler
   If FileExist(InFile) = False Then
      EncryptFile = False
      Exit Function
   End If
   If FileExist(OutFile) = True And Overwrite = False Then
      EncryptFile = False
      Exit Function
   End If
   Dim Buffer() As Byte, FileO As Integer
   FileO = FreeFile
   Open InFile For Binary As #FileO
      ReDim Buffer(0 To LOF(FileO) - 1)
      Get #FileO, , Buffer()
   Close #FileO
   Call EncryptByte(Buffer(), Key)
   If FileExist(OutFile) = True Then Kill OutFile
   FileO = FreeFile
   Open OutFile For Binary As #FileO
      Put #FileO, , Buffer()
   Close #FileO
   EncryptFile = True
   Exit Function

errorhandler:
   EncryptFile = False
End Function

Public Function DecryptFile(InFile As String, OutFile As String, Overwrite As
Boolean, Optional Key As String) As Boolean
   On Error GoTo errorhandler
   If FileExist(InFile) = False Then
      DecryptFile = False
      Exit Function
   End If
   If FileExist(OutFile) = True And Overwrite = False Then
      DecryptFile = False
      Exit Function
   End If
   Dim Buffer() As Byte, FileO As Integer
   FileO = FreeFile
   Open InFile For Binary As #FileO
      ReDim Buffer(0 To LOF(FileO) - 1)
      Get #FileO, , Buffer()
   Close #FileO
   Call DecryptByte(Buffer(), Key)
```

```
    Open OutFile For Binary As #FileO
      Put #FileO, , Buffer()
    Close #FileO
    DecryptFile = True
    Exit Function

errorhandler:
    DecryptFile = False
End Function

Private Function FileExist(FilePath As String) As Boolean
    On Error GoTo errorhandler
    Call FileLen(FilePath)
    FileExist = True
    Exit Function

errorhandler:
    FileExist = False
End Function
```

ASCII Table

(ASCII = American Standard Code for Information Interchange)

Decimal	Octal	Hex	Binary	Value	
000	000	000	00000000	NUL	(Null char \0)
001	001	001	00000001	SOH	(Start of Header)
002	002	002	00000010	STX	(Start of Text)
003	003	003	00000011	ETX	(End of Text)
004	004	004	00000100	EOT	(End of Transmission)
005	005	005	00000101	ENQ	(Enquiry)
006	006	006	00000110	ACK	(Acknowledgment)
007	007	007	00000111	BEL	(Bell \a)
008	010	008	00001000	BS	(Backspace \b)
009	011	009	00001001	HT	(Horizontal Tab \t)
010	012	00A	00001010	LF	(Line Feed \n)
011	013	00B	00001011	VT	(Vertical Tab \v)
012	014	00C	00001100	FF	(Form Feed \f)
013	015	00D	00001101	CR	(Carriage Return \r)
014	016	00E	00001110	SO	(Shift Out)
015	017	00F	00001111	SI	(Shift In)
016	020	010	00010000	DLE	(Data Link Escape)
017	021	011	00010001	DC1	(XON) (Device Control 1)
018	022	012	00010010	DC2	(Device Control 2)
019	023	013	00010011	DC3	(XOFF) (Device Control 3)
020	024	014	00010100	DC4	(Device Control 4)
021	025	015	00010101	NAK	(Negative Acknowledgement)
022	026	016	00010110	SYN	(Synchronous Idle)
023	027	017	00010111	ETB	(End of Trans. Block)
024	030	018	00011000	CAN	(Cancel)
025	031	019	00011001	EM	(End of Medium)
026	032	01A	00011010	SUB	(Substitute)
027	033	01B	00011011	ESC	(Escape)
028	034	01C	00011100	FS	(File Separator)
029	035	01D	00011101	GS	(Group Separator)
030	036	01E	00011110	RS	(Request to Send) (Record Separator)
031	037	01F	00011111	US	(Unit Separator)
032	040	020	00100000	SP	(Space)
033	041	021	00100001	!	(exclamation mark)
034	042	022	00100010	"	(double quote)
035	043	023	00100011	#	(number sign)
036	044	024	00100100	$	(dollar sign)
037	045	025	00100101	%	(percent)
038	046	026	00100110	&	(ampersand)
039	047	027	00100111	'	(single quote)
040	050	028	00101000	(	(left/opening parenthesis)
041	051	029	00101001	)	(right/closing parenthesis)
042	052	02A	00101010	*	(asterisk)
043	053	02B	00101011	+	(plus)
044	054	02C	00101100	,	(comma)
045	055	02D	00101101	-	(minus or dash)
046	056	02E	00101110	.	(dot)
047	057	02F	00101111	/	(forward slash)
048	060	030	00110000	0	
049	061	031	00110001	1	
050	062	032	00110010	2	
051	063	033	00110011	3	

052	064	034	00110100	4	
053	065	035	00110101	5	
054	066	036	00110110	6	
055	067	037	00110111	7	
056	070	038	00111000	8	
057	071	039	00111001	9	
058	072	03A	00111010	:	(colon)
059	073	03B	00111011	;	(semi-colon)
060	074	03C	00111100	<	(less than)
061	075	03D	00111101	=	(equal sign)
062	076	03E	00111110	>	(greater than)
063	077	03F	00111111	?	(question mark)
064	100	040	01000000	@	(AT symbol)
065	101	041	01000001	A	
066	102	042	01000010	B	
067	103	043	01000011	C	
068	104	044	01000100	D	
069	105	045	01000101	E	
070	106	046	01000110	F	
071	107	047	01000111	G	
072	110	048	01001000	H	
073	111	049	01001001	I	
074	112	04A	01001010	J	
075	113	04B	01001011	K	
076	114	04C	01001100	L	
077	115	04D	01001101	M	
078	116	04E	01001110	N	
079	117	04F	01001111	O	
080	120	050	01010000	P	
081	121	051	01010001	Q	
082	122	052	01010010	R	
083	123	053	01010011	S	
084	124	054	01010100	T	
085	125	055	01010101	U	
086	126	056	01010110	V	
087	127	057	01010111	W	
088	130	058	01011000	X	
089	131	059	01011001	Y	
090	132	05A	01011010	Z	
091	133	05B	01011011	[	(left/opening bracket)
092	134	05C	01011100	\	(back slash)
093	135	05D	01011101	]	(right/closing bracket)
094	136	05E	01011110	^	(caret/circumflex)
095	137	05F	01011111	_	(underscore)
096	140	060	01100000	`	
097	141	061	01100001	a	
098	142	062	01100010	b	
099	143	063	01100011	c	
100	144	064	01100100	d	
101	145	065	01100101	e	
102	146	066	01100110	f	
103	147	067	01100111	g	
104	150	068	01101000	h	
105	151	069	01101001	i	
106	152	06A	01101010	j	
107	153	06B	01101011	k	
108	154	06C	01101100	l	
109	155	06D	01101101	m	
110	156	06E	01101110	n	
111	157	06F	01101111	o	
112	160	070	01110000	p	

113	161	071	01110001	q	
114	162	072	01110010	r	
115	163	073	01110011	s	
116	164	074	01110100	t	
117	165	075	01110101	u	
118	166	076	01110110	v	
119	167	077	01110111	w	
120	170	078	01111000	x	
121	171	079	01111001	y	
122	172	07A	01111010	z	
123	173	07B	01111011	{	(left/opening brace)
124	174	07C	01111100	|	(vertical bar)
125	175	07D	01111101	}	(right/closing brace)
126	176	07E	01111110	~	(tilde)
127	177	07F	01111111	DEL	(delete)