SKRIPSI

ENKRIPSI DAN DEKRIPSI PENGIRIMAN PESAN SUARA DENGAN ALGORITMA *SERPENT*

Disusun Oleh

DWI PRASTANTO

07. 12. 625

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

CONTENTS

LEMBAR PERSETUJUAN

ENKRIPSI DAN DEKRIPSI PENGIRIMAN PESAN SUARA DENGAN ALGORITMA *SERPENT*

SKRIPSI

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

Disusun oleh :

DWI PRASTANTO

07. 12. 625

Mengetahui,

Ketua Jurusan Teknik Elektro S-1

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

Diperiksa dan Disetujui

Dosen Pembimbing I	Dosen Pembimbing II
Joseph Dedy Irawan, ST, MT	**M. Ibrahim Ashari, ST, MT**
NIP. 19740416.200501.1002	**NIP. P. 1030100358**

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

ABSTRAK

ENKRIPSI DAN DEKRIPSI PENGIRIMAN PESAN SUARA DENGAN ALGORITMA *SERPENT*

Dwi Prastanto, NIM 07.12.625
Dosen Pembimbing : Joseph Dedy Irawan, ST, MT dan M. Ibrahim Ashari, ST, MT

Dalam kemajuan teknologi saat ini sangat rentan terjadinya loss data. Dengan adanya masalah tersebut maka perlunya suatu penyandian atau enkripsi dan dekripsi data pesan suara sebelum dikirimkan sehingga bisa terjaga kerahasiaan dan privasi pengguna. Solusi yang ditawarkan ada berbagai macam.

Salah satu solusi pengamanan pesan data dengan algoritma serpent dimana memiliki blok *cipher simetris* kunci yang cukup kuat dalam hal keamanan dan memiliki tipe enkripsi *blok cipher* yang melakukan enkripsi dalam bentuk blok-blok bit.

Dalam pengiriman data di dukung dengan winsock control yang terdapat pada *visual basic 6.0* yang menyediakan layanan *protocol TCP/IP* sehingga memudahkan untuk bertukar informasi.

Kata Kunci: enkripsi, dekripsi, *serpent, winsock, TCP/IP*

SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini :

Nama	: Dwi Prastanto
NIM	: 07.12.625
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,

Dwi Prastanto
NIM : 07.12.625

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:

ENKRIPSI DAN DEKRIPSI PENGIRIMAN PESAN SUARA DENGAN ALGORITMA *SERPENT*

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 Joseph Dedy Irawan, ST, MT selaku Dosen Pembimbing I
4. Bapak M. Ibrahim Ashari, ST, MT selaku Dosen Pembimbing II
5. Bapak Ir. Yusuf Ismail Nahkoda, MT, selaku Dosen Wali.
6. Kedua orangtua dan kakakku yang telah memberikan dukungan untuk selalu berdoa, berusaha dan nasehat yang telah diberikan sampai saat ini.
7. Seluruh dosen dan pegawai ITN Kampus 2 Malang.
8. Semua teman-teman mahasiswa ITN Malang yang tidak mungkin

saya sebutkan satu-persatu, Anak-anak Budi jaya kos.

9. 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, Maret 2012

Dwi Prastanto

DAFTAR ISI

DAFTAR TABEL

BAB II LANDASAN TEORI

BAB III ANALISA DAN PERANCANGAN SISTEM

BAB IV IMPLEMENTASI DAN PENGUJIAN

DAFTAR GAMBAR

BAB II LANDASAN TEORI

BAB III ANALISA DAN PERANCANGAN SISTEM

BAB IV IMPLEMENTASI DAN PEMBAHASAN

BAB I
PENDAHULUAN

1.1. Latar Belakang

Saat ini komunikasi suara menjadi hal yang penting dalam kehidupan sehari-hari, seperti komunikasi suara dengan telepon yang berbasis analog dan telepon seluler yang berbasis digital. Berbagai alat komunikasi yang ada saat ini belum tentu aman untuk digunakan, karena belum ada standar keamanan yang dapat digunakan oleh alat-alat tersebut. Oleh karena itu, komunikasi ini sangat rentan terhadap serangan pihak ketiga yang seringkali sangat merugikan.

Salah satu solusi yang ditawarkan untuk permasalahan ini adalah enkripsi data komunikasi suara yang memiliki tingkat keamanan yang lebih tinggi dikarenakan dapat merubah isi bit pada pada setiap blok data dengan kunci. Enkripsi pada data digital ini dilakukan sebelum data dikirimkan sehingga pihak ketiga tidak dapat memahami arti dari data yang berhasil diambilnya. Proses enkripsi ini biasanya dilakukan oleh alat atau aplikasi pengenkripsi.

Ada berbagai macam algoritma enkripsi dengan karakteristiknya masing-masing yang dapat digunakan untuk proses enkripsi suara. Karena belum ada standart tertentu yang dapat digunakan, diperlukan usaha untuk menerapkan algoritma lain untuk mengetahui sebaik apa algoritma tersebut. Untuk proses enkripsi komunikasi dan pertukaran data suara digunakan algoritma yang mempunyai chiper aliran untuk mempercepat prosesnya.

1.2. Rumusan Masalah

Berdasarkan latar belakang pemilihan judul, maka yang menjadi permasalahan adalah bagaimana membuat enkripsi dan dekripsi pengiriman suara dengan menggunakan algoritma *serpent*.

1. Menganalisa kinerja dari algoritma *serpent* sejauh mana sistem keamanan dari data yang diterapkan dengan menggunakan algoritma tersebut dalam pertukaran data.

2. Mengimplementasi metode ini dengan menggunakan aplikasi sederhana yang telah dirancang untuk memberikan gambaran tentang model keamanan menggunakan algoritma enkripsi.

1.3. Tujuan

Tujuan penyusunan skripsi ini adalah untuk merancang enkripsi dan dekripsi pengiriman suara dengan algoritma yang digunakan yaitu :

1. Seperti apa algoritma serpent tersebut dan bagaimana proses enkripsi dan dekripsi.
2. Membuat aplikasi enkripsi suara berdasarkan algoritma serpent untuk enkripsi pengiriman pesan suara antara dua komputer melalui kabel jaringan.
3. Menganalisa algoritma ini untuk penyandian data sehingga nantinya dapat dikirim dan di dekripsi dengan baik.

1.4. Batasan Masalah

Dalam skripsi ini, ruang lingkup permasalahan dibatasi pada bagaimana membuat program aplikasi dalam bentuk perangkat lunak (software) yang dipergunakan untuk pengenkripsian pesan suara menggunakan algoritma *serpent* :

1. Dalam skripsi ini membahas mengenai proses penyandian pesan yang meliputi : proses enkripsi dan dekripsi pesan menggunakan algoritma *Serpent* serta mengimplementasikan dalam sebuah program sederhana.
2. Tidak membahas mengenai pemecahan kunci sandi (kriptanalisis).
3. Tidak membahas rumus perputaran kunci secara detail.
4. Jenis topologi dalam proses pengiriman pesan suara adalah peer to peer.
5. Program aplikasi yang dibuat hanya berfungsi untuk enkripsi 128 bit, perekaman suara dan pengiriman data.
6. Tidak membahas perbandingan dengan algoritma lain.
7. Pada pengiriman pesan saya asumsikan di simpan di drive C.
8. Tipe format suara hanya berbentuk WAV.
9. Effect setting suara hanya tambahan.

10. Aplikasi dibuat menggunakan visual basic 6.0.

1.5. Metodologi Penelitian

Langkah-langkah pembuatan perangkat lunak ini antara lain :

a. Studi literatur

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

b. Analisa Kebutuhan Sistim

Data dan informasi yang telah diperoleh akan dianalisa agar didapatkan kerangka global yang bertujuan untuk mendefinisikan kebutuhan sistim di mana nantinya akan digunakan sebagai acuan perancangan sistim.

c. Perancangan dan Implementasi

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

d. Eksperimen dan Evaluasi

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

1.6. Sistimatika Penulisan

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

Bab I : Pendahuluan

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

Bab II : Landasan Teori

Berisi tentang landasan teori mengenai permasalahan yang

10. Aplikasi dibuat menggunakan visual c# 2010

1.5. Metodologi Penelitian

Langkah-langkah pembuatan perangkat lunak ini antara lain :

a. Studi literatur

Pengumpulan data yang dilakukan antara melewati beberapa sub-bagian berkaitan dan refere[illegible] sebagai berikut pengertian yang ada hubungannya dengan pengenalan yang dijadikan objek penelitian.

b. Analisa Kebutuhan Sistem

Data dan informasi yang telah diperoleh akan dianalisa agar ditemukan beberapa global yang berpengaruh untuk memecahkan kebutuhan sistim. di mana tentunya akan dijumpai sebagai acuan pembangun sistim

c. Perancangan dan Implementasi

Berdasarkan data dan informasi yang telah diperoleh serta analisa kebutuhan untuk membangun sistim maka akan dilakukan perancangan sistim yang mengganbarkan mekanisme dan sistim secara umum dan lebih detail sehingga memudahkan dalam pemrograman sistim.

d. Eksperimen dan Evaluasi

Pada tahap ini sistim yang telah selesai dibuat akan diuji coba, yaitu pengujian berdasarkan fungsionalitas program, dan akan dilakukan evaluasi 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 Sistimatika Penulisan

Bab II : Landasan Teori

Berisi tentang landasan teori mengenai pengenalan yang

berhubungan dengan penelitian yang dilakukan.

Bab III : Anlisa Dan Perancangan Sistem

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

Bab IV : Implementasi dan Pengujian Sistem

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

berhubungan dengan penelitian yang dilakukan.

Bab III : Analisa Dan Perancangan Sistem

Dalam bab ini berisi pengenai analisa kebutuhan sistim software yang diperlukan untuk pembuatan komputer global yang menggambarkan mekanisme dari sistim yang akan dibuat.

Bab IV : Implementasi dan Penjelasan Sistem

Berisi tentang implementasi dari perancangan sistim yang telah dibuat serta penjelasan terhadap sistim tersebut.

Bab V : Penutup

Merupakan bab terakhir yang memuat isiasni dari hasil pembahasan yang berisikan kesimpulan dan saran yang dapat digunakan sebagai perimbangan untuk pengembangan penelitian selanjutnya.

BAB II
LANDASAN TEORI

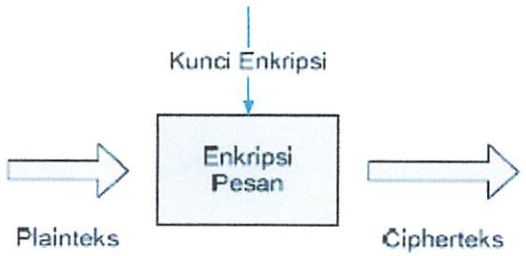
Dalam pembuatan aplikasi enkripsi dan dekripsi pengiriman pesan suara dengan algoritma *Serpent* ini, mengacu pada beberapa dasar teori yang mendukung sistem kerja dari aplikasi. Adapun dasar teori yang digunakan dalam perancangan aplikasi ini adalah sebagai berikut.

2.1 Sistem Kriptografi

Untuk menjamin keamanan pertukaran data, berbagai proses dilakukan terhadap data, salah satunya adalah proses penyandian. Proses penyandian dilakukan untuk membuat data yang dikirimkan tidak dapat dimengerti oleh pihak lain selain yang memiliki akses terhadap data tersebut. Proses penyandian terdiri atas dua tahapan, yaitu [2]:

1. Enkripsi

Enkripsi merupakan proses untuk mengubah plainteks menjadi cipherteks yang tidak bisa dimengerti. Proses enkripsi biasanya dilakukan sebelum pesan dikirimkan. Untuk meningkatkan keamanan enkripsi pesan, pada proses enkripsi ditambahkan kunci yang juga diperlukan untuk proses dekripsi, seperti pada Gambar 2.1.

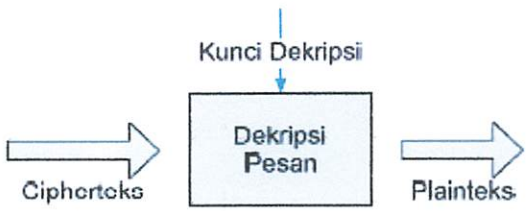


Gambar 2.1 Proses Enkripsi Dengan Kunci

2. Deskripsi

Dekripsi merupakan proses untuk mengubah cipherteks kembali menjadi plainteks agar pesan dapat dimengerti. Proses dekripsi biasanya dilakukan oleh penerima pesan agar pesan yang diterima dapat dimengerti.

Untuk proses enkripsi yang menggunakan kunci maka proses dekripsi harus dilakukan dengan menggunakan kunci, seperti pada Gambar 2.2.



Gambar 2.2 Proses Dekripsi Dengan Kunci

2.2 Algoritma Kriptografi

Algoritma kriptografi adalah algoritma yang berfungsi untuk melakukan tujuan kriptografis. Algoritma tersebut harus memiliki kekuatan untuk melakukan :

- Kerahasiaan, adalah layanan yang digunakan untuk menjaga isi dari informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka/mengupas informasi yang telah disandi.
- Integritas data, adalah berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak-pihak yang tidak berhak, antara lain penyisipan, penghapusan, dan pensubsitusian data lain kedalam data yang sebenarnya.
- Autentikasi, adalah berhubungan dengan identifikasi/pengenalan, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri. Informasi yang dikirimkan melalui kanal harus diautentikasi keaslian, isi datanya, waktu pengiriman, dan lain-lain.

2. Deskripsi

Deskripsi merupakan proses untuk menguraikan objek-objek kertasoli menjadi pisinoks agar pesan dapat dimengerti. Proses deskripsi biasanya dilakukan oleh penerima pesan agar pesan yang diterimanya dapat dimengerti.

Untuk proses enkripsi yang menggunakan kunci maka proses deskripsi harus dilakukan dengan menggunakan kunci, seperti pada Gambar 2.2.

Gambar 2.2 Proses Deskripsi Dengan Kunci

2.2 Algoritma Kriptografi

Algoritma kriptografi adalah algoritma yang berfungsi untuk melakukan tujuan kriptografis. Algoritma tersebut harus memiliki kekuatan untuk melakukan :

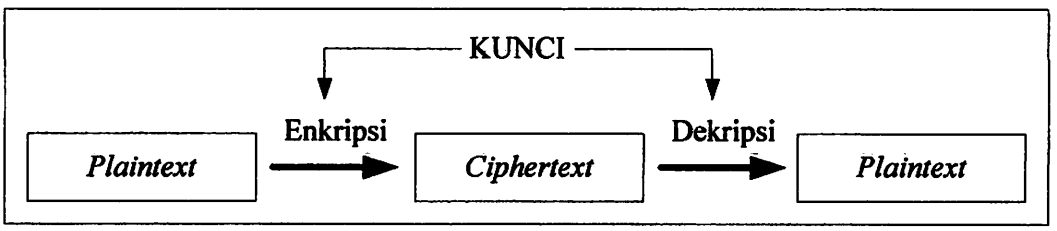
- Kerahasiaan, adalah layanan yang digunakan untuk menjaga isi dari informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membukanya/mengupas informasi yang telah disandi.
- Integritas data, adalah berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk itu pihak penerima pesan sistem harus mengetahui kemungkinan untuk mendeteksi manipulasi data oleh pihak-pihak yang tidak berhak, antara lain penyisipan, penghapusan, dan pensubstitusian data lain kedalam data yang sebenarnya.
- Autentikasi, adalah berhubungan dengan identifikasi/pengenalan, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling mengenalkan diri. Informasi yang didistribusikan melalui kanal harus diautentikasi keaslian, isi datanya, waktu pengiriman, dan lain-lain.

- Non-repudiasi., atau nirpenyangkalan adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman/terciptanya suatu informasi oleh yang mengirimkan/membuat.

Sehingga dapat digunakan untuk mengamankan informasi. Pada implementasinya sebuah algoritma sandi harus memperhatikan kualitas layanan/Quality of Service atau QoS dari keseluruhan sistem dimana dia diimplementasikan. Algoritma sandi yang handal adalah algoritma sandi yang kekuatannya terletak pada kunci, bukan pada kerahasiaan algoritma itu sendiri. Teknik dan metode untuk menguji kehandalan algoritma sandi adalah kriptanalisa [1].

2.2.1 Algoritma Simetris

Algoritma kriptografi simetris atau disebut juga algoritma kriptografi konvensional. Algoritma ini menggunakan kunci yang sama untuk proses enkripsi dan proses dekripsi.



Gambar 2.3 Prosedur kerja algoritma simetris

Algoritma kriptografi simetris dibagi menjadi 2 kategori yaitu algoritma aliran (*Stream Ciphers*) dan algoritma blok (*Block Ciphers*). Pada algoritma aliran, proses penyandiannya berorientasi pada satu bit atau satu *byte* data. Sedang pada algoritma blok, proses penyandiannya berorientasi pada sekumpulan *bit* atau *byte* data (per blok). Contoh algoritma kunci simetris yang terkenal adalah DES (*Data Encryption Standard*).

Block-cipher adalah skema algoritma sandi yang akan membagi-bagi teks terang yang akan dikirimkan dengan ukuran tertentu (disebut blok) dengan panjang t, dan setiap blok dienkripsi dengan menggunakan kunci yang sama.

Non-repudiasi, akan dipertanyakan apakah suatu pesan benar-benar berasal dari pengirim yang sah atau tidak. Hal ini dapat terjadi karena pengirim dapat menyangkal bahwa ia telah mengirimkan pesan tersebut.

Sebagian besar algoritma kunci publik berdasarkan pada permasalahan komputasi yang rumit. Misalnya, pemfaktoran bilangan bulat yang sangat besar, perhitungan logaritma diskret dan perhitungan aritmatika pada kurva eliptik. Algoritma kunci publik yang populer adalah algoritma RSA yang keamanannya terletak pada sulitnya melakukan faktorisasi bilangan yang sangat besar. Sedangkan algoritma ElGamal dan metode kunci rahasia Diffie-Hellman algoritma kunci publik [1].

2.2.1 Algoritma Simetris

Algoritma kriptografi simetris atau disebut juga algoritma kriptografi konvensional. Algoritma ini menggunakan kunci yang sama untuk proses enkripsi dan proses dekripsi.

Gambar 2.1 Proses dari kedua algoritma simetris

Algoritma kriptografi simetris dibagi menjadi 2 kategori yaitu algoritma aliran (Stream Cipher) dan algoritma blok (Block Cipher). Pada algoritma aliran, proses pengamanannya berorientasi pada satu bit atau satu byte data. Sedang pada algoritma blok, proses pengamanannya dilakukan pada sekumpulan bit atau byte data (per blok). Contoh algoritma kunci simetris yang terkenal adalah DES (Data Encryption Standard).

Block-cipher adalah skema algoritma sandi yang akan membagi-bagi teks terang yang akan dienkripsikan dengan ukuran tertentu (disebut blok) dengan panjang n dan setiap blok dienkripsi dengan menggunakan kunci yang sama.

Pada umumnya, block-cipher memproses teks terang dengan blok yang relatif panjang lebih dari 64 bit, untuk mempersulit penggunaan pola-pola serangan yang ada untuk membongkar kunci [12]. Ada beberapa algoritma modern yang menggunakan algoritma simetris diantaranya adalah : AES, SERPENT, BLOWFISH dan lain-lain.

Algoritma *Serpent* adalah blok cipher simetris kunci yang merupakan finalis dalam kontes Lanjutan Encryption Standard (AES), di mana ia datang kedua untuk Rijndael. *Serpent* dirancang oleh Ross Anderson, Eli Biham, dan Lars Knudsen [12].

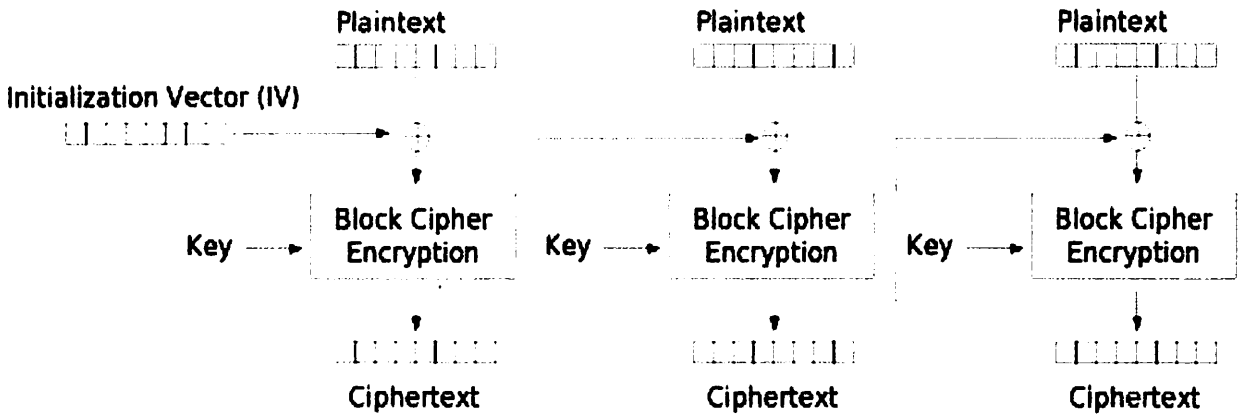
Serpent AES adalah *symmetric cryptography* yang memproses ukuran *block* 128 bit dengan ukuran *key* 128 bit. *Serpent* merupakan finalis dari lomba *AES* yang dimenangkan oleh *Rijndael*. *Serpent* mempunyai kompleksitas yang lebih rumit sehingga *Serpent* lebih lambat dibandingkan dengan *Rijndael*. Meskipun demikian keamanan *Serpent* lebih kompleks dibandingkan dengan kriptografi *Rijndael* sendiri.

Keamanan *Serpent* telah terbukti, hingga hanya dapat diserang dengan pencarian *brute-force*. Berdasarkan analisis yang ada, untuk mendapatkan sebuah *key* berukuran 256 bit, sebuah *supercomputer* membutuhkan waktu 256 bit. Hingga saat ini algoritma *Serpent* belum dipatenkan sebagai sebuah *AES* sehingga semua orang dapat menggunakan *Serpent* untuk mengamankan data data mereka ataupun mempelajari *Serpent* untuk mengembangkan ilmu kriptografi [12].

Algoritma cipher block *Serpent* adalah algoritma dengan 32 putaran jaringan SP yang beroperasi pada empat *word* 32 bit, yang berarti ukuran bloknya adalah 128 bit. Untuk komputasi internal, semua nilai direpresentasikan dalam *little-endian*, di mana word pertama adalah *least-significant word*, dan *word terakhir* adalah *most-significant word* [1].

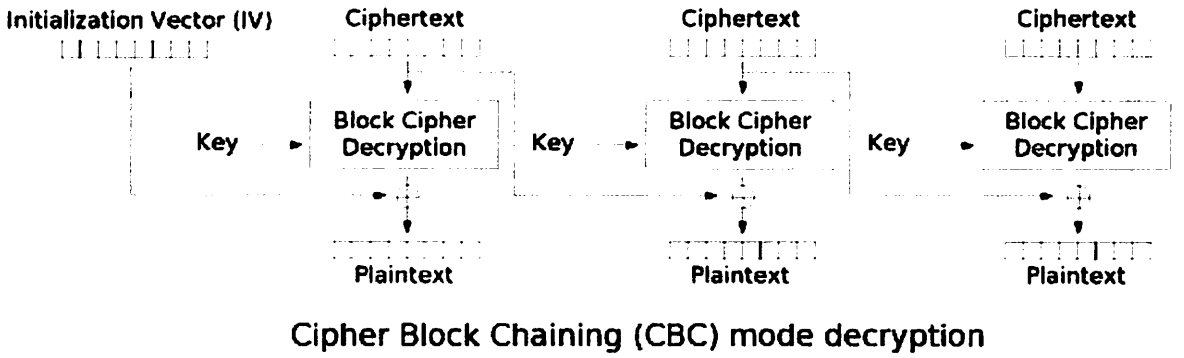
Chipper blok yang digunakan adalah CBC (Cipher-block chaining) Di dalam model ini, data tetap dibagi menjadi beberapa blok, masing-masing 64 bit. Perbedaannya adalah input dari algoritma DES adalah hasil operasi XOR dari chipertext yang telah dihasilkan sebelumnya dengan plaintext yang akan dienkripsi. Untuk enkripsi pada 64 bit blok pertama diperlukan vektor inisialisasi yang digunakan untuk operasi XOR dengan plaintext yang akan dienkripsi. Maka, baik pengirim maupun penerima harus mengetahui nila vektor inisialisasi untuk dapat melakukan

enkripsi maupun dekripsi. Aplikasi yang biasa diimplentasikan adalah aplikasi transmisi data yang berbasis blok atau aplikasi otentifikasi [12].



Gambar 2.4 CBC (Cipher-block chaining) mode enkripsi

Algoritma *Serpent* mengenkripsi plainteks P 128 bit menjadi cipherteks C 128 bit dalam 32 putaran dengan kontrol dari 33 sub-kunci 128 bit K0,...,K32. Panjang kunci masukan user 128, 192, dan 256 bit. Kunci yang lebih pendek dari 256 bit dipetakan menjadi kunci sepanjang 256 bit dengan menambahkan satu "1" bit pada akhir MSB, dan diikuti dengan "0" bit sampai mencapai 256 bit.



Gambar 2.5 CBC (Cipher-block chaining) mode dekripsi

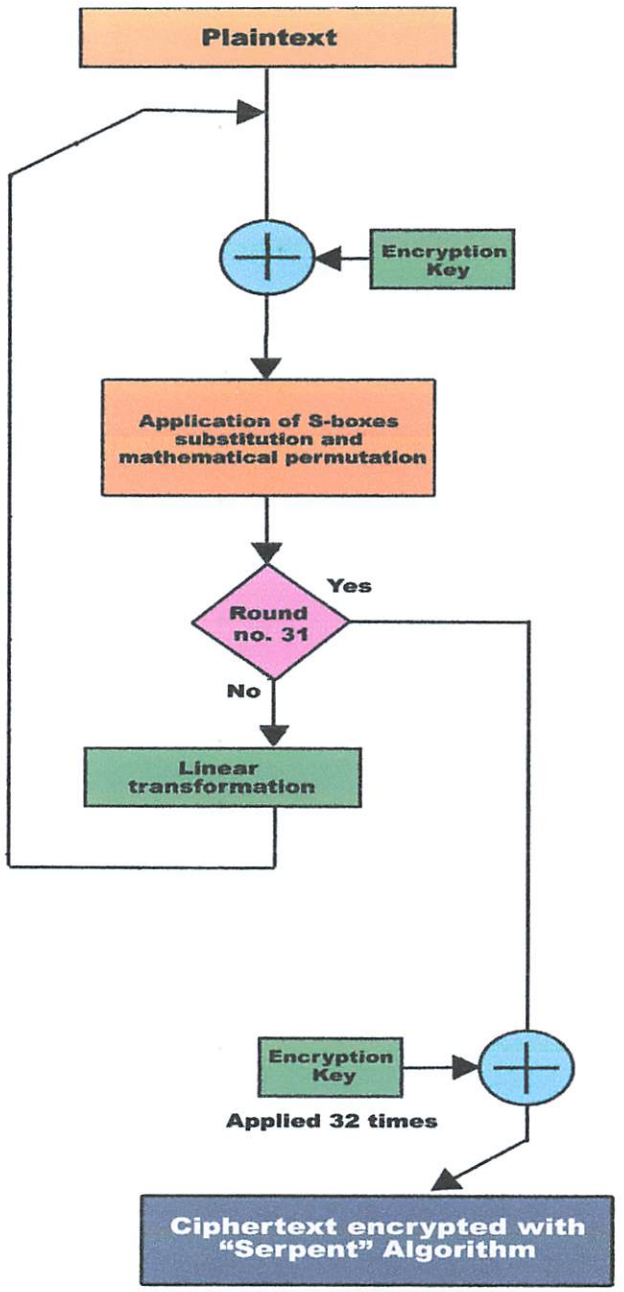
enkripsi maupun dekripsi. Aplikasi yang biasa diimplementasikan adalah aplikasi transmisi data yang berbasis blok non aplikasi enkripsi [12].

Gambar 2.4 CBC (Cipher-block chaining) mode enkripsi

Algoritma dengan menggunakan plaintext P 128 bit menjadi ciphertext C 128 bit dalam 32 putaran dengan control dari 32 sub-kunci 128 bit K0, ..K31. Panjang kunci masukan user 128, 192, dan 256 bit. Untuk yang lebih pendek dari 256 bit dipenuhkan menjadi kunci sepanjang 256 bit dengan menambahkan satu "1" bit pada akhir kunci dan diikuti dengan "0" bit sampai mencapai 256 bit.

Gambar 2.5 CBC (Cipher-block chaining) mode dekripsi

1. Arsitektur global enkripsi Algoritma *Serpent* :

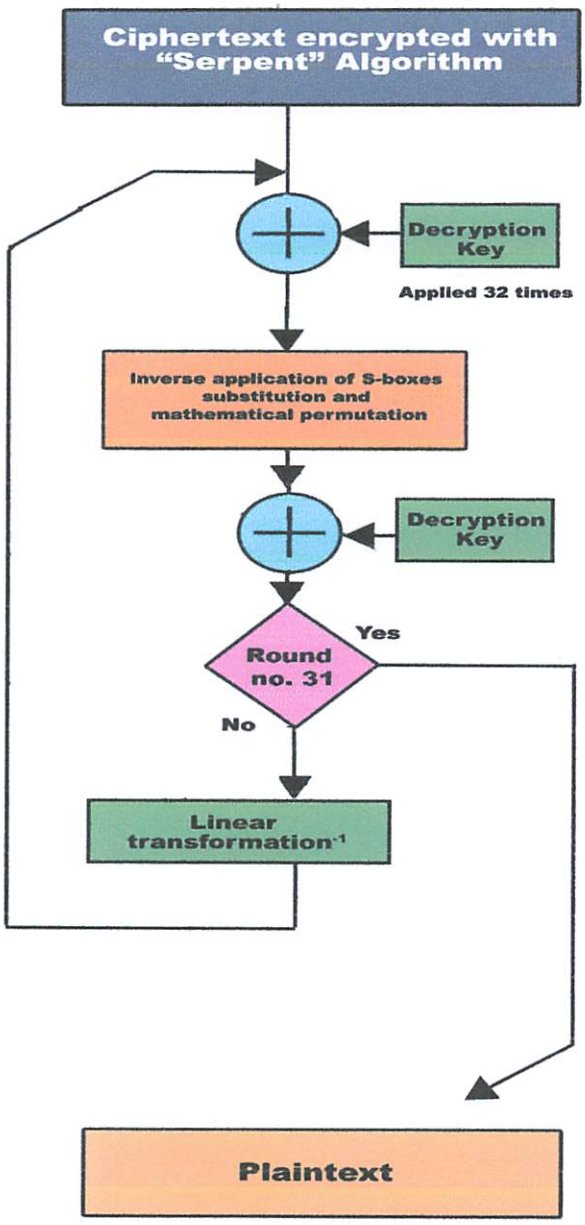


Gambar 2.6 Arsitektur global enkripsi Algoritma *Serpent*

1. Arsitektur global optimasi Algoritma Semut

Gambar 2.6 Arsitektur global optimasi Algoritma Semut

2. Arsitektur global dekripsi Algoritma *Serpent* :



Gambar 2.7 Arsitektur global dekripsi Algoritma *Serpent*

Menurut [9] Algoritma *Serpent* ini terdiri dari:

1. IP atau Initial Permutation diterapkan pada plainteks P menghasilkan B0, yang merupakan input dari putaran pertama, yaitu putaran-0 (putaran diberi nomor

dari 0 sampai 31). Hasil dari putaran pertama (putaran-0) dinamakan B1, hasil putaran kedua (putaran-1) dinamakan B2, dan seterusnya sampai B32. Permutasi akhir akan menghasilkan cipherteks C. Masing-masing fungsi putaran Ri (i = 0,...,31) hanya menggunakan sebuah S-Box ter-replikasi. Misalnya, R0 menggunakan S0, 32 copy yang diterapkan secara paralel, sehingga copy dari S0 menggunakan bit 0,1,2, dan 3 dari B0 ⊕ K0 sebagai input dan mengembalikan empat bit pertama dari vektor intermediate sebagai output, copy selanjutnya menerima masukan bit ke 4-7 dari B0 ⊕ K0 dan mengembalikan empat bit selanjutnya dari vektor intemediate, dan seterusnya. Vektor intermediate kemudian ditransformasi menggunakan linear transformasi, menghasilkan B1. Dengan cara yang sama, R1 menggunakan 32 copy S1 secara paralel pada B1 ⊕ K1 dan mentransformasi outputnya menggunakan transformasi linear, menghasilkan B2.

2. Himpunan delapan S-Box digunakan sebanyak empat kali. Oleh karena itu, setelah menggunakan S7 pada putaran-7, S0 digunakan kembali pada putaran-8, kemudian S1 pada putaran-9, dan seterusnya. Putaran terakhir R31 sedikir berbeda dengan lainnya, yaitu dengan menggunakan S7 pada B31 ⊕ K31, dan mengXORkan hasilnya dengan K32, bukan menggunakan transformasi linear. Hasil B32 kemudian dipermutasikan dengan FP, menghasilkan cipherteks. Setiap putaran menggunakan 9 S-Box yang berbeda yang memetakan empat input bit ke empat output bit. Masing-masing S-Box digunakan tepat pada empat putaran, dan masing-masing digunakan 32 kali secara paralel.
3. Final Permutation (FP) adalah invers dari permutasi inisial. Dengan demikian, cipher secara formal dapat dideskripsikan sebagai berikut.

Bo := IP(P)

Bi+1 := Ri(Bi)

C := FP(B32)

di mana:

Ri(X) = L(S(X ⊕ Ki)) i = 0,...,30

Ri(X) = Si(S ⊕ Ki) K32 i = 31

dan 0 sampai 31). Hasil dari putaran pertama (putaran-0) dinamakan B1. hasil putaran kedua (putaran-1) dinamakan B2. dan seterusnya sampai B32. Permutasi akhir akan menghasilkan ciphertek C. Masing-masing fungsi putaran Ri (i = 0,...,31) hanya menggunakan sebuah S-Box (s-replikasi. Misalnya. S0 menggunakan S0. S2 juga yang ditunjukan secara paralel. sehingga copy dari S0 menyumbang bit B1,2. dan 3 dari B0 ⊕ K0 sebagai input dan menghasilkan empat bit pertama dari vektor intermediate sebagai output. copy selanjutnya menerima masukan bit ke 4-7 dari B0 ⊕ K0 dan menghasilkan empat bit selanjutnya dari vektor intermediate. dan seterusnya. Vektor intermediate kemudian di transformasi menggunakan linear transformasi menghasilkan B1. Perjalan cara yang sama B1 menggunakan S2 copy S1 secara paralel pada B1 ⊕ K1 dan menghasilkan output. menggunakan transformasi linear menghasilkan B2.

2. Implemen dengan S-Box dipendekan sebanyak empat kali. Cara kerja ini sesuai menggunakan S7 pada putaran-7. S0 digunakan kembali pada putaran-8. kemudian S1 pada putaran-9. dan seterusnya. Putaran terakhir R31 sedikit berbeda dengan lainnya. yaitu dengan menggunakan S7 pada B31 ⊕ K31. dan menyXORkan hasilnya dengan K32. bukan menggunakan transformasi linear. Hasil B32 kemudian dipermutasikan dengan IP. menghasilkan ciphertek. Setiap putaran menggunakan 8 S-Box yang berbeda yang memetakan empat input bit ke empat output bit. Masing-masing S-Box dipetakan tepat pada empat putaran. dan masing-masing dipetakan 32 kali secara paralel.

3. Final Permutation (IP): adalah invers dari permutasi initial. Dengan demikian. cipher secara formal dapat didefinisikan sebagai berikut:

$B_0 := IP(P)$

$B_{i+1} := R_i(B_i)$

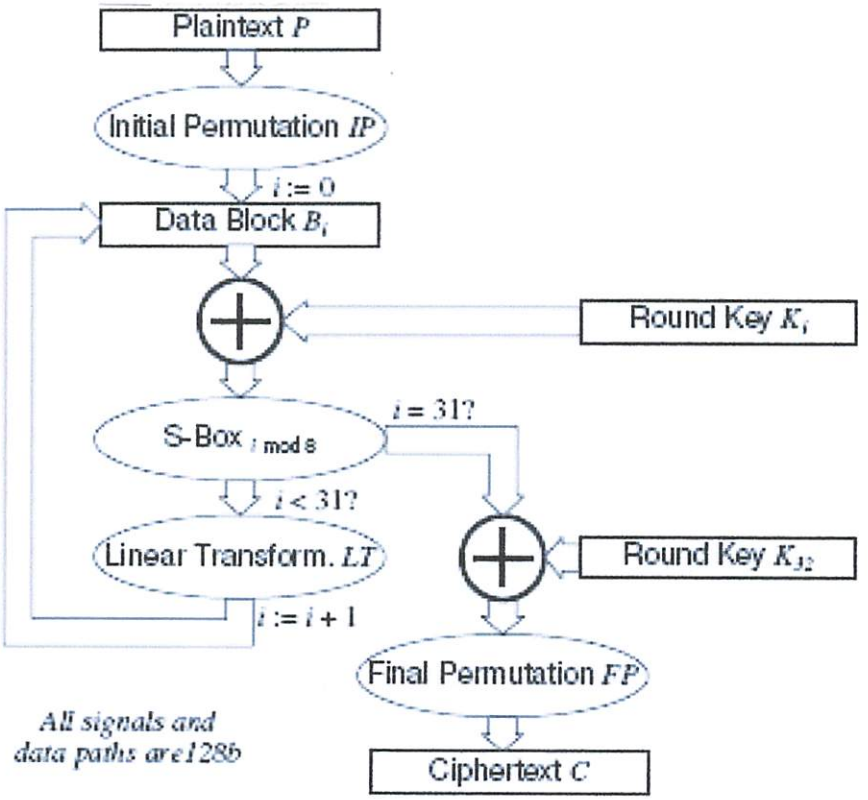
$C := IP(B_{32})$

di mana:

$R_i(X) = L_i(S_i(X \oplus K_i)), i = 0, \ldots, 31$

$R_{32}(X) = S_{31}(X \oplus K_{31}) \, K_{32}, i = 31$

di mana Si adalah aplikasi S-Box Si mod 8 32 kali secara paralel, dan L adalah transformasi linear.



Gambar 2.8

Pengolahan Data Pada Algoritma *Serpent*

Sumber : http://en.wikipedia.org/wiki/Serpent_(cipher)

di mana SI adalah aplikasi S-Box SI mod 8 32 kali secara paralel, dan L adalah transformasi linear.

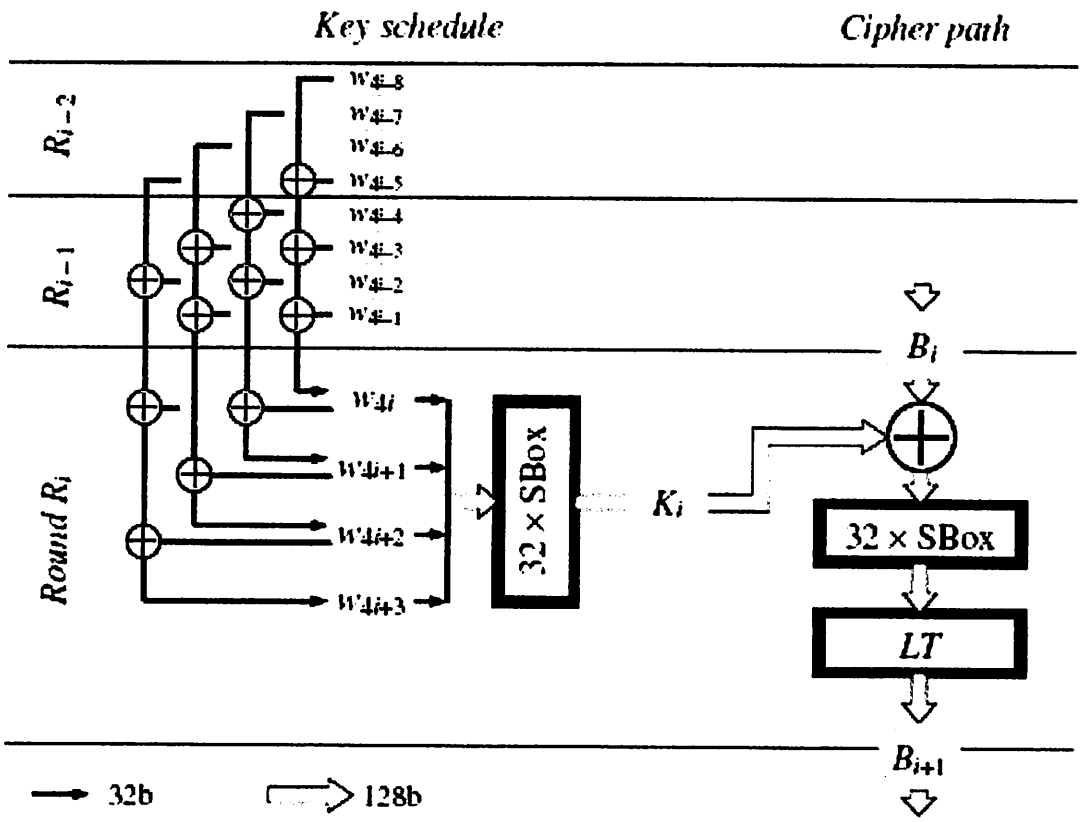
Gambar 2.8

Rangkaian Data Pada Algoritma Serpent

Sumber :

2.3 Pengacakan Kunci

Tugas utama dari pengacakan kunci adalah untuk menghasilkan 33 putaran Ki kunci dari kunci eksternal K yang diberikan oleh pengguna. Key K dapat dari hampir setiap panjang tetapi ketika usulan untuk AES standart dirumuskan tetap pada bit 128, 192 atau 256 dengan prosedur ekspansi khusus yang diterapkan untuk kunci [14].



Gambar 2.9 Pengacakan kunci (Round key)

Serpent memerlukan 133 32-bit word sebagai material kunci. Sebagai langkah awal, padding kunci input menjadi 256 bit, dengan menambahkan bit 1 diakhir kunci input dilanjutkan dengan bit0 sampai panjang bit menjadi 256 bit. Kemudian dilakukan pembangkitan kunci sebagai berikut :

- Tulis K sebagai 8 32-bit word w-8,..., w-1
- Expand ke bentuk prekey w0,..., w 131 melalui fungsi affine:

- wi:= (wi-8 wi-5 wi-3 wi-1 Ø i) < < < 1 1
- Transformasi Wi menjadi Ki dengan S-Box :

$$\{k_0, k_1, k_2, k_3\} := S_3(w_0, w_1, w_2, w_3)$$
$$\{k_4, k_5, k_6, k_7\} := S_2(w_4, w_5, w_6, w_7)$$
$$\{k_8, k_9, k_{10}, k_{11}\} := S_1(w_8, w_9, w_{10}, w_{11})$$
$$\{k_{12}, k_{13}, k_{14}, k_{15}\} := S_0(w_{12}, w_{13}, w_{14}, w_{15})$$
$$\{k_{16}, k_{17}, k_{18}, k_{19}\} := S_7(w_{16}, w_{17}, w_{18}, w_{19})$$
$$\ldots$$
$$\{k_{124}, k_{125}, k_{126}, k_{127}\} := S_4(w_{124}, w_{125}, w_{126}, w_{127})$$
$$\{k_{128}, k_{129}, k_{130}, k_{131}\} := S_3(w_{128}, w_{129}, w_{130}, w_{131})$$

- Subkey adalah Ki dimana Ki:= {k4i, k4i+1, k4i+2, k4i+3 }
- Roundkey adalah K^I = IP(Ki)

3.1.3 Subtitution-Box

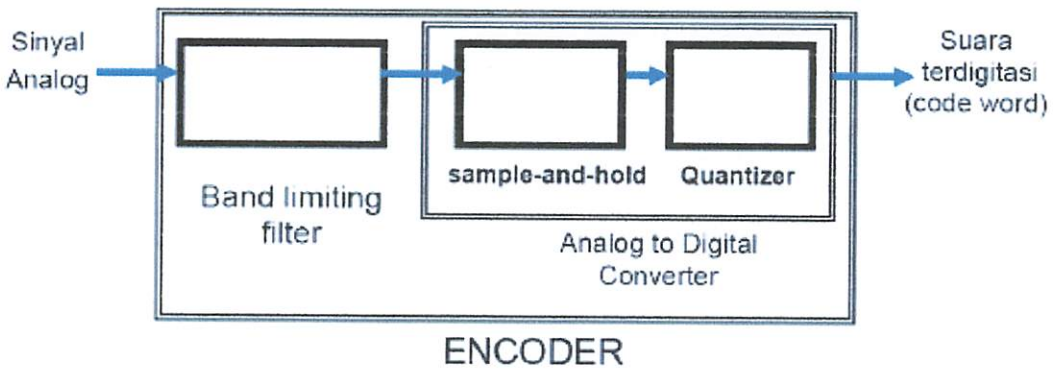
Substitution-Box atau S-Box *Serpent* adalah permutasi 4 bit dengan ketentuan :

1. Setiap karakteristik diferensial memiliki probabilitas maksimal ¼, dan sebuah input dengan perbedaan satu bit tidak akan meghasilkan output dengan perbedaan satu bit
2. Setiap karakteristik linear memiliki probabilitas antara ½ ± ¼, dan hubungan linear antara sebuah bit pada input dan sebuah bit pada output memiliki probabilitas ½ ±1/8
3. Urutan non-linear bit output sebagai fungsi dari bit input maksimal 3.

S-Box dibangkitkan dengan menggunakan matriks dengan 32 array yang masing-masing memiliki 16 entri. Matriks diinisialisasi dengan 32 baris S-Box DES dan ditransformasikan dengan menukar entri pada array ke-r bergantung pada nilai entri ke-(r+1) array dan pada inisial string yang merepresentasikan kunci. Jika array hasilnya memenuhi ketentuan yang telah disebutkan sebelumnya, maka simpan array sebagai *Serpent* S-Box. Ulangi prosedur tadi sampai 8 S-Box berhasil dibangkitkan.

2.4 SUARA atau SOUND (AUDIO)

Gelombang suara analog tidak dapat langsung direpresentasikan pada komputer. Komputer mengukur amplitudo pada satuan waktu tertentu untuk menghasilkan sejumlah angka. Tiap satuan pengukuran ini dinamakan "*sample*". *Analog To Digital Conversion* (ADC) adalah proses mengubah amplitudo gelombang bunyi ke dalam waktu interval tertentu (*sampling*), sehingga menghasilkan representasi digital dari suara. Dalam teknik *sampling* dikenal istilah *sampling rate* yaitu beberapa gelombang yang diambil dalam satu detik. Sebagai contoh jika kualitas CD Audio dikatakan memiliki frekuensi sebesar 44100 Hz, berarti jumlah *sampel* sebesar 44100 per detik.



Gambar 2.10 Proses *Sampling Audio Analog* ke *Digital*

Langkah-langkah dalam proses digitalisasi adalah:

1. Membuang frekuensi tinggi dari *source signal*.
2. Mengambil *sample* pada interval waktu tertentu (*sampling*).
3. Menyimpan amplitudo sampel dan mengubahnya ke dalam bentuk diskrit (kuantisasi).
4. Merubah bentuk menjadi nilai biner.

Teknik sampling yang umum pada file *audio* seperti *Nyquist Sampling Rate* dimana untuk memperoleh representasi akurat dari suatu sinyal analog secara *lossless*, amplitudonya harus diambil *sample*-nya setidaknya pada kecepatan (*rate*) sama atau lebih besar dari 2 kali lipat komponen frekuensi maksimum yang akan didengar.

2.4 SINYAL atau SOUND (Al DIG)

Sebelumnya suara analog tidak dapat langsung ditepresentasikan pada komputer. Komputer mengakui sinyal hanya pada bentuk sinyal tertentu untuk menghasilkan sejumlah angka. Tiap suatan perubahan ini dinamakan "sampel". Analog To Digital Conversion (ADC) adalah proses mengubah amplitudo gelombang sinyal ke dalam waktu interval tertentu (sampling), sehingga menghasilkan representasi digital dari suara. Dalam teknik sampling dikenal istilah sampling rate yaitu jumlah gelombang yang diambil dalam satu detik. Sehingga satuan jika kualitas CD Audio dikatakan memiliki frekuensi sebesar 44100 Hz berarti jumlah sampel sebesar 44100 per detik.

Gambar 2.10 Proses Sampling Audio Analog ke Digital

Langkah-langkah dalam proses digitalisasi adalah:

1. Membuang frekuensi tinggi dari sinyal asal.
2. Mengambil sampel pada interval waktu tertentu (sampling).
3. Menghitung amplitudo sampel dan mengubahnya ke dalam bentuk diskrit (kuantisasi).
4. Menyimpan bentuk tersebut ke dalam bit.

Teknik sampling yang umum pada file audio seperti .wav/.mp3 Sampling Rate dimana untuk menyatakan representasi kurva dari suatu sinyal analog secara kasar, amplitudonya harus diambil sampel-sampel sesekalinya pada kecepatan (rate) sama atau lebih besar dari 2 kali lipat komponen frekuensi maksimum yang akan direpre-

Mis : Untuk sinyal analog dengan *bandwith* 15 Hz – 10 kHz → *sampling rate* = 2 x 10 KHz = 20 kHz.

File output pada proses sampling analog ke digital ada beberapa

- WAV adalah format audio standar Microsoft dan IBM untuk PC.
- WAV biasanya menggunakan coding PCM (Pulse Code Modulation)
- WAV adalah data tidak terkompres sehingga seluruh sampel audio disimpan semuanya di harddisk.
- Software yang dapat menciptakan WAV dari Analog Sound misalnya adalah Windows Sound Recorder.
- WAV jarang sekali digunakan di internet karena ukurannya yang relative besar.
- Maksimal ukuran file WAV adalah 2GB.

Secara umum data *audio* digital memiliki karakteristik yang dapat dinyatakan dengan parameter-parameter berikut:

a. Laju sampel (*sampling rate)* dalam sampel/detik, misalnya 22050 atau 44100 sampel/detik.

b. Jumlah bit tiap sampel, misalnya 8 atau 16 bit.

c. Jumlah kanal, yaitu 1 untuk mono dan 2 untuk *stereo*.

Parameter-parameter tersebut menyatakan *setting* yang digunakan oleh ADC *(Analog-to-Digital Converter)* pada saat data *audio* direkam. Biasanya laju sampel juga dinyatakan dengan satuan Hz atau kHz. Sebagai gambaran, data *audio* digital yang tersimpan dalam CD *audio* memiliki karakteristik laju sampel 44100 Hz, 16 bit per sampel, dan 2 kanal (*stereo*), yang berarti setiap satu detik suara tersusun dari 44100 sampel, dan setiap sampel tersimpan dalam data sebesar 16-bit atau 2 *byte*. Laju sampel selalu dinyatakan untuk setiap satu kanal. Jadi misalkan suatu data *audio* digital memiliki 2 kanal dengan laju sampel 8000 sampel/detik, maka sesungguhnya di dalam setiap detiknya akan terdapat 16000 sampel.

Sebagaimana telah dijelaskan sebelumnya bahwa untuk *stream* data *audio* menggunakan *header* berupa struktur *PCMWAVEFORMAT. PCM* merupakan singkatan dari *Pulse Coded Modulation*, yaitu suatu metode yang digunakan untuk mengkonversikan sinyal *audio* dari bentuk analog ke bentuk digital [15].

Mis : Untuk sinyal analog dengan bandwidth 15 kHz → 10 kHz → sampling rate = 2 x 10 kHz = 20 kHz.

File output pada proses sampling analog ke digital ada beberapa

- WAV adalah format audio standar Microsoft dan IBM untuk PC.
- WAV biasanya menggunakan coding PCM (Pulse Code Modulation)
- WAV adalah data tidak terkompres sehingga sebuah sampel audio disimpan semuanya di harddisk.
- Software yang dapat menciptakan/WAV dari Analog Sound misalnya adalah Windows Sound Recorder
- WAV jarang sekali digunakan di internet karena ukurannya yang relative besar.
- Maksimal ukuran file WAV adalah 2GB.

Secara umum data audio digital memiliki karakteristik yang dapat dinyatakan dengan parameter-parameter berikut.

a. Laju sampel (sampling rate) dalam sampel/detik, misalnya 22050 atau 44100 sampel detik.

b. Jumlah bit tiap sampel, misalnya 8 atau 16 bit.

c. Jumlah kanal, yaitu 1 untuk mono dan 2 untuk stereo.

Parameter-parameter tersebut dipunyai oleh setiap yang digunakan oleh ADC (Analog-to-Digital Converter) pada saat data audio direkam. Biasanya laju sampel juga dinyatakan dengan satuan Hz atau kHz. Sebagai contohnya, dua audio digital yang tersimpan dalam CD audio memiliki karakteristik laju sampel 44100 Hz, 16 bit per sampel, dan 2 kanal (stereo), yang berarti setiap satu detik suara tersimpan dari 44100 sampel, dan setiap sampel tersimpan dalam data sebesar 16-bit atau 2 byte. Laju sampel sebuah ditujukan untuk setiap satu kanal. Jadi, misalkan suatu file audio digital memiliki 2 kanal dengan laju sampel 8000 sampel detik, maka sesungguhnya ia dalam setiap detiknya akan terdapat 16000 sampel.

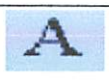
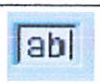
Sebagaimana telah dijelaskan sebelumnya bahwa untuk sistem data audio menggunakan beberapa standar PC [WIL 97] [MAK97] PCM merupakan singkatan dari Pulse Coded Modulation, yaitu salah metode yang digunakan untuk mengkonversikan sinyal audio dari bentuk analog ke bentuk digital [15].

2.5 Microsoft Visual Basic 6.0

Microsoft Visual Basic 6.0 adalah sebuah bahasa pemrograman komputer. Bahasa pemrograman adalah perintah-perintah atau instruksi yang berurutan, yang dimengerti oleh komputer untuk menyelesaikan masalah-masalah tertentu. *Microsoft Visual Basic* merupakan bahasa pemrograman yang menghasilkan aplikasi-aplikasi pada *windows* yang berbasis grafis (*GUI-Grafical User Interface*). *Microsoft Visual Basic* merupakan *event-driven programming* (pemrograman terkendali kejadian) yang mengandung arti bahwa program menunggu sampai adanya respon dari pemakai, yang berupa *event* atau kejadian tertentu. Ketika *event* terdeteksi, kode yang berhubungan dengan *event* akan dijalankan.

Bahasa pemrograman *Microsoft Visual Basic* merupakan pengembangan terakhir dari bahasa pemrograman *Basic* (*Beginner Of Purpose Simbolic Instruction Code*). Karena kemiripannya dengan pemrograman *basic*, bahasa pemrograman *Visual Basic* ini menjadi lebih mudah untuk dipahami dan dipelajari. *Microsoft Visual Basic 6.0* ini mempunyai kemampuan yang sangat besar dalam membuat program-program yang lebih kompleks. *Microsoft Visual Basic* terdiri dari beberapa versi, dan *Microsoft Visual Basic 6.0* merupakan penyempurnaan dari versi sebelumnya [8].

Tabel 1.1 Toolbox komponen

ICON	NAMA	KETERANGAN
	Picture Box	Kontrol untuk menampilkan file bitmaps (.bmp, :dib), words metafile (.wmf, cmf), icon (icon, .cur), GIF (.gif) dan JPG (.jpg) (awalan : pic)
	Label	Kontrol yang dapat menampilkan teks, namun tidak dapat dicdit olch uscr (awalan :lbl)
	Text Box	Unit kontrol yang dapat menampilkan teks, dmana user dapat

		berinteraksi dan mengeditnya (awalan : txt)
	Frame	Merupakan unit kontrol yang mengidentifikasi sebuah grup kontrol dan frame bertindak sebagai parent kontrol(awalan : fra)
	Command Button	Unit ini akan membuat sebuah tombol yang umumnya digunakan untuk mengeksekusi sebuah rutin (awalan : cmd)
	Timer	*Timer* digunakan untuk proses background yang diaktifkan berdasarkan interval waktu tertentu. Merupakan kontrol non-visual.
	List Box	Unit ini akan menampilkan data item dimana pemakai dapat memilih salah satu dari beberapa item yang ditampilkan (awalan : lst).
	Combo Box	Merupakan unit kombinasi antara TextBox dan ListBox. Dengan unit ini pemakai dapat mengetikkan atau memilih item lewat dropdown list (awalan : cbo).
	Toolbar	Merupakan control untuk membuat tampilan toolbar sehingga kita dapat dengan mudah memilih.
	*Common Dialog*	komponen yang berguna untuk membuka atau menyimpan file dari operating system.
	Winsock	Merupakan kontrol untuk menghubungkan antar computer untuk bertukar data.

2.6 Control *Winsock*

Control Winsock adalah control yang sering digunakan dalam pemrograman jaringan. Winsock ini bersifat dua arah. Dengan bantuan control ini, program bisa berfungsi sebagai client yang melakukan koneksi ke server di komputer tujuan, dan bisa juga menjadi server yang menerima koneksi dari banyak komputer. Properti-properti penting Winsock antara lain :

a. Protocol

Properti protocol mengatur bagaimana komunikasi yang akan dilakukan antara program dengan computer tujuan. Terdapat dua jenis protocol, yaitu protocol TCP dan UDP.

b. RemoteHost

Properti RemoteHost adalah alamat IP komputer tujuan.

c. RemotePort

Properti RemotePort adalah port di komputer tujuan. Adanya properti ini memungkinkan Winsock untuk mengakses beragam layanan TCP/IP.

d. LocalPort

Properti LocalPort berguna terutama untuk membuat program server. Sebuah program server umumnya membuka koneksi di port tertentu. Dengan mengeset property LocalPort, maka program yangdibuat akan menunggu masuknya koneksi dari luar.

Event-event yang dimiliki control Winsock adalah :

a. Connect

Event Connect terjadi saat Winsock membuka koneksi ke komputer tujuan yang ada dalam properti RemoteHost.

b. ConnectionRequest

Event ConnectionRequest terjadi saat Winsock menerima koneksi yang masuk ke computer tujuan, khususnya pada port yang diset pada property LocalPort.

c. DataArrival

Event DataArrival terjadi saat Winsock menerima paket data dari computer lain. Event ini adalah inti dari proses komunikasi jaringan.

d. Close

Event Close terjadi saat Winsock menutup koneksi. Event ini penting supaya pada saat penutupan koneksi, program bisa melepas resource-resource yang sudah tidak diperlukan.

Transmission control protocol (TCP) adalah protocol pada layer transport dari stack jaringan TCP/IP. TCP memecah stream byte yang kontinu menjadi segmen-segmen dan mengirimkannya sebagai frame IP.

Komponen *Winsock* mempunyai 2 jenis protokol:

- TCP – *Transmission Control Protocol.* Dengan TCP mengharuskan 2 atau lebih komputer yang terhubungan untuk mengirim/menerima data harus dalam kondisi terkoneksi. Jadi dengan menggunakan TCP sangat aman karena data akan selalu dicek sampai atau tidak.
- UDP – *User Datagram Protocol.* Dengan UDP tidak mengharuskan adanya koneksi yang aktif, hanya saja tidak ada jaminan data yang dikirim sampai atau tidak [7].

Dalam skripsi ini saya menggunakan protokol TCP.

TCP/IP (Transmission Control Protocol/Internet Protocol) adalah sekelompok protokol yang mengatur komunikasi data computer di internet. Computer-komputer yang terhubung ke internet berkomunikasi dengan protokol ini. Karena menggunakan bahasa yang sama, yaitu TCP/IP, perbedaan jenis computer dan system operasi menjadi tidak masalah [11].

Bab III
ANALISA DAN PERANCANGAN SISTEM

3.1 Analisa Sistem

Aplikasi yang akan dibuat pada tugas akhir ini adalah sebuah aplikasi enkripsi dan dekripsi pengiriman pesan suara dengan algoritma *Serpent*, dimana fungsi utama dari aplikasi pengiriman pesan suara dengan algoritma *Serpent* adalah agar pesan suara yang di enkripsi bisa memiliki tingkat keamanan dalam pengiriman.

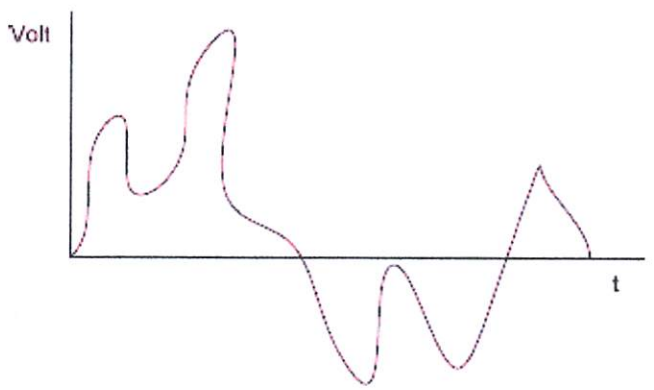
3.2 Analisa pada pemrosesan suara

Suara yang masih berupa sinyal analog harus diproses terlebih dahulu sebelum dapat diolah oleh program komputer. Berikut akan dibahas mengenai proses digitalisasi dan kompresi suara.

3.2.1 Digitalisasi Suara

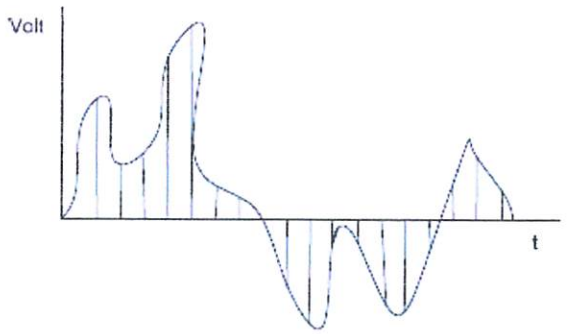
Gelombang suara analog sebelum dapat diproses oleh aplikasi dan dienkripsi harus diubah terlebih dahulu kedalam format digital dengan menggunakan sirkuit ADC (*Analog Digital Converter*). Karena pada saat ini semua pemrosesan suara melalui computer menggunakan suara digital. Pada ADC, sinyal analog akan diubah menjadi digital dengan cara *sampling*.

Misalnya sinyal analog diwakilkan seperti pada Gambar 3.1. Sumbu y melambangkan Volt, yang berarti tegangan volt. Sumbu x melambangkan t, yang berarti waktu.



Gambar 3.1 Sinyal Analog

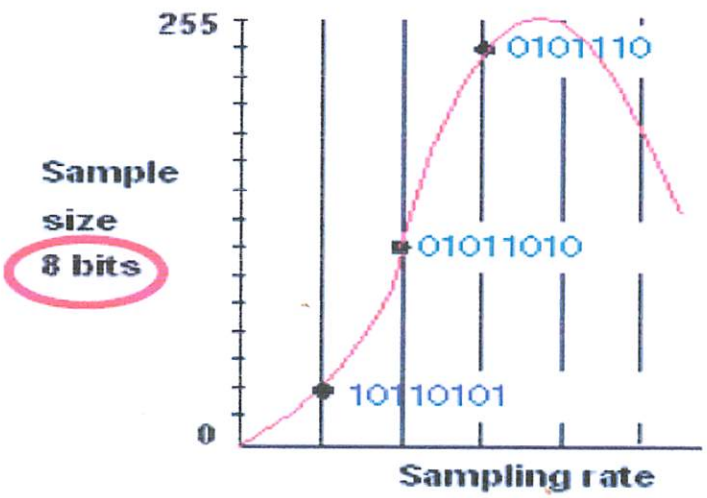
Sirkuit ADC akan mengambil *sample* atau bagian dari sinyal analog berdasarkan waktu tertentu. Bagian-bagian tersebut kemudian akan direpresentasikan dalam angka berdasarkan tegangan volt yang dimiliki masing-masing bagian. Gambar 3.2 menunjukkan contoh beberapa titik *sampling* yang diambil.



Gambar 3.2 Contoh *Sampling*

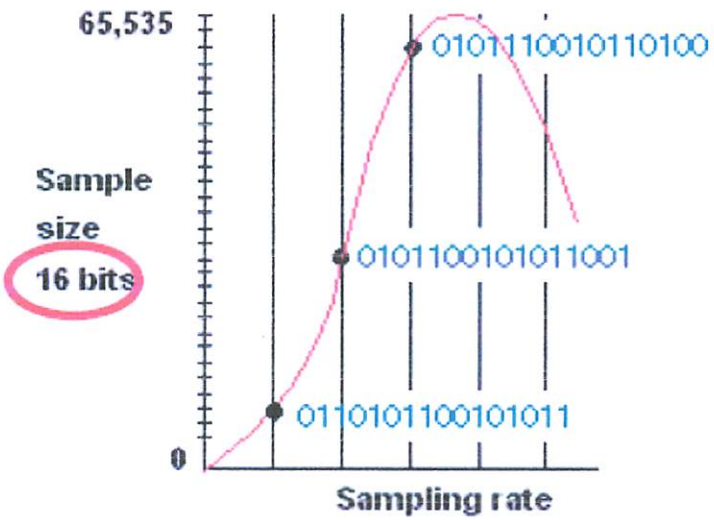
Data *sampling* yang diambil dapat bervariasi berdasarkan Dt yang digunakan. Jumlah *sample* yang diambil tiap detiknya disebut *sampling rate*. Semakin besar *sampling rate* maka data yang diperoleh akan semakin akurat namun *storage* yang dibutuhkan juga semakin besar. Dibutuhkan perhitungan yang tepat agar kualitas suara yang didapat baik dan *storage* yang dibutuhkan tidak terlalu besar.

Bentuk pada suara yang terdigitalisasi berbentuk bilangan biner 01, dimana pada keluaran suara terdapat 8bit dan 16Bit. Perbedaan tersebut mempengaruhi kualitas suara itu sendiri. Contoh biner pada keluaran suara 8 bit :



Gambar 3.3 Contoh Keluaran suara 8 bit

Contoh keluaran suara pada 16 bit :



Gambar 3.4 Contoh Keluaran suara 16 bit

Hasil dari keluaran suara yang berbentuk biner tersebut kemudian di jadikan bilangan hexa pada proses pengenkripsian pada langkah selanjutnya. Misalnya contoh keluaran sinyal hasil analog ke digital adalah 0110101100101011. Jadi Pesan (dalam bentuk rangkaian bit) dipecah menajdi beberapa blok.

Contoh: Plainteks 0110101100101011

Bila dibagi menjadi blok 4-bit

0110 1011 0010 1011

maka setiap blok menyatakan 0 sampai 15:

0110 = 6 ; 1011 = 11 ; 0010 = 2 ; 1011 = 11.

3.3 Analisa Algoritma Serpent

Dari hasil contoh perekaman pada 3.2.1 telah di dapat contoh suara atau plainteks yang akan di enkripsi yaitu : 0110|1011|0010|1011 = 6|11|2|11 =6B|2B

Misal pemasukan kunci "ITNMALANG" = "49544E4D414C414E47" (ASCII CODE dapat dilihat pada lampiran).

Dan hasil dari Tabel 3.1 Pemasukan kunci pada plainteks

Tipe	**Kunci Enkripsi**															
Ki	**I**	**T**	**N**	**M**	**A**	**L**	**A**	**N**	**G**	**I**	**T**	**N**	**M**	**A**	**L**	**A**
Hexa	49	54	4E	4D	41	4C	41	4E	47	49	54	4E	4D	41	4C	41
plainteks	0	1	1	0	1	0	1	1	0	0	1	0	1	0	1	1
Pi	6B								2B							

Jika sudah di dapat plainteks dan kunci enkripsi dapat kita lakukan proses inisial permutasi dimana penggabungan kunci dan plainteks tersebut.

S0...S32= (**Ki ⊕ Pi =IP**)

Kunci : I ⊕ 6B = IP

{01001001} ⊕ {01101011} = {00100010} (notasi biner)

{49} ⊕ {6B} = {22} (notasi hexadesimal)

Kunci : T ⊕ 6B = IP

{01010100} ⊕ {01101011} = {00111111} (notasi biner)

{54} ⊕ {6B} = {3F} (notasi hexadesimal)

Kunci : N ⊕ 6B = IP

{01001110} ⊕ {01101011} = {00100101} (notasi biner)

{4E} ⊕ {6B} = {45} (notasi hexadesimal)

Kunci : M ⊕ 6B = IP

{01001101} ⊕ {01101011} = {00100110} (notasi biner)

{4D} ⊕ {6B} = {26} (notasi hexadesimal)

Kunci : A ⊕ 6B = IP

{01000001} ⊕ {01101011} = {00101010} (notasi biner)

{41} ⊕ {6B} = {2A} (notasi hexadesimal)

Kunci : L ⊕ 6B = IP

{01001100} ⊕ {01101011} = {00100111} (notasi biner)

{4C} ⊕ {6B} = {27} (notasi hexadesimal)

Kunci : A ⊕ 6B = IP

{01000001} ⊕ {01101011} = {00101010} (notasi biner)

{41} ⊕ {6B} = {2A} (notasi hexadesimal)

Kunci : N ⊕ 6B = IP

{01001110} ⊕ {01101011} = {00100101} (notasi biner)

{4E} ⊕ {6B} = {45} (notasi hexadesimal)

Kunci : G ⊕ 2B = IP

{01000111} ⊕ {00101011} = {01101100} (notasi biner)

{47} ⊕ {2B} = {6C} (notasi hexadesimal)

Kunci : I ⊕ 2B = IP

{01001001} ⊕ {00101011} = {01100010} (notasi biner)

{49} ⊕ {2B} = {62} (notasi hexadesimal)

Kunci : T ⊕ 2B = IP

{01010100} ⊕ {00101011} = {01111111} (notasi biner)

{54} ⊕ {2B} = {7F} (notasi hexadesimal)

Kunci : N ⊕ 2B = IP

{01001110} ⊕ {00101011} = {01100101} (notasi biner)

{4E} ⊕ {2B} = {65} (notasi hexadesimal)

Kunci : M ⊕ 2B = IP

{01001101} ⊕ {00101011} = {01100110} (notasi biner)

{4D} ⊕ {2B} = {66} (notasi hexadesimal)

Kunci : A ⊕ 2B = IP

{01000001} ⊕ {00101011} = {01101010} (notasi biner)

{41} ⊕ {2B} = {6A} (notasi hexadesimal)

Kunci : L ⊕ 2B = IP

{01001100} ⊕ {00101011} = {01100111} (notasi biner)

{4C} ⊕ {2B} = {67} (notasi hexadesimal)

Kunci : A ⊕ 2B = IP

{01000001} ⊕ {00101011} = {01101010} (notasi biner)

{41} ⊕ {2B} = {6A} (notasi hexadesimal)

Jadi hasil dari penggabungan kunci 128 bit enkripsi dan plainteks adalah :

"22|3F|45|26|2A|27|2A|45|6C|62|7F|65|66|6A|67|6A"

Hasil berikut kemudian di masukan pada kotak subtitusi (S-BOX) dimana akan di dapat hasil dari S0 adalah = " 93|75|1b|3f|e5|f7|e5|1b|50|ef|d2|43|4d|02|a8|02"

Berikutnya proses dari S1 sampai dengan S32 sama seperti proses S0.

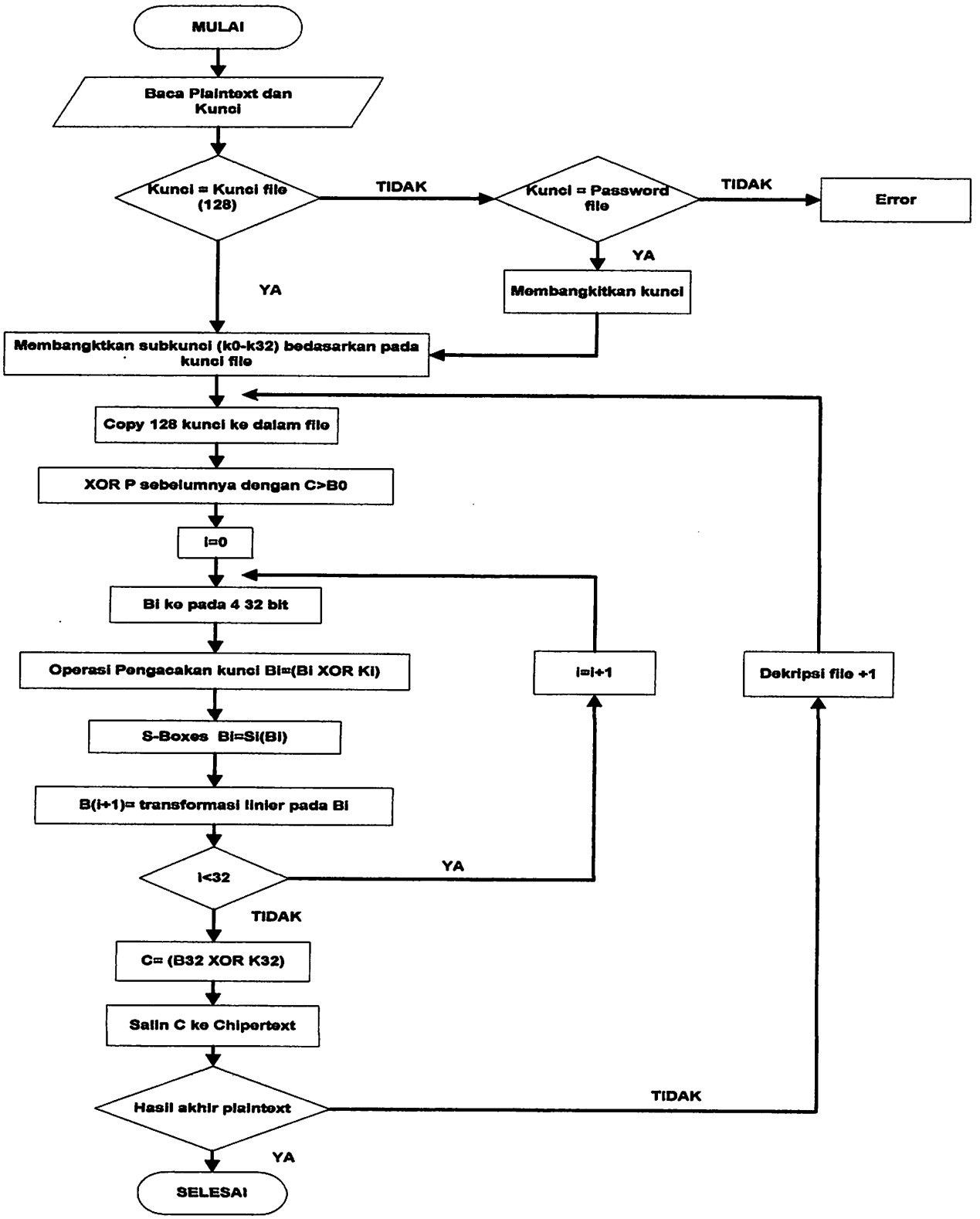
	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6B	6f	c5	30	01	67	2B	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6c	5a	a0	52	3b	d6	b3	29	c3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Tabel 3.2 LookUp *Subtitusi Bytes algoritma serpent*

Pada proses dekripsi algoritma serpent adalah pembalikan atau invers dari subtitusi hasil dari penggabungan kunci enkripsi dan plainteks sehingga di dapatkan chiperteks. Misal chiperteks S0 adalah "93|75|1b|3f|e5|f7|e5|1b|50|ef|d2|43|4d|02|a8|02" maka sub kunci adalah "22|3F|45|26|2A|27|2A|45|6C|62|7F|65|66|6A|67|6A". sehingga plainteks dapat di dekripsi dan dapat di baca.

		Y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
X	0	52	9	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
	1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
	2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
	3	8	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
	4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
	5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
	6	90	d8	ab	0	8c	bc	d3	0a	f7	e4	58	5	b8	b3	45	6
	7	d0	2c	1e	8f	ca	3f	0f	2	c1	af	bd	3	1	13	8a	6b
	8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
	9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
	a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
	b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
	c	1f	dd	a8	33	88	7	c7	31	b1	12	10	59	27	80	ec	5f
	d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
	e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
	f	17	2b	4	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Tabel 3.3 Invers atau dekripsi dari algoritma serpent

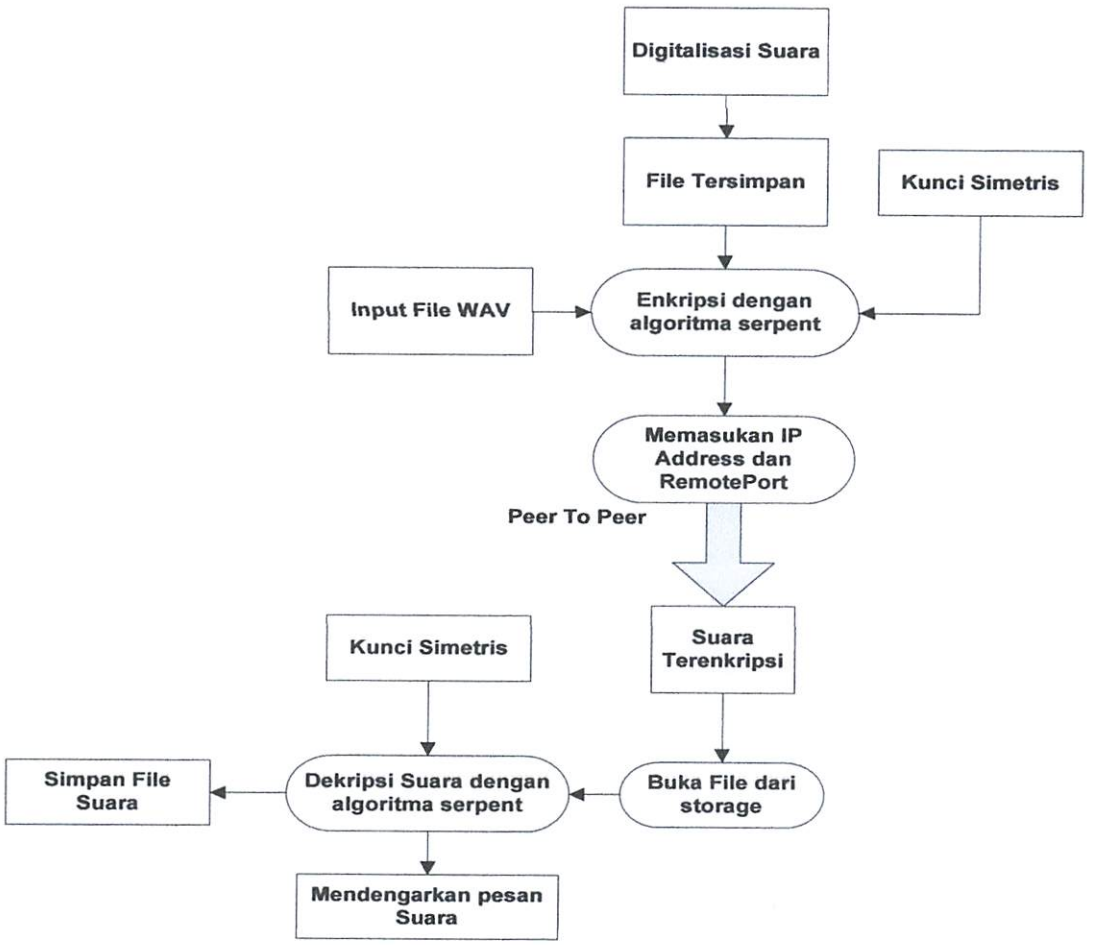


Gambar 3.5 Flowchart Algoritma *Serpent*

3.4 Alur Sistem

Aplikasi yang akan dibuat pada skripsi ini adalah Enkripsi dan Dekripsi Pengiriman Pesan Suara Dengan Menggunakan Algoritma *Serpent*, objek masukan untuk modul proses penyandian atau pengenkripsian yaitu File yang berformat WAV atau melakukan digitalisasi suara terlebih dahulu yang berupa *Audio Recording* yang yang berformat sama dan penginputan kunci simetris untuk menjaga keamanan suara. Kemudian dikirimkan ke user 2 melalui jaringan LAN yang menggunakan topologi *peer to peer* dan menggunakan *RemotePort*, sehingga dapat menikmati layanan *TCP/IP*.

Sedangkan pada modul penerima objek pesan berupa pesan suara yang terenkripsi dimana proses dekripsi di haruskan menginputkan kunci yang simetris atau sama pada waktu enkripsi. Berikut ini adalah arsitektur system enkripsi pengiriman suara dengan algoritma *Serpent*.



Gambar 3.6 Arsitektur sistem enkripsi pengiriman pesan suara

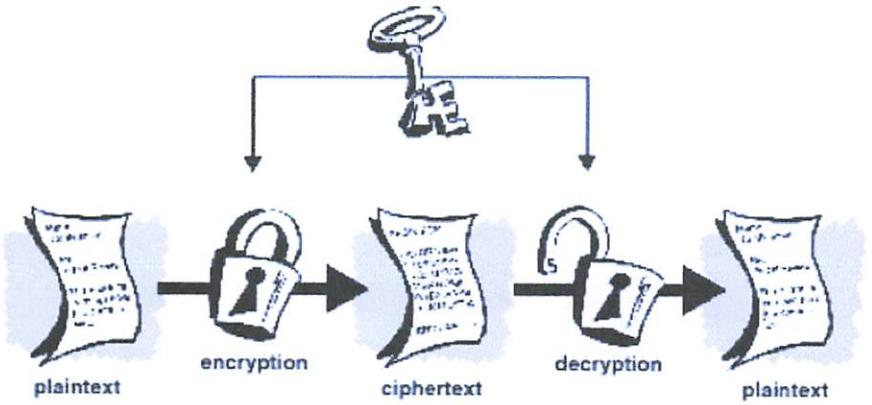
Dari gambar 3.4 dijelaskan bahwa pembangunan aplikasi dimulai dengan dalam garis besar dengan langkah pertama-tama kita melakukan digitalisasi suara, dimana kita merekam suara analog ke dalam bentuk digital. Kemudian inputkan file yang akan di enkripsi dan pastikan suara dalam kualitas yg bagus. Jika file suara sudah di enkripsi kemudian siapkan form dialog transfer untuk mengirim file tersebut. Pastikan alamat IP dan port tidak salah sehingga pengiriman bisa berjalan dengan baik.

3.5 Proses-Proses di dalam Perangkat Lunak

Pengirim akan menerima masukan data suara melalui microphone kemudian diubah menjadi bit digital. Untuk memperkecil ukuran data suara yang akan diproses, dilakukan proses kompresi. Hasil dari kompresi ini nantinya akan dienkripsi dengan algortima *Serpent* dan dikirimkan melalui kabel jaringan. Penerima akan melakukan dekripsi terhadap data suara yang dikirimkan. Hasil dekripsi kemudian di didekompresi dan dikeluarkan melalui speaker agar dapat didengarkan kembali. Masukan suara selain berasal dari microphone juga bisa berasal dari suatu file audio dengan format wav.

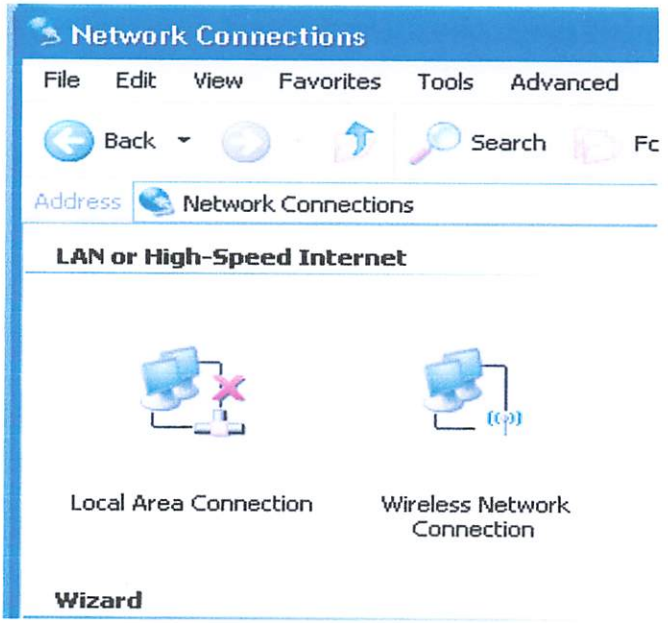
1. Pertama proses penginputan file, *user* membutuhkan masukan suara dengan cara merekam suara atau bisa juga menginputkan suatu file rekaman yang telah ada.
   a. Input file, *User* dapat menginputkan file rekaman yang sudah ada yang kemudian akan di enkripsi.
   b. Input suara, dimana *User* harus memilih device atau media yang akan di gunakan untuk merekam suara.
   c. Output suara, *User* dapat mengatur dimana file rekaman tersebut di simpan dan mensetting profil format file rekam (WAV).
2. Pemberian kunci pada file yang akan di enkripsi.

Key enkripsi, pemberian kunci simetris pada file WAV.



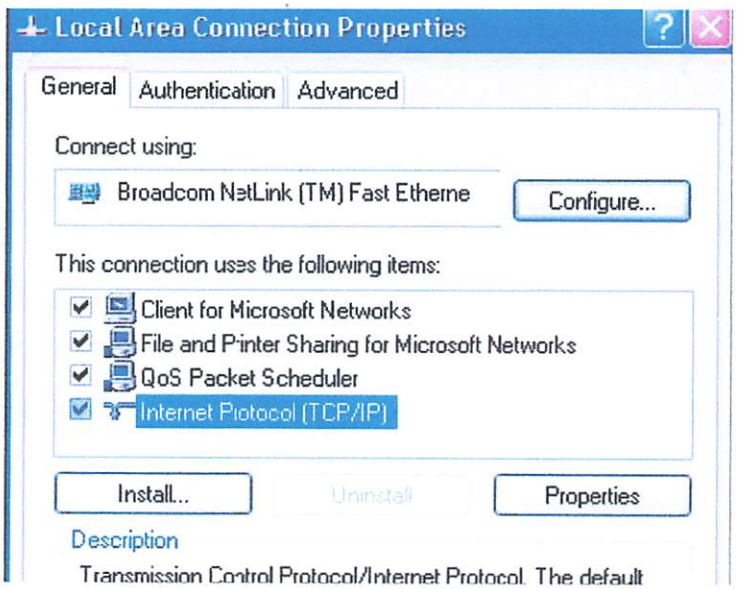
Gambar 3.7 Kunci Simetris

3. Pengiriman file yang telah terenkripsi dengan menggunakan *winsock* komponen yang terdapat pada visual basic 6.0. Berikut ini alur pengiriman file pada komputer lain :
   a. Pertama *User* harus mensetting IP address untuk identitas atau alamat user, dengan cara :
      - Click Start -Setting - Control Panel - Network Connection.



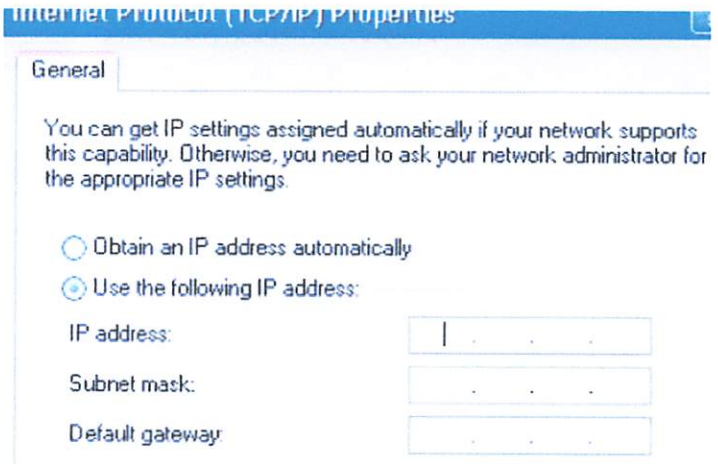
Gambar 3.8 Dialog Network Connection

- Double-click Icon Local Area Connection sampai keluar kotak dialog Local Connection Area Status.



Gambar 3.9 Dialog LAN area connection

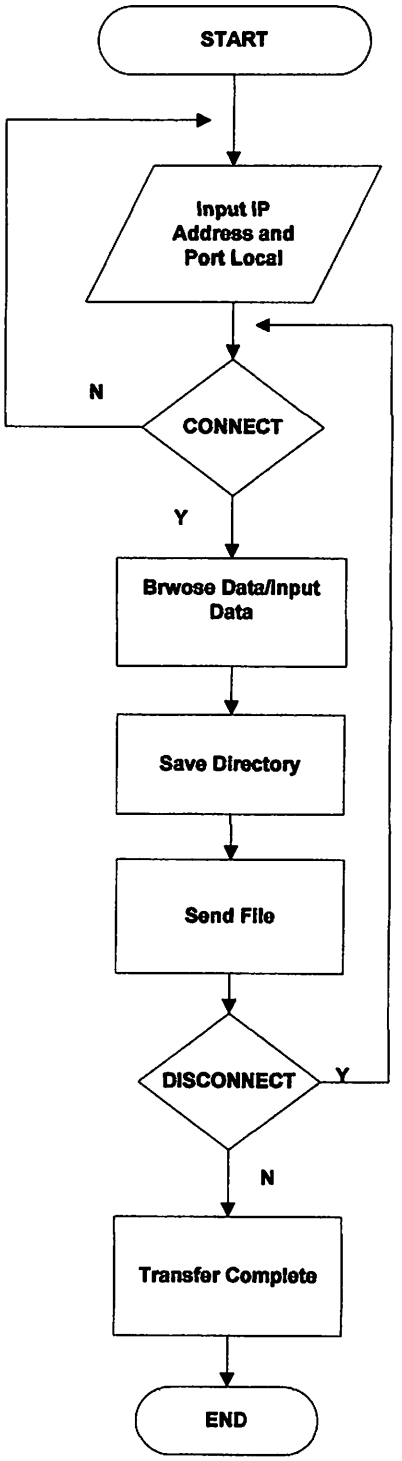
- Double click Internet Protocol TCP/IP. Sampai muncul kotak dialog baru.



Gambar 3.10 Dialog TCP/IP

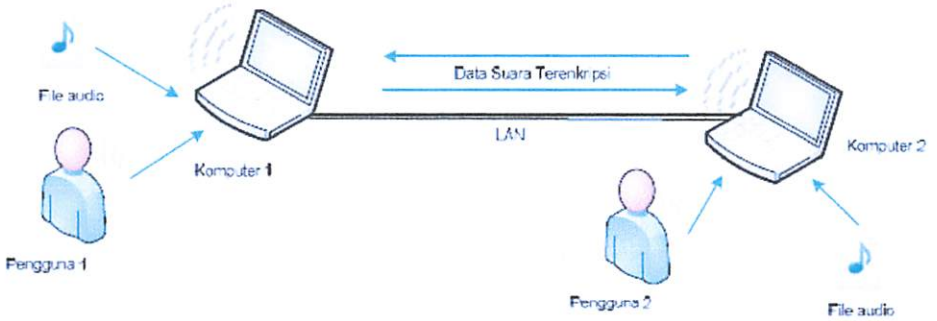
- Kita tinggal mengisikan IP Address.

b. Input file transfer, *User* menginputkan file yang telah di enkripsi.



Gambar 3.11 Flowchart Alur File Transfer

c. *remoteport* untuk koneksi dengan komputer lain *user* diharuskan mengisi port local . Disini port harus simetris atau sama dengan yang lain.
d. Save directory, Menentukan letak file yang akan dikirim ke komputer remote atau komputer lain.



Gambar 3.12 Arsitektur Sistem umum

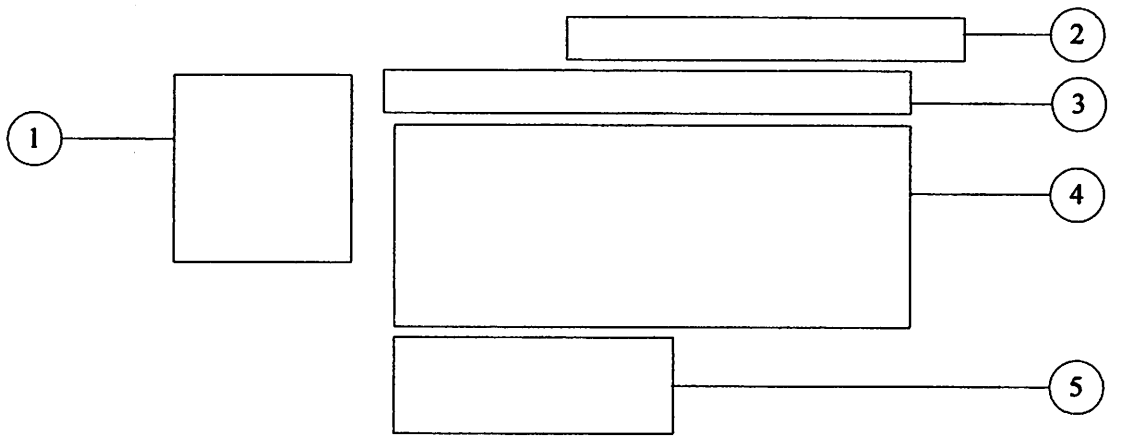
3.6 Perancangan Sistem

Dalam perancangan perangkat lunak enkripsi dan dekripsi pengiriman pesan suara dengan algoritma *Serpent* ini dirancang dengan menggunakan bahasa pemrograman *Microsoft Visual Basic 6.0* dengan beberapa komponen *standard.*
Perangkat lunak ini memiliki beberapa buah *form*, yaitu:

1. *Form* splash screen.
2. *Form* Utama.
3. *Form* Set Password.
4. *Form* Input Kunci.
5. *Form* Rekam Suara.
6. *Form* Output format WAV.
7. *Form* File Transfer.

1. *Form* splash screen

Form Splash Screen, berisi tentang pengenalan diri dari nama perangkat lunak dan nama serta NIM mahasiswa yang membuat perangkat lunak. Rancangan *form* Splash Screen dapat dilihat pada gambar 3.13 :



Gambar 3.13 Rancangan *Form* Splash Screen

Keterangan :

1 : *icon* atau logo perangkat lunak.

2 : nama jurusan dan tahun pembuatan tugas akhir.

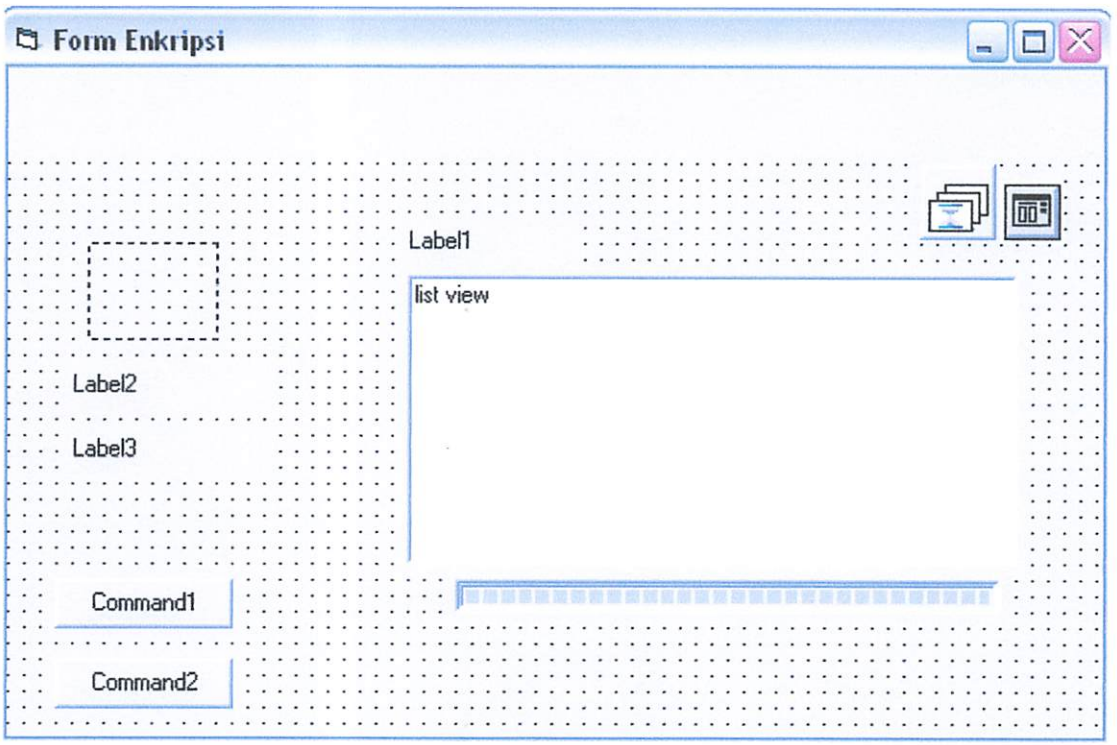
3 : nama kampus.

4 : nama perangkat lunak.

5 : nama pembuat program dan penyusun tugas akhir.

2. Form Utama

Form Utama menampilkan bagan dan *link* untuk membuka *form input* kunci, *form* rekam suara dan *form* file transfer. Rancangan *form* Utama dapat dilihat pada gambar 3.14 :



Gambar 3.14 Rancangan *form* utama

Keterangan :

1 : Form Enkripsi.

2 : *Icon toolbar list,* untuk menampilkan pilihan menu.

3 : komponen *Command* dialog untuk membuka file pada storage.

8 : *List View*, untuk menampilkan hasil urutan file yang telah terenkripsi.

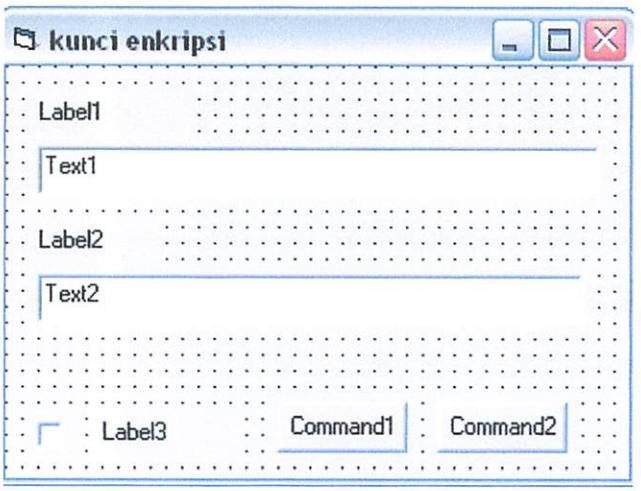
9 : *progressbar,* untuk indikator proses.

10 : *Command* 1, untuk membuka *form* Rekam Suara.

11 : *Command* 2, untuk membuka *form* Input File Transfer.

3. *Form* Kunci Enkripsi

Form Kunci Enkripsi ini menampilkan dialog yang berguna sebagai penginputan kunci sebelum di enkripsi. Rancangan *form* Kunci Enkripsi dapat di lihat pada gambar 3.15 :



Gambar 3.15 Rancangan Form Kunci Enkripsi

Keterangan :

1 : title bar form kunci enkripsi.

2 : *Text Box 1,* untuk memasukan text kunci.

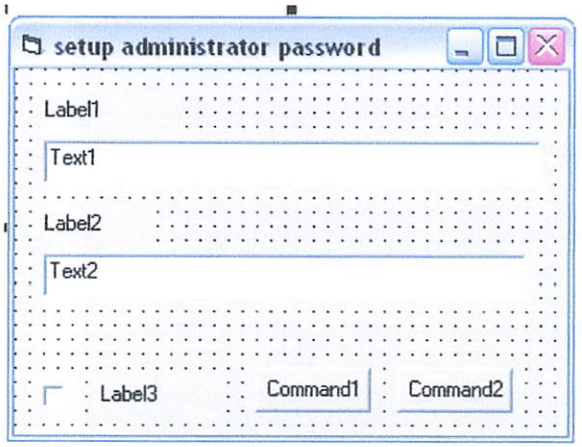
3 : *Text Box 2,* untuk perulangan kunci.

4 : *Check Box*, untuk melihat *password.*

5 : *Command* 1 untuk melanjutkan.

4. *Form* Set Password

Form Set Password ini digunakan untuk memberi atau merubah kata sandi sebelum masuk ke dalam perangkat lunak. Rancangan *Form* Set Password dapat di lihat pada gambar 3.16:



Gambar 3.16 *Form* Set Password

Keterangan :

1 : *Text Box 1,* memasukan password baru.

2 : *Label 1,* untuk nama *teks.*

3 : *Text Box 2,* pengulangan masukan password baru.

4 : *Label 2,* untuk nama *teks.*

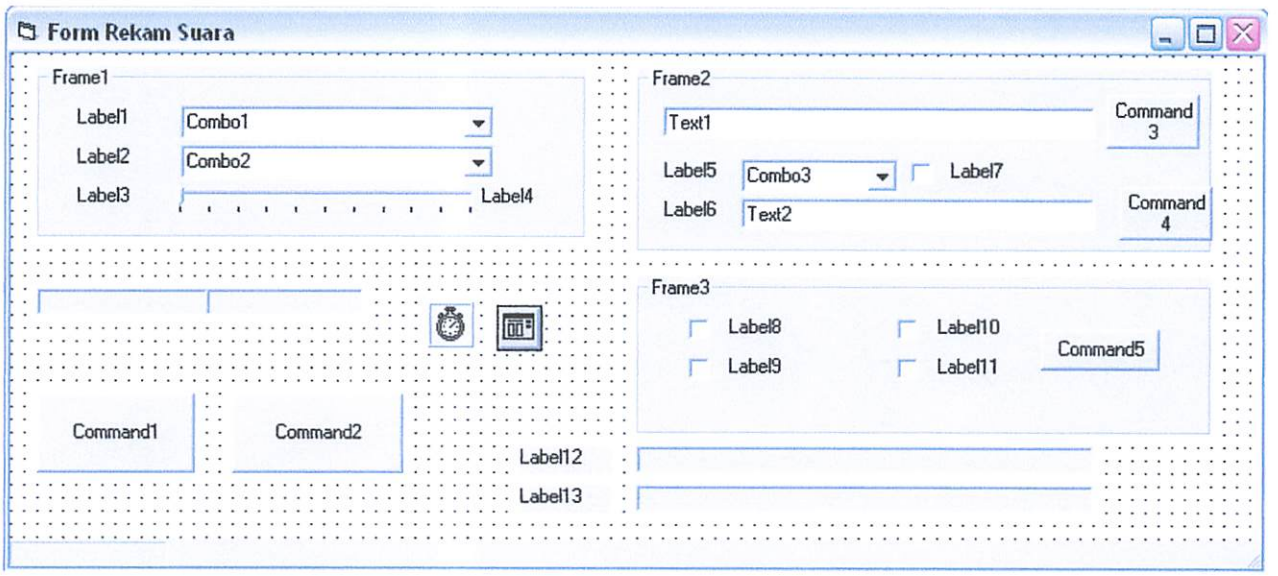
5 : *Check Box,* menampilkan hasil.

6 : *Command* 1 untuk merubah.

7 : *Command* 2 untuk membatalkan.

5. Form Rekam Suara

Form Rekam suara ini menampilkan bagan mengatur kualitas suara yang di dalamnya terdapat *form* Output format WAV dan *form* Efect settings. Rancangan *form* Rekam Suara dapat dilihat pada gambar 3.17 :



Gambar 3.17 Rancangan *form* Rekam Suara

Keterangan:

1 : *Form* Rekam Suara.

2 : *Frame 1* dari input suara

3 : *ComboBox 1,* untuk memilih *device.*

4 : *ComboBox 2,* untuk memilih line input suara.

5 : *Slider 1,* untuk mengatur volume suara.

6 : *Picture Box,* untuk indikator *equalizer.*

7 : *Command* 1 untuk merekam suara.

8 : *Command* 2 untuk kembali ke *Form* utama.

9 : *Frame 2* dari output

10 : *Command* 3, untuk menentukan output file

11 : *CheckBox Label* 7 untuk mengatur stereo atau mono.

12 : *Command 4* untuk memilih format WAV.

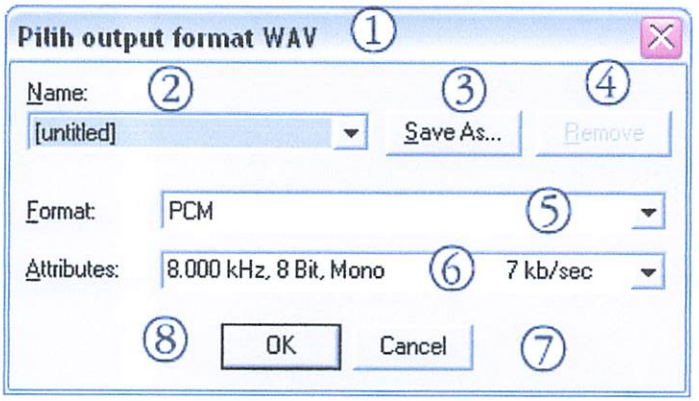
13 : *Frame* setting effect.

14 : *Picture Box,* sebagai indikator.

15 : komponen common dialog untuk membuka strorage.

6. *Form* Output format WAV

Form Output format WAV ini menampilkan dialog editor yang berguna untuk mengatur format dan attribute hasil rekaman atau output file. Rancangan *Form* Output format WAV dapat dilihat pada gambar 3.18:



Gambar 3.18 Rancangan *Form* Output format WAV

Keterangan:

1 : *Form* format keluaran file rekam.

2 : *ComboBox* untuk mimilih profil output.

3 : Tombol untuk menyimpan profil.

4 : Tombol untuk menghapus profil.

5 : *ComboBox* untuk format.

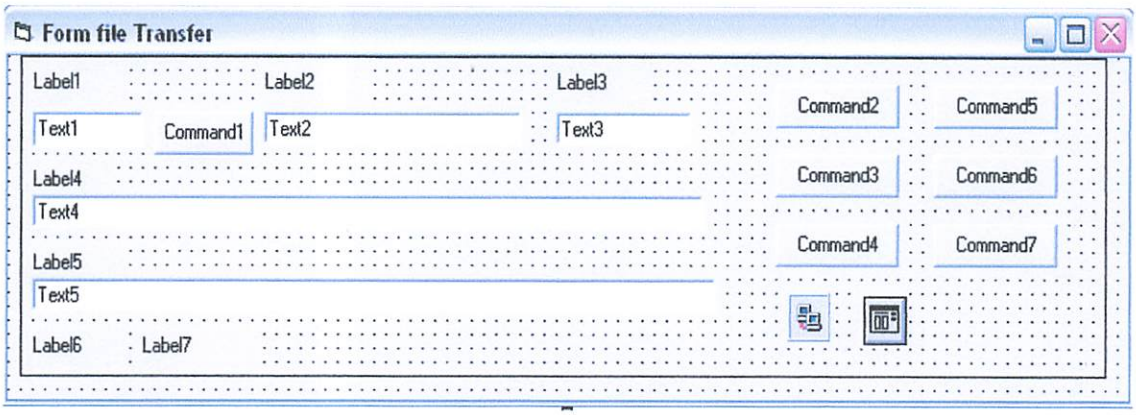
6 : *ComboBox* untuk memilih attribut.

7 : tombol, untuk menutup form.

8 : tombol, untuk menyelesaikan setting format.

7. Form File Transfer

Form File Transfer ini menampilkan bagan sebagai pengiriman file ke computer lain. Rancangan *form* File Transfer dapat di lihat pada gambar 3.19 :



Gambar 3.19 Rancangan *form* File Transfer

Keterangan:

1 : *Textbox 1*, untuk port yang terkoneksi.

2 : *Command* 1 'Listen', menerima koneksi dari port local.

3 : *Textbox 2*, untuk menentukan ip address tujuan.

4 : *Textbox 3*, untuk menentukan port local.

5 : *Command* 2, untuk membuka koneksi.

6 : *Command* 5, untuk menutup koneksi.

7 : *Command* 3, untuk memasukkan file yang akan dikirim.

8 : *Command* 6, untuk pengiriman file.

9 : *Textbox 4,* alamat file yang dikirim

10 : *Textbox 5,* directory penentuan file kiriman.

11 : *Label 6 dan Label 7,* menampilkan status.

BAB IV
IMPLEMENTASI DAN PENGUJIAN

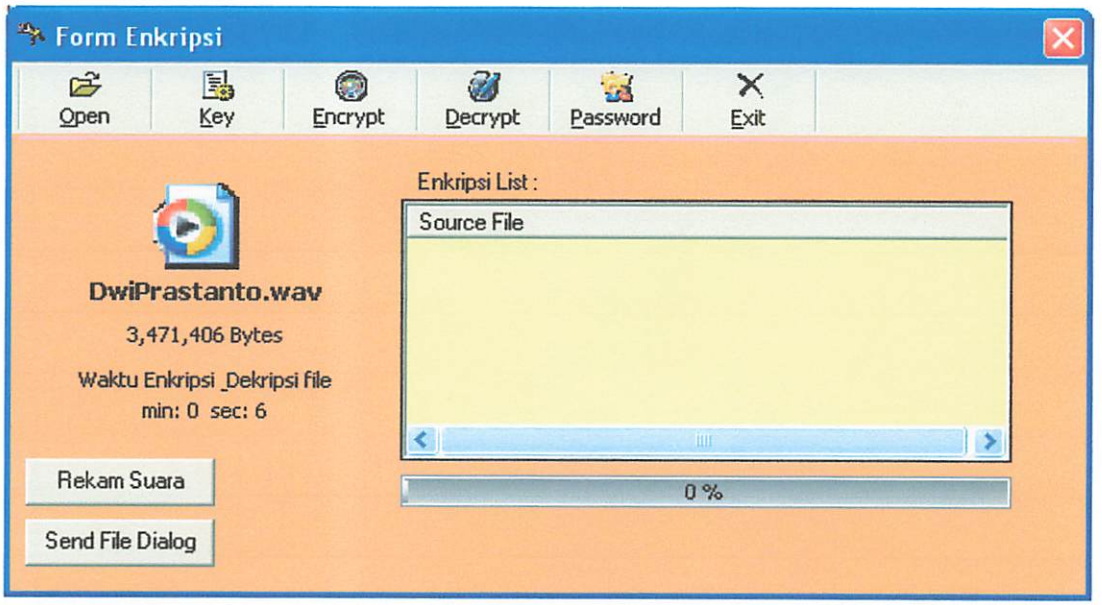
Tahap implementasi pengembangan perangkat lunak merupakan proses pengubahan spesifikasi sistem menjadi sistem yang yang dapat dijalankan. Tahap ini merupakan lanjutan dari proses perancangan sesuai dengan spesifikasi dan desain system Aplikasi Enkripsi dan Dekripsi pesan suara dengan menggunakan algoritma *Serpent* ini menggunakan Visual Basic 6.0.

4.1 Lingkungan Implementasi

Lingkungan implementasi meliputi proses-proses yang tedapat dalam aplikasi, bagan Input desain, proses desain, file transfer dan output yang sesuai dengan target yang diharapkan.

4.1.1 Form Utama

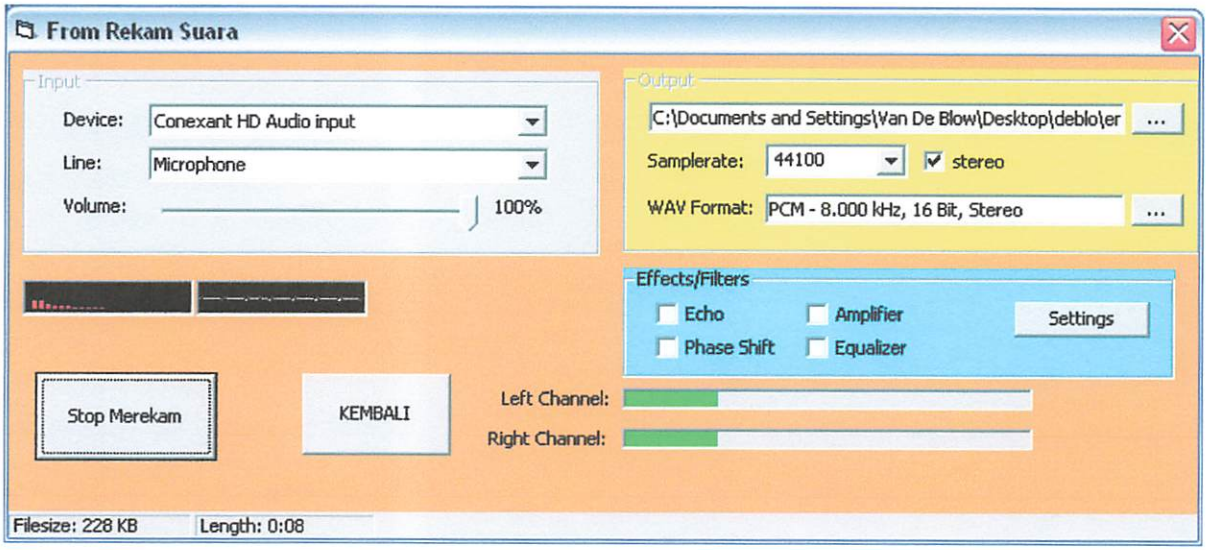
Form utama ini digunakan untuk enkripsi dan dekripsi file suara. Selain itu form tersebut juga berisi set password untuk merubah password admin. Dari form utama ini kita dapat mengacu ke form lainnya untuk menjalankan aplikasi ini sesuai harapan.



Gambar 4.1 Form utama

4.1.2 Form Rekam Suara

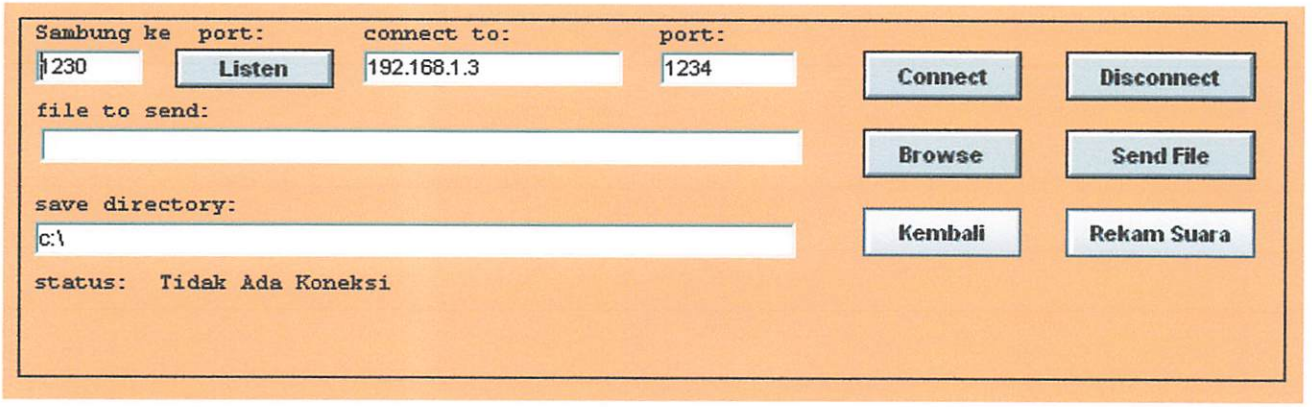
Form rekam suara ini digunakan untuk merekam mengatur kualitas suara yang akan di enkripsi.



Gambar 4.2 Form Rekam Suara

4.1.3 Form File Transfer (kirim data)

Form File Transfer (kirim data) ini digunakan untuk mengirim hasil enkripsi file suara rekaman kepada user lain.



Gambar 4.3 Form File Transfer (kirim data)

4.2 Pengujian Sistem

Untuk melakukan pengujian aplikasi ini menggunakan PC/Laptop dengan spesifikasi :

Untuk user 1 :

1. Prosesor : intel 2.2 Ghz, 800 Mhz FSB, Dual core
2. Memori :1 GB DDR2
3. Hardisk : 250 GB
4. Microphone
5. Kabel UTP crossover
6. Sistem Operasi : Windows
7. Microsoft Visual Basic 6.0

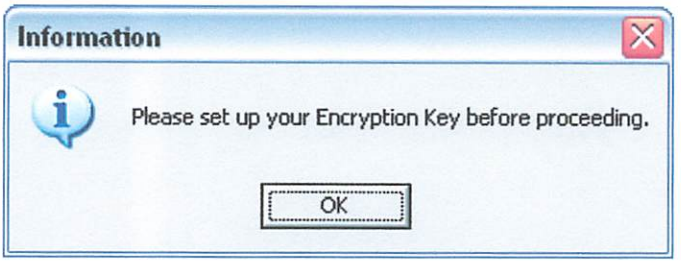
Untuk User 2 :

1. Prosesor
2. Memori RAM 2 GB DDR3
3. Microphone
4. Kabel UTP crossover
5. Sistem Operasi Windows
6. Microsoft Visual Basic 6.0

4.2.1 Alur Aplikasi Dari Proses Enkripsi, Pengiriman dan Dekripsi

Dalam menjalankan aplikasi ini terdapat beberapa langkah yang bisa dilakukan sehingga dapat di operasikan dengan baik. Adapun langkah-langkah dari awal sampai akhir aplikasi ini bekerja yaitu :

1. Pada saat awal pembukaan dari aplikasi ini kita akan dihadapkan pada pengaturan kunci enkripsi dan admin password. Admin password ini hanya berguna untuk menjaga aplikasi tersebut. Pembukaan berikutnya kita langsung dihadapkan pada form admin password.



Gambar 4.4 Informasi enkripsi kunci

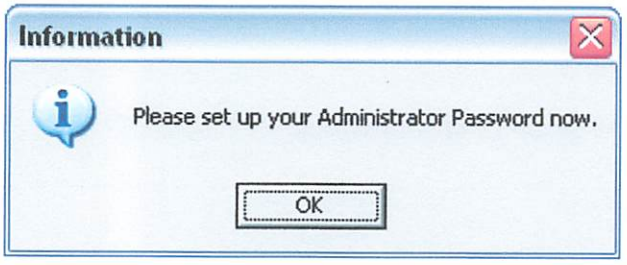
2. Pada tampilan form kunci enkripsi kita dapat melihat password yang kita masukan cocok apa tidak pada *check box Show Encryption Key.*

Gambar 4.5 input enkripsi kunci

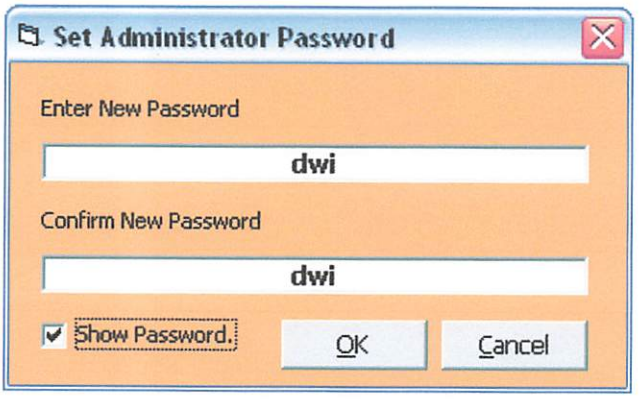
3. Jika kunci cocok kita telah berhasil menginputkan kunci dengan di tandai informasi pada tampilan gambar 4.6.

Gambar 4.6 Kunci Enkripsi Berhasil

4. Berikutnya pengaturan pada admin password dapat kita tentukan berikutnya.

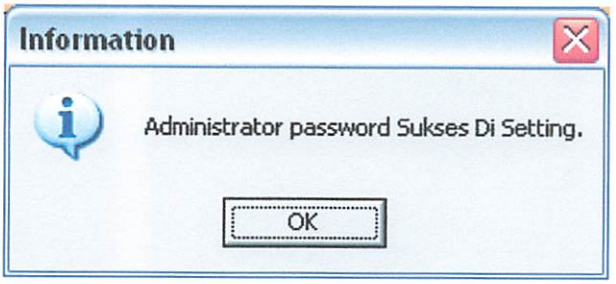


Gambar 4.7 Informasi pengaturan Admin Password



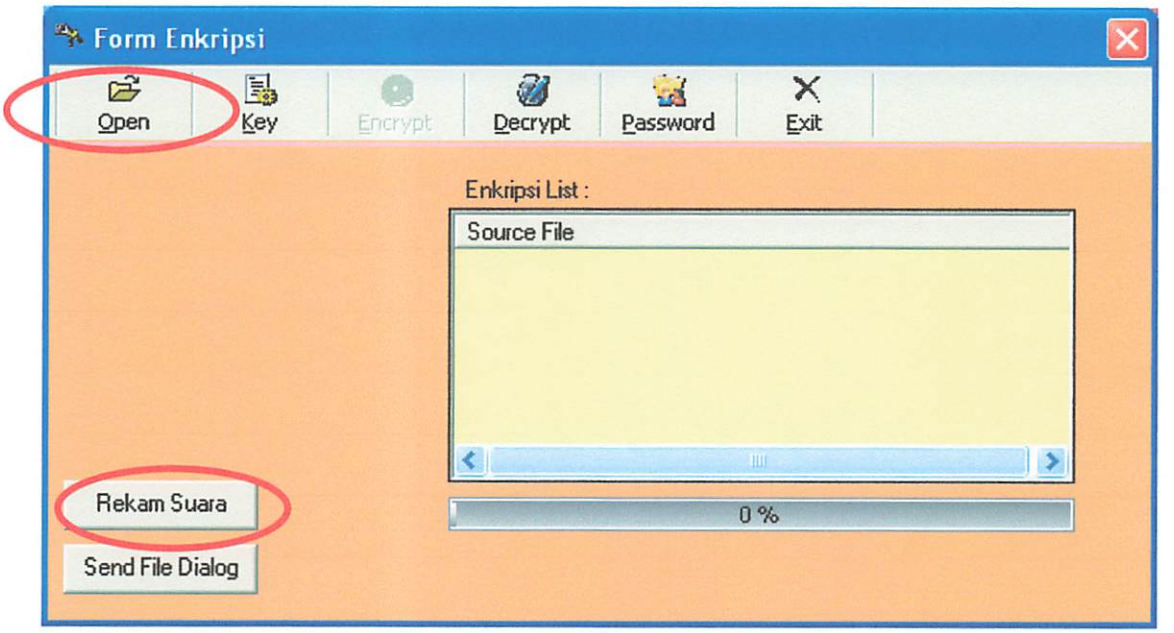
Gambar 4.8 Pengaturan password

5. Jika muncul informasi pada tampilan gambar 4.9 muncul, kita telah berhasil mengatur password.



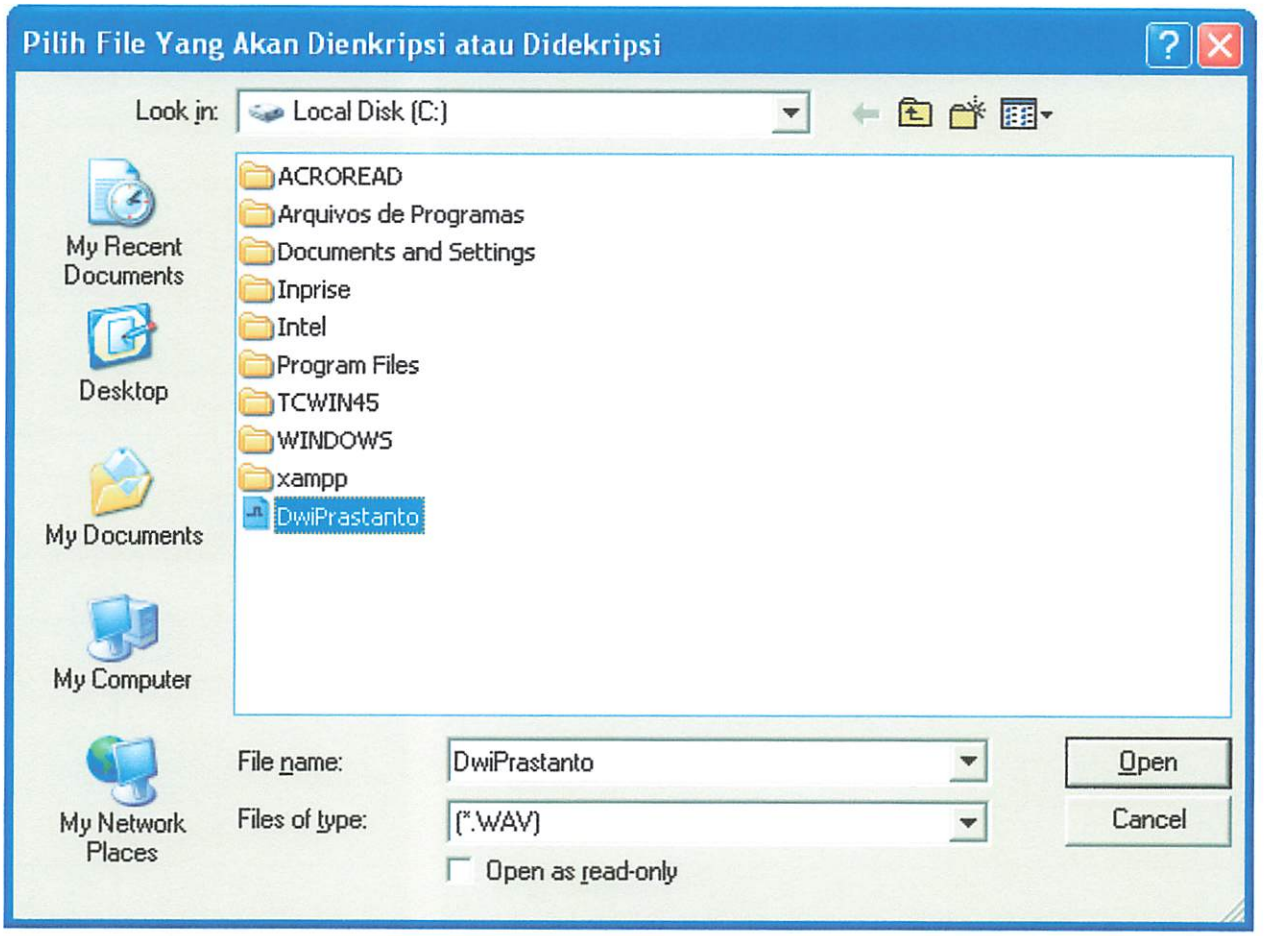
Gambar 4.9 Informasi Admin Password Berhasil

6. Tampilan Utama aplikasi di dalam tampilan ini kita dapat memasukan file rekaman yang sudah ada atau kita memulai merekam pesan baru dengan memilih button Rekam Suara.



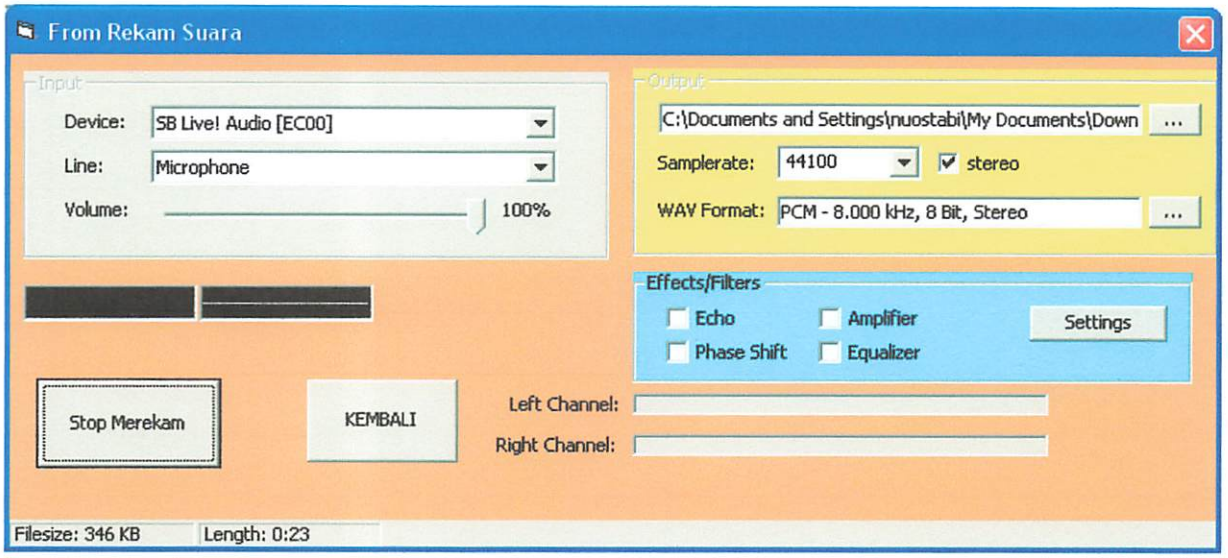
Gambar 4.10 Tampilan Utama

7. Pemasukan file yang sudah ada.



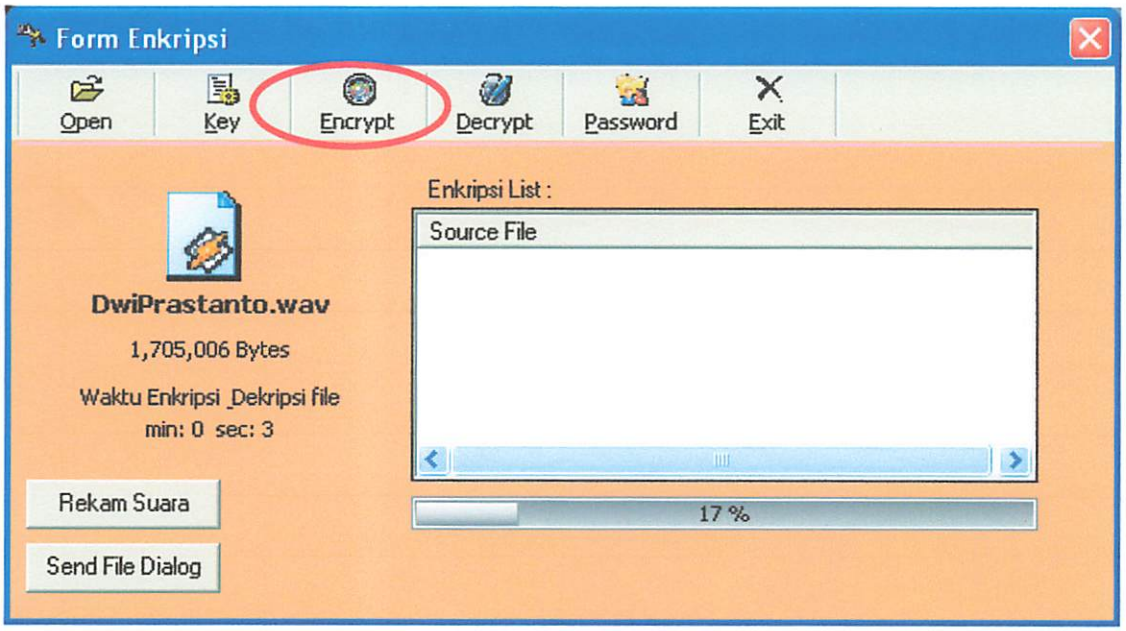
Gambar 4.11 Input file

8. Proses perekaman suara pada form rekam suara.



Gambar 4.12 Proses perekaman suara

9. Jika file sudah siap untuk di enkripsi, lakukan proses seperti pada tampilan gambar 4.13.

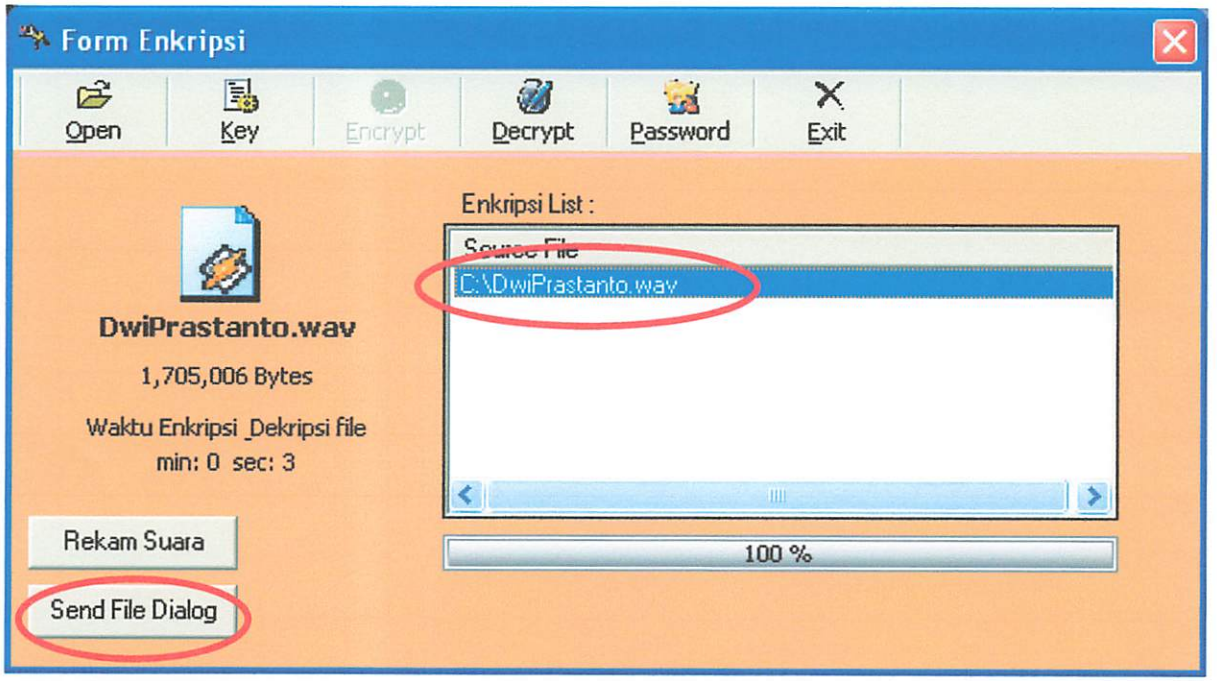


Gambar 4.13 Proses enkripsi

Jika enkripsi berhasil akan tampil informasi seperti tampilan gambar 4.14.

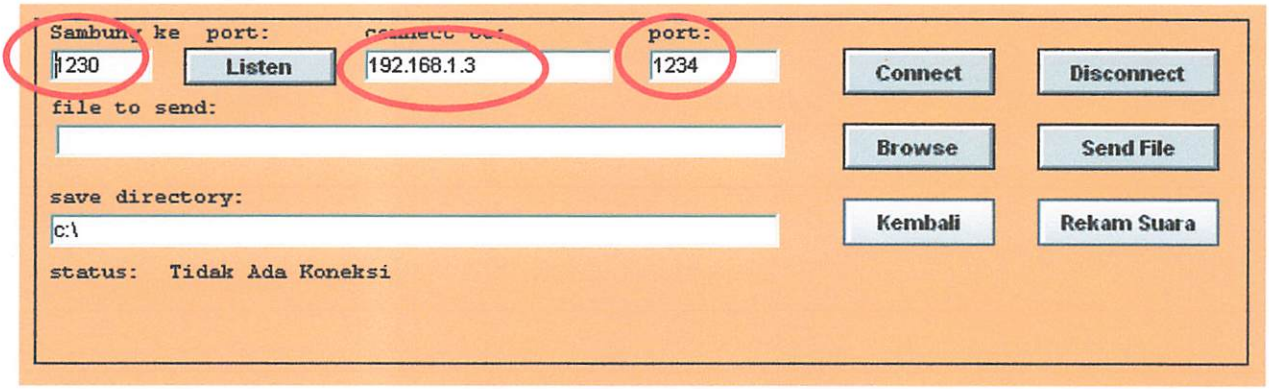
Gambar 4.14 Info Enkripsi berhasil

File yang telah terenkripsi akan tampil pada enkripsi list.



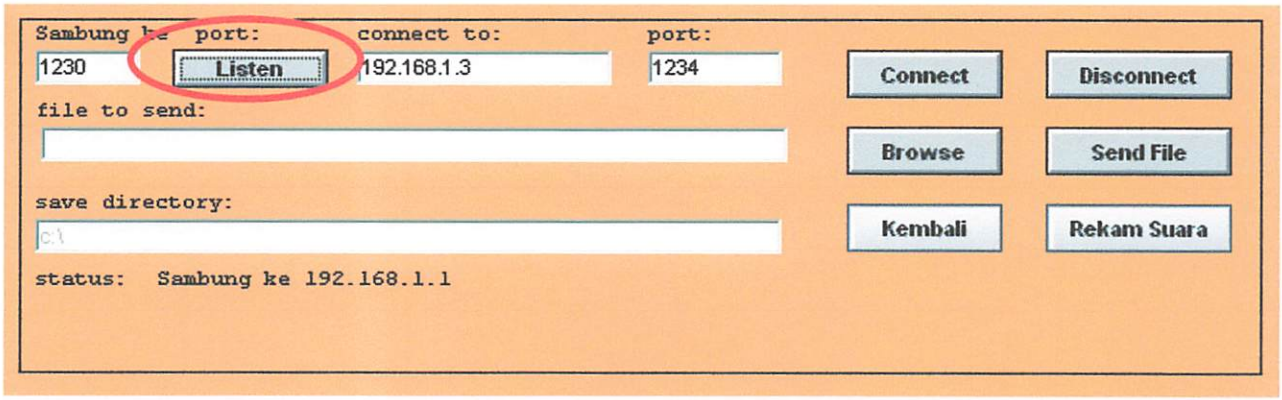
Gambar 4.15 Enkripsi sukses

10. Selanjutnya kita jika ingin mengirim hasil enkripsi suara kepada user 2 klik *button Send File Dialog* yang ada pada tampilan utama. Jika sudah masuk pada *form Send File Dialog*, kita di haruskan mengisi Local Port, Ip Address dan Remote Port (Port user 2). Jika semua sudah terisi pastikan Local Port dan Remote Port tidak salah, jika salah komunikasi data tidak akan terjadi.



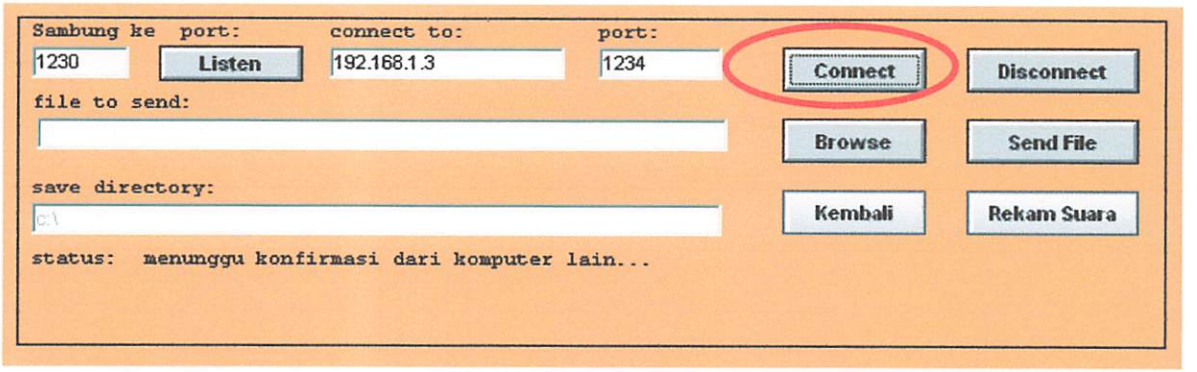
Gambar 4.16 Tampilan Form File Transfer

11. Jika semua sudah benar terisi, silahkan tekan tombol Listen untuk membuka port.



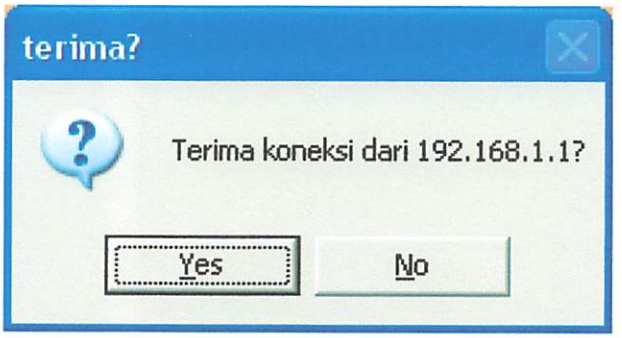
Gambar 4.17 Tampilan membuka koneksi

12. Jika sudah muncul status sambung ke alamat tujuan, kita bisa menekan tombol connect dan akan muncul status menunggu user 2 menerima koneksi kita.



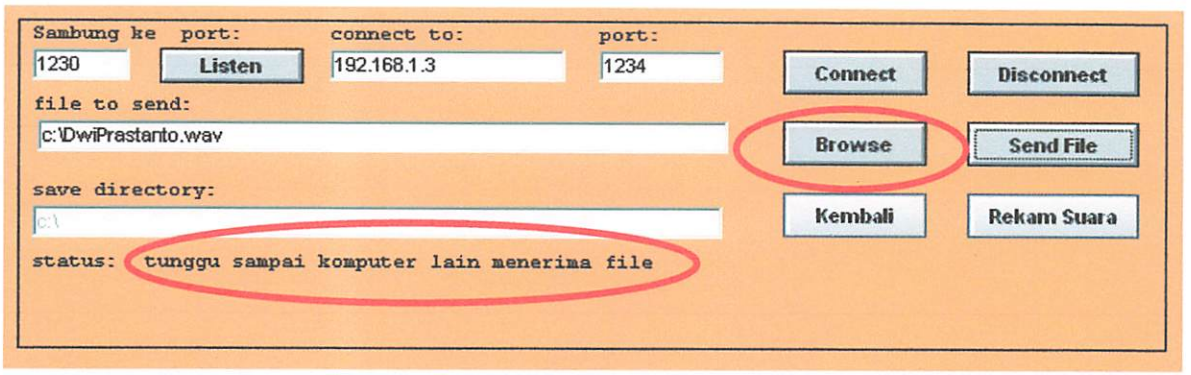
Gambar 4.18 Melakukan koneksi ke computer lain

13. Pada user 2 akan muncul tampilan menerima atau menolak koneksi dari user 1, dengan catatan user 2 sudah membuka port.



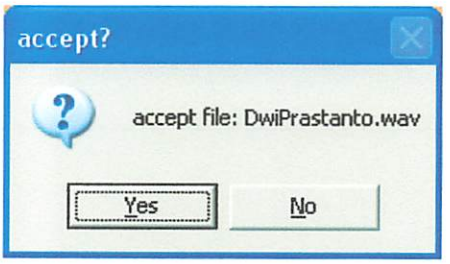
Gambar 4.19 Tampilan pada user 2

14. Jika user 2 menerima koneksi dari user 1 maka kita dapat melakukan transfer file yang kita enkripsi sebelumnya dengan memilih tombol browse pada form. Jika user 1 sudah memilih file yang sudah terenkripsi, user 1 sudah siap melakukan pengiriman file kepada user 2 dengan menekan tombol Send File.



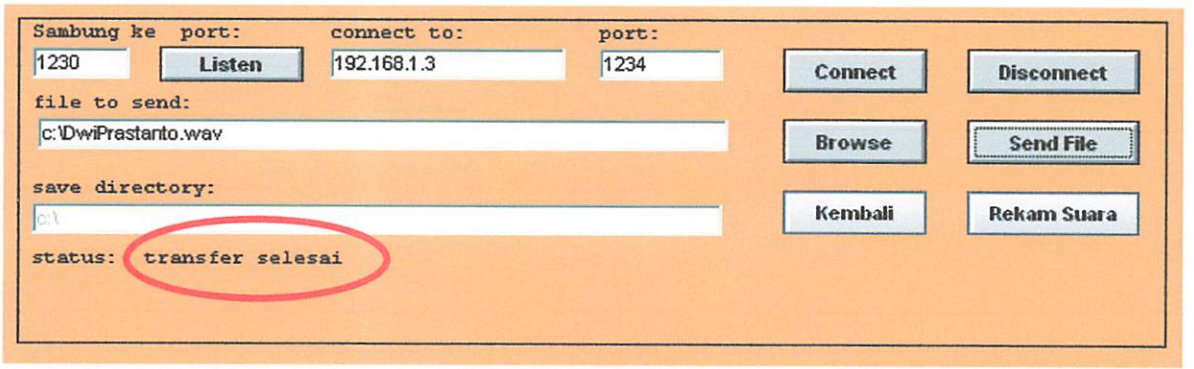
Gambar 4.20 Tampilan pengiriman file pada user 1

15. Pada user 2 akan muncul tampilan menerima file yang dikirimkan pada user 1 atau menolak pengiriman. Pada tampilan 4.21 saya asumsikan menerima file dari user 1.



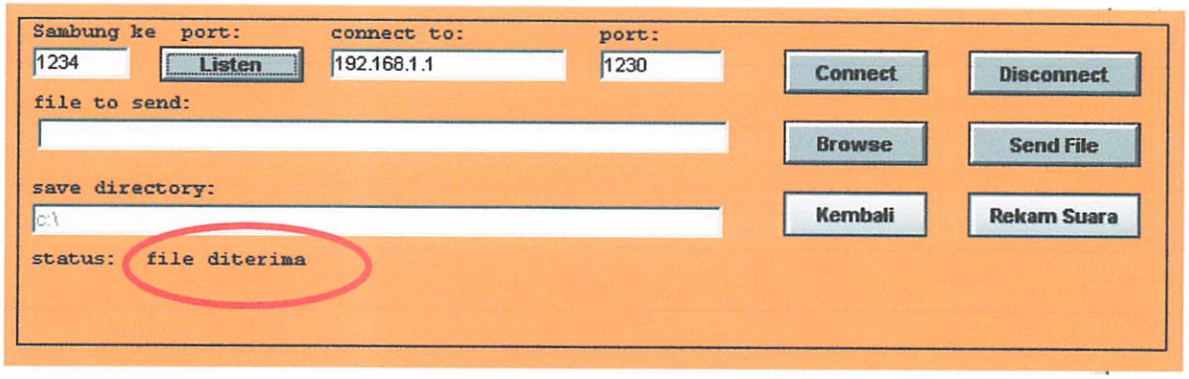
Gambar 4.21 Penerimaan file

16. Jika file sudah berhasil terkirim status pada user 1 akan mengidentifikasikan bahwa file transfer selesai.



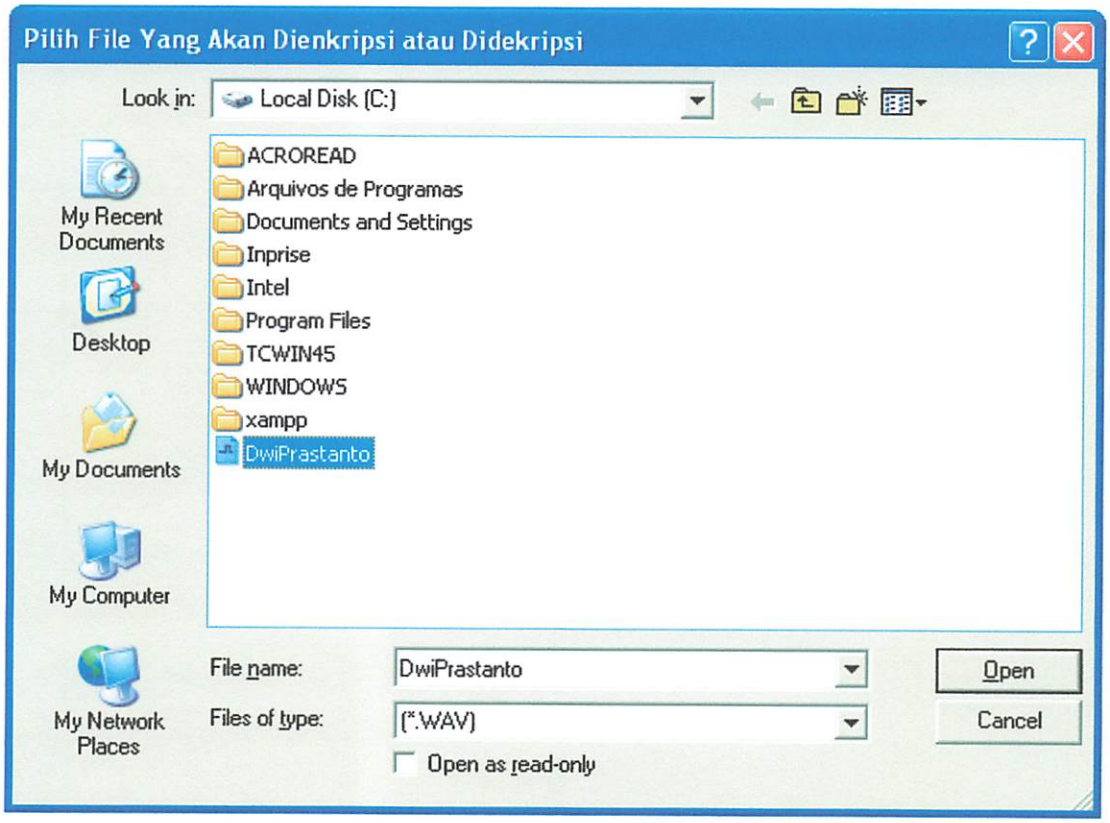
Gambar 4.22 File berhasil dikirim

17. Sedangkan pada user 2 akan muncul tampilan file telah diterima.



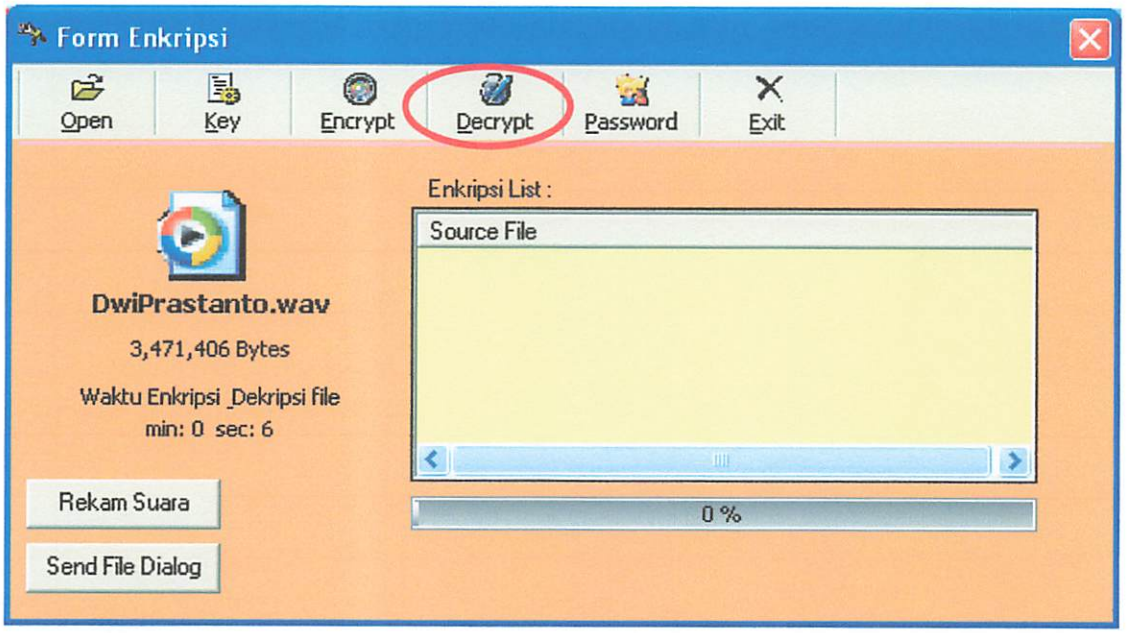
Gambar 4.23 File diterima

18. Jika user 2 ingin mendengarkan pesan suara yang telah dikirim dari user 1. Maka file tersebut harus di dekripsikan terlebih dahulu.



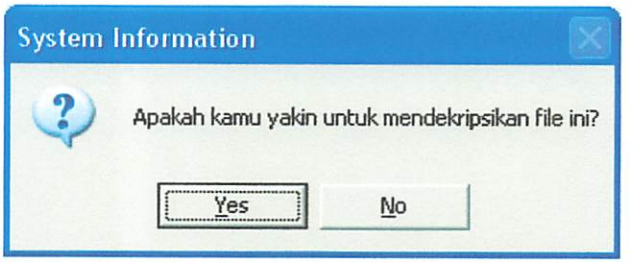
Gambar 4.24 Ambil File yang akan di dekripsi

19. Jika sudah di masukan file yang telah di enkripsi pada user 1. Lakukan prose dekripsi file.



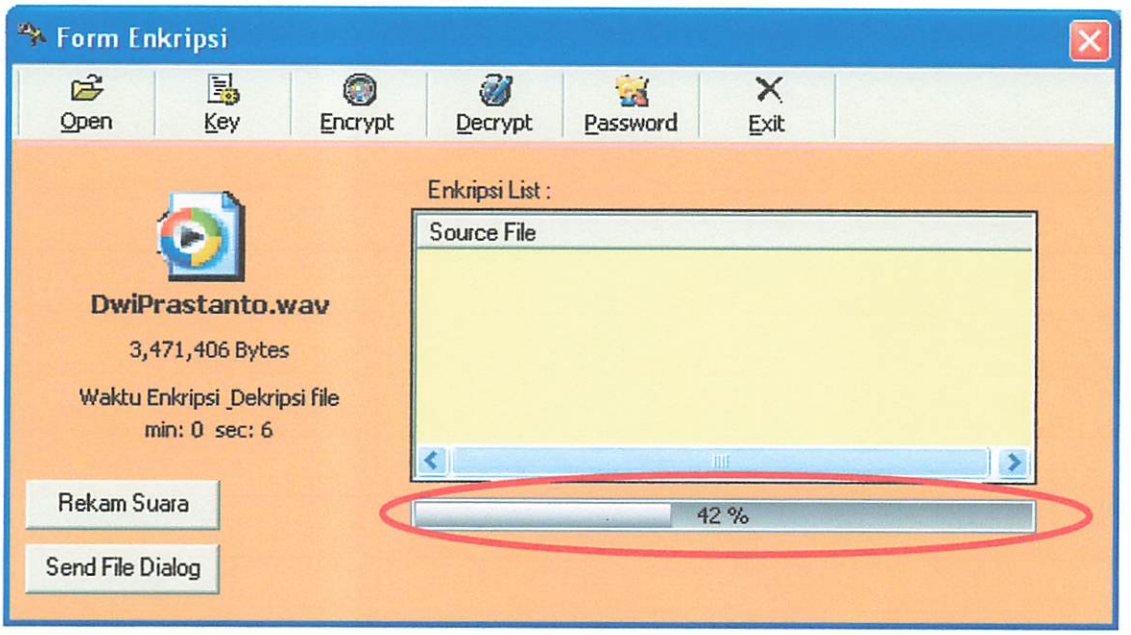
Gambar 4.25 Input file yang akan di dekripsi

20. Tampilan jika proses dekripsi telah berhasil dilakukan.



Gambar 4.26 Informasi dekripsi

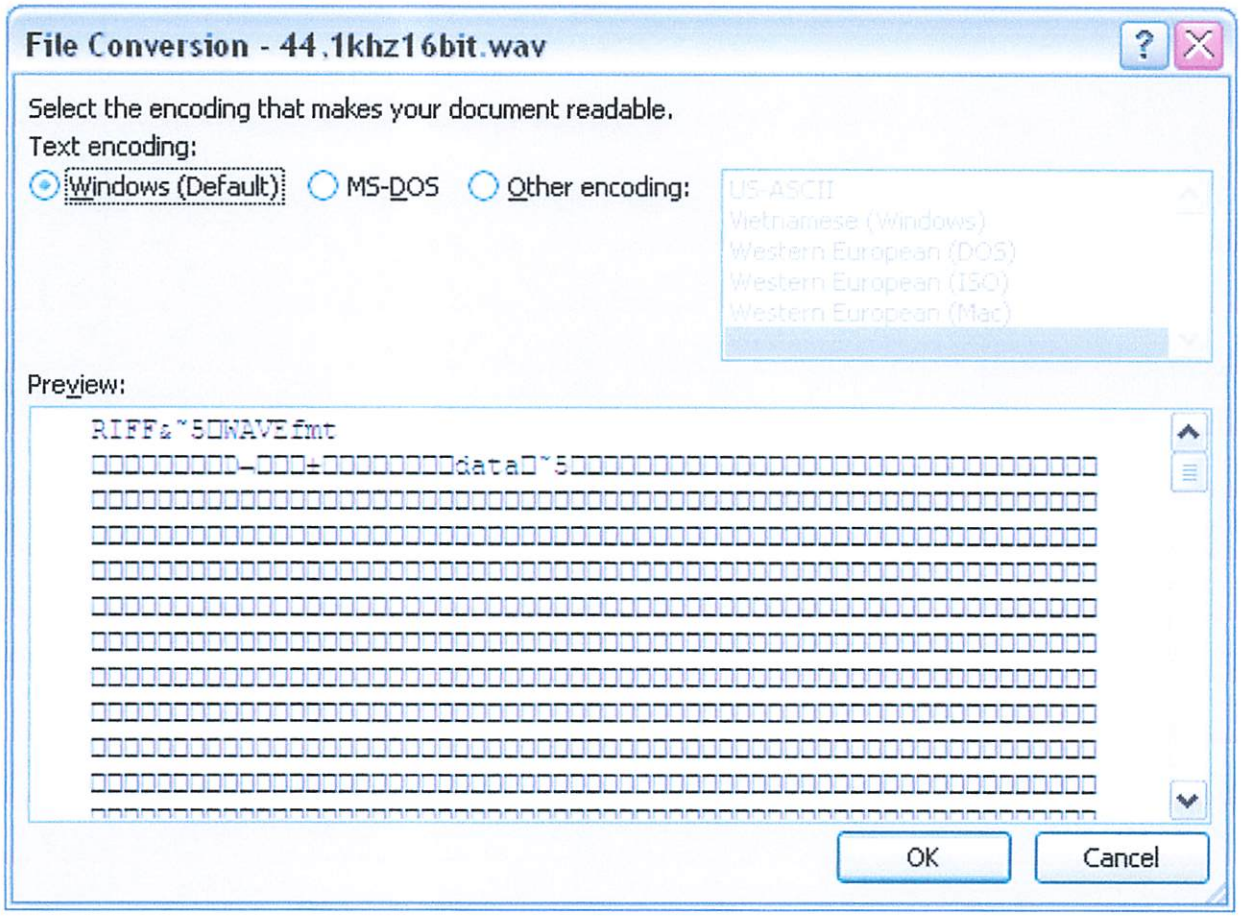
21. Proses dekripsi dijalankan telah selesai.



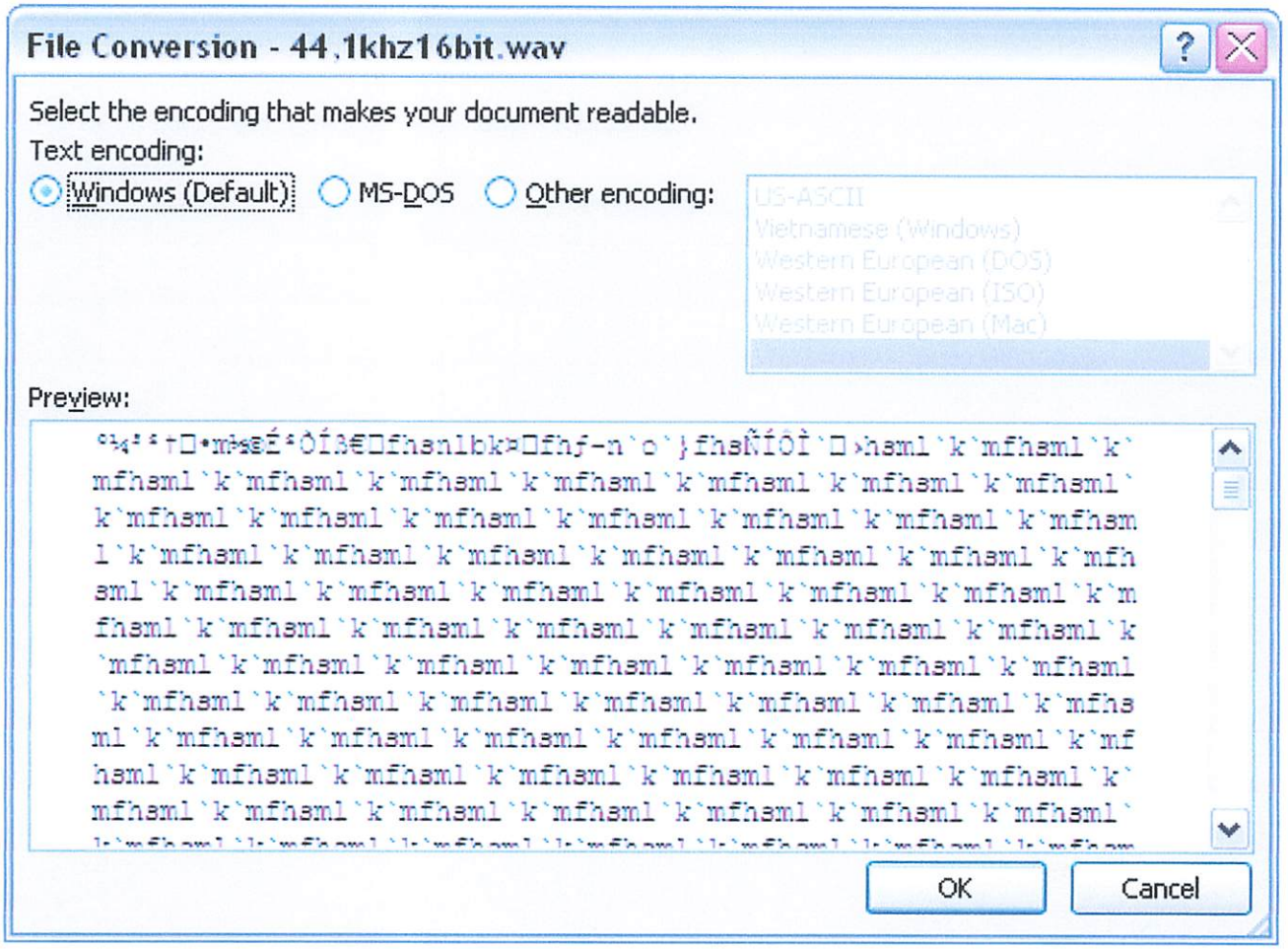
Gambar 4.27 tampilan proses dekripsi file

4.2.2 Hasil Enkripsi dan Dekripsi

Dari hasil percobaan terhadap file rekaman yang sudah terenkripsi dapat kita lihat dengan membuka file tersebut dengan Microsoft word, telah di dapat perbedaan hasil struktur dari file yang telah di jelaskan pada subbab 3.1.1

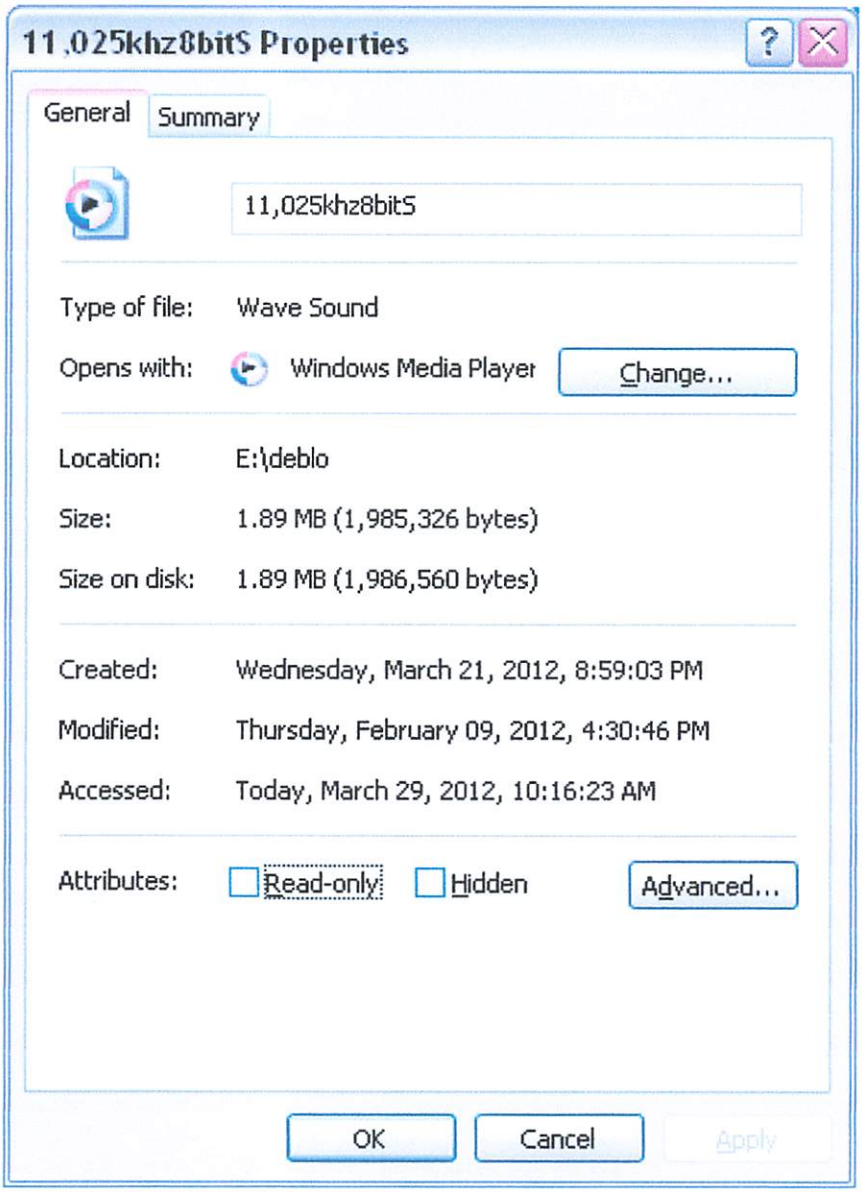


Gambar 4.28 Sebelum dilakukan enkripsi



Gambar 4.29 Sesudah dilakukan enkripsi

Ukuran file saat sebelum di lakukan proses enkripsi sesudah di enkripsi dan setelah di dekripsi mempunyai ukuran yang sama.



Gambar 4.30 Ukuran file

4.2.3 Perbandingan ukuran file

- Percobaan resolusi 8 bit, sample rate 8000 kHz WAV :

Size in bytes	1 menit	2 menit	3 menit
Mono	466 Kb	932 Kb	1.398 (1,4 Mb)
Stereo	934 Kb	1.868 (1,9 Mb)	2.802 (2,8 Mb)

- Percobaan resolusi 16 bit, sample rate 8000 kHz WAV :

Size in bytes	1 menit	2 menit	3 menit
Mono	934 Kb	1.868 (1,9 Mb)	2.802 (2,8 Mb)
Stereo	1,82 Mb	364 Mb	546 Mb

- Percobaan resolusi 8 bit, 11,025 kHz WAV :

Size in bytes	1 menit	2 menit	3 menit
Mono	644 Kb	1.288 (1,3 Mb)	1.932 (1,9 Mb)
Stereo	1,25 Mb	2,5 Mb	3,75 Mb

- Percobaan resolusi 16 bit, sample rate 11.025 kHz WAV :

Size in bytes	1 menit	2 menit	3 menit
Mono	1,25 Mb	2,5 Mb	3,75 Mb
Stereo	2,47 Mb	4,94 Mb	7,41 Mb

- Percobaan resolusi 8 bit, sample rate 22.050 kHz WAV :

Size in bytes	1 menit	2 menit	3 menit
Mono	1,25 Mb	2,5 Mb	3,75 Mb
Stereo	2,5 Mb	5 Mb	7,5 Mb

- Percobaan resolusi 16 bit, sample rate 22.050 kHz WAV :

Size in bytes	1 menit	2 menit	3 menit
Mono	2,5 Mb	5 Mb	7,5 Mb
Stereo	5,02 Mb	10,04 Mb	20,08 Mb

- Percobaan resolusi 8 bit, sample rate 44.100 kHz WAV :

Size in bytes	1 menit	2 menit	3 menit
Mono	2,51 Mb	5,02 Mb	7,53 Mb
Stereo	5,03 Mb	10,06 Mb	15,1 Mb

- Percobaan resolusi 16 bit, sample rate 44.100 kHz WAV :

Size in bytes	1 menit	2 menit	3 menit
Mono	5,02 Mb	10,04 Mb	20,08 Mb
Stereo	10 Mb	20 Mb	30 Mb

Berdasarkan beberapa percobaan yang telah dilakukan dapat di simpulkan ukuran file sangat bervariasi. Perbedaan resolusi dan sample rate sangat mempengaruhi besar ukuran file.

4.2.4 Pengujian Enkripsi file terhadap waktu dan frekuensi

Ada beberapa Format pada perekaman WAV yaitu :

3. PCM

Pengujian ini dilakukan untuk mengetahui waktu yang di butuhkan dalam beberapa frekuensi dan dua tipe berbeda yaitu Stereo dan Mono. Dalam pengujian ini saya asumsikan membatasi waktu 20 detik sebagai bahan pengukuran beberapa frekuensi.

Table 4.1 Hasil pengujian Enkripsi file tipe Stereo terhadap waktu dan frekuensi

NO.	FREKUENSI (kHz)	BIT	UKURAN FILE (Kb)	MANUAL WAKTU (s)	
				ENKRIPSI	**DEKRIPSI**
1.	**8.000**	**8**	**311**	**4,49**	**4,89**
2.	**8.000**	**16**	**617**	**5,34**	**5,41**
3.	**11.025**	**8**	**431**	**3,89**	**3,88**
4.	**11.025**	**16**	**858**	**7,59**	**7,89**
5.	**22.050**	**8**	**853**	**7,46**	**7,37**
6.	**22.050**	**16**	**1,711**	**13,2**	**15,32**
7.	**44.100**	**8**	**1,711**	**13,59**	**15,53**
8.	**44.100**	**16**	**3,431**	**28,06**	**29,73**

Table 4.2 Hasil pengujian Enkripsi file terhadap waktu rekam 20 detik tipe Mono dan frekuensi

NO.	FREKUENSI (kHz)	BIT	UKURAN FILE (Kb)	WAKTU (s)	
				ENKRIPSI	**DEKRIPSI**
1.	**8.000**	**8**	**156**	**1,51**	**1,46**
2.	**8.000**	**16**	**311**	**2,74**	**2,76**
3.	**11.025**	**8**	**215**	**1,97**	**2,01**
4.	**11.025**	**16**	**428**	**3,66**	**3,69**
5.	**22.050**	**8**	**429**	**3.66**	**3,76**
6.	**22.050**	**16**	**856**	**7,43**	**7,41**
7.	**44.100**	**8**	**851**	**7,59**	**7,62**
8.	**44.100**	**16**	**1,701**	**11,63**	**14,15**

Setelah di lakukan pengujian enkripsi file terhadap waktu dan frekuensi yang di dapat adalah pengaruh frekuensi, bit dan besar file rekaman mempengaruhi waktu dari enkripsi dan dekripsi. Pada pengukuran secara manual menggunakan stopwatch tidak sesuai dengan waktu yang diperkirakan. Pengukuran secara manual menimbulkan perbedaan yang signifikan.

4. Microsoft ADPCM

   Format ini merupakan bawaan dari windows itu sendiri yang hanya mempunyai satu frekuensi dan tipe stereo saja.

   Frekuensi = 44.100 Khz

   Bit = 4 bit

   Ukuran file = 692 kb

Lama waktu enkripsi = 4,5 sec

Lama waktu dekripsi = 4,7 sec

5. Windows media Audio v1 dan v2

Format ini merupakan bawaan dari windows itu sendiri yang hanya mempunyai satu frekuensi dan tipe stereo saja.

Frekuensi = 44.100 Khz

Bit rate = 160 kbps

Ukuran file = 274 kb

Lama waktu enkripsi = 2 sec

Lama waktu dekripsi = 1,9 sec

4.2.5 Pengujian frekuensi terhadap kualitas suara

Pengujian ini dilakukan untuk mengetahui frekuensi yang tepat untuk merekam suara, sehingga dapat mengetahui noise paling sedikit dalam ukuran file yang kecil.

Table 4.3 Hasil Pengujian frekuensi terhadap kualitas suara dengan tipe Stereo

NO.	FREKUENSI (kHz)	BIT	Rate (kb/sec)	JERNIH	JELEK
1.	**8.000**	**8**	**15**	**Tidak**	**Ya**
2.	**8.000**	**16**	**31**	**Ya**	**Tidak**
3.	**11.025**	**8**	**21**	**Tidak**	**Ya**
4.	**11.025**	**16**	**43**	**Ya**	**Tidak**
5.	**22.050**	**8**	**43**	**Tidak**	**Ya**
6.	**22.050**	**16**	**86**	**Ya**	**Tidak**

7.	**44.100**	**8**	**86**	**Tidak**	**Ya**
8.	**44.100**	**16**	**172**	**Ya**	**Tidak**

Dari hasil pengujian pada suara dengan tipe stereo serta perubahan bit dan frekuensi didapat hasil dimana pada saat perekaman menggunakan 8 bit suara tidak begitu jelas dan tingkat kejernihan suara rendah.

Table 4.4 Hasil Pengujian frekuensi terhadap kualitas suara dengan tipe Mono

NO.	FREKUENSI (kHz)	BIT	Rate (kb/sec)	JERNIH	JELEK
1.	**8.000**	**8**	**7**	**Tidak**	**Ya**
2.	**8.000**	**16**	**15**	**Tidak**	**Ya**
3.	**11.025**	**8**	**10**	**Tidak**	**Ya**
4.	**11.025**	**16**	**21**	**Tidak**	**Ya**
5.	**22.050**	**8**	**21**	**Tidak**	**Ya**
6.	**22.050**	**16**	**43**	**Tidak**	**Ya**
7.	**44.100**	**8**	**43**	**Tidak**	**Ya**
8.	**44.100**	**16**	**86**	**Tidak**	**Ya**

Dari pengujian pada tabel 4.3 dan 4.4 frekuensi terhadap kualitas suara maka didapat pengaruh tipe suara stereo dan mono, bit dan frekuensi suara ternyata mempengaruhi kualitas suara yang di rekam. Perbandingan tipe stereo lebih baik dari pada tipe mono.

BAB V
PENUTUP

5.1 Kesimpulan

Kesimpulan yang dapat diambil dari setelah pambahasan tentang penerapan algoritma *Serpent* pada aliran pesan dua arah melalui jaringan adalah:

1. Proses enkripsi, dekripsi, perekaman suara dan pengiriman dapat berjalan dengan baik pada *Microsoft visual basic 6.0.*
2. Dari hasil pengujian, perubahan frekuensi dan besar file suara berpengaruh pada waktu penyandian atau enkripsi.
3. Dari pengujian pada tabel 4.3 dan 4.4 frekuensi terhadap kualitas suara maka didapat pengaruh tipe suara stereo dan mono, bit dan frekuensi suara ternyata mempengaruhi kualitas suara yang di rekam. Perbandingan tipe stereo lebih baik dari pada tipe mono.
4. Hasil enkripsi suara rekaman memiliki format yang sama dengan aslinya.
5. Pada proses pertukaran data tidak di wajibkan untuk menerima file yang akan dikirim, kita bisa juga menolaknya.

5.2 Saran

Aplikasi enkripsi dan dekripsi pengiriman pesan suara dengan algoritma *serpent* ini masih jauh dari kata sempurna dan kurang efisien. Mungkin dalam pengembangannya dapat menggunakan metode penyandian atau komponen dan bahasa pemrograman lain dalam menjaga keamanan pesan data yang dikirim dalam suatu jaringan komputer.

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] Brandau, Albert, Markus. (2008), *Implementation of a real-time voice encryption system.*

[4] Hall, Martin. (1997). *Windows Sockets 2 Application Programming Interface.*

[5] Mangkulo, Alexander, Hengky. (2005). *Pemrograman pada jaringan computer dengan Visual Basic 6.0.*

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

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

[8] Syahrizal, Muhammad. (2007). *Mahir dan professional Visual Basic.*

[9] Sosemanuk, *a fast software-oriented stream cipher.*

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

URL :

[11] http://electronical-instrument.blogspot.com/2010/06/control-winsock-pada-visual-basic.html Tanggal akses : 3 januari 2012.

[12] *http://en.wikipedia.org/wiki/Serpent_(cipher)* Tanggal akses : 24 November 2011.

[13] http://pubercity.wordpress.com/2011/02/03/download-ebook-pemograman/ tanggal akses : 15 desember 2011.

[14] R.J Anderson, E. bilham, LR Knudsen,*"Serpent : "A proposal for the Advanced Encryption Standard"*, submitted to NIST as AES candidate,1998. Lebih lengkapnya dapat dilihat pada http://www.cl.cam.ac.uk/~rja14/serpent.html. Tanggal akses : 10 desember 2011.

[15] Torres, Gabriel. How Analog-to-Digital Converter (ADC) Works, URL: http://www.hardwaresecrets.com/article/317/, Tanggal akses : 1 januari 2012

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	:	Dwi Prastanto
Nim	:	07.12.625
Jurusan	:	Teknik Elektro S-1
Konsentrasi	:	Teknik Komputer dan Informatika S-1
Judul	:	**ENKRIPSI DAN DEKRIPSI PENGIRIMAN PESAN SUARA DENGAN ALGORITMA *SERPENT*.**

Dipertahankan dihadapan Tim Penguji Skripsi Jenjang Program Strata Satu (S-1)

Pada Hari : Rabu
Tanggal : 22 Februari 2012
Dengan Nilai : 79,3 (B+)

PANITIA UJIAN SKRIPSI

KETUA

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

SEKRETARIS

Dr. Aryuanto S, ST. MT
NIP.P.1030800417

ANGGOTA PENGUJI

Dosen Penguji I

Irmalia Suryani Faradisa, ST, MT
NIP.P. 1030000365

Dosen Penguji II

Ahmad Faisol, ST
NIP.P. 1031000431

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
Jl. Karanglo KM.2 Malang

FORMULIR PERBAIKAN SKRIPSI

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

Nama : Dwi Prastanto
Nim : 07.12.625
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer dan Informatika S-1
Judul : **ENKRIPSI DAN DEKRIPSI PENGIRIMAN PESAN SUARA DENGAN ALGORITMA *SERPENT*.**

Tanggal	Uraian	Paraf
Penguji I 22 Februari 2012	1. Bab 3 : Perancangan algoritmanya tidak tercantum di bab 3. 2. Analisa tidak terlihat. 3. Bab 4 : pengujian tidak sesuai dengan bab 1. 4. Demo Ulang	
Penguji II 22 Februari 2012	1. Tambah pengujian filter dan format file. 2. Tambah pengujian untuk ukuran file setelah di enkripsi dan setelah di dekripsi. 3. Uji ulang dekripsi file yang dikirim. 4. Perjelas tujuan masalah. 5. Ganti rancangan program.	

Disetujui :

Dosen Penguji I

Irmalia Suryani Faradisa, ST, MT
NIP.P. 1030000365

Dosen Penguji II

Ahmad Faisol, ST
NIP.P. 1031000431

Mengetahui :

Dosen Pembimbing I

Josep Dedy Irawan, ST, MT
NIP. 19740416.200501.1002

Dosen Pembimbing II

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

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

PERMOHONAN PERSETUJUAN SKRIPSI

Yang betanda tangan dibawah ini :

Nama	: Dwi Prastanto
NIM	: 0712625
Semester	: Sembilan (IX)
Fakultas	: Teknologi Industri
Jurusan	: Teknik Elektro S-1
Konsentrasi	: ~~TEKNIK ELEKTRONIKA~~ ~~TEKNIK ENERGI LISTRIK~~ TEKNIK KOMPUTER DAN INFORMATIKA ~~TEKNIK KOMPUTER~~ ~~TEKNIK TELEKOMUNIKASI~~
Alamat	: .....

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

(Ismi Handayani)

Malang, 19 Oktober 201
Pemohon

(Dwi Prastanto)

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. IP 423.5 / 138 = 3.07
2. .....
3. Praktikum = lengkap.

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi

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

NAMA : Dwi Prastanto
N I M : 0712 625
Perbaikan meliputi :

BAB 3 =
- Perancangan ~~dkk~~ algoritmanya tdk tercantum di bab 3.
- Analisa tidak terlinat.

BAB 4.
- pengujian tdk sesuai dg bab 1.

- Demo ulang.
- Sebisi revisi thg maju dg dosen pembimbing.

Malang,

(_____)

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Jenjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika / Telekomunikasi / Komputer, saya yang bertanda tangan dibawah ini :

NAMA : Dwi Prastanto
NIM : 07.12.625
Perbaikan meliputi :

1. Aplikasi kurang ketelitian dlm pengiriman file, untuk proses pengiriman file yg proses scanning otomatis.
2. Tambah pengujian tiap filter & format file
3. Tambah pengujian untuk ukuran file sebelum, setelah di enkrip, & setelah di dekrip.
4. Uji ulang dekripsi file yg dikirim.
5. Perlu penjelasan tentang bagaimana & apa saja Rancangan Program

Malang, 22-02-12

A. Priyanto

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: DWI PRASTANTO			Nim: 0712625
2.	Waktu Pengajuan	Tanggal: 27	Bulan: Oktober	Tahun: 2011
3.	Spesifikasi Judul (berilah tanda silang)**) 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 komputer dan informatika			
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:	ENKRIPSI DAN DEKRIPSI PENGIRIMAN PESAN SUARA DENGAN ALGORITMA SERPENT		
6.	Perubahan judul yang disetujui Dosen sesuai materi bidang ilmu			
7.	Catatan: Persetujuan Judul skripsi yang dikonsultasikan kepada Dosen materi bidang ilmu	Disetujui 28/10/201 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

Jln. Raya Karanglo, Km. 2 Malang

Lampiran : 1 (satu) berkas

Pembimbing Skripsi

Kepada : Yth. Bapak Ibrahim Ashari, ST, MT

Dosen Institut Teknologi Nasional Malang

Yang bertanda tangan di bawah ini:

Nama : Dwi Prastanto

NIM : 07.12.625

Jurusan : Teknik Elektro S-1

Konsentrasi : Teknik Komputer dan Informatika

Dengan ini mengajukan permohonan, kiranya Bapak bersedia menjadi Dosen Pembimbing Pendamping untuk peyusunan Skripsi dengan judul (proposal terlampir):

" ENKRIPSI DAN DEKRIPSI PENGIRIMAN PESAN SUARA DENGAN ALGORITMA *SERPENT* "

Adapun tugas tersebut sebagai salah satu syarat untuk menempuh Ujian Akhir Sarjana Teknik.

Demikian permohonan kami buat dan atas kesediaaan Bapak kami ucapkan terima kasih.

Mengetahui

Ketua Jurusan Teknik Elektro

Ir.Yusuf Ismail Nakhoda,MT

NIP. Y. 101.880.0189

Malang ,

Hormat kami

Dwi Prastanto

Form S-3a

INSTITUT TEKNOLOGI NASIONAL

Jln. Raya Karanglo, Km. 2 Malang

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa/i:

Nama : Dwi Prastanto

NIM : 07.12.625

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:

"ENKRIPSI DAN DEKRIPSI PENGIRIMAN PESAN SUARA DENGAN ALGORITMA *SERPENT* "

Demikian surat pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Malang,

Kami yang membuat pernyataan

Yoseph Dedy Irawan,ST ,MT
NIP. [redacted]19740416.200501.1002

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 : Dwi Prastanto

NIM : 07.12.625

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:

"ENKRIPSI DAN DEKRIPSI PENGIRIMAN PESAN SUARA DENGAN ALGORITMA *SERPENT* "

Demikian surat pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Malang,

Kami yang membuat pernyataan

Ibrahim Ashari, ST,MT
NIP.103.010.0358

Catatan :

Setelah disetujui agar formulir ini

Diserahkan mahasiswa/I yang bersangkutan

Kepada Jurusan untuk diproses lebih lanjut.

*) Coret yang tidak perlu

Form S-3b

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: Dwi Prastanto			Nim: 0712625
2.	Keterangan Pelaksanaan	Tanggal: 30 November 201	Waktu:	Tempat Ruang:
3.	Spesifikasi Judul (berilah tanda silang)**) 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 Komputer dan Informatika			
4.	Judul Proposal yang diseminarkan Mahasiswa	ENKRIPSI DAN DESKRIPSI PENGIRIMAN PESAN SUARA DENGAN ALGORITMA SERPENT		
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
A. Fauzi	Bima Arya F	Sotyohadi

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

Perhatian:

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

Form S-3c

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG

INSTITUT TEKNOLOGI NASIONAL MALANG

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

. 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- 876/I.TA/2/11
Lampiran : -
Perihal : BIMBINGAN SKRIPSI

Kepada : Yth. Sdr/I. **JOSEPH DEDY IRAWAN, 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	: DWI PRASTANTO
Nim	: 0712625
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 :

30 Nopember 2011 s/d 30 Mei 2012

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

INSTITUT TEKNOLOGI NASIONAL MALANG FAKULTAS TEKNOLOGI INDUSTRI JURUSAN TEKNIK ELEKTRO STRATA-1

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
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- 875/I.TA/2/11
Lampiran : -
Perihal : BIMBINGAN SKRIPSI

Kepada : Yth. Sdr/I. **M.IBRAHIM ASHARI, 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	: DWI PRASTANTO
Nim	: 0712625
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 :

30 Nopember 2011 s/d 30 Mei 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

FORMULIR BIMBINGAN SKRIPSI

Nama : DWI PRASTANTO
Nim : 07.12.625
Masa Bimbingan : 30 NOPEMBER 2011 s/d 30 MEI 2012
Judul Skripsi : **ENKRIPSI DAN DEKRIPSI PENGIRIMAN PESAN SUARA DENGAN ALGORITMA *SERPENT***

No.	Tanggal	Uraian	Paraf Pembimbing
1.	2-12-2011	ACC Proposal Seminar	
2.	2-02-2012	Revisi Bab II, IV, V	
3.	6-02-2012	Acc bab II, IV, V	
4.	8-02-2012	Revisi Bab I, II	
5.	9-02-2012	Acc Bab I, II	
6.	12-02-2012	Revisi Makalah Seminar Hasil	
7.	13-02-2012	Acc Makalah Seminar Hasil	
8.		Acc kompre	
9.			
10.			

Malang,
Dosen Pembimbing I

Joseph Dedy Irawan ,ST,MT
NIP.19740416.200501.1002

Form S-4b

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

FORMULIR BIMBINGAN SKRIPSI

Nama : DWI PRASTANTO
Nim : 07.12.625
Masa Bimbingan : 30 NOPEMBER 2011 s/d 30 MEI 2012
Judul Skripsi : **ENKRIPSI DAN DEKRIPSI PENGIRIMAN PESAN SUARA DENGAN ALGORITMA *SERPENT***

No.	Tanggal	Uraian	Paraf Pembimbing
1.	1-02-2012	ACC Bab I .	
2.	1-02-2012	ACC Bab II .	
3.	10-02-2012	Revisi Bab III dan IV .	
4.	13-02-2012	ACC Bab III dan IV .	
5.	17-02-2012	ACC Bab V .	
6.	18-2-2012	Revisi makalah Seminar hasil	
7.	19-2-2012	ACC makalah Seminar hasil	
8.	20/2/2012	ACC makalah kompre.	
9.			
10.			

Malang,
Dosen Pembimbing II

M. Ibrahim Ashari ,ST,MT
NIP.103.010.0358

Form S-4b

LISTING PROGRAM

SOURCE CODE

FORM ENKRIPSI DAN DEKRIPSI

```
Option Explicit

Private WithEvents Encryptserpent As clsSerpent

Dim Key As String

Dim crate As String

Dim tmplen As String

Dim cSec As String

Dim cMin As String

Dim Append As Long

Private Sub Form_Load1()

Me.Move (Screen.Width - Me.Width) / 2, (Screen.Height - Me.Height) / 2

Set Encryptserpent = New clsSerpent

End Sub

Private Sub Command1_Click()

Form1.Show

Unload Me

End Sub

Private Sub Command2_Click()

Form2.Show

frm_Thesis.Hide

Form1.Hide

End Sub

Private Sub Form_Load()

   Init_List

   CheckMenu

   Call PKencrypt(Ekey)

   Ekey = strConData

End Sub

Private Sub lvwList_DblClick()

If Me.lvwList.ListItems.Count <> 0 Then
```

```
    Call PopupMenu(mnupopup)
    Me.ToolBar.Buttons(6).Enabled = False
  End If
  End Sub
  Private Sub lvwList_ItemClick(ByVal item As MSComctlLib.ListItem)
    cd.Filename = Empty
    strFile = Me.lvwList.SelectedItem
    lblSource.Caption = Me.lvwList.SelectedItem
    strSourceFile = Me.lvwList.SelectedItem.Text
    Me.lblFileName.Caption = Me.lvwList.SelectedItem.SubItems(1)
    Me.lblBytes.Caption = Format(FileLen(strFile), "###,###,###,###") & " Bytes"
    Me.lblConversionTime.Caption = "Waktu Dekripsi File"
    GetConvertionTime
    intIndex = Me.lvwList.SelectedItem.Index
    SetIcon
    Me.ToolBar.Buttons(8).Enabled = True
    Me.ToolBar.Buttons(6).Enabled = False
  End Sub
  Private Sub mnuDecrypt_Click()
    dCrypto
  End Sub
  Private Sub mnuRemove_Click()
    RMList
  End Sub

  Private Sub ToolBar_ButtonClick(ByVal Button As MSComctlLib.Button)
  Select Case Button.Index
    Case 2
      GetFile
      frm_Thesis.xp_Prog.value = 0
    Case 4
```

```
        frm_Key.Show 1
    Case 6
        crypto
    Case 8
        dCrypto
    Case 10
        frm_SetPassword.Show 1
    Case 12
        Unload Me
End Select
End Sub
Private Sub GetFile()
On Error GoTo err:
Dim cSec As Integer
Dim cMin As Integer
With cd
.DialogTitle = "Pilih File Yang Akan Dikirim"
.Filter = "(*.WAV)|*.wav|"
.ShowOpen
End With
    strSourceFile = cd.Filename
    strFile = cd.FileTitle
    If strFile <> "" Then
        Me.lblFileName.Caption = strFile
        strByte = Format(FileLen(strFile), "###,###,###,###")
        Me.lblBytes.Caption = strByte & " Bytes"
        Me.lblConversionTime.Caption = "Waktu Enkripsi & Dekripsi file"
        'Me.lvwList.ListItems.Clear
        'Me.lvwList.ListItems.Add , , strSourceFile
        'Me.lvwList.ListItems(Me.lvwList.ListItems.Count).SubItems(1) = strFile
        'Me.lvwList.ListItems(Me.lvwList.ListItems.Count).SubItems(2) = Format(FileLen(strFile),
"###,###,###,##0") & " Bytes"
```

```
        GetConvertionTime
        SetIcon
        Me.ToolBar.Buttons(6).Enabled = True
    End If
    Exit Sub
err:
MsgBox err.Description, vbCritical, "Error"
End Sub
Private Sub GetConvertionTime()
    crate = RATE_ENCRYPT
    If FileLen(strFile) > 75000000 Then
        tmplen = FileLen(strFile) / 20
        cSec = tmplen / crate
    Else
        cSec = FileLen(strFile) / crate
    End If
    cMin = Int(cSec / 60)
    cSec = Int(cSec - (cMin * 60))
    If cSec = 0 Then cSec = 1
    lblTime.Caption = "min: " & Trim(Str(cMin)) & "  sec: " & Trim(Str(cSec))
    'Me.lvwList.ListItems(Me.lvwList.ListItems.Count).SubItems(3) = "min: " & Trim(Str(cMin)) & "  sec: " &
Trim(Str(cSec))
End Sub
Private Sub crypto()
If MsgBox("Apakah kamu yakin untuk mengenkripsi file ini?", vbQuestion + vbYesNo, "System Information") =
vbYes Then
    strSourceFile = cd.Filename
    LookUp = True
    Call EncFile(strSourceFile, strSourceFile)
    strSourceFile = ""
    cd.Filename = Empty
Else
```

```
    Exit Sub
End If
Me.ToolBar.Buttons(6).Enabled = False
End Sub
Private Sub dCrypto()
If MsgBox("Apakah kamu yakin untuk mendekripsikan file ini?", vbQuestion + vbYesNo, "System Information")
= vbYes Then
    Me.ToolBar.Buttons(8).Enabled = True
    LookUp = False
    Call EncFile(strSourceFile, strSourceFile)
    If strSourceFile = "" Then
        Exit Sub
    End If
    If strSourceFile = cd.Filename Then
        MsgBox "Buka File." & vbCrLf & "Jika File ada pada Urutan File terenkripsi, Tolong Remove Secara
Manual.", vbInformation, "Information"
        strSourceFile = ""
        cd.Filename = Empty
        Exit Sub
    End If
    RMList
Else
    Exit Sub
End If
End Sub
Private Sub CheckMenu()
If Me.lvwList.ListItems.Count = 0 Then
    Me.ToolBar.Buttons(8).Enabled = True
Else
    Me.ToolBar.Buttons(8).Enabled = True
End If
    Me.ToolBar.Buttons(6).Enabled = False
```

```
End Sub
Private Sub SaveEnc()
   Open frm_Thesis.cd.Filename For Binary As #6
      Put #6, , strConData
   Close #6
End Sub
```

FORM REKAM SUARA

```
Option Explicit
' VU gradient colors
Private Const COLOR_START      As Long = vbGreen
Private Const COLOR_MIDDLE     As Long = &H22A8E1
Private Const COLOR_END        As Long = vbRed
' the bigger, the smoother
Private Const VU_SMOOTH_LEN    As Long = 3
Private WithEvents clsRecorder  As WaveInRecorder
Private clsVUSmoothL          As clsSmooth
Private clsVUSmoothR          As clsSmooth
Private clsVis                As clsDraw
Private clsDSP                As clsDSP
Private intSamples()          As Integer
Private clsEncoder            As EncoderWAV
Private lngMSEncoded          As Long
Private lngBytesPerSec        As Long
Private blnLoaded             As Boolean
Private Sub cboDev_Click()
   cboRecLine.Clear
   If Not clsRecorder.SelectDevice(cboDev.ListIndex) Then
      MsgBox "Tidak bisa memilih perangkat!", vbExclamation
      Exit Sub
   End If
   ShowLines
```

```
End Sub
Private Sub cboRecLine_Click()
    If Not clsRecorder.SelectMixerLine(cboRecLine.ListIndex) Then
        MsgBox "Tidak Bisa Memilih Microphone!", vbExclamation
    End If
    ' MixerLineType can be used to automaticaly find and set
    ' the line you want to record from, e.g. microphone.
    ' MixerLine also accepts a line id as a parameter,
    ' pass -1 and the currently selected line is returned.
    Debug.Print "Line Type: ";
    Select Case clsRecorder.MixerLineType
        Case MIXERLINE_ANALOG:     Debug.Print "Analog"
        Case MIXERLINE_AUXILIARY:  Debug.Print "Auxiliary"
        Case MIXERLINE_COMPACTDISC: Debug.Print "Compact Disc"
        Case MIXERLINE_DIGITAL:    Debug.Print "Digital"
        Case MIXERLINE_LINE:       Debug.Print "Line-In"
        Case MIXERLINE_MICROPHONE: Debug.Print "Microphone"
        Case MIXERLINE_PCSPEAKER:  Debug.Print "PC Speaker"
        Case MIXERLINE_SYNTHESIZER: Debug.Print "Synthesizer"
        Case MIXERLINE_TELEPHONE:  Debug.Print "Telephone"
        Case MIXERLINE_UNDEFINED:  Debug.Print "Undefined"
        Case MIXERLINE_WAVEOUT:    Debug.Print "WaveOut"
        Case Else:                 Debug.Print "Unknown"
    End Select
    sldVol.value = clsRecorder.MixerLineVolume
    sldVol_Scroll
End Sub
Private Sub cboSamplerate_Click()
    ' the currently selected WAV output codec
    ' couldn't support resampling, so force the
    ' user to select a new format which is
```

```
    ' compatible to the new samplerate
    If blnLoaded Then
      cmdSelFmt_Click
      clsDSP.samplerate = CLng(cboSamplerate.Text)
    End If
  End Sub
  Private Sub cboSamplerate_KeyPress( _
    KeyAscii As Integer _)
    If KeyAscii <> vbKeyBack Then
      If Not IsNumeric(Chr(KeyAscii)) Then
        KeyAscii = 0
      Else
        If Len(cboSamplerate.Text) = 5 Then
          KeyAscii = 0
        End If
      End If
    End If
  End Sub
  Private Sub chkFX_Click( _
    Index As Integer _)
    If chkFX(Index).value Then
      clsDSP.EffectsUsed = clsDSP.EffectsUsed Or (2 ^ Index)
    Else
      clsDSP.EffectsUsed = clsDSP.EffectsUsed And (Not (2 ^ Index))
    End If
  End Sub
  Private Sub chkStereo_Click()
    'Saat ini dipilih WAV keluaran codec
    'Tidak dapat mendukung saluran pencampuran, sehingga memaksa
    'Pengguna untuk memilih format baru yang
    'Kompatibel ke saluran baru menghitung
```

```
    If blnLoaded Then
        cmdSelFmt_Click
        clsDSP.Channels = chkStereo.value + 1
    End If
End Sub
' waveIn returns a buffer to the app
Private Sub clsRecorder_GotData( _
    intBuffer() As Integer, _
    lngLen As Long _
)
    ' menyimpan buffer saat ini untuk memvisualisasikan
    intSamples = intBuffer
    clsDSP.ProcessSamples intSamples
' Got Data juga bisa dinaikkan setelah perekaman
    'Mendapat dihentikan karena buffer hanya selesai
    'Ketika Rekam Berhenti mendapat disebut
    If Not clsRecorder.IsRecording Then Exit Sub
    ' memperbarui waktu direkam
    lngMSEncoded = lngMSEncoded + ((lngLen / lngBytesPerSec) * 1000)
    If Not clsEncoder Is Nothing Then
        ' mengirim data PCM untuk encoder WAV
        If clsEncoder.Encoder_Encode(VarPtr(intSamples(0)), lngLen, 0) = SND_ERR_WRITE_ERROR Then
            cmdDo_Click
            MsgBox "Kesalahan(tidak ada ruang disk?)!", vbExclamation
        End If
    End If
End Sub
Private Sub cmdBrowse_Click()
    Dim strExt  As String

    strExt = "*." & clsEncoder.Encoder_Extension
```

```
    With dlg
      .Filename = vbNullString
      .Flags = cdlOFNOverwritePrompt
      .Filter = clsEncoder.Encoder_Description & " (" & strExt & ")|" & strExt
      .ShowOpen
    End With
    If dlg.Filename <> vbNullString Then
      If Not Right$(LCase$(dlg.Filename), 4) = ".wav" Then
        txtFile.Text = dlg.Filename & ".wav"
      Else
        txtFile.Text = dlg.Filename
      End If
    End If
End Sub
Private Sub cmdDo_Click()
    Dim sndres  As SND_RESULT
    If clsRecorder.IsRecording Then
      If Not clsRecorder.StopRecord Then
        MsgBox "Tidak dapat berhenti merekam!", vbExclamation
      End If
      clsEncoder.Encoder_EncoderClose
      lngMSEncoded = 0
      lngBytesPerSec = 0
      cmdDo.Caption = "Mulai Merekam"
      SetFormEnabled True
    Else
      If txtFile.Text = "" Then
        MsgBox "Masukan Output File!", vbExclamation
        Exit Sub
      End If
      If cboDev.ListIndex = -1 Then
```

```
        MsgBox "Device Belum Dipilih!", vbExclamation
        Exit Sub
    End If
    ' Blockalign = (Bits/Sample / 8) * Channels
    ' Bytes/Sec = Samplerate * Blockalign
    ' Kita hanya menggunakan 16 sampel bit (bilangan bulat) di mana kita bisa,
    'Itu cara yang paling nyaman untuk bekerja dengan
    'Sampel di VB
    lngBytesPerSec = (CLng(cboSamplerate.Text) * (2 * (chkStereo.value + 1)))

    sndres = clsEncoder.Encoder_EncoderInit(CLng(cboSamplerate.Text), _
                                chkStereo.value + 1, _
                                txtFile.Text)
    If sndres <> SND_ERR_SUCCESS Then
        MsgBox "Could not init the encoder!", vbExclamation
        Exit Sub
    End If
    clsDSP.samplerate = CLng(cboSamplerate.Text)
    clsDSP.Channels = chkStereo.value + 1
    If Not clsRecorder.StartRecord(cboSamplerate.Text, chkStereo.value + 1) Then
        MsgBox "Tidak dapat mulai merekam!", vbExclamation
    End If

    cmdDo.Caption = "Stop Merekam"
    SetFormEnabled False
  End If
End Sub
Private Sub SetFormEnabled( _
  bln As Boolean _)
  frmOut.Enabled = bln
  frmInp.Enabled = bln
```

```
End Sub
Private Sub cmdFXOpts_Click()
   frmFXOpts.ShowEx clsDSP, Me
End Sub
Private Sub cmdSelFmt_Click()
   clsEncoder.SelectFormat CLng(cboSamplerate.Text), _
                  chkStereo.value + 1, _
                  Me.hWnd, _
                  "Pilih output format WAV"
   UpdateWAVFmtDisp
End Sub
Private Sub UpdateWAVFmtDisp()
   With clsEncoder
      txtWAVFmt.Text = .FormatTag & " - " & .FormatID
   End With
End Sub
Private Sub Command1_Click()
frm_Thesis.Show
Unload Me
End Sub

Private Sub Form_Load()
   Set clsVUSmoothL = New clsSmooth
   Set clsVUSmoothR = New clsSmooth
   Set clsRecorder = New WaveInRecorder
   Set clsEncoder = New EncoderWAV
   Set clsVis = New clsDraw
   Set clsDSP = New clsDSP
   ' buffer 3 peaks for the VU meter
   clsVUSmoothL.SmoothNew VU_SMOOTH_LEN
   clsVUSmoothR.SmoothNew VU_SMOOTH_LEN
```

```
    ' std. format is 44.1 kHz stereo (16 bit, of course)
    cboSamplerate.Text = "44100"
    chkStereo.value = 1
    ReDim intSamples(FFT_SAMPLES - 1) As Integer
    blnLoaded = True
    UpdateWAVFmtDisp
    ShowDevices
    clsDSP.samplerate = CLng(cboSamplerate.Text)
    clsDSP.Channels = chkStereo.value + 1
    Me.Show
End Sub
Private Sub ShowDevices()
    Dim i  As Long
    cboDev.Clear
    For i = 0 To clsRecorder.DeviceCount - 1
      cboDev.AddItem clsRecorder.DeviceName(i)
    Next
End Sub
Private Sub ShowLines()
    Dim i  As Long
    cboRecLine.Clear
    For i = 0 To clsRecorder.MixerLineCount - 1
      cboRecLine.AddItem clsRecorder.MixerLineName(i)
    Next
    cboRecLine.ListIndex = clsRecorder.SelectedMixerLine
End Sub
Private Sub Form_Unload( _
    Cancel As Integer _
)
    If clsRecorder.IsRecording Then
      cmdDo_Click
```

```
    End If
    Set clsRecorder = Nothing
    Set clsEncoder = Nothing
End Sub
Private Sub sldVol_Click()
    sldVol_Scroll
End Sub
Private Sub sldVol_Scroll()
    clsRecorder.MixerLineVolume = sldVol.value
    lblVolPer.Caption = Fix(sldVol.value / sldVol.max * 100) & "%"
End Sub
Private Sub tmrVis_Timer()
    Dim lngMaxL As Long
    Dim lngMaxR As Long

    If clsRecorder.IsRecording Then
        ' frequency spectrum
        clsVis.DrawAmplitudes intSamples, picAmpl
        ' amplitude curve
        clsVis.DrawFrequencies intSamples, picFreq
        ' VU meter
        If chkStereo.value = 1 Then
            lngMaxL = GetArrayMaxAbs(intSamples, 0, 2)
            lngMaxR = GetArrayMaxAbs(intSamples, 1, 2)
        Else
            lngMaxL = GetArrayMaxAbs(intSamples)
            lngMaxR = lngMaxL
        End If
        If lngMaxL = 0 Then lngMaxL = 1
        If lngMaxR = 0 Then lngMaxR = 1
        clsVUSmoothL.SmoothAdd lngMaxL
```

```
        clsVUSmoothR.SmoothAdd lngMaxR
        DrawVU picVUL, clsVUSmoothL.SmoothGetMax / 32768#
        DrawVU picVUR, clsVUSmoothR.SmoothGetMax / 32768#
        ' Info
        sbar.Panels(1).Text = "Filesize: " & FormatFileSize(FileLen(txtFile.Text))
        sbar.Panels(2).Text = "Length: " & FmtTime(lngMSEncoded)
    Else
        sbar.Panels(1).Text = "Filesize: 0 Bytes"
        sbar.Panels(2).Text = "Length: 0:00"
    End If
End Sub

Private Function FmtTime( _
    ByVal lngMS As Long _
) As String
    Dim lngMin  As Long
    Dim lngSec  As Long
    lngSec = lngMS / 1000
    lngMin = lngSec \ 60
    lngSec = lngSec Mod 60
    FmtTime = lngMin & ":" & Format(lngSec, "00")
End Function
Private Function FormatFileSize( _
    ByVal dblFileSize As Double, _
    Optional ByVal strFormatMask As String _
) As String
    Select Case dblFileSize
        Case 0 To 1023 ' Bytes
            FormatFileSize = Format(dblFileSize) & " bytes"
        Case 1024 To 1048575 ' KB
            If strFormatMask = Empty Then strFormatMask = "###0"
```

```
        FormatFileSize = Format(dblFileSize / 1024#, strFormatMask) & " KB"
      Case 1024# ^ 2 To 1073741823 ' MB
        If strFormatMask = Empty Then strFormatMask = "###0.0"
        FormatFileSize = Format(dblFileSize / (1024# ^ 2), strFormatMask) & " MB"
      Case Is > 1073741823# ' GB
        If strFormatMask = Empty Then strFormatMask = "###0.0"
        FormatFileSize = Format(dblFileSize / (1024# ^ 3), strFormatMask) & " GB"
    End Select
End Function

' mendapatkan nilai absolut terbesar dalam berbagai sampel
Private Function GetArrayMaxAbs( _
    intArray() As Integer, _
    Optional ByVal offStart As Long = 0, _
    Optional ByVal steps As Long = 1 _
) As Long
    Dim lngTemp As Long
    Dim lngMax  As Long
    Dim i       As Long
    For i = offStart To UBound(intArray) Step steps
      lngTemp = Abs(CLng(intArray(i)))
      If lngTemp > lngMax Then
        lngMax = lngTemp
      End If
    Next
    GetArrayMaxAbs = lngMax
End Function
Private Sub DrawVU( _
    ByVal picbox As PictureBox, _
    ByVal value As Single _)
    Dim lngColor    As Long
```

```
    lngColor = GetGradColor(1, value, COLOR_START, COLOR_MIDDLE, COLOR_END)
    picbox.Cls
    clsVis.DrawRect picbox.hdc, 0, 0, value * picbox.Width, picbox.Height, lngColor
End Sub
```

FORM PENGIRIMAN FILE

```
Private Type Version
  major As Integer
  minor As Integer
  revision As Integer
End Type
Private Declare Function PathFileExists Lib "shlwapi.dll" Alias "PathFileExistsA" (ByVal pszPath As String) As
Long
Private Declare Function PathIsDirectory Lib "shlwapi.dll" Alias "PathIsDirectoryA" (ByVal pszPath As String) As
Long
Dim fileData As String, fileSize As Long, lFileName As String, iWritePos As Long
Dim Sending As Boolean, Receiving As Boolean, sProgress As Long
Dim Lft As Long, Tp As Long, noSave As Boolean, noLoad As Boolean
Dim acceptConnections As Boolean, acceptTransfers As Boolean, autoListen As Boolean
Private Type POINT
  x As Long
  y As Long
End Type
Dim Sutculmam As POINT, dRagging As Boolean
Private Sub CommandXP1_Click()
On Error Resume Next
If PathIsDirectory(Text1.Text) = 0 Then
  MsgBox "sebelum kamu melakukan koneksi, tentukan save directory terlebih dahulu", vbOKOnly Or
vbCritical, "error"
  Label4.Caption = "Tentukan save directory"
  Text1.SetFocus
  Exit Sub
End If
```

```
If Right(Text1.Text, 1) <> "\" Then Text1.Text = Text1.Text & "\"


If Text2.Text = "" Then
  MsgBox "masukan local port"
  Exit Sub
Else
  If Winsock1.State = 7 Then _
    If MsgBox("kamu telah tersambung, apakah anda mau memutuskan?", vbYesNo Or vbQuestion,
"disconnect?") = vbNo Then Exit Sub
  Winsock1.Close
  Sending = False
  Receiving = False
  Winsock1.LocalPort = Text2.Text
  Winsock1.Listen
  Label4.Caption = "Sambung ke " & Winsock1.LocalIP
  Text1.Enabled = False
End If
End Sub
Private Sub CommandXP2_Click()
frm_Thesis.Show
Unload Me
End Sub
Private Sub CommandXP3_Click()
Form1.Show
Unload Me
End Sub
Private Sub Form_MouseDown(Button As Integer, Shift As Integer, x As Single, y As Single)
SutcuImam.x = x
SutcuImam.y = y
dRagging = True
End Sub
Private Sub Form_MouseMove(Button As Integer, Shift As Integer, x As Single, y As Single)
```

```
If dRagging = True Then
  Form1.Left = Form1.Left + x - Sutculmam.x
  Form1.Top = Form1.Top + y - Sutculmam.y
  DoEvents
End If
End Sub
Private Sub Form_MouseUp(Button As Integer, Shift As Integer, x As Single, y As Single)
dRagging = False
End Sub
Private Sub Command2_Click()
If PathIsDirectory(Text1.Text) = 0 Then
  MsgBox "sebelum kamu melakukan koneksi, tentukan save directory terlebih dahulu", vbOKOnly Or
vbCritical, "error"
  Label4.Caption = "Tentukan save directory"
  Text1.SetFocus
  Exit Sub
End If
If Right(Text1.Text, 1) <> "\" Then Text1.Text = Text1.Text & "\"
If Text3.Text = "" Or Text4.Text = "" Then
  MsgBox "please fill in both boxes"
  Exit Sub
Else
  If Winsock1.State = 7 Then _
    If MsgBox("kamu telah tersambung, apakah anda mau memutuskan?", vbYesNo Or vbQuestion,
"disconnect?") = vbNo Then Exit Sub
  Winsock1.Close
  Winsock1.Connect Text3.Text, Val(Text4.Text)
  Label4.Caption = "connecting..."
  Text1.Enabled = False
End If
End Sub
Private Sub Command3_Click()
```

```
On Error Resume Next
With CD1
.DialogTitle = "Pilih File Yang Akan Dikirim"
.Filter = "Rekaman Suara (*.WAV)|*.wav|"
.ShowOpen

End With
If err.Number = cdlCancel Then Exit Sub
Text5.Text = CD1.Filename
End Sub
Private Sub Command4_Click()
If Winsock1.State <> 7 Then
 MsgBox "kamu tidak terkoneksi", vbOKOnly Or vbCritical, "error"
 Exit Sub
Else
 If Sending = True Or Receiving = True Then
  MsgBox "kamu telah siap untuk pengiriman dan penerimaan", vbOKOnly Or vbCritical, "error"
  Exit Sub
 End If
  If PathFileExists(Text5.Text) = 0 Then
  MsgBox "File tidak ada", vbOKOnly Or vbCritical, "error"
  Exit Sub
 End If
  Label4.Caption = "loading file ke memory..."
 Text5.Enabled = False
 fileSize = FileLen(Text5.Text)
 fileData = String(fileSize, Chr(0))
 Open Text5.Text For Binary As #1
  Get #1, , fileData
 Close #1
 Label4.Caption = "tunggu sampai komputer lain menerima file"
```

```
  Winsock1.SendData "f=" & Mid(Text5.Text, InStrRev(Text5.Text, "\") + 1)
  Text5.Enabled = True
  End If
End Sub
Private Sub Command5_Click()
Winsock1.Close
Label4.Caption = "disconnect @ " & Now
Reset
fileSize = 0
Sending = False
Receiving = False
Text1.Enabled = True
End Sub
Private Sub Form_Activate()
Dim c As String
c = Command
If InStr(1, c, "-min") <> 0 Then Form1.WindowState = 1

End Sub
Private Sub Form_Load()
Dim c As String
c = Command
ChDir App.path
If InStr(1, c, "-noload") <> 0 Then noLoad = True
If InStr(1, c, "-nosave") <> 0 Then noSave = True
If InStr(1, c, "-acceptconnections") <> 0 Then acceptConnections = True
If InStr(1, c, "-accepttransfers") <> 0 Then acceptTransfers = True
If PathFileExists("Odesa.ini") = 1 And Not noLoad Then
  Dim s As String
  Open "Odesa.ini" For Input As #1
    DoEvents
```

```
    Line Input #1, s
    Lft = Val(s)
    Line Input #1, s
    Tp = Val(s)
    Line Input #1, s
    Text1.Text = s
    Line Input #1, s
    Text2.Text = s
    Line Input #1, s
    Text3.Text = s
    Line Input #1, s
    Text4.Text = s
  Close #1
End If

If InStr(1, c, "-listenport=") <> 0 Then Text2 = Val(Mid(c, InStr(1, c, "-listenport=") + 12))
If InStr(1, c, "-savedir=") <> 0 Then Text1 = Mid(c, InStr(1, c, "-savedir=") + 10, InStr(InStr(1, c, "-savedir=") + 10,
c, """) - (InStr(1, c, "-savedir=") + 10))
If InStr(1, c, "-listen") <> 0 Then autoListen = True
If autoListen Then Command1_Click
End Sub
Private Sub Command1_Click()
End Sub
Private Sub Form_Resize()
If Lft = -5 And Tp = -5 Then Exit Sub
Form1.Left = Lft
Form1.Top = Tp
Lft = -5
Tp = -5
End Sub
Private Sub Form_Unload(Cancel As Integer)
If Not noSave Then
```

```
  ChDir App.path
  Open "Odesa.ini" For Output As #1
    Print #1, CStr(Form1.Left)
    Print #1, CStr(Form1.Top)
       Print #1, Text1.Text
    Print #1, Text2.Text
    Print #1, Text3.Text
    Print #1, Text4.Text
  Close #1
End If

End Sub
Private Sub Winsock1_Close()
If Left(Label4.Caption, 12) <> "disconnected" Then Label4.Caption = "disconnect @ " & Now
Reset
Beep
Sending = False
Receiving = False
Text1.Enabled = True
If autoListen Then Command1_Click
End Sub
Private Sub Winsock1_Connect()
Label4.Caption = "menunggu konfirmasi dari komputer lain..."
End Sub
Private Sub Winsock1_ConnectionRequest(ByVal requestID As Long)
Winsock1.Close
Winsock1.Accept requestID
Beep
If acceptConnections Then GoTo skip0
If MsgBox("Terima koneksi dari " & Winsock1.RemoteHostIP & "?", vbYesNo Or vbQuestion, "terima?") = vbNo
Then
  Winsock1.Close
```

```
  Label4.Caption = "keluar dari koneksi"
 Else
 skip0:
  If PathIsDirectory(Text1.Text) = 0 Then
    MsgBox "sebelum kamu melakukan koneksi, tentukan save directory terlebih dahulu", vbOKOnly Or
vbCritical, "error"
    Label4.Caption = "tentukan save directory"
    Text1.SetFocus
    Winsock1.Close
    Exit Sub
  End If
  If Right(Text1.Text, 1) <> "\" Then Text1.Text = Text1.Text & "\"
    Label4.Caption = "connect @ " & Now
  Winsock1.SendData "c=" & CStr(App.major) & "." & CStr(App.minor) & " " & CStr(App.revision)
  Text1.Enabled = False
 End If
 End Sub
 Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
 On Error Resume Next
 Dim dR As String, A As VbMsgBoxResult
 Winsock1.GetData dR
 If Sending = True Then
  Winsock1.SendData fileData
  DoEvents
  Exit Sub
 End If
 If Receiving = True Then
  Put #1, iWritePos, dR
  iWritePos = iWritePos + Len(dR)
  Label4.Caption = "menerima file... " & Format((fileSize - iWritePos) / 1000000, "0.0") & " Mbs remaining... " &
CStr(Int(100 * (iWritePos / fileSize))) & "%"
  DoEvents
```

```
  If iWritePos >= fileSize Then
    Close #1
    Label4.Caption = "file diterima"
    Receiving = False
    fileSize = 0
  End If
  Exit Sub
 End If
 If fileSize = 0 Then
  If Left(dR, 2) = "s=" Then
    fileSize = Val(Mid(dR, 3))
    fileData = ""
    Receiving = True
    Winsock1.SendData "send"
     ElseIf Left(dR, 2) = "f=" Then
    Beep
    If acceptTransfers Then GoTo skip0
    A = MsgBox("accept file: " & Mid(dR, 3), vbYesNo Or vbQuestion, "accept?")
    If A = vbYes Then
skip0:
     lFileName = Text1.Text & Mid(dR, 3)
     Open lFileName For Binary As #1
     iWritePos = 1
     Label4.Caption = "recieving file... 0%"
     Winsock1.SendData "sendSize"
        Else
     Winsock1.SendData "no"
     Label4.Caption = "denied file"
   End If
     ElseIf Left(dR, 2) = "c=" Then
   Dim v As Version
```

```
    v.major = Mid(dR, 3, InStr(1, dR, ".") - 3)
    v.minor = Mid(dR, InStr(1, dR, ".") + 1, InStr(1, dR, " ") - InStr(1, dR, ".") - 1)
    v.revision = Mid(dR, InStr(1, dR, " ") + 1)
      If App.major <> v.major Or App.minor <> v.minor Or App.revision <> v.revision Then
     Winsock1.SendData "dsc_ver"
     Label4.Caption = "disconnected... versi tidak compatible"
     Exit Sub
    End If
     Label4.Caption = "connect @ " & Now
     ElseIf dR = "dsc_ver" Then
    Winsock1.Close
    Label4.Caption = "disconnected... versi tidak compatible"
    Exit Sub
  End If
  Else
  If dR = "sendSize" Then
   Winsock1.SendData "s=" & fileSize
   Label4.Caption = "negotiating transfer..."
    ElseIf dR = "no" Then
   Label4.Caption = "komputer tujuan menolak file"
   fileSize = 0
    ElseIf dR = "send" Then
   Sending = True
   sProgress = 0
   Winsock1.SendData fileData
   Label4.Caption = "sending... 0%"
  End If
End If

End Sub
Private Sub Winsock1_SendComplete()
```

```
If Sending = False Then Exit Sub

Sending = False

fileSize = 0

Label4.Caption = "transfer selesai"

End Sub

Private Sub Winsock1_SendProgress(ByVal bytesSent As Long, ByVal bytesRemaining As Long)

If Sending = False Then Exit Sub

sProgress = sProgress + bytesSent

Label4.Caption = "sending... " & Format(bytesRemaining / 1000000, "0.0") & " Mbs remaining... " &
CStr(Int(100 * (sProgress / (sProgress + bytesRemaining)))) & "%"

DoEvents

End Sub
```

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)

AUDIO BOX, INC.

Digital Audio Compression Table

CD Quality Formats	COMPRESSION	MPEG BIT RATE	SIZE IN BYTES PER MINUTE
PCM WAV Data	1 : 1	N/A	10,584,048 Stereo 5,292,024 Mono
MPEG Layer 2	3 : 1	384	2,878,222 Stereo 1,439,111 Mono
NEAR CD Quality Formats	**COMPRESSION**	**MPEG BIT RATE**	**SIZE IN BYTES PER MINUTE**
ADPCM	4 : 1	N/A	2,661,692 Stereo 1,330,846 Mono
MPEG 1 Layer 2	6 : 1	256	1,918,076 Stereo 959,038 Mono
MPEG 1 Layer 3	10 : 1	128	957,034 Stereo 478,517 Mono
Less than CD Quality Formats	**COMPRESSION**	**MPEG BIT RATE**	**SIZE IN BYTES PER MINUTE**
MPEG 1 Layer 2	8 : 1	192	1,437,376 Stereo 718,688 Mono
MPEG 1 Layer 3	12 : 1	112	837,654 Stereo 418,827 Mono

Total Music Storage Time	1 GB HARD DRIVE	2 GB HARD DRIVE	4 GB HARD DRIVE	6GB HARD DRIVE	10 GB HARD DRIVE
PCM WAV	1:41:26	3:22:53	6:45:47	10:08:41	16:54:30
MPEG 2, 3 : 1	6:13:03	12:26:07	24:52:13	37:18:20	62:10:34
ADPCM	6:43:24	13:26:07	26:53:37	40:20:26	67:14:03
MPEG 2, 6 : 1	9:19:48	18:39:36	37:19:12	55:58:48	93:18:01
MPEG 3, 10 : 1	18:41:56	37:23:53	74:47:47	112:11:41	186:59:28
MPEG 2, 8 : 1	12:27:01	24:54:02	49:48:04	74:42:05	124:30:26
MPEG 3, 12 : 1	21:21:50	42:43:40	85:27:22	128:11:04	213:38:26

Number of 3 minute songs	1 GB HARD DRIVE	2 GB HARD DRIVE	4 GB HARD DRIVE	6GB HARD DRIVE	10 GB HARD DRIVE
PCM WAV	33	67	135	202	338
MPEG 2, 3 : 1	124	248	497	746	1243
ADPCM	134	268	537	806	1344
MPEG 2, 6 : 1	186	373	746	1119	1866
MPEG 3, 10 : 1	373	747	1495	2243	3739
MPEG 2, 8 : 1	249	498	996	1494	2490
MPEG 3, 12 : 1	427	854	1709	2563	4272