APLIKASI ENKRIPSI PENGIRIMAN FILE SUARA MENGGUNAKAN ALGORITMA BLOWFISH

Novrido Charibaldi[1], ,Fitrianty[2], Bambang Yuwono[3]
[1,2,3]) Jurusan Teknik Informatika UPN "Veteran" Yogyakarta
Jl.Babarsari no.2 Tambakbayan 55281 Yogyakarta Telp (0274)-485323
email : novrido@gmail.com, vitrimail@yahoo.com, bambangy@gmail.com

Abstrak
Untuk menjaga keamanan data ataupun informasi yang tersimpan dalam bentuk file, salah satunya dengan menggunakan metode kriptografi untuk mengenkripsi data file tersebut sehingga tidak dapat dilihat atau dibaca oleh yang tidak berhak. Pada skripsi ini dirancang suatu aplikasi yang dapat menunjukkan solusi keamanan pesan suara dengan menggunakan algoritma enkripsi. Pesan suara dilakukan dengan menggunakan algoritma kunci simetri dengan cipher blok Blowfish yang dibuat oleh Bruce Scheiner. Keamanan diukur dengan melakukan pengecekan kebenaran proses enkripsi dan dekripsi.
Kata kunci : cipher, AlgoritmaBlowfish, Enkripsi

1. PENDAHULUAN

Saat ini komunikasi suara telah menjadi bagian dari kehidupan sehari-hari. Dimulai dengan komunikasi suara melalui telepon berbasis analog, kemudian telepon selular yang berbasis digital. Teknologi terbaru dalam komunikasi suara merupakan komunikasi melalui internet atau jaringan yang biasa disebut *voice over internet protocol* (VOIP).

Berbagai macam jenis komunikasi suara tersebut belum tentu aman untuk digunakan, karena belum tentu ada suatu standar keamanan yang diterapkan untuk masing-masing fasilitas komunikasi suara tersebut. Komunikasi suara menjadi sangat rentan terhadap gangguan pihak ketiga karena biasanya jaringan internet dapat dengan mudah diakses oleh pihak umum dan jaringan telepon juga tidak dapat dikatakan sepenuhnya aman dari penyadapan.

Salah satu solusi untuk mengamankan data suara tersebut adalah dengan enkripsi suara. Pada enkripsi suara, proses dilakukan dengan berbasis data digital. Enkripsi dilakukan pada data suara yang akan dikirimkan, sehingga pihak lain yang tidak berhak tidak dapat memahami data suara yang dikirimkan tersebut meskipun data suara berhasil diakses. Biasanya enkripsi dilakukan oleh suatu alat atau aplikasi pengenkripsi.

Salah satu algoritma kriptografi yang memiliki tingkat keamanan tinggi adalah algoritma *blowfish*. *Blowfish* atau yang disebut juga "*OpenPGP.Cipher.4*" adalah algoritma kunci simetrik cipher blok yang dirancang pada tahun 1993 oleh Bruce Schneider untuk menggantikan DES (*Data Encryption Standard*). Algoritma *blowfish* dibuat untuk digunakan pada komputer yang mempunyai microposesor besar (32-bit keatas dengan cache data yang besar). Pada saat itu banyak sekali rancangan algoritma yang ditawarkan, namun hampir semua terhalang oleh paten atau kerahasiaan pemerintah Amerika. Schneier menyatakan bahwa *blowfish* bebas paten dan akan berada pada domain publik. Dengan pernyataan Schneier tersebut *blowfish* telah mendapatkan tempat di dunia kriptografi, khususnya bagi masyarakat yang membutuhkan algoritma kriptografi yang cepat, kuat, dan tidak terhalang oleh lisensi. Untuk itu, sistem yang akan dibuat pada aplikasi enkripsi ini akan menggunakan teknik pengamanan data menggunakan algoritma *blowfish*.

Tujuan penelitian ini untuk menghasilkan sebuah aplikasi enkripsi pengiriman pesan suara menggunakan algoritma *blowfish*. Adapun manfaat dari penelitian ini diharapkan mampu menjaga kerahasiaan data pesan atau suara bagi para pengguna atau *user,* sehingga hanya orang yang berwenang saja yang dapat menerima pesan suara tersebut.

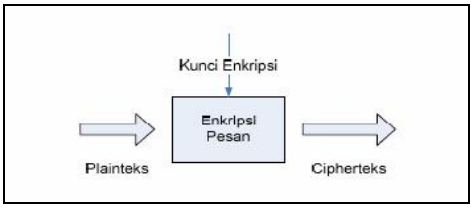
2. TINJAUAN PUSTAKA

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 :

a. Enkripsi

Enkripsi merupakan proses untuk mengubah plainteks menjadi *ciphertext* yang tidak dapat 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 1

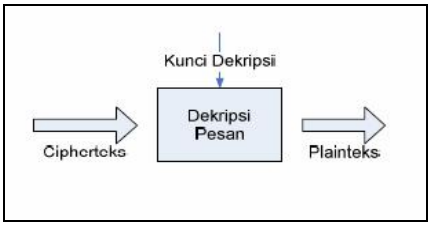


Gambar 1. Proses Enkripsi dengan Kunci

b. Dekripsi

Dekripsi merupakan proses untuk mengubah c*iphertext* 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.

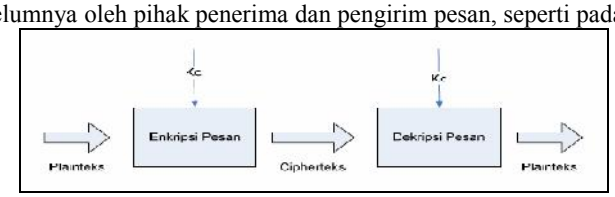


Gambar 2. Proses Dekripsi dengan Kunci

Kunci yang digunakan pada proses dekripsi dapat berbeda dengan kunci yang digunakan pada proses enkripsi, disebut juga kriptografi kunci publik. Sebaliknya jika kunci yang digunakan sama, disebut juga kriptografi kunci simetri.

2.2 Kriptografi Kunci Simetri

Kriptografi kunci simetri menggunakan kunci yang sama untuk proses enkripsi dan dekripsi. Kunci tersebut harus ditentukan sebelumnya oleh pihak penerima dan pengirim pesan, seperti pada Gambar 3



Gambar 3. Skema Kriptografi Kunci Simetri

Kelemahan dari kriptografi kunci simetri adalah kesulitan dalam pendistribusian kunci, karena pengirim dan penerima pesan harus mengetahui kunci yang sama maka kunci harus dikiriman atau diberitahukan pada pihak lainnya. Masalah lainnya adalah pada banyaknya kunci yang harus digunakan, untuk setiap pasangan pengirim dan penerima pesan harus ada paling tidak satu buah kunci yang dapat digunakan untuk melakukan enkripsi antara mereka.

2.3 Tipe dan Mode Algoritma Kunci Simetri

Algoritma kunci simetri memiliki dua tipe yang umum digunakan, yaitu:

a. *Cipher* blok, yang beroperasi pada blok-blok plainteks. Ukuran blok dapat bermacam-macam tergantung pada algoritma enkripsi yang digunakan, biasanya berkisar antara 64 atau 128 bit. Hasil enkripsi dari suatu plainteks yang sama, akan menghasilkan *ciphertex t*yang sama.
b. *Cipher* aliran, yang beroperasi pada aliran plainteks satu bit atau byte setiap waktu. Hasil enkripsi dari suatu plainteks yang sama belum tentu menghasilkan *ciphertext* yang sama.

2.4 Algoritma *Blowfish*

Blowfish merupakan blok cipher 64-bit dengan panjang kunci variabel. Algoritma ini terdiri dari dua bagian: *key expansion* atau perluasan kunci dan enkripsi data.

a. *Key-Expansion* berfungsi merubah kunci (Minimum 32-bit, Maksimum 448-bit) menjadi beberapa array subkunci (*subkey*) dengan total 4168 byte.
b. Enkripsi data terdiri dari iterasi fungsi sederhana (*Feistel Network*) sebanyak 16 kali putaran. Setiap putaran terdiri dari permutasi kunci-dependent dan substitusi kunci- dan data *dependent*. Semua operasi adalah penambahan (*addition*) dan XOR pada variabel 32-bit. Operasi tambahan lainnya hanyalah empat penelusuran tabel (*table lookup*) array berindeks untuk setiap putaran.

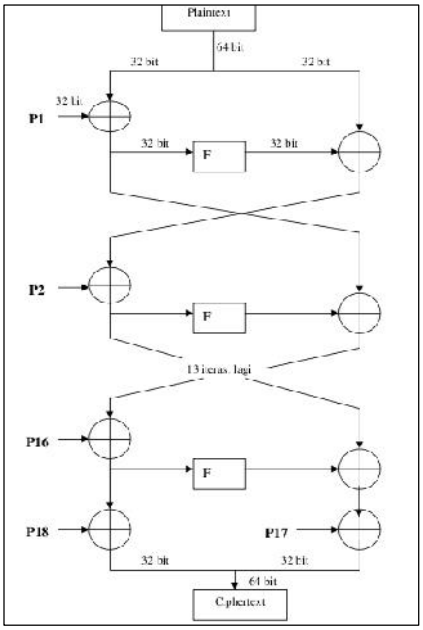
Untuk alur algoritma enkripsi dengan metoda *Blowfish* dijelaskan sebagai berikut :

1. Bentuk inisial array P sebanyak 18 buah (P1,P2, ……..P1 masing-msing bernilai 32-bit. Array P terdiri dari delapan belas kunci 32-bit subkunci : P1,P2,……..,P18
2. Bentuk S-box sebanyak 4 buah masing-masing bernilai 32-bit yang memiliki masukan 256.

Empat 32-bit S-box masing-masing mempunyai 256 entri :
S1,0,S1,1,……..,S1,255
S2,0,S2,1,……..,S2,255
S3,0,S3,1,……..,S3,255
S4,0,S4,1,……..,S4,255

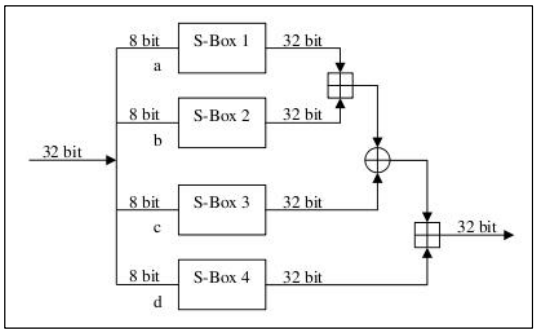
3. Plainteks yang akan dienkripsi diasumsikan sebagai masukan, Plainteks tersebut diambil sebanyak 64-bit, dan apabila kurang dari 64-bit maka kita tambahkan bitnya, supaya dalam operasi nanti sesuai dengan datanya.
4. Hasil pengambilan tadi dibagi 2, 32-bit pertama disebut XL, 32-bit yang kedua disebut XR.
5. Selanjutnya lakukan operasi XL = XL xor Pi dan XR = F(XL) xor XR
6. Hasil dari operrasi diatas ditukar XL menjadi XR dan XR menjadi XL.
7. Lakukan sebanyak 16 kali, perulangan yang ke-16 lakukan lagi proses penukaran XL dan XR.
8. Pada proses ke-17 lakukan operasi untuk XR = XR xor P17 dan XL = XL xor P18.
9. Proses terakhir satukan kembali XL dan XR sehingga menjadi 64-bit kembali.

Blowfish menggunakan jaringan Feistel yang terdiri dari 16 buah putaran. Skema jaringan Feistel dapat dilihat di gambar 4.



Gambar 4. Jaringan Feistel untuk Algoritma *Blowfish*

Algoritma *Blowfish* memiliki keunikan dalam hal proses dekripsi, yaitu proses dekripsi dilakukan dengan urutan yang sama persis dengan proses enkripsi, hanya saja pada proses dekripsi P1, P2, …, P18 digunakan dalam urutan yang terbalik. Dalam algoritma *Blowfish* juga terdapat fungsi f. Berikut ini gambar mengenai fungsi f tersebut.



Gambar 5. Fungsi f dalam algoritma *Blowfish*

Sebelumnya dijelaskan bahwa Array P terdiri dari delapan belas subkunci. Subkunci dihitung menggunakan algoritma *Blowfish*, metodenya adalah sebagai berikut :

1. Pertama-tama inilialisasi P-array dan kemudian empat S-box secara berurutan dengan string yang tetap. String ini terdiri atas digit hexadesimal dari Pi.
2. XOR P1 dengan 32-bit pertama kunci, XOR P2 dengan 32-bit kedua dari kunci dan seterusnya untuk setiap bit dari kunci (sampai P18).Ulangi terhadap bit kunci sampai seluruh P-array di XOR dengan bit kunci.
3. Enkrip semua string nol dengan algoritma *Blowfish* dengan menggunakan subkunci seperti dijelaskan pada langkah (1) dan (2).
4. Ganti P1 dan P2 dengan keluaran dari langkah (3).
5. Enkrip keluaran dari langkah (3) dengan algoritma *Blowfish* dengan subkunci yang sudah dimodifikasi.
6. Ganti P3 dan P4 dengan keluaran dari langkah (5).
7. Lanjutkan proses tersebut, ganti seluruh elemen dari P-array, kemudian seluruh keempat S-box berurutan, dengan keluaran yang berubah secara kontiyu dari algoritma *Blowfish*.

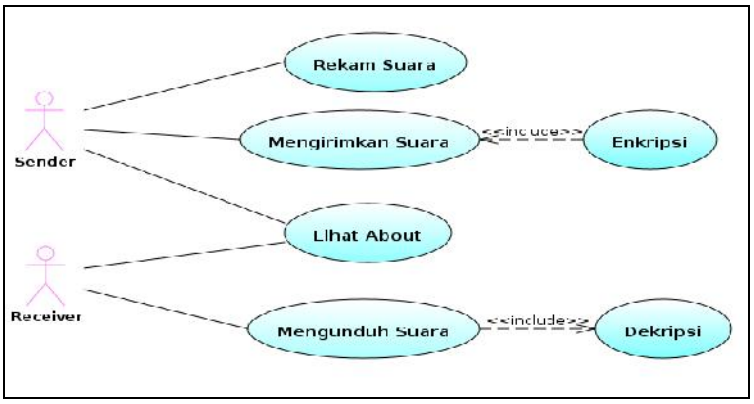
Secara keseluruhan terdapat 521 iterasi atau putaran yang dibutuhkan untuk membangkitkan seluruh upa-kunci yang dibutuhkan. Aplikasi kemudian dapat menyimpan upa-kunci yang telah dihasilkan. Proses pembangkitan kunci ini tidak perlu selalu dilakukan setiap saat.

3. METODE PENELITIAN

Metode pengembangan perangkat lunak yang digunakan adalah metode pengembangan sistem berorientasi objek *Guidelines for Rapid APPlication Diagram* (GRAPPLE) yang terdiri dari Pengumpulan Kebutuhan (*Requirement Gathering*), Analisis *(Analysis)* Perancangan *(Design),* Pengembangan (*Development*), dan Penyebaran (*Deployment*).

3.1 Diagram Use Case

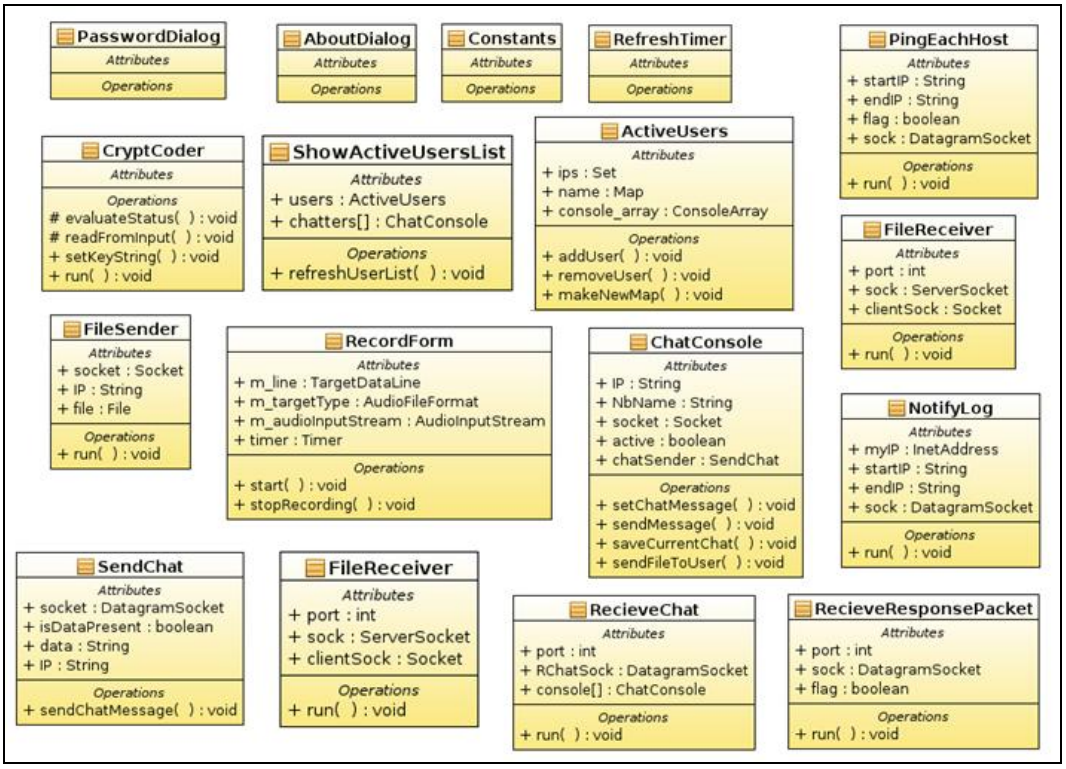
Dalam sistem ini pengguna dapat melakukan empat hal yaitu rekam suara, mengirim suara, mengunduh suara dan lihat about. Diagram use case dapat dilihat pada gambar 6.



Gambar 6. Diagram *Use Case*

3.2 Diagram Class

Identifikasi kelas berdasarkan analisis kelas memiliki kelas-kelas utama seperti pada diagram *class* berikut:



Gambar 7. Diagram *Class*

4. HASIL DAN PEMBAHASAN

4.1 Implementasi Form Utama

Form *Cryptic Voice* adalah tampilan yang muncul pertama saat program dijalankan. Terdapat dua layanan menu yaitu dan Menu Tools dimana didalam Menu Tools terdapat *MenuItem Record Voice* dan *MenuItem Encrypt/Decrypt* dan *Menu Help* yang didalamnya terdapat *MenuItem* About. *Button "Refresh Users List"* digunakan untuk me-*refresh* list IP yang sedang aktif di jaringan lokal dan menampilkannya di JList.

Gambar 8. Tampilan *Form Cryptic Voice*

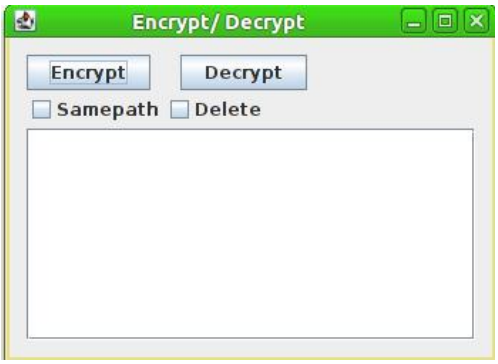
4.2 Implementasi Form Recording

Form recording digunakan ketika user akan merekam suara. Proses rekam akan berjalan ketika menekan *button* Start dan akan berhenti ketika menekan *tombol* Stop. File yang dihasilkan berekstensi (*.wav).

Gambar 9. Tampilan *Form Recording*

4.3 Implementasi Form Encrypt dan Decrypt

Form Encrypt/Decrypt tampil ketika user akan melakukan proses enkripsi atau dekripsi.. Hasil file yang terenkripsi berekstensi (*.enc).



Gambar 10. Tampilan *Form Encrypt/ Decrypt*

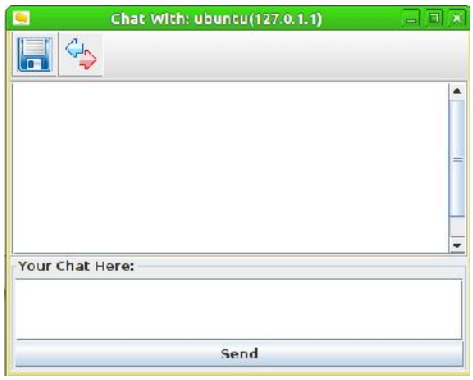
4.4 Implementasi Form Password

Form Enter Key digunakan untuk memasukkan kata kunci ketika melakukan proses enkripsi dan dekripsi pada file.

Gambar 11. Tampilan *Form Enter Key*

4.5 Implementasi Form Chat, Save Log dan Kirim File

Form ini akan muncul ketika user menekan salah satu list yang sedang aktif di form utama. User dapat melakukan komunikasi chat dan kirim file pada user yang dituju. Ukuran file yang dikirimkan memiliki batas maksimal 2,5 MB.



Gambar 12. Tampilan *Form Chat*

5. KESIMPULAN

Kesimpulan yang dapat diambil setelah mengerjakan penelitian ini adalah :

1. Telah berhasil dibangun sebuah aplikasi enkripsi pengiriman pesan suara menggunakan algoritma blowfish.
2. Kualitas suara setelah mengalami enkripsi dan dekripsi tetap memiliki kualitas yang baik dan ukuran file suara tidak berubah.
3. Delay yang dihasilkan saat proses enkripsi dekripsi berlangsung tergantung dari ukuran file yang akan di enkripsi dan dekripsi, sedangkan delay yang dihasilkan ketika melakukan proses pengiriman juga tidak memakan waktu yang lama.
4. Pada aplikasi ini, tidak menutup kemungkinan untuk mengenkripsi atau mendekripsi file lain seperti (*.doc), (*.jpg), (*.pdf) dan file lainnya.

DAFTAR PUSTAKA

Munir, Rinaldi, 2006, *KRIPTOGRAFI*, Informatika, Bandung.

Pressman, R.S, 1997, *Rekayasa Perangkat Lunak; Pendekatan Praktisi,Edisi Pertama*, Andi, Yogyakarta,

Schneier, Bruce, 1996, *APPLIED CRYPTOGRAPHY protocols, Algorithm, and Source Code in C*, Second Edition, John Wiley and Sons, United State of America.

Randyu, Aditya, 2006, *Studi dan Perbandingan Algoritma Blowfish dan Twofish*, diakses tanggal 20 Januari 2011.

Kadir, Abdul, 2003, *Pengenalan Sistem Informasi*, Andi, Yogyakarta.

Hartono, Jogiyanto, 1999, *Pengenalan Komputer Dasar Ilmu Komputer, Pemrograman, Sistem Informasi dan Intelegensi Buatan*, Andi, Yogyakarta.

Wahana, 2003, *Memahami Model Enkripsi dan Security Data*, Andy Offset, Yogyakarta.