DINAMIKA Jurnal Ilmiah Teknik Mesin

IMPLEMENTASI ALGORITMA *BLOWFISH* UNTUK KEAMANAN DATA SUARA

Sutardi

Staf Pengajar Program Studi Teknik Informatika Fakultas Teknik Universitas Halu Oleo
Kampus Hijau Bumi Tridarma Andounohu Kendari 93232

E-mail: sutardi@gmail.com

Abstrak

Teknologi terbaru dalam komunikasi suara merupakan bagian dari kehidupan sehari-hari, Pada beberapa kasus pencurian data, para pelaku pencurian dapat dengan mudah mengambil data dan menguggahnya ke berbagai media secara luas tanpa sepengetahuan dari pemilik data, sehingga mengakibatkan kerugian pada pemiliknya. Salah satu solusi untuk mengamankan data suara tersebut adalah dengan melakukan *voice scrambling*, yaitu perubahan pada sinyal telekomunikasi untuk membuatnya menjadi tidak dapat diketahui oleh siapapun kecuali oleh pihak yang memiliki alat penerima khusus. Akan tetapi teknik ini memiliki tingkat keamanan yang sangat rendah karena alat penerima sinyal tersebut bisa dimiliki oleh siapa saja. Solusi lain yang memiliki tingkat keamanan jauh lebih tinggi adalah enkripsi suara. Enkripsi dilakukan pada data suara sebelum data suara dikirimkan, sehingga pihak lain yang tidak berhak, tidak dapat memahami data suara yang dikirimkan tersebut meskipun data suara berhasil diakses. Salah satu metode yang digunakan untuk enkripsi dan dekripsi data suara adalah metode *Blowfish*. Tujuan penelitian ini adalah untuk mengetahui penerapan kunci algoritma *Blowfish* untuk keamanan suara data. Metode yang digunakan dalam penelitian ini adalah pengujian pada pembuatan suatu aplikasi dengan enkripsi dan dekripsi data suara. Hasil penelitian ini menunjukkan bahwa kerahasiaan data menggunakan algoritma Blowfish bergantung pada panjang kunci. Penggunaan kunci lebih fleksibel antara 1 sampai 16 karakter, dimana enkripsi dan dekripsi data dapat berjalan dengan baik pada berbagai operasi sistem. Selain itu, untuk kecepatan enkripsi dan dekripsi tidak membutuhkan waktu yang lama, sesuai dengan besar ukuran data yang dienkrispi dan didekripsikan.

Kata Kunci *: Blowfish, enkripsi, dekripsi, komunikasi, data, suara*

Abstract

The implementation of the Blowfish algorithm for the security of a voice data. The latest technology in the voice communication is a part of everyday life. In some cases of the data theft, the culprit can easily retrieve and upload the data to various media widely without being known the knowledge by the owner, resulting in a loss. One of the solutions to secure the voice data is to make a voice scrambling, ie changing in the telecommunications signal to make it not be known by anyone except by those who have a special receiver. However, this technique has a very low level of security because the signal receiver can be owned by anyone. Another solution which has a much higher level of security is a voice encryption. The encryption is performed on voice data before the voice data is sent, thus the other party who is not entitled, can not understand the voice data transmitted even though the voice data successfully accessed. One of the methods that is used for the encryption and decryption of the voice data is the Blowfish. The purpose of this study is to determine the application of the security key on the Blowfish algorithm for the voice data. The method used in this study is a test on the making of an application by encryption and decryption of voice data. The results show that the confidentiality of the data using the Blowfish algorithm depends on the key length. It is more flexible to use the keys between 1 and 16 characters, at which the encryption and decryption of data may operate properly at the variety of system operation. Indeed, the speed of encryption and decryption requires no long time, according to the large size of the encrypted or decrypted data.

Keywords: *Blowfish, encryption, decryption, communication, data, voice*

1. Pendahuluan

Saat ini komunikasi suara telah menjadi bagian dari kehidupan sehari-hari. Salah satu 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 (Hidayanto, 2007)

Salah satu contoh kasusnya adalah pencurian data suara berupa rekaman lagu dari seorang penyanyi internasional yang terjadi pada tahun 2014. Data suara ini belum dipublikasikan atau dirilis secara resmi, lalu kemudian berhasil dicuri dan disebar luaskan dengan bebas, sehingga mengakibatkan kerugian pada pemiliknya (Hidayanto, 2007).

Salah satu solusi untuk mengamankan data suara tersebut adalah dengan melakukan *voice scrambling*, yaitu perubahan pada sinyal telekomunikasi untuk membuatnya menjadi tidak dapat diketahui oleh siapapun kecuali pihak yang memiliki alat penerima khusus. Akan tetapi teknik ini memiliki tingkat keamanan yang sangat rendah karena alat penerima sinyal tersebut bisa dimiliki oleh siapa saja (Sadikin, 2012).

Solusi lain yang memiliki tingkat keamanan jauh lebih tinggi adalah enkripsi suara. Enkripsi dilakukan pada data suara sebelum data suara dikirimkan, sehingga pihak lain yang tidak berhak tidak dapat memahami data suara yang dikirimkan tersebut meskipun data suara berhasil diakses. Salah satu metode yang digunakan untuk enkripsi dan dekripsi adalah metode *Blowfish* (Kelsey, 1998)

Tujuan penelitian ini adalah untuk mengetahui penerapan kunci algoritma *Blowfish* untuk keamanan suara data.

2 Tinjauan Pustaka

Blowfish

Blowfish alias "*OpenPGP.Cipher.4*" merupakan enkripsi yang termasuk dalam golongan *Symmetric Cryptosistem*, yang dibuat untuk digunakan pada komputer yang mempunyai microposesor besar (32-bit keatas dengan *cache* data yang besar). Sampai saat ini belum ada *attack* yang dapat memecahkan *Blowfish*. *Blowfish* merupakan *cipher block*, yang berarti selama proses enkripsi dan dekripsi, *Blowfish* akan membagi pesan menjadi blok-blok dengan ukuran yang sama panjang. Panjang blok untuk algoritma *Blowfish* adalah 64-bit. Pesan yang bukan merupakan kelipatan delapan *byte* akan ditambahkan bit-bit tambahan (*padding*) sehingga ukuran untuk tiap blok sama (Sutanto, 2010).

Blowfish adalah algoritma yang tidak dipatenkan dan *license free*, dan tersedia secara gratis untuk berbagai macam kegunaan. *Blowfish* dirancang dan diharapkan mempunyai kriteria perancangan yang diinginkan, misalnaya yang pertama cepat, *Blowfish* melakukan enkripsi data pada *microprocessor* 32-bit dengan *rate* 26 *clock cycles* per *byte* (Trisnawati, 2008).

Kriptografi

Kriptografi (*cryptography*) berasal dari Bahasa Yunani: "*cryptós*" artinya "*secret*" (rahasia), sedangkan "*gráphein*" artinya "*writing*" (tulisan). Jadi, kriptografi berarti "*secret writing*" (tulisan rahasia) (Scheiner, 1996).

Definisi yang dipakai di dalam buku -buku yang lama (sebelum tahun 1980an) menyatakan bahwa kriptografi adalah ilmu dan seni untuk menjaga kerahasian pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Definisi ini mungkin cocok pada masa lalu di mana kriptografi digunakan untuk keamanan komunikasi penting seperti komunikasi di kalangan militer, diplomat, dan mata-mata. Namun saat ini kriptografi lebih dari sekadar *privacy*, tetapi juga untuk tujuan *data integrity*, *authentication*, dan *non-repudiation* (Kromodimoeljo, 2009).

Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan. (*Cryptography is the art and science of keeping messages secure*) (Scheiner, 1996). Kata "seni" di dalam definisi di atas berasal dari fakta sejarah bahwa pada masa-masa awal sejarah kriptografi, setiap orang mungkin mempunyai cara yang unik untuk merahasiakan pesan. Cara-cara unik tersebut mungkin berbeda-beda pada setiap pelaku kriptografi sehingga setiap cara menulis pesan rahasia pesan mempunyai nilai estetika tersendiri

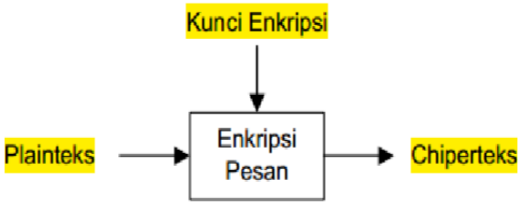
sehingga kriptografi berkembang menjadi sebuah seni merahasiakan pesan (kata "*graphy*" di dalam "*cryptography*" itu sendiri sudah menyiratkan sebuah seni).

Pada perkembangan selanjutnya, kriptografi berkembang menjadi sebuah disiplin ilmu sendiri karena teknik-teknik kriptografi dapat diformulasikan secara matematik sehingga menjadi sebuah metode yang formal (Munir, 2004).

Sistem Kriptografi

Untuk menjamin keamanan pertukaran data, berbagai proses dilakukan terhadap data. Salah satu proses adalah 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 ekskripsi dan dekripsi (Scheiner, 1996).

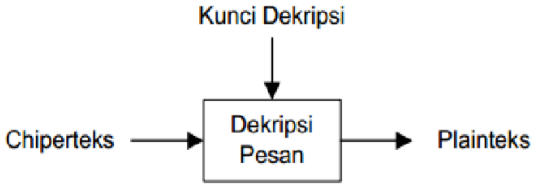
Enkripsi merupakan proses untuk mengubah *plainteks* menjadi *cipherteks* yang tidak bisa dimengerti. Proses enkripsi biasanya dilakukan sebelum pesan dikirimkan. Untuk meningkatkan keamanan enkripsi pesan, pada proses enkripsi ditambahkan kunci yang juga diperlukan untuk proses dekripsi, seperti pada Gambar 1. (Scheiner, 1996).



Gambar 1 Proses enkripsi dengan kunci

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.

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 (Ariyus, 2008).

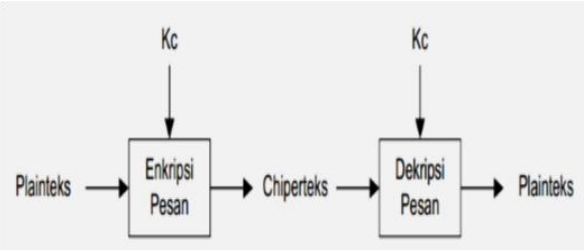


Gambar 2 Proses dekripsi dengan kunci

Algoritma Kunci Simetri

Algoritma kunci simetri adalah algoritma kriptografi yang memiliki kunci yang sama untuk proses enkripsi dan dekripsinya. Kunci tersebut merupakan satu-satunya jalan untuk proses enkripsi, sehingga kerahasiaan kunci menjadi nomor satu. Mengirimkan kunci tersebut ke suatu pihak tanpa diketahui pihak yang lain merupakan masalah awal dari algoritma kunci simetrik (Sutatnto, 2010).

Algoritma kunci simetri terbagi menjadi dua buah bergantung pada datanya, yaitu cipher aliran (*stream cipher*) dan cipher blok (*block cipher*). Cipher aliran memproses satu bit pesan sekali dalam satu waktu, sedangkan cipher blok memproses sekumpulan bit sekaligus sebagai satu unit. Ukuran blok yang umum dipakai adalah 64 bit. Dari segi kecepatan komputasi, algoritma kunci simetri lebih cepat daripada algoritma asimetrik. Kelemahan utama dari opsi ini adalah dalam mendistribusikan kunci ke pihak-pihak yang berkepentingan. Namun jika dipakai dalam suatu lingkungan yang tidak membutuhkan pendistribusian kunci (seperti penggunaan pribadi), maka algoritma ini merupakan algoritma yang terbaik (Sadikin, 2012).



Gambar 3 Skema Algoritma Kunci Simetri

Enkripsi Algoritma *Blowfish*

Blowfish menggunakan sub-kunci berukuran besar. Kunci-kunci tersebut harus dikomputasikan pada saat awal, sebelum pengkomputasian enkripsi dan dekripsi data (Sutanto, 2010).

Langkah awal enkripsi algoritma *Blowfish* adalah menentukan kotak permutasi (P-box). Kotak ini terdiri dari 18 buah 32 bit subkunci, yaitu P1 sampai P18 (Sutanto, 2010).

P-box ini telah ditetapkan sejak awal dan 4 buah P-box awal adalah sebagai berikut:

Tabel 1 P-box awal

P1	*0x243f6a88*
P2	*0x85a308d3*
P3	*0x13198a2e*
P4	*0x03707344*

Langkah selanjutnya adalah meng-xorkan P1 dengan 32 bit awal kunci, meg-xorkan P2 dengan 32 bit berikutnya dari kunci, dan teruskan hingga seluruh panjang kunci telah ter-xorkan (kemungkinan sampai P14, 14x32 = 448, panjang maksimal kunci).

Terdapat 64 bit dengan isi kosong, bit-bit tersebut dimasukkan ke langkah dua. Setelah itu menggantikan P1 dan P2 dengan keluaran dari langkah ke-tiga. Selanjutnya, meng-enkripsikan keluaran langkah ketiga dengan langkah kedua kembali, namun kali ini dengan subkunci yang berbeda. Selanjutnya adalah menggantikan P3 dan P4 dengan keluaran dari langkah lima dan tujuh dan melakukanya seterusnya sehingga seluruh Pbox teracak sempurna

Secara total keseluruhan, terdapat 521 iterasi untuk menghasilkan subkunci-subkunci yang dibutuhkan. Aplikasi hendaknya menyimpannya daripada menghasilkan ulang subkunci-subkunci tersebut.

Kunci- kunci yang digunakan antara lain terdiri dari, 18 buah 32-bit *subkey* yang tergabung dalam P-array. Selain itu, ada pula empat 32-bit S-box yang masing-masingnya memiliki 256 entri sebagai berikut

Tabel 2 S-box 32-bit yang masing-masingnya memiliki 256 entri

S1,0,S1,1,..., S1,255;
S2,0, S2,1,..,, S2,255;
S3,0, S3,1,..., S3,255;
S4,0, S4,1,..,, S4,255.

Pada jaringan feistel, *Blowfish* memiliki 16 iterasi, dengan masukannya adalah 64-bit elemen data, X. Untuk melakukan proses enkripsi langkah pertama adalah membagi X menjadi dua bagian yang masing-masing terdiri dari 32-bit: XL, XR (For i = 1 to 16 : = XL XOR Pi XR = F(XL) XOR XR Tukar XL dan XR). Selanjutnya adalah iterasi ke-enam belas, tukar XL dan XR lagi untuk melakukan *undo* pertukaran terakhir. Selanjutnya adalah melakukan XR = XR XOR P17 = XL XOR P18. Tahap akhir adalah menggabungkan kembali XL dan XR untuk mendapatkan cipherteks.

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

F(XL) = ((S1,a + S2,b mod 232) XOR S3,c) + S4,d mod 232.

Dekripsi Algoritma *Blowfish*

Dekripsi sama persis dengan enkripsi, naum P_1 sampai P_{18} yang digunakan pada urutan yang terbalik. Dekripsi untuk *Blowfish* bersifat maju kedepan yang mengakibatkan dekripsi bekerja dalam arah algoritma yang sama seperti halnya dengan enkripsi, namun sebagai masukannya adalah *chipertext* (Sutanto, 2010).

Langkah awal subkunci dihitung menggunakan algoritma*Blowfish*, metodenya adalah inisialisasi P-array dan kemudian empat S-box secara berurutan dengan string yang tetap.String ini terdiri digit *hexadesimal* dari pi.

Selanjutnya meng-XOR P_1 dengan 32 bit pertama kunci, meng-XOR P_2 dengan 32 bit kedua dari kunci dan seterusnya untuk setiap bit dari kunci (sampai P_{18}) dan mengulangi terhadap bit kunci sampai seluruh Parray di XOR dengan bit kunci. Setelah itu, meng-enkripsi semua string nol dengan algoritma *Blowfish* dengan

menggunakan subkunci. Selanjutnya adalah mengganti P1 dan P2 dengan keluaran dari langkah tiga dan meng-enkripsi keluaran dari langkah tiga dengan algoritma *Blowfish* dengan subkunci yang sudah dimodifikasi. Selanjutnya adalah mengganti P_3 dan P_4 dengan keluaran dari langkah lima. Selanjutnya adalah mengganti seluruh elemen dari P-array, dan kemudian seluruh keempat S-box berurutan, dengan keluaran yang berubah secara kontinyu dari algoritma *Blowfish* Total yang diperlukan 521 iterasi untuk menghasilkan semua subkunci yang dibutuhkan. Selanjutnya menyimpan subkunci ini. Langkah-langkah proses penurunan berulang kali tidak dibutuhkan, kecuali kunci yang digunakan berubah.

3 Metodologi Penelitian

Metode yang digunakan dalam penelitian ini adalah pengujian pada pembuatan suatu aplikasi dengan ekskipsi dan dekripsi data suara.

Metode Algoritma

Blowfish dapat dilakukan dengan cara membalikkan 18 sub ke yang ada. Langkah awal yang dilakukan adalah masalah ini nampak tidak dapat dipercaya, karena terdapat dua XOR operasi yang mengikuti pemakaian f--fungsi yang sebelumnya dan hanya satu.

Walupun jika kita memodifikasi algoritma tersebut sehingga pemakaian subkey 2 sampai 17 menempatkan sebelum output f--fungsi yang di-XOR-kan ke sebelah kanan blok dan dilakukan ke data yang sama sebelum XOR itu, meskipun itu berarti ia sekarang berada di sebelah kanan blok. Hal ini karena XORc subkey tersebut telah dipindahkan sebelum swap (tukar) kedua belah blok tersebut (tukar separuh blok kiri dan separuh blok kanan).

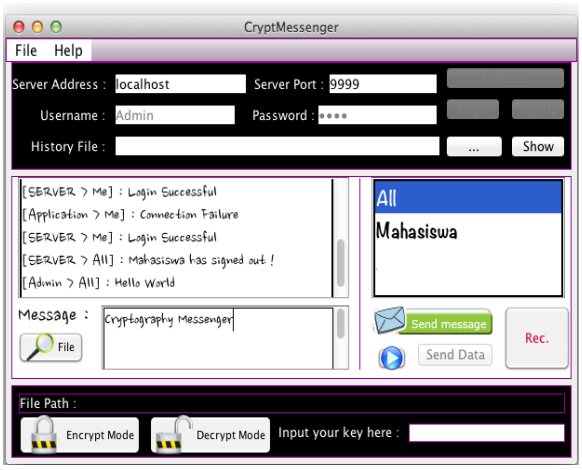
Metode ini tidak merubaah suatu apapun karena informasi yang sama di-XOR-kan ke separuh blok kiri antara setiap waktu, informasi ini digunakan sebagai input f-fungsi. Kenyataannya, hal ini memiliki kebalikan yang pasti dari barisan dekripsi (Sutanto, 2010).

Implementasi dan Pengujian Sistem.

Database server merupakan nama database dari aplikasi ini yang berfungsi sebagai tempat penyimpanan informasi pengguna (*user*) agar terdaftar dalam *server* (Sutatnto, 2010).

App *Server* merupakan lalulintas penghubung antara pengirim dan penerima pesan yang sebelumnya telah terdaftar dalam database server yang disimpan kedalama file Data.xml. Pengguna yang sudah terdaftar dapat masuk kedalam sistem.

App *Client* merupakan aplikasi yang bertugas untuk melakukan proses enkripsi dan ekripsi terhadap pesan atau *PlainData* yang akan dikirimkan. Proses enkripsi dimulai dengan merekam suara atau sinyal analog dari pengguna menggunakan *mic* yang kemudian sistem dari aplikasi ini akan merubah sinyal analog tersebut kedalam data digital (*digitalisasi*) yang kemudian akan disimpan dalam bentuk *Plain Data* dengan ekstensi file berupa .wav (*PlainData.wav*). Setelah proses digitalisasi suara telah berhasil dan mengeluarkan data yang belum terenkripsi atau *Plain Data*. Kemudian data tersebut akan di masukkan kedalam mode enkripsi dengan menggabungkan dengan kunci. Apabila proses enkripsi selesai maka sistemakan memberitahukan kepada pengguna bahwa proses enkripsi telah berhasil, kemudian data siap dikirimakan dalam bentuk file yang telah terenkripsi (*EncryptedFile.wav*).



Gambar 4. Interface crypt messenger

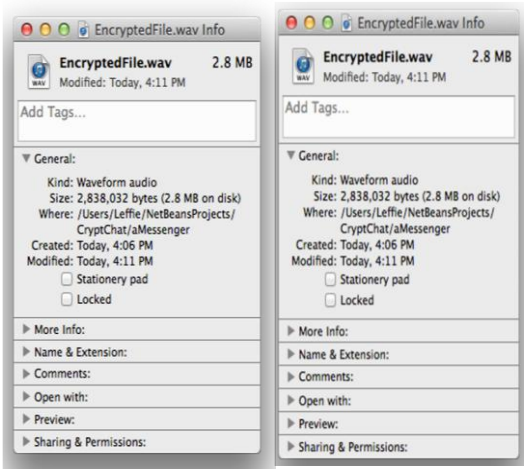
Setelah file dikirimkan ke penerima maka sistemakan memberikan notifikasi atau pemberitahuan untuk menerima pengiriman data yang terenkripsi. Setelah data diterima maka

pengguna yang bertugas menerima data hasil enkripsi masuk kedalam mode dekripsi dengan menggabungkan file yang telah diterima sebelumnya dalam bentuk *EncryptedFile.wav* untuk digabungkan dengan kunci dan akan mengeluarkan *Plain Data* yang telah didekripsi dalam bentuk *DecryptedFile.wav*. Perlu diperhatikan bahwa kunci yang dimasukkan untuk proses dekripsi harus sama untuk menghasilkan *Plain Data* semula apabila kunci yang dimasukkan salah maka output dari proses dekripsi tidak akan menghasilkan *Plain Data* semula dan *file* yang dihasilkan tetap terenkripsi.

Analisis Enkripsi Data

Pertama *Plain Data* berbentuk *Wave Form* dan Kunci di inputkan terlebih dahulu sebelum mengenkrip data. Kemudian pesan akan dienkripsi menggunakan kunci yang telah diinputkan dengan metode penyandian *Blowfish* (Syafari, 2007).

Gambar 4 menunjukan analisis enkripsi pesan pada pesan Plain data.wav dengan ukuran 2,838,028 bytes (2.8 MB on disk), kunci : 12345678 dan ukuran CipherData : 2,838,032 bytes

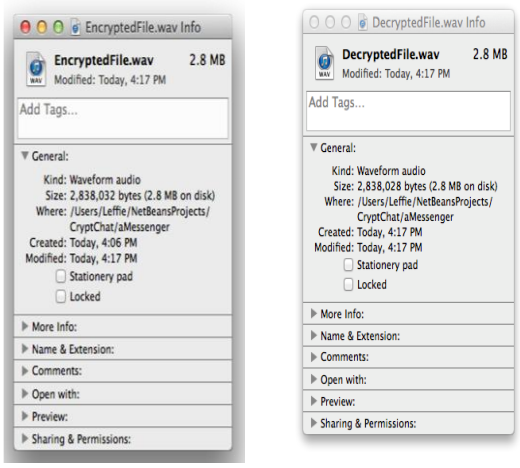


Gambar 5 Analisis enkripsi pesan

Analisis Dekripsi Pesan

CipherData tidak bisa didekripsi jika user salah menginputkan kunci. Hasil pendekripsian dengan kunci yang benar akan kembali seperti *Plain Data* semula. Dengan ukuran yang sama persis dengan *Plain Data* (Syafari, 2007)

Gambar 5 menunjukan analisis dekripsi pesan pada pesan EncryptedFile.wav , dengan kunci : 12345678, ukuran *CipherData* : 2,838,032 bytes (2.8 MB on disk), dan ukuran DecryptedFile : 2,838,028 bytes.



Gambar 6 Analisis dekripsi pesan

Jika pengguna mendekripsi data dengan kunci yang salah maka data tidak akan bisa didengar. dan file masih tetap terenkripsi dengan aman.

4 Hasil Pengujian Enkripsi Dekripsi

Pengujian merupakan tahap yang utama dalam pembuatan suatu aplikasi. Hasil pengujian yang didapat, akan dijadikan sebagai tolak ukur dalam proses pengembangan selanjutnya. Pengujian ini dilakukan untuk mengetahui hasil yang didapat dari aplikasi yang telah dibuat.

Tabel 3. Enkripsi data suara

No	Input	Output
1	- Data: Michael Bubble - Home.wav - Ukuran: 1,821,761 bytes (1.73 MB on disk) - kunci: qwertyui1234 (12 Karakter)	Data Terenkripsi - *cipherdata*: EncryptedFile.wav - *Ukuran* : 1,821,768 bytes (1.73 MB on disk)
2	- Data: Michael Bubble - Home.wav - Ukuran: 1,821,761 bytes (1.73 MB on disk) - kunci: enkripsi (8 Karakter)	Data Terenkripsi *cipherdata*: EncryptedFile.wav *Ukuran* : 1,821,768 bytes (1.73 MB on disk)
3	- Data: Michael Bubble - Home.wav - Ukuran: 1,821,761 bytes (1.73 MB on disk) - kunci: kunciabdi1234qwer (16 Karakter)	Data Terenkripsi - *cipherdata*: EncryptedFile.wav - *Ukuran* : 1,821,768 bytes (1.73 MB on disk)
4	- Data: Indahnya Cinta.wav - Ukuran: 3,733,584 bytes (3.56 MB on disk) - kunci: abdianci12345678 (16 Karakter)	Data Terenkripsi - *cipherdata*: EncryptedFile.wav - *Ukuran* : 3,733,592 bytes (3.56 MB on disk)
5	- Data: Indahnya Cinta.wav - Ukuran: 3,733,584 bytes (3.56 MB on disk) - kunci: 12345678 (8 Karakter)	Data Terenkripsi - *cipherdata*: EncryptedFile.wav - *Ukuran* : 3,733,592 bytes (3.56 MB on disk)

Tabel 4. Deskripsi data suara

No	Input	Output
1	- Data: EncryptedFile.wav - Ukuran: 1,821,768 bytes (1.73 MB on disk) - Kunci : enkripsi (8 Karakter)	Data Terdekripsi *cipherdata*: DecryptedFile.wav *Ukuran* : 1,821,761 bytes (1.73 MB on disk)
2	- Data: EncryptedFile.wav - Ukuran: 1,821,768 bytes (1.73 MB on disk) - kunci: kunciabdi1234qwer (16 Karakter)	Data Terdekripsi *cipherdata*: DecryptedFile.wav *Ukuran* : 1,821,761 bytes (1.73 MB on disk)
3	- Data: EncryptedFile.wav - Ukuran: 3,733,592 bytes (3.56 MB on disk) - kunci: abdianci12345678 (16 Karakter)	Data Terdekripsi *cipherdata*: DecryptedFile.wav *Ukuran* : 3,733,584 bytes (3.56 MB on disk)
4	- Data: EncryptedFile.wav - Ukuran: 3,733,592 bytes (3.56 MB on disk) - kunci: 12345678 (8 Karakter)	Data Terdekripsi *cipherdata*: DecryptedFile.wav *Ukuran* : 3,733,584 bytes (3.56 MB on disk)
5	- Data: EncryptedFile.wav - Ukuran: 3,733,592 bytes (3.56 MB on disk) - kunci: abdi (4 Karakter)	Data Terdekripsi - *cipherdata*: DecryptedFile.wav - *Ukuran* : 3,733,584 bytes (3.56 MB on disk)

6 Kesimpulan dan Saran

Penerapan kunci algoritma *Blowfish* yang fleksibel semakin memudahkan pengguna. Hal ini karane tidak-perlunya untuk menginputkan kunci sesuai dengan standar yaitu 16 *character*. Algoritma *Blowfish* menerapkan jaringan Feistel (Feistel Network) yang terdiri dari 16 putaran. *Blowfish* merupakan *cipher block*, yang berarti selama proses enkripsi dan dekripsi, *Blowfish* bekerja dengan membagi pesan menjadi blok-blok bit dengan ukuran sama panjang yaitu 64-bit dengan panjang kunci bervariasi yang meng-enkripsi data dalam 8 byte blok. Pesan yang bukan merupakan kelipatan 8 byte akan ditambahkan bit-bittambahan (*padding*) sehingga

ukuran untuk tiap blok sama. Algoritma *Blowfish* terdiri dari dua bagian, yaitu *key expansion* dan enkripsi data. Keamanan data bergantung pada kunci yang diinputkan. Semakin panjang atau semakin rumit kunci maka semakin aman pula file *enkripsi* yang dihasilkan. Proses enkripsi dan dekripsi pada data suara tergolong cepat. Dapat dilihat dari pengujian aplikasi yang mengenkripsi *file/data* dengan berbagai ukuran penggunaan kunci yang berbeda-beda pada tiap data yang sama menghasilkan perbedaan waktu dan ukuran file enkripsi dan dekripsi yang tidak terlalu *besar*, hanya beberapa detik saja. Diharapkan pada penelitian selanjutnya dapat dilakukan pengembangan aplikasi ini dapat dibuat berbasis *android chatting application.*

Daftar Pustaka

Ariyus, D. 2008. *"Pengantar Ilmu Kriptografi : Teori, Analsis & Implementasi",* Penerbit Andi, Yogyakarta.

Hidayanto, A 2007. *"Teknik Pengolahan Suara Digital"*, Penerbit Universitas Diponegoro, Yogyakarta.

Kelsey, J, 1998, "*Analisis Sistem Blowfish*: a 64-Bit Block *Cipher".Edisi 6*", *Mc Graw Hill Education*, Penerbit ANDI

Kromodimoeljo, S 2009. *"Teori dan Aplikasi Kriptografi"*, Penerbit SPK IT *Consulting*, Yogyakarta.

Munir, R, 2004. "*Bahan Kuliah IF5054 Kriptograf"*, Departemen Teknik Informatika, Institut Teknologi Bandung.

Sadikin, R, 2012. " *Kriptografi untuk Keamanan Jaringan dan Implementasinya dalam Bahasa Java"*. Yogyakarta: Andi.

Schneier, Bruce. 1996, "*Applied Cryptography Second Edition : Protocol, Algorithm, and Source Code in C"*, John Willey & Sons. Inc., New York

Sutanto CA, 2010. "*Penggunaan Algoritma Blowfish dalam kriptografi"*, halaman 6. Tersedia: http://webmail.informatika.org/~rinaldi/Matdis/20092010/Makalah0910/MakalahStrukdis09 10-070.pdf [27 Juni 2014].

Syafari A, 2007. "*Tentang Enkripsi Blowfish"*, 14 halaman. Tersedia: http://ilmukomputer.org/wp--content/uploads/2007/07/anjar--enkripsi--*Blowfish*.doc [2 September 2014].

Trisnawati.,2007. "*Sistem Kemanan Menggunakan Blowfish Advance CS Pada File dan Folder Data"*, Fakultas Ilmu computer, Universitas Sriwijaya, Palembang.