Kriptografi Audio Dengan Teknik Interferensi Data Non Biner

Muhamad Fajrin Rasyid [1)]

1) Program Studi Teknik Informatika ITB, Bandung 40132, email: if14055@students.if.itb.ac.id

Abstract **–** *Kriptografi audio adalah teknik untuk menyembunyikan informasi dalam berkas audio dengan membagi informasi tersebut menjadi n (n > 1) bagian. Untuk mendengarkan berkas audio asal, dapat dilakukan dengan memperdengarkan k (1 < k <= n) berkas audio. Saat ini aplikasi kriptografi audio yang ada hanya dapat membagi informasi pada berkas audio biner yang terdiri atas dua jenis suara, yaitu suara panjang dan pendek atau suara tinggi dan rendah. Makalah ini bertujuan untuk menerapkan kriptografi audio pada data non biner sehingga suara apapun dapat dirahasiakan. Metode yang digunakan adalah interferensi data non biner, yaitu mengambil sample audio non biner dari berkas audio asal dan membaginya ke dalam n sample audio acak sehingga jika k sample audio diputar bersamaan, dihasilkan sample audio semula. Batasan masalah yaitu aplikasi diterapkan pada berkas WAV yang merupakan format yang tidak mengalami pemampatan. Hasil analisis menunjukkan bahwa nilai k harus sama dengan n. Implementasi dilakukan dengan menggunakan bahasa pemrograman Java dan kakas pengembangan Netbeans 6.5. Pengujian menunjukkan bahwa penggabungan sejumlah berkas WAV yang dihasilkan ke dalam satu berkas WAV sebelum memutarnya akan memberikan hasil yang lebih baik daripada tanpa proses penggabungan. Disimpulkan bahwa kriptografi audio dengan teknik interferensi data non biner dapat diterapkan pada berkas WAV.*

Kata kunci*: kriptografi audio, interferensi, data non biner, sample audio, berkas audio WAV.*

1. PENDAHULUAN

Teknologi informasi saat ini semakin populer digunakan dalam seluruh aspek kehidupan. Hampir seluruh informasi kini dikelola dalam bentuk data digital. Akan tetapi, penggunaan data digital belum tentu meningkatkan keamanan pesan tersebut. Berbagai teknik penyerangan muncul sehingga pihak yang tidak bertanggungjawab dapat mengetahui informasi rahasia yang terkandung dalam pesan. Oleh karena itu, faktor keamanan menjadi salah satu isu penting dalam pengelolaan data digital.

Kriptografi hadir untuk meningkatkan aspek keamanan pesan [9]. Hal ini dilakukan dengan menyandikan pesan ke dalam bentuk yang tidak dapat dimengerti lagi maknanya [7]. Di dalam kriptografi, algoritma yang menentukan bagaimana pesan asal yang dapat dimengerti (*plaintext*) diubah menjadi pesan acak (*ciphertext*) dan selanjutnya diubah kembali menjadi pesan asal tidak dapat dirahasiakan karena pihak-pihak yang berhak mengetahui pesan asal dapat berubah sewaktu-waktu. Jika algoritma dirahasiakan, algoritma harus berubah setiap terjadi pergantian pihak yang terlibat. Oleh karena itu, keamanan pesan bergantung pada kerahasiaan kunci.

Kunci perlu dikelola sehingga hanya dapat diketahui oleh pihak yang berhak mengetahui pesan asal. Berbagai skema untuk menjawab berbagai persoalan terkait pengelolaan kunci ini telah dikembangkan. Salah satu persoalan yang muncul adalah bagaimana membagi data menjadi n ($n > 1$) data terpisah sedemikian hingga untuk mengembalikan data semula, diperlukan setidaknya k ($1 < k <= n$) data.

Kriptografi audio merupakan teknik yang digunakan untuk menyembunyikan informasi yang dikandung dalam berkas audio dengan cara membagi informasi tersebut menjadi n bagian. Kriptografi audio membagi berkas audio ke dalam n ($n > 1$) berkas audio acak dengan membagi informasi yang terkandung dalam setiap *sample* audio. Untuk mendengarkan berkas audio asli, dapat dilakukan dengan memperdengarkan k ($1 < k <= n$) berkas audio secara bersamaan.

Hingga saat ini aplikasi yang menerapkan prinsip kriptografi audio belum banyak dikembangkan. Aplikasi kriptografi audio yang ada hanya dapat membagi informasi yang terdapat pada berkas audio biner, yaitu berkas suara yang terdiri atas dua jenis suara, yaitu suara panjang dan suara pendek (seperti kode Morse) atau suara tinggi dan suara rendah. Dengan demikian, informasi yang ingin dirahasiakan harus berupa data biner.

Makalah ini membahas gagasan untuk mengimplementasikan kriptografi audio pada data non biner sehingga suara apapun, termasuk bahasa alami manusia, dapat dirahasiakan. Hal ini dapat dilakukan dengan mengambil setiap *sample* audio non biner dari berkas audio asal dan membaginya ke dalam *n sample* audio acak. Pembagian ini dilakukan sedemikian hingga jika *k sample* audio diputar secara bersamaan melalui satu sumber, dihasilkan *sample* audio semula. Penggabungan suara dengan cara diputar bersamaan ini disebut dengan interferensi terhadap data non biner dari suatu berkas audio [6]. Berkas audio yang digunakan adalah berkas WAV karena berkas ini merupakan format paling umum digunakan dan dapat diputar oleh hampir seluruh perangkat multimedia.

2. ANALISIS

2.1. Kriptografi

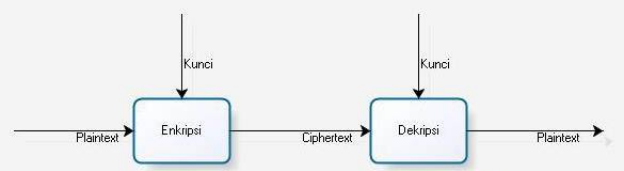
Kriptografi berasal dari bahasa Yunani dan terdiri atas dua kata, *kryptos* yang berarti tersembunyi dan *graphein* yang berarti menulis. Kriptografi pada awalnya didefinisikan sebagai ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara

menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya [7]. Namun demikian, kriptografi berkembang sehingga tidak hanya terbatas pada menyandikan pesan, tetapi juga memberikan aspek keamanan lain. Oleh karena itu, definisi kriptografi diperbarui menjadi ilmu dan seni untuk meningkatkan aspek keamanan pesan [9].

2.1.1. Elemen-elemen Kriptografi

Sebuah sistem yang mengimplementasikan kriptografi secara umum memiliki elemen sebagai berikut [13]:

1. *Plaintext*; yaitu pesan asal yang dapat dimengerti maknanya dan hendak dirahasiakan,
2. *Ciphertext*; yaitu pesan yang merupakan hasil penyandian yang tidak dimengerti maknanya,
3. Algoritma enkripsi; yaitu suatu aturan untuk menyandikan *plaintext* menjadi *ciphertext*,
4. Algoritma dekripsi; yaitu aturan untuk mengembalikan *ciphertext* menjadi *plaintext*,
5. Kunci; yaitu parameter yang menjadi masukan.



2.1.2. Manajemen Kunci

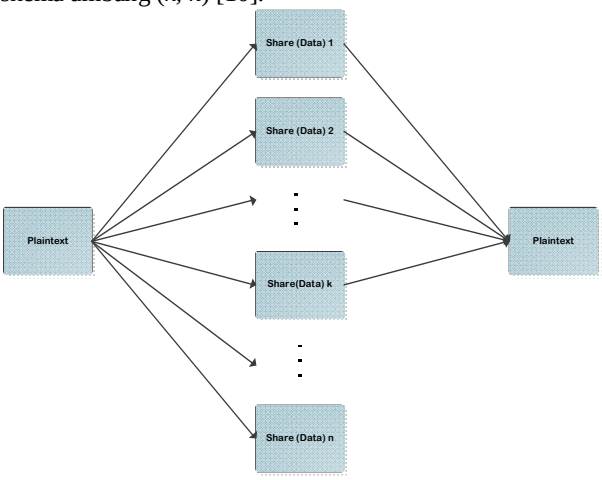
Kerahasiaan kunci merupakan aspek paling sulit di dalam ilmu kriptografi [9]. Pentingnya kunci telah melahirkan manajemen kunci, yaitu teknik yang dikembangkan untuk menjaga kerahasiaan kunci. Beberapa permasalahan yang ditangani di dalam manajemen kunci adalah sebagai berikut [9]:

1. Pembangkitan kunci
   Kunci dibangkitkan oleh proses otomatis yang bekerja berdasarkan suatu algoritma. Hal ini akan menghasilkan kunci acak yang lebih aman daripada kunci yang disepakati bersama oleh pengirim dan penerima pesan.
2. Pengiriman kunci
   Pengiriman kunci ini dapat diamankan salah satunya dengan membagi kunci menjadi beberapa bagian sehingga apabila hanya salah satu bagian diketahui oleh orang lain, hal itu tidak masalah.
3. Pemeriksaan kunci
   Ketika salah satu pihak menerima kunci, ia perlu memastikan apakah kunci tersebut asli atau tidak.
4. Penyimpanan kunci
   Kunci yang digunakan di dalam kriptografi tidak selalu dapat dihafalkan. Oleh karena itu, penyimpanan kunci merupakan hal yang harus dilakukan.
5. Pembuatan cadangan kunci
   Cadangan kunci diperlukan apabila kunci yang disimpan tidak dapat diambil kembali.

2.1.3 Skema Pembagian Data Rahasia

Skema pembagian rahasia merupakan salah satu teknik di dalam manajemen kunci khususnya dalam pengiriman kunci. Skema ini diajukan melalui persoalan Bank dan empat perampok. Misalkan sekelompok perampok yang beranggotakan empat orang ingin menyimpan hasil curian mereka di rekening Bank Swiss. Empat perampok tidak mempercayai satu sama lain karena mereka adalah orang-orang jahat. Secara spesifik, mereka tidak ingin ada satu anggota di antara mereka yang dapat menarik uang dari rekening tersebut dan melarikan diri. Namun, mereka menganggap bahwa penarikan uang oleh dua anggota bukan merupakan suatu konspirasi, melainkan suatu otoritas yang sah. Oleh karena itu, mereka memutuskan untuk membagi kode rahasia rekening Bank tersebut menjadi empat bagian sehingga dua atau lebih bagian dapat digunakan untuk merekonstruksi kode rahasia tersebut. Bagaimanakah hal ini dapat dilakukan? [2].

Skema pembagian data rahasia terdiri atas satu administrator dan sejumlah n ($n > 1$) pemain. Administrator membagikan data rahasia kepada para pemain tersebut dengan ketentuan sebagai berikut. Sejumlah k ($1 < k <= n$) pemain tersebut atau lebih selalu dapat secara bersama-sama merekonstruksi data rahasia semula, sedangkan tidak ada sekelompok pemain yang berjumlah kurang dari k orang yang dapat merekonstruksinya. Skema ini disebut dengan skema ambang (k, n) [10].



.

2.2. Audio

Audio, atau suara, merupakan gelombang yang merambat melalui medium tertentu [6]. Perambatan tersebut tiba di telinga sehingga kita dapat mendengar.

2.2.1. Interferensi Audio

Interferensi merupakan fenomena yang terjadi ketika dua atau lebih gelombang bertemu ketika melintasi medium yang sama. Interferensi menyebabkan terbentuknya gelombang yang merupakan hasil dari perpaduan dua atau lebih gelombang tersebut. Apabila dua gelombang yang bertemu memiliki bentuk yang sama, maka interferensi yang dihasilkan bersifat konstruktif. Sebaliknya, apabila dua gelombang yang bertemu memiliki bentuk yang berkebalikan, maka

interferensi yang dihasilkan bersifat destruktif.

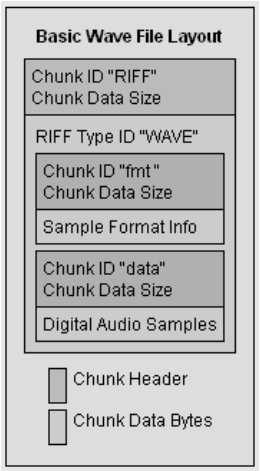
2.2.2. Audio Digital

Audio digital merupakan representasi audio yang disimpan dalam komputer secara biner (terdiri dari 0 dan 1). Apabila direpresentasikan dalam bentuk gelombang, audio digital memiliki sejumlah hingga nilai tekanan dan titik waktu.Terdapat dua faktor yang menentukan kualitas audio digital:

1. *Sample rate*
   Hal ini mengukur banyaknya *sample* yang dihasilkan, diukur dengan satuan Hz atau *sample* per detik.
2. *Sample format / sample size*
   Hal ini merupakan banyaknya digit yang tersedia untuk merepresentasikan *sample*.

2.2.3. Berkas Audio WAV

Berkas audio WAV, atau WAVE, merupakan standard format berkas audio yang digunakan oleh IBM dan Microsoft dalam menyimpan aliran data audio pada PC. Berkas audio WAV menerapkan teknik *Linear Pulse Code Modulation (LPCM)* dalam merepresentasikan data. LPCM merupakan salah satu jenis PCM yang menggunakan metode *lossless* dan tanpa kompresi, yaitu metode yang menyimpan seluruh *sample* audio, sehingga berkas WAV merupakan berkas mentah (sesuai dengan aslinya). Format berkas WAV secara umum menggunakan standard format RIFF seperti ditampilkan pada gambar dibawah ini.



2.3. Kriptografi Audio

Kriptografi audio muncul berdasarkan pemikiran bahwa prinsip skema pembagian data rahasia tidak hanya dapat diterapkan pada berkas gambar, tetapi juga pada berkas suara. Kriptografi audio bertujuan untuk membagi data suara menjadi *n* data suara. Hal ini bermanfaat antara lain dalam pembagian warisan dan dalam pengamanan data suara pada jaringan.

Aplikasi kriptografi audio yang berkembang terdiri atas dua jenis:

1. Skema yang dicetuskan oleh Desmedt, Hou, dan Quisquater dalam skema yang disebut Desmedt, Hou, and Quisquater Audio Secret Sharing Scheme (DHQ ASS) [3]. yang menerapkan prinsip interferensi konstruktif dan destruktif pada suatu gelombang harmonis sederhana yang berulang-ulang dan terdiri atas suara tinggi dan rendah,
2. Audio Cryptography Scheme (ACS) yang dicetuskan oleh Daniel Socek dan Spyros Magliveras [11] yang bekerja berdasarkan prinsip kode Morse (suara panjang dan suara pendek).

2.4. Algoritma Penanganan Permasalahan

Penggunaan lebih dari satu bit untuk merepresentasikan tiap *sample* audio mengakibatkan kesulitan dalam membagi *sample* audio tersebut menjadi *n sample* audio acak. Namun, dengan menerapkan prinsip interferensi yang diterapkan pada data non biner, pembagian yang dilakukan untuk tiap *sample* audio tetap berpedoman pada fakta bahwa berkas audio asal merupakan penjumlahan Dengan demikian, untuk membagi *sample* audio menjadi *n sample* audio, hal ini dapat dirumuskan sebagai:

$$M = x_1 + x_2 + \cdots + x_n \qquad \text{[III-I]}$$

dengan *M* adalah *sample* audio asal, x_1 adalah *sample* audio hasil 1, x_2 adalah *sample* audio hasil 2, … , dan x_n adalah *sample* audio hasil ke-*n*.). Hal yang perlu diperhatikan adalah implikasi dari rumus [III-I] mengakibatkan diperlukan tepat sejumlah *n sample* audio untuk merekonstruksi *sample* audio semula. Dengan demikian, untuk pembagian *sample audio* menjadi *n sample* audio, skema ambang yang mungkin adalah skema ambang (*n*, *n*).

Pembagian *sample* audio menjadi *n sample* audio ini dapat dilakukan dengan melakukan operasi XOR antara berkas audio semula dengan kunci untuk mendapatkan *share* atau data hasil ke-1 hingga ke-(*n* – 1), dan menggunakan rumus [III-I] untuk mendapatkan *share* ke-*n*. Prosedur selengkapnya adalah sebagai berikut:

1. Tentukan kunci yang digunakan sebagai parameter dalam proses pembagian *sample* audio. Hal ini dapat dilakukan dengan dua cara, berdasarkan masukan dari pengguna atau dibangkitkan secara acak oleh program. Apabila kunci yang digunakan berasal dari masukan pengguna, kunci ini terlebih dahulu diubah menjadi bilangan biner yang bersesuaian, sebagai contoh kunci ‘suara’ yang dimasukkan oleh pengguna akan direpresentasikan sebagai 01110011011101010110000101110010011000001 . Sementara itu, kunci yang berasal dari pembangkitan secara acak oleh program akan berupa suatu bilangan yang dapat langsung diubah menjadi bilangan biner.
2. Untuk tiap *share* ke-*i* ($1 < i < n$), bangkitkan kunci ke-*i* yang merupakan permutasi dari kunci yang menjadi parameter yang dihasilkan oleh langkah 1) dan di-XOR dengan bilangan acak,
3. Lakukan operasi XOR antara kunci yang dihasilkan dari langkah 3) dengan barisan *sample*

audio. Apabila panjang kunci lebih kecil daripada barisan *sample* audio yang menjadi masukan, maka gunakan kunci tersebut berulang-ulang seperti pada *vigenere cipher*. Sebagai contoh dibawah ini:

Sample audio : 1010101011110100-0110100100001101…
Kunci : 1100001111001011-1100001111001011…
Hasil XOR : 0110100100111111-1010101011000100…

4. Simpan hasil XOR ini sebagai *share* ke-*i*. Perhatikan bahwa hasil XOR tidak akan menambah jumlah bit yang diperlukan. Dengan demikian, hasil XOR ini akan selalu terletak dalam rentang yang diizinkan (pada berkas suara 16 bit, *sample* audio terletak pada rentang -32768 hingga +32767),
5. Tentukan *share* ke-*n* dengan rumus [III-II], yaitu $x_n = M - (x_1 + x_2 + \cdots + x_{n-1})$.

3. HASIL DAN PEMBAHASAN

3.1. Perancangan dan Implementasi

Secara umum, perangkat lunak yang dibangun memiliki fungsi sebagai berikut:

No	Deskripsi
1	Sistem mampu membagi berkas audio WAV menjadi *n* berkas WAV acak dengan kunci
2	Sistem mampu membagi berkas audio WAV menjadi *n* berkas WAV acak tanpa kunci
3	Sistem mampu memutar sejumlah lebih dari satu berkas WAV secara bersamaan
4	Sistem mampu memberikan pilihan berkas audio WAV yang ingin diputar dari sejumlah berkas WAV yang dibuka
5	Sistem mampu memperdengarkan berkas WAV semula jika dan hanya jika *n* berkas WAV diputar secara bersamaan
6	Sistem mampu menampilkan grafik representasi audio yang dikandung dalam tiap berkas WAV

Deskripsi kelas-kelas yang dibutuhkan untuk diimplementasikan pada perangkat lunak adalah sebagai berikut:

1) Kelas *UserPanel*
   Kelas *UserPanel* merupakan kelas yang berperan sebagai penghubung perangkat lunak dengan pengguna. Fungsi-fungsi yang terdapat di dalam kelas ini adalah fungsi-fungsi yang dijalankan ketika pengguna memilih suatu aksi tertentu.
2) Kelas *AudioDistribution*
   Kelas *AudioDistribution* merupakan kelas yang menyediakan fungsi-fungsi terkait peran utama perangkat lunak dalam membagi berkas audio WAV.
3) Kelas *AudioPlayer*
   Kelas *AudioPlayer* merupakan kelas yang menyediakan fungsi-fungsi terkait memainkan berkas audio WAV.
4) Kelas *GraphRepresentation*
   Kelas *GraphRepresentation* merupakan kelas yang menyediakan fungsi-fungsi terkait menampilkan grafik representasi berkas audio WAV.
5) Kelas *WaveFile*
   Kelas *WaveFile* merupakan kelas yang merepresentasikan berkas audio WAV, yaitu tipe data yang dikelola oleh perangkat lunak.

3.2. Pengujian Perangkat Lunak

3.2.1. Skenario Pengujian Perangkat Lunak

Terdapat dua skenario pengujian yang dilakukan sebagai berikut:

1) Skenario utama; yaitu tanpa menggabungkan terlebih dahulu berkas-berkas audio WAV yang dihasilkan. Dengan demikian, sejumlah berkas audio WAV yang dihasilkan diputar secara bersama-sama,
2) Skenario alternatif; dengan menggabungkan berkas-berkas audio WAV yang dihasilkan ke dalam satu berkas audio sementara terlebih dahulu sebelum diputar.

Untuk setiap skenario pengujian, akan diuji kebenaran dan kinerja perangkat lunak. Adapun langkah yang diuji untuk setiap skenario adalah sebagai beriku:

1) Pengguna membuka berkas audio WAV
2) Pengguna memasukkan parameter kunci
3) Pengguna memilih untuk membagi berkas audio WAV
4) Pengguna menyimpan berkas audio WAV keluaran
5) Pengguna membuka *n* berkas audio WAV yang dihasilkan
6) Pengguna memilih sejumlah kurang dari *n* berkas audio WAV yang dibuka tersebut untuk dimainkan
7) Pengguna memainkan sejumlah berkas audio WAV yang dipilih
8) Pengguna mengulangi langkah 6) dan 7) namun kali ini dipilih tepat *n* berkas audio WAV
9) Pengguna mengulangi langkah 1) – 8) untuk berbagai nilai *n* dan pilihan untuk memasukkan kunci atau tidak

3.2.2. Data Pengujian Perangkat Lunak

Data yang digunakan dalam pengujian adalah sebagai berikut:

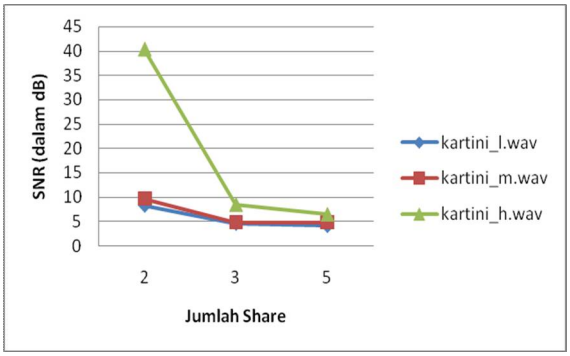
Nomor	Nama berkas	Deskripsi
1	kartini_h.wav	Suara manusia menyanyikan lagu
2	freedom.wav	Musik
3	adaband.wav	Suara manusia menyanyikan lagu + musik

Nomor	Nama berkas	*Sample rate*	Nilai Sinyal
1	kartini_h.wav	44,100 Hz	-16.73 dB
2	kartini_m.wav	22,050 Hz	-10.71 dB
3	kartini_l.wav	11.025 Hz	-10.71 dB

Dalam hal ini, kelompok data pertama digunakan untuk menguji kinerja perangkat lunak, sedangkan kelompok data kedua digunakan untuk menguji kebenaran perangkat lunak.
Pengujian kebenaran dilakukan dengan menghitung SNR, yaitu nilai sinyal berkas audio yang dihasilkan dan dibandingkan dengan nilai sinyal berkas audio semula.

3.2.3. Hasil dan Analisis Pengujian Perangkat Lunak

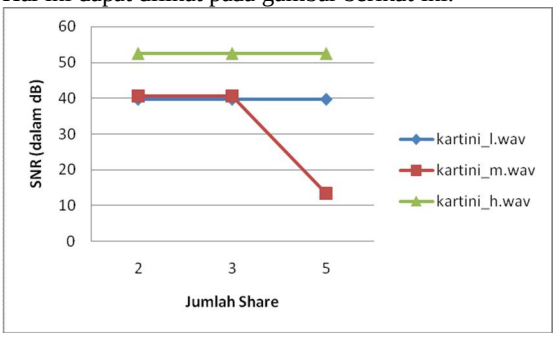
Pada skenario utama, berkas audio WAV semula dapat didengar ketika sejumlah berkas audio WAV hasil pembagian diperdengarkan secara bersamaan. Namun, suara yang dihasilkan pada umumnya kurang jelas, terutama untuk berkas audio WAV "adaband.wav" yang mengandung perpaduan beberapa jenis suara (suara manusia dan musik). Dengan demikian, berkas audio WAV yang layak menjadi masukan adalah berkas audio WAV yang hanya mengandung satu jenis suara.
Selain itu, dari pengujian yang dilakukan, semakin besar jumlah *n* maka nilai SNR akan semakin rendah, yang berarti bahwa kualitas audio yang dihasilkan akan semakin menurun seperti pada gambar dibawah ini.



Hal lain yang dapat dilihat dari gambar tersebut adalah fakta bahwa secara umum, untuk kondisi yang sama (jumlah *share* yang sama), maka berkas audio WAV yang memiliki kualitas audio paling tinggi akan menghasilkan nilai SNR yang paling tinggi pula, dan sebaliknya. Dalam hal ini, berkas audio yang dihasilkan yang masih layak didengar adalah berkas audio WAV yang dihasilkan oleh kasus dengan menggunakan masukan berkas audio WAV "**kartini_h.wav**" dan jumlah *share* dua (dengan ataupun tanpa kunci). Hal ini disebabkan berkas audio WAV "**kartini_h.wav**" adalah berkas audio dengan kualitas terbaik dibandingkan dengan kedua berkas audio WAV lainnya. Selain kasus tersebut, nilai SNR yang dihasilkan kurang dari 30 dB sehingga berkas audio yang dihasilkan mengandung *noise* yang tinggi.

Hasil pengujian skenario utama yang kurang baik tersebut menjadi dasar untuk diajukannya teknik untuk menggabungkan berkas audio WAV yang dihasilkan tersebut terlebih dahulu sebelum diputar. Hal ini akan menghasilkan keluaran yang mendekati berkas audio WAV semula. Teknik inilah yang diuji pada skenario alternatif.

Pada skenario alternatif, , berkas audio WAV semula dapat didengar dengan kualitas yang jelas dan hampir sama seperti suara asli ketika berkas audio WAV hasil penggabungan tersebut diperdengarkan untuk seluruh kasus uji yang dilakukan.
Selain itu, dari pengujian yang dilakukan, hampir seluruh berkas audio yang dihasilkan layak didengar karena memiliki nilai SNR lebih dari 30 dB, kecuali satu kasus, yaitu kasus dengan menggunakan masukan berkas audio WAV "**kartini_m.wav**" dan jumlah *share* lima serta tidak menggunakan kunci. Dengan demikian, secara umum teknik menggabungkan berkas audio WAV sebelum diputar akan memberikan nilai SNR yang lebih baik daripada jika tidak diputar. Hal ini dapat dilihat pada gambar berikut ini.



4. KESIMPULAN

Dari pembahasan di atas, dapat ditarik kesimpulan sebagai berikut.

1. Kriptografi audio dengan teknik interferensi data non biner dapat diterapkan pada berkas audio WAV tanpa kompresi,
2. Perangkat lunak yang mengimplementasikan kriptografi audio dengan teknik interferensi data non biner pada berkas audio WAV tanpa kompresi dapat dibangun,
3. Teknik untuk menggabungkan sejumlah berkas audio WAV yang dihasilkan ke dalam satu berkas audio WAV terlebih dahulu sebelum memutarnya memberikan hasil yang lebih baik daripada memutar sejumlah berkas audio WAV yang dihasilkan secara bersamaan tanpa melalui proses penggabungan,
4. Kualitas berkas audio WAV yang dihasilkan bergantung pada jumlah jenis suara yang digunakan sebagai masukan. Semakin banyak jenis suara yang dikandung di dalam berkas audio WAV tersebut maka kualitas berkas audio WAV yang dihasilkan akan semakin menurun.
5. Kualitas berkas audio WAV yang dihasilkan

bergantung pada jumlah *share* dan kualitas berkas audio WAV yang digunakan sebagai masukan. Semakin banyak jumlah *share* maka kualitas berkas audio WAV yang dihasilkan akan semakin menurun. Sementara itu, semakin baik kualitas audio WAV yang digunakan sebagai masukan (yang ditunjukkan dengan tingginya nilai *sample rate* dan *sample size* berkas audio WAV tersebut) maka kualitas berkas audio WAV yang dihasilkan akan semakin baik,

6. Kualitas berkas audio WAV yang dihasilkan tidak dapat disimpulkan dari kunci yang digunakan,
7. Dapat dilakukan peningkatan performansi terkait teknik yang digunakan dalam membagi berkas audio WAV sehingga kualitas berkas audio WAV yang dihasilkan ketika diputar secara bersamaan menjadi lebih baik dan dapat berlaku pada skema ambang (*k*, *n*),
8. Dapat dilakukan analisis terkait implementasi kriptografi audio dengan teknik interferensi data non biner pada berkas audio dengan format yang berbeda sehingga dapat ditentukan format yang paling sesuai,

DAFTAR REFERENSI

[1] http://www.bcae1.com/sig2nois.htm, diakses tanggal 1 Juni 2009 pukul 13.30.

[2] Cai, Jim. *Short Survey on Visual Cryptography Schemes*. 2005.

[3] Desmedt, dkk. *Audio Cryptographic Scheme*. Asiacrypt. 1998.

[4] Forouzan, Behrouz A. *Data Comunications and Netwoking: 3rd Edition*. McGraw Hill. 2004.

[5] http://www.garykessler.net/library/crypto.html , diakses tanggal 12 Februari 2009 pukul 11.54.

[6] http://www.glenbrook.k12.il.us/GBSSCI/ PHYS/Class/waves/u10l3c.html, diakses tanggal 3 Februari 2009 pukul 21.35.

[7] Munir, Rinaldi. *Kriptografi*. Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung. 2006.

[8] Naor, Moni dan Adi Shamir. *Visual Cryptography*. Eurocrypt. 1994

[9] Schneier, Bruce. *Applied Cryptography 2nd*. John Wiley & Sons. 1996.

[10] Shamir, Adi. *Secret Sharing Scheme*. 1979.

[11] Socek, Daniel dan Spyros S. Magliveras. *General Access Structure in Audio Cryptography*. 2006.

[12] http://www.sonicspot.com/guide /wavefiles.html, diakses tanggal 25 Februari 2009 pukul 14.27.

[13] Stallings, William. *Network Security Essentials.*1999.