Journal of Informatics Management and Information Technology
ISSN 2774-4744 (Media Online)
Vol 2, No 1, Januari 2022
Hal 18−34
https://hostjournals.com/jimat

Impelementasi Metode N-Hash Untuk Mendeteksi Keaslian File Audio

Udiyani

Program Studi Teknik Informatika, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Budi Darma, Medan
Jl. Sisingamangaraja No.338, Siti Rejo I, Kec. Medan Kota, Kota Medan, Sumatera Utara, Indonesia
Email: udiyaniskom@gmail.com

Abstrak−File audio merupakan suatu sarana informasi dari satu orang ke orang lain atau dari suatu kelompok-kelompok lain. Perkembangan teknologi komputerisasi ini sudah sangat meninggkatkan. File audio sangat rentan terhadap penipuan, penyadapan maupun pencurian data oleh pihak-pihak yang tidak bertanggung jawab. Demi menjaga keamanan file audio dapat dilakukan dengan pemanfaatan teknik kriptografiKriptografi adalah sekumpulan teknik yang berguna untuk mengamankan informasi, Selain mengamankan informasi juga menjaga kerahasian dan keutuhan informasi tersebut. kriptografi digunakan untuk komunikasi penting seperti komunikasi dikalangan militer, diplomat, dan mata-mata. Namun saat ini kriptografi lebih nonrepudiation.N-Hash adalah keluarga algoritma enskripsi yang dirancang untuk memiliki dan sederhana implementasi perangkat lunak yang efesien pada miropesesor 128 bit anggota asli dari keluarga iniyang disebut N-hash yang memiliki 8 putaran

Kata Kunci: Kriptografi; Metode Hash; File Audio

Abstract−Audio file is a means of information from one person to another or from a group to another. The development of computerized technology has greatly increased. Audio files are very vulnerable to fraud, eavesdropping or data theft by irresponsible parties. In order to maintain the security of audio files, it can be done by using cryptographic techniques. Cryptography is a collection of techniques that are useful for securing information. Apart from securing information, it also maintains the confidentiality and integrity of the information. Cryptography is used for important communications such as communications among the military, diplomats, and spies. But nowadays cryptography is more non-repudiation.N-Hash is a family of encryption algorithms designed to have an efficient and simple software implementation on 128-bit microprocessors. The original member of this family is called N-hash which has 8 rounds.

Keywords: Cryptography; Hash Method; Audio Files

1. PENDAHULUAN

Dalam era teknologi informasi yang berkembang sangat pesat, penggunaan file audio sudah banyak diterapkan secara digital melalui orisinalitas file audio. Seiring munculnya kebutuhan *orisinalitas* suatu data atau berkas yang digunakan secara digital. Saat ini, pemanfaat mendeteksi sudah banyak diterapkan pada distribusi perangkat lunak.

Keaslian file audio pada saat ini menjadi hal yang sangat penting dan terus berkembang, karena itu file audio dijadikan sebagai sumber utama informasi. Berbagai *sofware editing* audio menyulitkan seseorang untuk membedakan antara file audio asli atau audio palsu. Audio sering kali disalah gunakan oleh pihak-pihak tertentu untuk memanipulasi sebuah rekaman dengan memalsukan file audio yang bertujuan untuk menghilangkan bukti-bukti yang ada didalamnya.

Untuk membedakan audio yang asli atau audio yang palsu maka diperlukan mendeteksi terhadap file audio tersebut menggunakan kriptografi. Kriptografi merupakan salah satu metode pengamanan file yang dapat digunakan menjaga keaslian file. Oleh karena itu dibutuhkan suatu cara untuk mendeteksi keaslian dari file audio tersebut. Salah satu solusi untuk mengatasi asli atau sudah di maodifikasi file audio tersebut dengan menggunakan metode *N-hash*.

Algoritma *N-Hash* pada tahun 1990 dan algoritma *N-Hash* merupakan fungsi *hash* kriptografi yang memiliki pesan panjang hingga nilainya mencapai 128 bit. Pesan- pesan ini dibagi menjadi blok 128 bit, dan setiap blok dicampur dengan nilai hash yang dihitung sejauh ini dengan pengacakan dari fungsi g. Nilai *hash* yang baru adalah xor dari output fungsi-g dengan blok itu sendiri dan dengan nilai hash yang lama [1]. Fungsi-g berisi delapanputaran, dan masing-masing dari mereka memanggil fungsi f. Secara kriptografis memecahkan fungsi *hash* berarti menemukan dua pesan yang berbeda dengan nilai yang sama. Algoritma *N-Hash* mempunyai kelebihan yaitu, kecepatan dalam berkomputasi dan kemudahan dalam implementasi dari algoritma lain.

Fungsi *n-hash* sering juga disebut enkripsi satu arah, atau disebut juga *messagediggest.* Fungsi *hash* digunakan untuk menjamin servis otentikasi dan integritas suatu pesan atau file. Suatu fungsi *hash* memetakan bit-bit *string* dengan panjang sembarang ke sebuah *string* dengan panjang tertentu misal n. Dengan domain D dan range R maka: Proses *hashing* merupakan proses pemetaan suatu input string menjadi *output*disebut ouput dari fungsi *hash* disebut nilai *hash* atau hasil *hash*.

2. METODOLOGI PENELITIAN

2.1 Kriptografi

Kriptografi (*cryptoraphy)* berasal dari bahasa Yunani: "*cryptos"* artinya *"secret"* (rahasia), sedangkan "*graphein"* artinya *"writing"* (tulisan). Jadi kriptografi berarti *"secret writting"* (tulisan rahasia). Ada berbagai pemahaman yang menjelaskan mengenai arti dari kriptografi. Ada beberapa defenisi kriptografi yang telah dikemukakan didalam berbagai literature. Defenisi yang dipakai dalam buku-buku yang lama (sebelum tahun 1980-an) menyatakan bahwa kriptografi

adalah ilmu dan seni untuk menjaga kerahasian pesan dengan cara menyandikannya kedalam bentuk yang tidak dapat dimengerti lagi maknanya

Pada masa lalu kriptografi digunakan untuk komunikasi penting seperti komunikasi dikalangan militer, diplomat, dan mata-mata. Namun saat ini kriptografi lebih *nonrepudiation* [2].

2.2 Fungsi *Hash*

Fungsi *hash (has function)* merupakan salah satu teknik kriptografi untuk menghitung nilai unik dari sebuah data. Fungsi *hash* dapat diberatkan sebagai sidik jari elektronik berguna untuk menentukan orisinalitas sebuah dokumen elektronik. Dua dokumen elektronik yang berbeda akan memilikinilai *hash* yang berbeda, itulah sebabnya apabila sebuah dokumen telah mengalami perubahan, maka nilai *hash* juga akan berubah. Sebuah dokumen dengan panjang berapapun akan menghasilkan nilai *hash* dengan panjang tertentu sesuai dengan spesifikasi fungsi *hash* yang digunakan [1]. Fungsi hash adalah fungsi yang menerima masukan string yang panjangnya sembarangan dan mengonverinya menjadi string keluaran yang panjangnya tetap *(fixed)*, umumnya berukuran jauh lebih kecil dari pada ukuran string semula.
Sifat-sifat fungsi satu arah diantaranya sebagai berikut:

1. Fungsi H dapat diterapkan pada blok data dengan ukuran berapa saja.
2. H menghasikan nilai (h) dengan panjang tetap *(fixed-lenghth output).*
3. H(x) mudah dihitung untuk setiap nilai x yang diberikan.
4. Untuk setiap h yang diberikan, tidak mungkin menemukan sedemikiansehingga H (x) = h. Itu sebabnya fungsi H dikatakan fungsi fungsi hash satu-arah *(one-way hash function).*
5. Untuk setiap x yang diberikan, tidak mungkin mencari pasangan y # x sedemikian sehingga H(y) = H(x).
6. Tidak mungkin secara komputasi mencari pasangan x dan y sedemikian sehingga H(y).

Sifat-sifat diatas sangat penting untuk sebuah fungsi *hash*, sebab sebuah fungsi *hash* seharusnya berlaku seperti fungsi acak. Sebab fungsi *hash*dianggap tidak aman jika (1) secara komputasi dimungkinkan menemukan pesan yang bersesuaian dengan pesan ringkasnya *(massage digest)*, dan (ii) terjadi kolusi *(collision)*, yaitu terdapat beberapa pesan berbeda yang mempunyai pesan ringkas yang sama.

2.2.1 Struktur Fungsi *hash*

Kebanyakan fungsi *hash* memiliki kemiripan dalam struktur interasi. Struktur tersebut berlandaskan pada kaidah-kaidah pada fungsi kompresi. Pendek kata, komputasi untuk menghasilkan fungsi hash sangat bergantung pada pesan input yang diberikan *(massage depend)* yang biasa disebut dengan variabel berantai *(chaining variable)*. Pada permulaan proses *hashing, chaining variable* tersebut diinisialisasi dengan suatu nilai tertentu yang sifatnya tetap. Hal ini merupakan salah satu dari spesifikasi algoritma fungsi *hash* [1]. Lalu fungsi kompresi digunakan untuk mengubah atau update nilai-nilai yang terdapat *chaining variable* dengan menggunakan cara algoritma yang cukup rumit. Proses ini berlanjut secara rekursif dengan *chaining variable* selalu di-update terus menerus untuk setiap bagian pada pesan tersebut, hingga semua pesan telah digunakan. Nilai akhir dari *chaining variable* akan menjadi nilai hash yang berkorespondensi dengan pesan tersebut.

2.3 Metode *N-hash*

N-Hash pada tahun 1990 algoritma N-Hash ini ditemukan oleh para peneliti di Nippon Telephone dan Telegraph, yaitu orang yang menemukan *FEAL*, pada tahun 1990.*N-Hash* menggunakan blok pesan 128 bit, fungsi pengacakan yang rumit mirip dengan *FEAL*, dan menghasilkan nilai *hash* 128 bit. *Hash* dari setiap blok 128 bit adalah fungsi dari blok dan *hash* dari blok sebelumnya [1].
H0 = I, di mana I adalah nilai awal acak

$$H_i = g(M_i, H_{i-1}) \bullet M_i \bullet H_{i-1} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

Hash dari seluruh pesan adalah *hash* dari blok pesan terakhir. Itu nilai awal acak, dapat berupa nilai apa pun yang ditentukan oleh pengguna (bahkan semuanol). Fungsi g adalah yang rumit. *Hash* 128-bit dari blok pesan sebelumnya, Hi-1, memilikisetengah kiri 64-bit dan setengah kanan 64-bit bertukar, kemudian XORed dengan ulangi satu / nol pola (senilai 128 bit), dan kemudian XOR dengan arus blok pesan, nilai ini kemudian mengalir ke N (N = 8. Input lain ke tahap pemrosesan adalah *hash* sebelumnya nilai XOR dengan satu dari delapan nilai konstanta biner. Blok pesan dipecah menjadi empat nilai 32-bit. Nilai *hash* sebelumnya juga dipecah menjadi empat 32-bit nilai-nilai. Bertden Boer menemukan cara untuk menghasilkan tabrakan dalam fungsi putaran *N-Hash*. Biham dan Shamir menggunakan pembacaan sandi diferensial untuk memecah 6-putaran *N-Hash.* Serangan yang sama dapat menemukan pasangan pesan yang *hash* dengan nilai yang sama12-putaran *N-Hash* dalam 256 operasi, dibandingkan dengan 264 operasi untuk aserangan membabi buta. *N-hash* dengan 15 putaran aman dari diferensial pembacaan sandi, serangan itu membutuhkan 272 operasi. *Hash* dengan nilai yang sama12-putaran N-Hash dalam 256 operasi, dibandingkan dengan 264 operasi untuk serangan membabi buta. N-hash dengan 15 putaran aman dari diferensial pembacaan sandi, Serangan itu membutuhkan 272 operasi.

2.4 File Audio

Journal of Informatics Management and Information Technology
ISSN 2774-4744 (Media Online)
Vol 2, No 1, Januari 2022
Hal 18−34
https://hostjournals.com/jimat

Audio adalah suara/bunyi yang dihasilkan oleh getaran benda. Agar dapat tertangkap oleh telinga manusia, getaran tersebut harus cukup kuat yaitu minimal 20 kali per detik. Banyaknya getaran suatu benda diukur dengan satuan cycles persecond atau cps.

3. HASIL DAN PEMBAHASAN

Pengujian dilakukan untuk mengukur kinerja algoritma n-hash dalam mendeteksi keaslian file audio:

1. Putaran Pertama

   Pada putaran pertama ini proses yang dilakukan adalah mengambil nilai input biner dari file audio. Setelah itu nilai input biner tersebut dibuat dengan variabel M_1. Kemudian dari variabel M_1 tersebut, dibagi lagi menjadi variabel X_1,X_2,X_3,X_4,. Setelah itu variabel X_1,X_2,X_3,X_4, dibagi lagi menjadi Fo,F1,F2 variabel Fo,F1,F2, F_0,di XOR kan dengan F_1,dan F_2,di XOR kan dengan F_3, dari hasil XOR tersebut, kemudian dilakukan penjumlahan untuk mencari nilai S_0,dan S_1,. Lalu hasil dari S_0,dan S_1, digabungkan menjadi satu dan menjadi nilai biner variabel P_1,P_2,P_3,P_4,. Setelah itu nilai biner variabel dari P_1,P_2,P_3,P_4, di Xor kan dengan nilai biner variabel X_1,X_2,X_3,X_4, sehingga menghasilkan nilai biner dari variabel Y_1,Y_2,Y_3,Y_4,.

   M_1= 00000000000000000000000000011000
   01101001011100110110111101101101
   00000000000000000000000001101100
   11010101100011110110010000000001
   X_1 = 00000000000000000000000000011000
   X_2 = 01101001011100110110111101101101
   X_3 = 00000000000000000000000001101100
   X_4 = 11010101100011110110010000000001
   F_0 = 00000000
   F_1 = 00000000
   F_2 =00000000
   F_3 = 00011000
   Keterangan : ⊕ adalah simbol untuk XOR

   F_0 ⊕ F_1
   F_0 = 00000000
   F_1 = 00000000
   ⊕
   00000000

   F_2 ⊕ F_3
   F_2= 00000000
   F_3= 00011000
   ⊕
   00011000

   F_0 = A + B + δ mod 256
   S_0 = 00000000
   00011001
   00000000
   +
   00011001

   F_2 = A+ B + δ mod 256
   S_0 = 00011000
   00011001
   00000000
   +
   00000001

   F_1 = A+B + δ = mod 256
   S_1 = 00000000
   00011000
   00000001
   +
   00011001

   F_3 = A +B + δ = mod 256
   S_1 = 00011000
   00110001
   00000001
   +
   00101000

   P_1 = 00011001000110010000000100101000
   X_4 = 11010101100011110110010000000001
   ⊕
   Y_4 = 11001100100101100110010000101000
   F_0 = 01101001
   F_1 = 01110011
   F_2 = 01101111
   F_3 = 01101101
   Keterangan : ⊕ adalah simbol untuk XOR

   F_0 ⊕ F_1
   F_0 = 01101001
   F_1 = 01110011
   ⊕
   00011010
   00111110

   F_2 ⊕ F_3
   F_2= 01101111
   F_3= 01101101
   ⊕
   00000010
   01011011

P_4 = 11101011 00111110010110110101101 1
X_1 = 00000000000000000000000000011000
$\oplus$
Y_1 = 11101011001111100101101101000011
Y_1 = 11101011001111100101101101000011
Y_2 = 00000100000111100110111001000001
Y_3 = 01110000000110010001101100011011
Y_4 = 11001100100101100110010000101000

2. Putaran Kedua

Pada putaran kedua ini proses yang dilakukan adalah mengambil nilai input biner dari file audio. Setelah itu nilai input biner tersebut dibuat dengan variabel M_1. Kemudian dari variabel M_1 tersebut, dibagi lagi menjadi variabel X_1,X_2,X_3,X_4,. Setelah itu variabel X_1,X_2,X_3,X_4, dibagi lagi menjadi Fo,F1,F2 variabel Fo,F1,F2, F_0,di XOR kan dengan F_1,dan F_2,di XOR kan dengan F_3, dari hasil XOR tersebut, kemudian dilakukan penjumlahan untuk mencari nilai S_0,dan S_1,. Lalu hasil dari S_0,dan S_1, digabungkan menjadi satu dan menjadi nilai biner variabel P_1,P_2,P_3,P_4,. Setelah itu nilai biner variabel dari P_1,P_2,P_3,P_4, di Xor kan dengan nilai biner variabel X_1,X_2,X_3,X_4, sehingga menghasilkan nilai biner dari variabel Y_1,Y_2,Y_3,Y_4,.

X_1 = 11101011001111100101101101000011
X_2 = 00000100000111100110111001000001
X_3 = 01110000000110010001101100011011
X_4 = 11001100100101100110010000101000
F_0 = 11101011
F_1 = 00111110
F_2 = 01011011
F_3 = 01000011
Keterangan : $\oplus$ adalah simbol untuk XOR

$F_0 \oplus F_1$: F_0 = 11101011, F_1 = 00111110, $\oplus$ → 11010101
$F_0 = A + B + \delta \bmod 256$
S_0 = 11101011 + 11001101 + 00000000 → 00100110
$F_1 = A+B+\delta = \bmod 256$
S_1 = 11010101 + 00011001 + 00000001 → 11001101 + 01001010
$F_0 = A + B + \delta \bmod 256$
S_0 = 11010101 + 00000111 + 00000000 → 11101011
$F_1 = A+B+\delta = \bmod 256$
S_1 = 01001010 + 01001100 + 00000001 → 00000111
P_4 = 11101011000001110100101101100010
X_1 = 11101011001111100101101101000010
$\oplus$
Y_1 = 00000000001110010001000000100010
Y_1 = 00000000001110010001000000100010

$F_2 \oplus F_3$: F_2 = 01011011, F_3 = 01000010, $\oplus$ → 00011001
$F_2 = A + B + \delta \bmod 256$
S_0 = 00011001 + 11001101 + 00000000 → 11010100
$F_3 = A +B + \delta = \bmod 256$
S_1 =01000011 + 11010100 + 00000001 → 10010110
$F_2 = A + B + \delta \bmod 256$
S_0 = 01001100 + 00000111 + 00000000 → 01001011
$F_3 = A +B + \delta = \bmod 256$
S_1 = 00101000 + 01011011 + 00000001 → 01100010

Y_2 = 00011100011101100000011000110011
Y_3 = 01000000001011010000000001000001
Y_4 =11101010010110111011000010111110

3. Putaran Ketiga

Pada putaran ketiga ini proses yang dilakukan adalah mengambil nilai input biner dari file audio. Setelah itu nilai input biner tersebut dibuat dengan variabel M_1. Kemudian dari variabel M_1 tersebut, dibagi lagi menjadi variabel $X_1,X_2,X_3,X_4,$. Setelah itu variabel $X_1,X_2,X_3,X_4,$ dibagi lagi menjadi Fo,F1,F2 variabel Fo,F1,F2, F_0,di XOR kan dengan F_1,dan F_2,di XOR kan dengan F_3, dari hasil XOR tersebut, kemudian dilakukan penjumlahan untuk mencari nilai S_0,dan S_1,. Lalu hasil dari S_0,dan S_1, digabungkan menjadi satu dan menjadi nilai biner variabel $P_1,P_2,P_3,P_4,$. Setelah itu nilai biner variabel dari $P_1,P_2,P_3,P_4,$ di Xor kan dengan nilai biner variabel $X_1,X_2,X_3,X_4,$ sehingga menghasilkan nilai biner dari variabel $Y_1,Y_2,Y_3,Y_4,$.

X_1 = 00000000001110010001000000100010
X_2 = 00011100011101100000011000110011
X_3 = 01000000001011010000000001000001
X_4 = 11101010010110111011000010111110
F_0 = 00000000
F_1 = 00111001
F_2= 00010000
F_3= 00100010

Keterangan : ⊕ adalah simbol untuk XOR

F_0 ⊕ F_1 F_2 ⊕ F_3

F_0 = 00000000 F_2= 00010000
F_1 = 00111001 F_3= 00100010
00111001 00110010
F_0 = A + B + δ mod 256 F_2 = A+ B + δ mod 256
S_0 = 00000000 S_0 = 00110010
11001100 11001100
00000000 00000000
11001100 11111110
F_1 = A+B + δ = mod 256 F_3 = A +B + δ = mod 256
S_1 = 00111001 S_1 = 00100010
00110010 11111111
00000001 00000001
11001100 11011100

P_1 = 11001101110011011111111111011101
X_4 = 11101010010110111011000010111110
Y_4 = 00100111100101100100111101100011
F_0 =01000000
F_1 =00101101
F_2 =00000000
F_3 =01000001

Keterangan : ⊕ adalah simbol untuk XOR

F_0 ⊕ F_1 F_2 ⊕ F_3

F_0 = 01000000 F_2= 00000000
F_1 = 00101101 F_3= 01000001
01101101 01000001
F_0 = A + B + δ mod 256 F_2 = A+ B + δ mod 256
S_0 =01000000 S_0 = 01000001
00101101 00101101
00000000 00000000
01101101 01101101
F_1 = A+B + δ = mod 256 F_3 = A +B + δ = mod 256
S_1 = 01101101 S_1 =01000001
01000001 01101101

ISSN 2774-4744 (Media Online)
Vol 2, No 1, Januari 2022
Hal 18−34
https://hostjournals.com/jimat

00000001 + 00101101 = 00000001 + 00101101 =

P_3 = 01101100001011000110110100101100

X_2 = 00011100011101100000011000110011 ⊕

Y_2 = 01110000010110100110101100011111

F_0 =11001010

F_1 =01011011

F_2 =10110000

F_3 =10111111

Keterangan : ⊕ adalah simbol untuk XOR

F_0 ⊕ F_1 : F_0 = 11001010, F_1 = 01011011 ⊕ = 10010001

F_2 ⊕ F_3 : F_2 = 10110000, F_3 = 10111111 ⊕ = 00001111

F_0 = A + B + δ mod 256 = 11010101

S_0 = 10011111 + 00000000 + 01001010

F_2 = A+ B + δ mod 256 = 00001111

S_0 = 10011111 + 00000000 + 10010000

F_1 = A+B + δ = mod 256 = 10010001

S_1 = 00001111 + 00000001 + 10011111

F_3 = A +B + δ = mod 256 = 10111111

S_1 = 10010000 + 00000001 + 00101110

P_4 =01001010100111111001000000101110

X_1 =00000000001110010001000000100010 ⊕

Y_1 =01000110101001101000000000001100

Y_1 = 01000110101001101000000000001100

Y_2 = 01110000010110100110101100011111

Y_3 = 00000010011010110111001100000000

Y_4 = 00100111100101100100111101100011

4. Putaran Keempat

Pada putaran keempat ini proses yang dilakukan adalah mengambil nilai input biner dari file audio. Setelah itu nilai input biner tersebut dibuat dengan variabel M_1. Kemudian dari variabel M_1 tersebut, dibagi lagi menjadi variabel X_1,X_2,X_3,X_4,. Setelah itu variabel X_1,X_2,X_3,X_4, dibagi lagi menjadi Fo,F1,F2 variabel Fo,F1,F2, F_0, di XOR kan dengan F_1, dan F_2, di XOR kan dengan F_3, dari hasil XOR tersebut, kemudian dilakukan penjumlahan untuk mencari nilai S_0, dan S_1,. Lalu hasil dari S_0, dan S_1, digabungkan menjadi satu dan menjadi nilai biner variabel P_1,P_2,P_3,P_4,. Setelah itu nilai biner variabel dari P_1,P_2,P_3,P_4, di Xor kan dengan nilai biner variabel X_1,X_2,X_3,X_4, sehingga menghasilkan nilai biner dari variabel Y_1,Y_2,Y_3,Y_4,.

X_1 = 01000110101001101000000000001100

X_2 = 01110000010110100110101100011111

X_3 = 00000010011010110111001100000000

X_4 = 00100111100101100100111101100011

F_0 = 01000110

F_1 = 10100110

F_2 = 10000000

F_3 = 00001100

Keterangan : ⊕ adalah simbol untuk XOR

F_0 ⊕ F_1 : F_0 = 01000110, F_1 = 10100110 ⊕ = 11100000

F_2 ⊕ F_3 : F_2 = 10000000, F_3 = 00001100 ⊕ = 10001100

F_0 = A + B + δ mod 256

F_2 = A+ B + δ mod 256

ISSN 2774-4744 (Media Online)
Vol 2, No 1, Januari 2022
Hal 18−34
https://hostjournals.com/jimat

S_0 = 01000111 S_0 = 10001101
01101101 01101101
00000000 00000000
+ +
00101011 11100000

F_1 = A+B + δ = mod 256 F_3 = A +B + δ = mod 256
S_1 = 11100000 S_1 = 00001100
10001100 11100000
00000001 00000001
+ +
01101101 11101101

P_1 = 00101011011011011110000011101101
X_4 = 00100111100101100100111101100011
⊕
Y_4 =00001111111100111010111110001110
F_0 = 01110000
F_1 = 01011010
F_2 = 01101011
F_3 = 00011111
Keterangan : ⊕ adalah simbol untuk XOR

F_0 ⊕ F_1 F_2 ⊕ F_3
F_0 = 01110000 F_2 = 01101011
F_1 = 01011010 F_3 = 00011111
⊕ ⊕
00101010 01110100

F_0 = A + B + δ mod 256 F_2 = A+ B + δ mod 256
S_0 = 01110000 S_0 = 01110100
01011111 01011111
00000000 00000000
+ +
00101111 00101011

F_1 = A+B + δ = mod 256 F_3 = A +B + δ = mod 256
S_1 = 00101010 S_1 = 00011111
01110100 00101011
00000001 00000001
+ +
01011111 00110101

P_2 = 00101111010111110010101100110101
X_3 = 00000010011010110111001100000000
⊕
Y_3 =00101101001101000101100000110101
F_0 = 00000010
F_1 = 01101011
F_2 = 01101011
F_3 = 00000000
Keterangan : ⊕ adalah simbol untuk XOR

F_0 ⊕ F_1 F_2 ⊕ F_3
F_0 = 00000010 F_2 = 01101011
F_1 = 01101011 F_3 = 00000000
⊕ ⊕
01101001 01101011

F_0 = A + B + δ mod 256 F_2 = A+ B + δ mod 256
S_0 = 00000010 S_0 = 01110100
00000011 00000011
00000000 00000000
+ +
00000001 01110111

F_1 = A+B + δ = mod 256 F_3 = A +B + δ = mod 256
S_1 = 01101001 S_1 = 00000000
01101011 01110111

ISSN 2774-4744 (Media Online)
Vol 2, No 1, Januari 2022
Hal 18−34
https://hostjournals.com/jimat

00000001 00000001
+ +
00000011 01110110

P_3 = 00000001000000110111011101110110
X_2 = 01110000010110100110101100011111
⊕
Y_2 =01110001010110010001110001101001
F_0 = 00100111
F_1 = 10010110
F_2 = 01001111
F_3 = 01100011
Keterangan : ⊕ adalah simbol untuk XOR

F_0 ⊕ F_1 F_2 ⊕ F_3
F_0 = 00100111 F_2 = 01001111
F_1 = 10010110 F_3 = 01100011
⊕ ⊕
10110001 00101100

F_0 = A + B + δ mod 256 F_2 = A+ B + δ mod 256
S_0 = 00100111 S_0 = 01110100
10011100 10011100
00000000 00000000
+ +
10111011 11101000

F_1 = A+B + δ = mod 256 F_3 = A +B + δ = mod 256
S_1 = 10110001 S_1 = 01100011
00101100 11101000
00000001 00000001
+ +
10011100 10001010

P_4 = 10111011100111001110100010001010
X_1 = 01000110101001101000000000001100
⊕
Y_1 = 11111101001110100110100010000110
Y_1 = 11111101001110100110100010000110
Y_2 = 01110001010110010001110001101001
Y_3 = 00101101001101000101100000110101
Y_4 = 00001111111100111010111110001110

5. Putaran Kelima
   Pada putaran kelima ini proses yang dilakukan adalah mengambil nilai input biner dari file audio. Setelah itu nilai input biner tersebut dibuat dengan variabel M_1. Kemudian dari variabel M_1 tersebut, dibagi lagi menjadi variabel $X_{1,}X_{2,}X_{3,}X_{4,}$. Setelah itu variabel $X_{1,}X_{2,}X_{3,}X_{4,}$ dibagi lagi menjadi Fo,F1,F2 variabel Fo,F1,F2, $F_{0,}$di XOR kan dengan $F_{1,}$dan $F_{2,}$di XOR kan dengan $F_{3,}$ dari hasil XOR tersebut, kemudian dilakukan penjumlahan untuk mencari nilai $S_{0,}$dan $S_{1,}$. Lalu hasil dari $S_{0,}$dan $S_{1,}$ digabungkan menjadi satu dan menjadi nilai biner variabel $P_{1,}P_{2,}P_{3,}P_{4,}$. Setelah itu nilai biner variabel dari $P_{1,}P_{2,}P_{3,}P_{4,}$ di Xor kan dengan nilai biner variabel $X_{1,}X_{2,}X_{3,}X_{4,}$ sehingga menghasilkan nilai biner dari variabel $Y_{1,}Y_{2,}Y_{3,}Y_{4,}$.

X_1 = 11111101001110100110100010000110
X_2 = 01110001010110010001110001101001
X_3 = 00101101001101000101100000110101
X_4 = 00001111111100111010111110001110
F_0 = 11111101
F_1 = 00111010
F_2 = 01101000
F_3 = 10000110
Keterangan : ⊕ adalah simbol untuk XOR

F_0 ⊕ F_1 F_2 ⊕ F_3
F_0 = 11111101 F_2 = 01101000
F_1 = 00111010 F_3 = 10000110
⊕ ⊕
11000111 11101110

ISSN 2774-4744 (Media Online)
Vol 2, No 1, Januari 2022
Hal 18−34
https://hostjournals.com/jimat

F_0 = A + B + δ mod 256 F_2 = A+ B + δ mod 256
S_0 = 11111101 S_0 = 11101110
00101000 00101000
00000000 00000000
11010101 11101000
F_1 = A+B + δ = mod 256 F_3 = A +B + δ = mod 256
S_1 = 11000111 S_1 = 10000110
11101110 11101000
00000001 00000001
00101000 01101111
P_1 = 11010101001010001110100001101111
X_4 = 00001111111100111010111110001110
Y_4 =11011010110110110100011111100001

P_3 = 01011000011101010000000000110100
X_2 = 01110001010110010001110001101001
Y_2 =00101001001011000001110001011101
F_0 = 00001111
F_1 = 11110011
F_2 = 10101111
F_3 = 10001110
Keterangan : ⊕ adalah simbol untuk XOR
F_0 ⊕ F_1 F_2 ⊕ F_3
F_0 = 00001111 F_2 = 10101111
F_1 = 11110011 F_3 = 10001110
11111100 00100001
F_0 = A + B + δ mod 256 F_2 = A+ B + δ mod 256
S_0 = 00001111 S_0 = 00100001
11011100 11011100
00000000 00000000
11010011 11111101
F_1 = A+B + δ = mod 256 F_3 = A +B + δ = mod 256
S_1 = 11111100 S_1 = 10001110
00100001 11111101
00000001 00000001
11011100 01110010
P_4 = 11010011110111001111110101110010
X_1 = 11111101001110100110100010000110
Y_1 = 00101110111001101001010111110100
Y_1 = 00101110111001101001010111110100
Y_2 = 00101001001011000001110001011101
Y_3 = 011011000111000 0000010100000000
Y_4 = 11011010110110110100011111100001

6. Putaran Keenam
   Pada putaran keenam ini proses yang dilakukan adalah mengambil nilai input biner dari file audio. Setelah itu nilai input biner tersebut dibuat dengan variabel M_1. Kemudian dari variabel M_1 tersebut, dibagi lagi menjadi variabel $X_1,X_2,X_3,X_4,$. Setelah itu variabel $X_1,X_2,X_3,X_4,$ dibagi lagi menjadi Fo,F1,F2 variabel Fo,F1,F2, $F_0,$di XOR kan dengan $F_1,$dan $F_2,$di XOR kan dengan $F_3,$ dari hasil XOR tersebut, kemudian dilakukan penjumlahan untuk mencari nilai $S_0,$dan $S_1,$. Lalu hasil dari $S_0,$dan $S_1,$ digabungkan menjadi satu dan menjadi nilai biner variabel $P_1,P_2,P_3,P_4,$. Setelah itu nilai biner variabel dari $P_1,P_2,P_3,P_4,$ di Xor kan dengan nilai biner variabel $X_1,X_2,X_3,X_4,$ sehingga menghasilkan nilai biner dari variabel $Y_1,Y_2,Y_3,Y_4,$.

X_1 =00101110111001101001010111110100

ISSN 2774-4744 (Media Online)
Vol 2, No 1, Januari 2022
Hal 18−34
https://hostjournals.com/jimat

X_2 =00101001001011000001110001011101
X_3 =01101100011100000000010100000000
X_4 =11011010110110110100011111100001
F_0 = 00101110
F_1 = 11100110
F_2 = 10010101
F_3 = 11110100
Keterangan : ⊕ adalah simbol untuk XOR

$F_0 \oplus F_1$ $F_2 \oplus F_3$
F_0 = 00101110 F_2 = 10010101
F_1 = 11100110 F_3 = 11110100
11001000 01100001

F_0 = A + B + δ mod 256 F_2 = A+ B + δ mod 256
S_0 = 00101110 S_0 = 01100001
10101000 10101000
00000000 00000000
10000110 11001001

F_1 = A+B + δ = mod 256 F_3 = A +B + δ = mod 256
S_1 = 11001000 S_1 = 10001110
01100001 11001001
00000001 00000001
10101000 01000110

P_1 = 10000110101010001100100101000110
X_4 = 11011010110110110100011111100001
Y_4 =01011100011100111000111010100111
F_0 = 00101001
F_1 = 00101100
F_2 = 00011100
F_3 = 01011101
Keterangan : ⊕ adalah simbol untuk XOR

$F_0 \oplus F_1$ $F_2 \oplus F_3$
F_0 = 00101001 F_2 = 00011100
F_1 = 00101100 F_3 = 01011101
00000101 11000001

F_0 = A + B + δ mod 256 F_2 = A+ B + δ mod 256
S_0 = 00101001 S_0 = 11000001
11000101 11000101
00000000 00000000
11101100 00000100

F_1 = A+B + δ = mod 256 F_3 = A +B + δ = mod 256
S_1 = 00000101 S_1 = 01011101
11000001 00000100
00000001 00000001
11000101 01011000

P_2 = 11101100110001010000010001011000
X_3 = 01101100011100000000010100000000
Y_3 =10000000101101010000000101011000
F_0 = 01101100
F_1 = 01110000
F_2 = 00000101
F_3 = 00000101
Keterangan : ⊕ adalah simbol untuk XOR

ISSN 2774-4744 (Media Online)
Vol 2, No 1, Januari 2022
Hal 18−34
https://hostjournals.com/jimat

$F_0 \oplus F_1$ $F_2 \oplus F_3$
F_0 = 01101100 F_2 = 00000101
F_1 = 01110000 F_3 = 00000101
00011100 00000000
F_0 = A + B + δ mod 256 F_2 = A+ B + δ mod 256
S_0 = 01101100 S_0 = 00000101
00011101 00011101
00000000 00000000
01110001 00011000
F_1 = A+B + δ = mod 256 F_3 = A +B + δ = mod 256
S_1 = 00011100 S_1 = 01011101
00000000 00011000
00000001 00000001
00011101 01000100
P_3 = 01110001000111010001100001000100
X_2 = 00101001001011000001110001011101
Y_2 =01011000001100010000010000011001
F_0 = 11011010
F_1 = 11011011
F_2 = 01000111
F_3 = 11100001
Keterangan : ⊕ adalah simbol untuk XOR
$F_0 \oplus F_1$ $F_2 \oplus F_3$
F_0 = 11011010 F_2 = 01000111
F_1 = 11011011 F_3 = 11100001
00000001 10100110
F_0 = A + B + δ mod 256 F_2 = A+ B + δ mod 256
S_0 = 11011010 S_0 = 10100110
10100110 10100110
00000000 00000000
01111100 00000000
F_1 = A+B + δ = mod 256 F_3 = A +B + δ = mod 256
S_1 = 00000001 S_1 = 01011101
10100110 00000000
00000001 00000001
10100110 01011100
P_4 = 01111100101001100000000001011100
X_1 = 00101110111001101001010111110100
Y_1 = 01010010010000001001010110101000
Y_1 = 01010010010000001001010110101000
Y_2 = 01011000001100010000010000011001
Y_3 = 10000000101101010000000101011000
Y_4= 01011100011100111000111010100111

7. Putaran Ketujuh
   Pada putaran ketujuh ini proses yang dilakukan adalah mengambil nilai input biner dari file audio. Setelah itu nilai input biner tersebut dibuat dengan variabel M_1. Kemudian dari variabel M_1 tersebut, dibagi lagi menjadi variabel X_1, X_2, X_3, X_4,. Setelah itu variabel X_1, X_2, X_3, X_4, dibagi lagi menjadi Fo,F1,F2 variabel Fo,F1,F2, F_0,di XOR kan dengan F_1,dan F_2,di XOR kan dengan F_3, dari hasil XOR tersebut, kemudian dilakukan penjumlahan untuk mencari nilai S_0,dan S_1,. Lalu hasil dari S_0,dan S_1, digabungkan menjadi satu dan menjadi nilai biner variabel P_1, P_2, P_3, P_4,. Setelah itu nilai biner variabel dari P_1, P_2, P_3, P_4, di Xor kan dengan nilai biner variabel X_1, X_2, X_3, X_4, sehingga menghasilkan nilai biner dari variabel Y_1, Y_2, Y_3, Y_4,.

X_1 = 01010010010000001001010110101000

ISSN 2774-4744 (Media Online)
Vol 2, No 1, Januari 2022
Hal 18−34
https://hostjournals.com/jimat

X_2 = 01011000001100010000010000011001
X_3 = 10000000101101010000000101011000
X_4 = 01011100011100111000111010100111
F_0 = 01010010
F_1 = 01000000
F_2 = 10010101
F_3 = 10101000
Keterangan : ⊕ adalah simbol untuk XOR

$F_0 \oplus F_1$ $F_2 \oplus F_3$
F_0 = 01010010 F_2 = 10010101
F_1 = 01000000 F_3 = 10101000
⊕ ⊕
00010010 00111001
F_0 = A + B + δ mod 256 F_2 = A+ B + δ mod 256
S_0 = 01010010 S_0 = 00111001
00101010 00101010
00000000 00000000
+ +
01111000 00010011
F_1 = A+B + δ = mod 256 F_3 = A +B + δ = mod 256
S_1 = 00010010 S_1 = 10101000
00111001 00010011
00000001 00000001
+ +
00101010 10111010
P_1 = 01111000001010100001001110111010
X_4= 01011100011100111000111010100111
⊕
Y_4 = 00100100010110011001110100011101
F_0 = 01011000
F_1 = 00110001
F_2 = 00000100
F_3 = 00011001

$F_0 \oplus F_1$ $F_2 \oplus F_3$
F_0 = 01011000 F_2 = 00000100
F_1 = 00110001 F_3 = 00011001
⊕ ⊕
01101001 00011101
F_0 = A + B + δ mod 256 F_2 = A+ B + δ mod 256
S_0 = 01011000 S_0 = 00011101
01110101 01110101
00000000 00000000
+ +
00101101 01101000
F_1 = A+B + δ = mod 256 F_3 = A +B + δ = mod 256
S_1 = 01101001 S_1 = 00011001
00011101 01101000
00000001 00000001
+ +
01110101 01110000
P_2 = 00101101011101010110100001110000
X_3= 10000000101101010000000101011000
⊕
Y_3 =10101101110000000110100100101000
F_0 = 10000000
F_1 = 10110101
F_2 = 00000001
F_3 = 01011000
Keterangan : ⊕ adalah simbol untuk XOR
$F_0 \oplus F_1$ $F_2 \oplus F_3$

ISSN 2774-4744 (Media Online)
Vol 2, No 1, Januari 2022
Hal 18−34
https://hostjournals.com/jimat

F_0 = 10000000 F_2 = 00000001
F_1 = 10110101 F_3 = 01011000
00110101 01011001
F_0 = A + B + δ mod 256 F_2 = A+ B + δ mod 256
S_0 = 10000000 S_0 = 11110000
01101101 01101101
00000000 00000000
11101101 10011101
F_1 = A+B + δ = mod 256 F_3 = A +B + δ = mod 256
S_1 = 00110101 S_1 = 01011000
01011001 10011101
00000001 00000001
01101101 11000100
P_3 = 11101101011011011001110111000100
X_2= 01011000001100010000010000011001
Y_2 = 10110101010111001001100111011101
F_0 = 01011100
F_1 = 01110011
F_2 = 10001110
F_3 = 10100111
Keterangan : ⊕ adalah simbol untuk XOR
F_0 ⊕ F_1 F_2 ⊕ F_3
F_0 = 01011100 F_2 = 10001110
F_1 = 01110011 F_3 = 10100111
00101111 00101001
F_0 = A + B + δ mod 256 F_2 = A+ B + δ mod 256
S_0 = 01011100 S_0 = 01000101
00000111 00000111
00000000 00000000
01011011 01000010
F_1 = A+B + δ = mod 256 F_3 = A +B + δ = mod 256
S_1 = 00101111 S_1 = 00100011
00101001 01000010
00000001 00000001
00000111 01100000
P_4 = 01011011000001110100001001100000
X_1= 01010010010000001001010110101000
Y_1 = 00001001010001111101011111001000
Y_1 = 00001001010001111101011111001000
Y_2 = 10110101010111001001100111011101
Y_3 = 10101101110000000110100100101000
Y_4 = 00100100010110011001110100011101

8. Putaran Kedelapan
   Pada putaran kedelapan ini proses yang dilakukan adalah mengambil nilai input biner dari file audio. Setelah itu nilai input biner tersebut dibuat dengan variabel M_1. Kemudian dari variabel M_1 tersebut, dibagi lagi menjadi variabel $X_1,X_2,X_3,X_4,$. Setelah itu variabel $X_1,X_2,X_3,X_4,$ dibagi lagi menjadi Fo,F1,F2 variabel Fo,F1,F2, $F_0,$di XOR kan dengan $F_1,$dan $F_2,$di XOR kan dengan $F_3,$ dari hasil XOR tersebut, kemudian dilakukan penjumlahan untuk mencari nilai $S_0,$dan $S_1,$. Lalu hasil dari $S_0,$dan $S_1,$ digabungkan menjadi satu dan menjadi nilai biner variabel $P_1,P_2,P_3,P_4,$. Setelah itu nilai biner variabel dari $P_1,P_2,P_3,P_4,$ di Xor kan dengan nilai biner variabel $X_1,X_2,X_3,X_4,$ sehingga menghasilkan nilai biner dari variabel $Y_1,Y_2,Y_3,Y_4,$. sehingga menghasilkan nilai biner dari variabel $Y_1,Y_2,Y_3,Y_4,$. Dari hasil nilai biner variabel $Y_1,Y_2,Y_3,Y_4,$ pada putaran kedelapan ini kemudian di XOR kan dengan nilai biner variabel $M_1,$ sehingga menghasilkan nilai $H_1,$atau nilai hash value nya.

ISSN 2774-4744 (Media Online)
Vol 2, No 1, Januari 2022
Hal 18−34
https://hostjournals.com/jimat

X_1 = 00001001010001111101011111001000
X_2 = 10110101010111001001100111011101
X_3 = 10101101110000000110100100101000
X_4 = 00100100010110011001110100011101
F_0 = 00001001
F_1 = 01000110
F_2 = 10000101
F_3 = 10000001
Keterangan : ⊕ adalah simbol untuk XOR

$F_0 \oplus F_1$ $F_2 \oplus F_3$
F_0 = 00001001 F_2 = 10000101
F_1 = 01000110 F_3 = 10000001
⊕ ⊕
01001111 00000100
F_0 = A + B + δ mod 256 F_2 = A+ B + δ mod 256
S_0 = 00001001 S_0 = 00000100
01001010 01001010
00000000 00000000
+ +
01000011 01001110
F_1 = A+B + δ = mod 256 F_3 = A +B + δ = mod 256
S_1 = 01001111 S_1 = 10000001
00000100 01001110
00000001 00000001
+ +
01001010 11001110
P_1 = 01000011010010100100111011001110
X_4= 00100100010110011001110100011101
⊕
Y_4 = 01100111000100111101001111010011
F_0 = 10110101
F_1 = 01011100
F_2 = 10011001
F_3 = 11011101
Keterangan : ⊕ adalah simbol untuk XOR

$F_0 \oplus F_1$ $F_2 \oplus F_3$
F_0 = 10110101 F_2 = 10011001
F_1 = 01011100 F_3 = 11011101
⊕ ⊕
11101001 01000100
F_0 = A + B + δ mod 256 F_2 = A+ B + δ mod 256
S_0 = 10110101 S_0 = 01000100
10101100 10101100
00000000 00000000
+ +
00011001 11101000
F_1 = A+B + δ = mod 256 F_3 = A +B + δ = mod 256
S_1 = 11101001 S_1 = 11011101
01000100 11101000
00000001 00000001
+ +
10101100 00110100
P_2 = 00011001101011001110100000110100
X_3 = 10101101110000000110100100101000
⊕
Y_3 = 10110100011011001000000100011100
F_0 = 10101101
F_1 = 11000000
F_2 = 01101001
F_3 = 00101000

Keterangan : ⊕ adalah simbol untuk XOR

$F_0 \oplus F_1$ $F_2 \oplus F_3$

F_0 = 10101101 F_2 = 01101001

F_1 = 11000000 F_3 = 00101000

01101101 01000001

F_0 = A + B + δ mod 256 F_2 = A+ B + δ mod 256

S_0 = 10101101 S_0 = 10010111

00101101 00101101

00000000 00000000

10000000 10111010

F_1 = A+B + δ = mod 256 F_3 = A +B + δ = mod 256

S_1 = 01101101 S_1 =00101000

01000001 10111010

00000001 00000001

00101101 10010011

P_3 = 10000000001011011011101010010011

X_2 = 10110101010111001001100111011101

Y_2 = 00110101011100010010001101001110

F_0 = 00100100

F_1 = 01011001

F_2 = 10011101

F_3 = 00011101

Keterangan : ⊕ adalah simbol untuk XOR

$F_0 \oplus F_1$ $F_2 \oplus F_3$

F_0 = 00100100 F_2 = 10011101

F_1 = 01011001 F_3 = 00011101

01111101 10000000

F_0 = A + B + δ mod 256 F_2 = A+ B + δ mod 256

S_0 = 00100100 S_0 = 01100110

11111100 11111100

00000000 00000000

11001000 10011010

F_1 = A+B + δ = mod 256 F_3 = A +B + δ = mod 256

S_1 = 01111101 S_1 = 00011101

10000000 10011010

00000001 00000001

11111100 10000110

P_4 = 11001000111111001001101010000110

X_1 = 01010010010000001001010110101000

Y_1 = 10011010101111000000111100101110

Y_1 = 10011010101111000000111100101110

Y_2 = 00110101011100010010001101001110

Y_3 = 10110100011011001000000100011100

Y_4 = 01100111000100111101001111010011

M_1= 00000000000000000000000000011000

Y_1 = 10011010101111000000111100101110

H_1= 10011010101111000000111100110110

M_1= 01101001011100110110111101101101

Y_2 = 00110101011100010010001101001110

H_1 = 01011100000000100100110000010011

Journal of Informatics Management and Information Technology
ISSN 2774-4744 (Media Online)
Vol 2, No 1, Januari 2022
Hal 18−34
https://hostjournals.com/jimat

M_1 = 00000000000000000000000001101100
Y_3 = 10110100011011001000000100011100
$\oplus$
H_1 = 10110100011011001000000101110000
M_1 = 11010101100011110110010000000001
Y_4 = 01100111000100111101001111010011
$\oplus$
H_1 = 10110010100111001011010011010010
H_1 = 10011010 10111100 00001111 00110110
01011100 00000010 01001100 00010011
10110100 01101100 10000001 01110000
10110010 10011100 10110100 11010010

Berdasarkan perhitungan yang telah dilakukan nilai sebelumnya 00000018 69736F6D 00 00006C D58F6401 maka di dapatkan nilai *hash* 9ABC0F36 5C024C13 B46C8170 B29CB4D2. Pada pengujian ini menggunakan beberapa parameter yaitu meningkatkan nada bass pada file audio, meningkatkan nada trable pada file audio, meninggikan vocal pada file audio, hasil pengujiannya dapat dilihat pada tabel dibawah ini.

Tabel 1. Hasil pengujian pada file audio menggunakan aplikasi matlab

Prameter	File audio asli	File audio modif	Nilai hash awal	Nilai hash modif	Hasil
Meningkatkan nada bass			9ABC0F 365C024 C13B46 C8170B2 9CB4D2	170B29 CB4D2 624C13 BBA42 ABCD 442276	Dari hasil perbandingan nilai hash yang sudah di modifikasi hasilnya berbeda, maka file audio hasilnya berbeda, maka file audio dapat diteksi.
Meningkatkan nada trable			9ABC0F 365C024 C13B46 C8170B2 9CB4D2	AA207 E663D 681292 207E66 DD432 0215A	Dari hasil perbandingan nilai hash yang sudah di modifikasi hasilnya berbeda, maka file audio hasilnya berbeda, maka file audio dapat diteksi.
Meninggikan vocal			9ABC0F 365C024 C13B46 C8170B2 9CB4D2	48FD1 436812 922EA DFA20 7EE21 34580 A	Dari hasil perbandingan nilai hash yang sudah di modifikasi hasilnya berbeda, maka file audio hasilnya berbeda, maka file audio dapat diteksi.

Berdasarkan hasil pengujian yang telah dilakukan algoritma n-hash berhasil 100% mendeteksi perubahan yang terjadi pada file audio, sehingga dapat dilakukan mana file yang asli dan mana file yang telah dimodifikasi

4. KESIMPULAN

Adapun kesimpulan yang diperoleh dari penelitian proses mendeteksi file audio dilakukan menggunakan metode n-hash dan telah berhasil melakukan proses mendeteksi keaslian *file audio* yang berformat mp3 dan berjalan sesuai dengan teknik mendeteksinya keasliannya. Penerapan metode n-hash telah berhasil membedakan *file audio* asli dengan *file audio* yang dimodifikasi yaitu dengan membandingkan nilai hash yang dihasilkan dari *file audio* tersebut. Mendeteksi *file audio* dengan metode n-hash menggunakan matlab dapat dilakukan dengan membandingkan nilai hash antara *file audio* dengan *file audio* yang dimodifikasi. Apabila nilai hash berbeda dari nilai hash *file audio* asli maka dapat diperoleh keputusan bahwa file tersebut sudah dimodifikasi. Algoritma n-hash dapat medeteksi perubahan yang terjadi pada file audio dengan tingkat keberhasilan 100%.

REFERENCES

[1] S. miyaguchi, k.ohta, M. Iwata, *128-bit hash Function (N-Hash)*, proceedings of SECURICOM 90, pp. 123-137, March 1990.
[2] Rinaldi Munir, *Kriptografi*, Bandung, 2006.
[3] Dony Ariyus, *Pengantar Ilmu Kriptografi Teori Analisis dan Impelemetasi*, FI. Sigit Suyantoro, Ed. Yogyakarta, Indonesia: Andi, 2008.

ISSN 2774-4744 (Media Online)
Vol 2, No 1, Januari 2022
Hal 18−34
https://hostjournals.com/jimat

[4] Harun Mukhtar, *Kriptografi untuk Keamanan Data*. Yogyakarta: Deepublish, 2018.
[5] Stinson, D. R. *Cryptografi Theory and Practice*, (2006).
[6] Eli Biham, Adi samir, *Differential Cryptanalysis of Des-like Cryptosystems*, accepted by the journal of Crytology,1990