Implementasi Metode Kriptografi Advanced Encryption Standard (AES) untuk Proteksi Pesan Audio

[1)]Marto Sihombing
STMIK Kaputama, Jl. Veteran No. 4A-9A, Binjai, Sumatera Utara, Indonesia
E-Mail : martosihombing@yahoo.com

[2)]Juliana Naftali Sitompul
STMIK Kaputama, Jl. Veteran No. 4A-9A, Binjai, Sumatera Utara, Indonesia
E-Mail : joellyanna07@gmail.com

Tia Anggia Putri
STMIK Kaputama, Jl. Veteran No. 4A-9A, Binjai, Sumatera Utara, Indonesia
E-Mail : jtia.anggia@gmail.com

ABSTRACT
This study is to solve the problem Cryptographic techniques used to protect and maintain the confidentiality of audio messages by changing audio messages into audio messages that cannot be played by doing the encryption process and can be played back after the audio message is decrypted. To protect audio messages, it is used in the process of encrypting and decrypting audio messages using the Advanced Encryption Standard (AES) method. The Advanced Encryption Standard (AES) uses symmetric key block passwords with varying key sizes of 128 bits, 192 bits, and 256 bits. The Advanced Encryption Standard (AES) method used is 128 bit key size and there are 10 rounds in the process of encryption and decryption. Audio messages that have been encrypted using the Advanced Encryption Standard (AES) method cannot be played by parties who do not know the key to encrypt and decrypt the audio message. Audio messages are protected and protected from irresponsible parties who want to damage or just listen to the audio message.

Keywords: AES, Endcription, and Decryption

PENDAHULUAN

Kemajuan di bidang teknologi informasi telah memungkinkan berbagai pihak untuk saling bertukar pesandari jarak jauh. Pesan tersebut dapat berupa pesan teks, pesan gambar, pesan video, dan pesan audio. Sebagian orang lebih memilih untuk mengirim pesan audio karena lebih mempersingkat waktu dibandingkan harus mengetik pesan yang ingin dikirimkan. Pesan audio yang dikirim dari satu orang ke orang lain sering terkendala dengan permasalahan kerahasiaan. Apalagi jika pesan audio tersebut merupakan pesan yang rahasia, sehingga tidak boleh sembarangan orang mengetahui isi dari pesan audio tersebut. Agar pesan audio lebih aman maka diperlukan suatu cara untuk memproteksi pesan audio tersebut [1]. Salah satu cara yang digunakan untuk memproteksi pesan audio agar lebih aman adalah menggunakan teknik kriptografi.Teknik kriptografi yang digunakan untuk memproteksi dan menjaga kerahasiaan pesan audio adalah dengan Metode *Advanced Encryption Standard* (AES). Pesan audio yang telah di enkripsi menggunakan metode AES mendukung perkembangan zaman yang semakin canggih dan pemilihan algoritma harus yang tepat salah satunya adalah algoritma AES kerena cukup sulit dipecahkan [2].

DASAR TEORI

2. 1. Pengertian Implementasi

Implementasi sebagai pelaksanaan atau penerapan. Implementasi adalah suatu tindakan atau pelaksanaan dari sebuah rencana yang sudah disusun secara matang dan terperinci. Implementasi biasanya dilakukan setelah perencanaan sudah dianggap sempurna [3].

2. 2. Kriptografi

Kriptografi berasal dari bahasa Yunani, *crypto* dan *graphia*. *Crypto* berarti *secret* (rahasia) dan *graphia* berarti *writing* (tulisan). Menurut terminologinya, Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan atau data dikirim dari suatu tempat ke tempat yang lain [4], [5].

Menurut penelitian tentang Mengenal Proses Perhitungan Enkripsi Menggunakan

Algoritma Kriptografi *Advanced Encryption Standard* (AES) [3] diperoleh tujuan utama dari kriptografi yang merupakan aspek kemanan sistem informasi yaitu:

1. Kerahasiaan.
2. Integritas Data.
3. *Autentifikasi*.
4. *Non Repudiasi*.

2.3. *Advanced Encryption Standard* (AES)

Advanced Encryption Standard (AES) dipublikasikan oleh NIST (*National Institute of Standard and Technology*) pada tahun 2001.

Menurut penelitian Tentang Perancangan Pengamanan Data Menggunakan Algoritma *AES* diperoleh pengertian AES merupakan algoritma *cryptographic* yang dapat digunakan untuk mengamankan data. Algoritma AES adalah blok *chipertext* simetrik yang dapat mengenkripsi dan dekripsi informasi. Enkripsi merubah data yang tidak dapat lagi dibaca disebut *ciphertext*, sebaliknya dekripsi adalah merubah *ciphertext* data menjadi bentuk semula yang dikenal sebagai *plaintext* [2][6] [7].

Proses yang dilakukan pada setiap rondenya identik sama (dari ronde ke-1 sampai dengan ronde ke Nr-1) kecuali untuk ronde Nr. Proses yang identik tersebut terdiri atas *SubBytes, ShiftRows, MixColumns dan AddRoundKey*. Sedangkan pada ronde Nr, proses *MixColumns* tidak dilakukan. Tiap ronde memilki *roundkey* yang dihasilkan dari ekspansi dari kunci utama.

Proses enkripsi yang dilakukan menggunakan algoritma AES yaitu:

1. ***AddRoundKey***
   Proses ini dilakukan di awal ronde dengan melakukan operasi XOR tiap *byte* pada matriks *state*(*plaintext*)dengan tiap *byte* pada *cipherkey*, tahap ini disebut juga *initial round*.
2. ***SubBytes***
   SubBytes merupakan transformasi *byte* dimana setiap elemen pada *state* akan dipetakan dengan menggunakan sebuah tabel substitusi (S-Box). Untuk setiap *byte* pada *state* dinyatakan dengan *S*'[*r*, *c*]. *S*'[*r*, *c*] adalah elemen di dalam tabel subtitusi yang merupakan perpotongan baris (x)dengan kolom (y).
   Shiftrows
   Shiftrows pada dasarnya adalah proses pergeseran *bit* dimana *bit* paling kiri akan dipindahkan menjadi *bit* paling kanan (rotasi *bit*). Namun jumlah pergeseran yang dilakukan berbeda, tergantung untuk setiap barisnya. Baris pertama tidak terjadi pergeseran. Setiap *byte* dari baris kedua pada matriks *state* digeser satu *byte* ke kiri. Selanjutnya baris ketiga digeser ke kiri sebanyak dua *byte* dan pada baris keempat digeser ke kiri sebanyak tiga byte. Proses ini bertujuan untuk menghasilkan *diffusion* yakni dengan menyebarkan pengaruh transformasi nonlinear pada baris-baris matriks *state* untuk putaran selanjutnya.
3. ***MixColumns***
   Pada proses *MixColumns*, tiap kolom dari matriks *state* dilakukan operasi perkalian. Hal ini bertujuan untuk menyebarkan pengaruh setiap bit *plaintext* dan *cipherkey* terhadap *ciphertext* yang dihasilkan, pada arah kolom matriks *state.* Setiap kolom matriks *state* diperlakukan sebagai polinomial empat suku dalam *Galois field,* kemudian dikalikan dengan modulo $(X^8 + X^4 + X^3 + X + 1)$. Operasi *MixColumns* juga dapat dipandang sebagai perkalian matriks, dengan mengalikan empat bilangan di dalam *Galois field*
   MixColumns juga disebut sebagai proses mengalikan setiap kolom dengan matriks berikut: $\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}$
4. ***AddRoundKey***
   Dalam tahap *AddRoundKey* ini, *cipherkey* yang telah ada di ekspansikan terlebih dahulu maka akan di dapat *roundkey* yang akan digunakan untuk proses selanjutnya. Kemudian setiap *byte* dari matriks *state* keluaran proses *MixColumns* dilakukan operasi XOR dengan setiap *byte* dari *roundkey*.
   Proses *round* atau proses *SubBytes*, *ShiftRows*, *MixColumns*, dan *AddRoundKey* dilakukan hingga putaran ke-n dengan cara yang sama. Sedangkan utuk putaran terakhir atau disebut juga *final round* proses *SubBytes*, *ShiftRows,* dan *AddRoundKey* tetap dilakukan tetapi proses *MixColumns* tidak dilakukan.

Proses dekripsi yang dilakukan menggunakan algoritma *Advanced Encryption Standard* (AES) yaitu:

1. ***AddRoundKey***
   Inverse atau kebalikan dari tahap *AddRoundKey* adalah operasi XOR antara byte-byte matriks *state* yangdisusun dari *ciphertext* dengan byte-byte *roundkey* yang dibangkitkan

sebelumnya. *Roundkey* yang digunakandi setiap iterasinya berkebalikan dengan *roundkey* yang ada pada proses enkripsi. *Inverse* dari transformasiini digunakan untuk proses dekripsi.

2. ***Inverse SubBytes***
   Inverse SubBytes juga merupakan transformasi *bytes* yang berkebalikan dengan transformasi *SubBytes*. Pada *InvSubBytes*, tiap elemen pada state dipetakan dengan menggunakan tabel *Inverse S-Box.*
3. ***InverseShiftRow***
   Untuk proses dekripsinya dilakukan proses *Inverse* dari transformasi *ShiftRows. InverseShiftRow s*adalah transformasi byte yang berkebalikan dengan transformasi *ShiftRows*. Pada transformasi *InverseShiftRows*, dilakukan pergeseran *bit* ke kanan sedangkan pada *ShiftRows* dilakukan pergeseran *bit* ke kiri.
4. ***Inverse MixColumns***
   Untuk melakukan dekripsi pesan, dilakukan *inverse* dari transformasi *MixColumns* yakni mengalikan setiap kolom hasil dari *Inverse AddRoundKey* dengan matriks berikut:

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix}$$

2.4. Kunci Ekspansi

Kunci ini biasa disebut *key schedule* dilakukan untuk mendapatkan kunci ronde atau *round key* yang akan digunakan untuk proses enkripsi dan dekripsi. Proses ini terdiri dari beberapa operasi yaitu :

1. *RotWord* atau *rotate* : operasi perputaran
2. *SubBytes atau SubRound* : mensubtitusikan dengan tabel S-Box
3. Operasi *Rcon* : operasi ini diterjemahkan sebagai operasi pangkat 2 nilai tertentu dari user. Operasi ini menggunakan nilai-nilai dalam *Galois Field.* Nilai-nilai dari *rcon* akan di xor kan dengan hasil operasi *SubBytes.*

2.5. Simulasi Proses Perhitungan AES

PlainText: 32 43 f6 a8 | 88 5a 30 8d | 31 31 98 a2 | - - - -
Karena kurang dari 16 karakter maka ditambah dengan 04 04 04 04 sehingga menjadi
PlainText: 32 43 f6 a8 | 88 5a 30 8d | 31 31 98 a2 | 04 04 04 04
Chiper Key : 2b 7e 15 16 | 28 ae d2 a6 | - - - - | - - - -
Karena kurang dari 16 karakter maka ditambah dengan 08 08 08 08 | 08 08 08 08sehingga menjadi
ChiperKey: 2b 7e 15 16 | 28 ae d2 a6 | 08 08 08 08 | 08 08 08 08
Masukkan Ke Kolom 4 X 4
Plainteks :

32	88	31	04
43	5a	31	04
f6	30	98	04
a8	8d	a2	04

Chiperkey :

2b	28	08	08
7e	ae	08	08
15	d2	08	08
16	a6	08	08

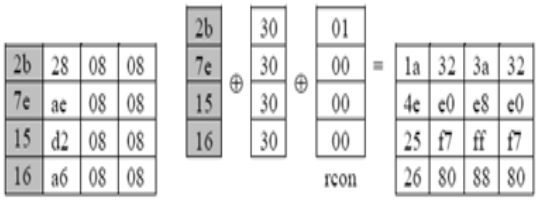
Konversikan Ke Biner
Plainteks, Chiperkey.

Kitahitung dulu *key schedule*, hasil dari perhitungan akan digunakan pada proses selanjutnya untuk proses enkripsi. Langkah-langkah ekspansi kunci (*key* schedule) sebagai berikut:

1. Kolom *cipherkey* paling kanan dirotasi dan dikonversi menggunakan tabel S-Box.
2. Kemudian hasilnya akan di xor kan dengan kolom *cipherkey* paling kiri dan di xor kan lagi dengan tabel rcon pada kolom 1.
3. Lakukan proses ini sampai semua kolom dihitung, sehingga akan diperoleh *roundkey* ke-n.

Contoh:

2b	28	08	08
7e	ae	08	08
15	d2	08	08
16	a6	08	08



Enkripsi

AddRoundKey atau juga bisa disebut sebagai *initial round*
Plainteks:

32	88	31	04
43	5a	31	04
f6	30	98	04
a8	8d	a2	04

Chiperkey :

2b	28	08	08
7e	ae	08	08
15	d2	08	08
16	a6	08	08

32 XOR 2b = 00110010 XOR 00101011 = 00011001
Lakukan dengan cara yang sama sampai semua terhitung, maka diperoleh:
Konversi Ke Hexadesimal menjadi 19 3d e3 be | a0 f4 e2 2b | 39 39 90 aa | 0c 0c 0c 0c

Round 1
State : 19 3d e3 be | a0 f4 e2 2b | 39 39 90 aa | 0c 0c 0c 0c
Tahap selanjutnya adalah *SubBytes*diketahui *byte* pertama 19, y = 1 dan x = 9,Begitu seterusnya sampai matriks tersubtitusi
After SubBytes : d4 27 11 ae | e0 bf 98 f1 | 12 12 60 ac | fe fe fe fe
Tahap selanjutnya adalah *ShiftRows*
After ShiftRows : d4 bf 60 fe | e0 12 fe ae | 12 fe 11 f1 | fe 27 98 ac
Tahap selanjutnya adalah *MixColumns*
After MixColumns : f7 ef b2 5f | bd 73 fc 90 | dd 37 c6 20 | ba af 1d e5, Di xor kan dengan*Round Key* : 1a 4e 25 26 | 32 e0 f7 80 | 3a e8 ff 88 | 32 e0 f7 80
Maka hasilnya adalah
After AddRoundKey : ed a1 97 79 | 8f 93 0b 10 | e7 df 39 a8 | 88 4f ea 65

Proses seperti pada *round* 1 dilakukan sampai *round* 9 dan pada *final round* proses *MixColumns* tidak dilakukan.

Round 9
After SubBytes : c6 86 ea 84 | a8 38 b9 5b | bb 8f f4 19 | 03 55 58 50
After ShiftRows : c6 38 f4 50 | a8 8f 58 84 | bb 55 ea 5b | 03 86 b9 19
After MixColumns : 7b e1 fd 3d | 1d c1 00 27 | 23 6f cc df | 37 dd c7 08
Round Key 9 : 94 9b ba 31 | 89 a3 8c 3d | 36 a0 42 82 | 41 a0 0a 11
After AddRoundKey : ef 7a 47 0c | 94 62 8c 1a | 15 cf 8e 5d | 76 7d cd 19

Final Round
After SubBytes : df da a0 fe | 22 aa 64 a2 | 59 8a 19 4c | 38 ff bd d4
After ShiftRows : df aa 19 d4 | 22 8a bd fe | 59 ff a0 a2 | 38 da 64 4c
After AddRoundKey : 9d 56 21 66 | e9 d5 09 71 | a4 00 56 af | 84 85 98 50
Round Key 10 : 42 fc 38 b2 | cb 5f b4 8f | fd ff f6 0d | bc 5f fc 1c

Hasil *CipherText* adalah 9d 56 21 66 | e9 d5 09 71 | a4 00 56 af | 84 85 98 50

Deskripsi
Block : 9d 56 21 66 | e9 d5 09 71 | a4 00 56 af | 84 85 98 50
Key : 2b 7e 15 16 | 28 ae d2 a6 | 08 08 08 08 | 08 08 08 08

Key Schedule
Round Key 1 : 7b e1 fd 3d | 1d c1 00 27 | 23 6f cc df | 37 dd c7 08
Round Key 2 : 2b 15 dd 8b | 72 4e ed 5b | 4170 74 31 | a2 ed 16 ff, *Round Key* 3 : 9e 3a be 10 27 44 7d 7a 75 10 69 30 81 dc 55 c4, *Round Key* 4 : d7 1a 74 5e 85 e3 f7 c5 ec 43 9a 56 96 11 9a b5, *Round Key* 5 : e7 4a 55 f7 e8 52 8e 83 d8 1a df 8f 7b 3a c7 4c, *Round Key* 6 : bcb7 77 de 4e 57 09 0c da 14 25 2e f1 ae 3f d0, *Round Key* 7 : 33 ba 58 fe 57 a9 c5 bf 05 ae 5e 87 51 bc bc be, *Round Key* 8 : 8a 8d 7a ab 6e 70 39 fb e6 ce 5e 24 2c1f fd d1, *Round Key* 9 : f7 ef b2 5f bd 73 fc 90 dd 37 c6 20 ba af 1d e5

Initial Round atau *AddRoundKey*
Round Key 10 : 42 fc 38 b2 | cb 5f b4 8f | fd ff f6 0d | bc 5f fc 1c di xor kan dengan hasil enkripsi: 9d 56 21 66 | e9 d5 09 71 | a4 00 56 af | 84 85 98 50 maka
After AddRoundKey : df aa 19 d4 | 22 8a bd fe | 59 ff a0 a2 | 38 da 64 4c
Round 1
Tahap selanjutnya *InvShiftRow* yaitu terjadi pergeseran baris ke kanan pada baris ke 2, 3, dan 4.
After InvShiftRow : df da a0 fe | 22 aa 64 a2 | 59 8a 19 4c | 38 ff bd d4

Tahap selanjutnya *InvSubByte* yaitu subtitusi menggunakan tabel *InvSubByte*
After InvSubByte : ef 7a 47 0c | 94 62 8c 1a | 15 cf 8e 5d | 76 7d cd 19

Tahap selanjutnya *InvMixColumn* yaitu perkalian state dengan matriks yang telah ditentukan
After InvMixColumn : 94 9b ba 31 | 89 a3 8c 3d | 36 a0 42 82 | 41 a0 0a 11
Tahap selanjutnya *AddRoundKey* yaitu hasil *InvMixColumn* di xor kan dengan *RoundKey*
After AddRoundKey : c6 38 f4 50 | a8 8f 58 84 | bb 55 ea 5b | 03 86 b9 19
Proses pada ronde 1 dilakukan sebanyak 9 kali.

Round 2
After InvShiftRow : c6 86 ea 84 | a8 38 b9 5b | bb 8f f4 19 | 03 55 58 50, *After InvSubByte* : c7 dc bb 4f | 6f 76 db 57 | fe 73 ba 8e | d5 ed

5e 6c, *After InvMixColumn* : ec c9 66 c4 | 1d 38 36 0c | bf 03 ce bf | 77 00 48 93, *After AddRoundKey* : a7 d5 81 9b | f6 f1 66 eb | 0e 9e d3 37 | 3b a9 d8 ec
*Demikian seterusnya hingga Round*8 dan,

Round 9
After InvShiftRow : 55 32 88 b6 | 73 dc 2b ca | 94 12 c2 | c4 84 87 4d, *After InvSubByte* : ed a1 97 79 | 8f 93 0b 10 | e7 df 39 a8 | 88 4f ea 65, *After InvMixColumn* : 1a 4e 25 26 | 32 e0 f7 80 | 3a e8 ff 88 | 32 e0 f7 80
After AddRoundKey : d4 bf 60 fe | e0 12 fe ae | 12 fe 11 f1 | fe 27 98 ac

Final Round
After InvShiftRow : d4 27 11 ae | e0 bf 98 f1 |2 12 60 ac | fe fe fe fe, *After InvSubByte* : 19 3d e3 be | a0 f4 e2 2b | 39 39 90 aa | 0c 0c 0c 0c, *After AddRoundKey* : 32 43 f6 a8 | 88 5a 30 8d | 31 31 98 a2 |04 04 04 04,
Berdasarkan perhitungan maka hasil dekripsi yang dihasilkan adalah 32 43 f6 a8 | 88 5a 30 8d | 31 31 98 a2 | 04 04 04 04.

Metode AES menggunakan representasi *byte*, maka pesan audio yang akan dienkripsi harus dikonversi atau diubah terlebih dahulu ke bilangan heksadesimal kemudian diubah lagi ke bilangan biner [3]. Mengubah atau mengkonversi pesan audio ke bilangan heksadesimal untuk melakukan perhitungan manual metodeAES bisa menggunakan bantuan *software binary viewer* dan untuk mengubah bilangan heksadesimal ke bilangan biner bisa menggunakan tabel kode ASCII. Pada sistem yang akan dibuat untuk memproteksi pesan audio, akan terdapat koding untuk mengubah pesan audio tersebut ke bilangan heksadesimal ataupun bilangan biner tanpa menggunakan bantuan dari *software binary viewer/.*

2.6. Proses Perhitungan Manual Dengan Metode AES

Untuk lebih memahami proses enkripsi menggunakan metode AES, maka akan dijelaskan perhitungan manual dari metode AES 128 bit untuk pesan audio menggunakan rekamanan suara dengan nama file pesan.mp3 [2]

Gambar 1. Pesan Audio Yang Akan Di Enkripsi

Sebelum melakukan perhitungan secara manual, file tersebut diubah terlebih dahulu ke dalam bilangan hexadesimal menggunakan *binary viewer*. File pesan.mp3 yang telah diubah ke dalam bilangan hexadesimal.

Dari data tersebut diambil beberapa bilangan hexadesimal untuk dijadikan *plaintext* yaitu 61 06 22 1C 44 70 6E 63 2C 4A 25 6C F2 48 A9 33

Contoh perhitungan metode AES secara manual adalah sebagai berikut:
Plaintext : 61 06 22 1C 44 70 6E 63 2C 4A 25 6C F2 48 A9 33
Kunci : sistem_informasi ➔ diubah ke bilangan hexadesimal menjadi 73 69 73 74 65 6D 5F 69 6E 66 6F 72 6D 61 73 69
Masukkan *plaintext* dan kunci ke kolom 4 x 4, *Plaintex,* Kunci.
Ubah bilangan hexadesimal ke bilangan biner, *Plaintext,* Kunci :

Sebelum melakukan enkripsi hitung ekspansi kunci (*key schedule*) terlebih dahulu.
Round 1:Hasil
Key schedule round 1: 9D E6 8A 48 | F8 8B D5 21 | 96 ED BA 53 | FB 8C C9 3A, *Key schedule round* 2 : FB 3B 0A 47 | 03 B0 DF 66 | 95 5D 65 35 | 6E D1 AC 0F
Key schedule round 3 : C1 AA 7C D8| C2 1A A3 BE| 57 47 C6 8B| 39 96 6A 84, *Key schedule round 4* : 58 A8 23 CA | 9A B2 80 74 | CD F5 46 FF | F4 63 2C 7B, *Key schedule round 5* : B3 D9 02 75 | 29 6B 82 01 | E4 9E C4 FE | DD 08 AE 7A, *Key schedule round 6*: A3 3D D8 B4 | 8A 56 5A B5 | 6E C8 9E 4B | B3 C0 30 31
Key schedule round 7 : 59 39 1F D9 | D3 6F 45 6C | BD A7 DB 27 | 0E 67 EB 16, *Key schedule round 8*: 5C D0 58 72 | 8F BF 1D 1E | 32 18 C6 39 | 3C 7F 2D 2F, *Key schedule round 9*: 95 08 4D 99 | 1A B7 50 87 | 28 AF 96 BE | 14 D0 BB 91, *Key schedule round 10*: D3 E2 CC 63 | C9 55 9C E4 | E1 FA 0A 72 | F5 2A B1 E3

Tahap Enkripsi
AddRoundKey atau juga bisa disebut sebagai *initial round*
Plaintext, Kunci :
61 xor 73 = 01100001 xor 01110011 =00010010, hexa: 12, 06 xor 69 = 00000110 xor 01101001 =01101111,hexa: 6F, 22 xor 73 = 00100010 xor 01110011 =01010001, hexa: 51, 1C xor 74 = 00011100 xor 01110100 =01101000, hexa: 68

Demikian seterusnya sampai :
F2 xor 6D = 11110010 xor 01101101 =10011111, hexa: 9F, 48 xor 61 = 01001000 xor 01100001 =00101001, hexa: 29, A9 xor 73 = 10101001 xor 01110011 =11011010, hexa:

DA, 33 xor 69 = 00110011 xor 01101001 =01011010, hexa: 5A

Hasil *AddRoundKey* dimasukkan ke kolom 4 x 4:

12	21	42	9F
6F	1D	2C	29
51	31	4A	DA
68	0A	1E	5A

Round 1
Hasil *AddRoundKey*

12	21	42	9F
6F	1D	2C	29
51	31	4A	DA
68	0A	1E	5A

Tahap selanjutnya adalah *SubBytes* yaitu mengubah hasil *AddRoundKey* menggunakan tabel S-Box. Hasil *AddRoundKey* yang pertama adalah 1D berartibaris ke 1, kolom d sehingga diperoleh a4.Lakukan dengan cara yang sama sampai semua hasil *AddRoundKey* berubah sesuai dengan tabel S-Box.

Tahap selanjutnya adalah *ShifRows* yaitu hasil *SubBytes* digeser ke kiri mulai dari baris ke 1 dilakukan 0 pergeseran, baris ke 2 dilakukan 1 pergeseran, baris ke 3 dilakukan 2 pergeseran, dan baris ke 4 dilakukan 3 pergeseran. Tahap selanjutnya adalah *MixColumns* yaitu mengalikan hasil *ShifRows* dengan 02, 03 dan seterusnya seperti berikut ini:

02	03	01	01
01	02	03	01
01	01	02	03
03	01	01	02

X

C9	FD	2C	DB
A4	71	A5	A8
D6	57	D1	C7
BE	45	67	72

Baris ke 1 dan kolom ke 1

- **02 x C9** diubah ke bilangan biner 00000010 x 11001001. Bilangan biner tersebut diubah ke bilangan polinomial (X) x $(X^7 + X^6 + X^3 + 1)$, untuk perkalian 02 setiap pangkat dari bilangan polinomial hasil *ShifRows* ditambah 1, $(X^7 + X^6 + X^3 + 1)$ menjadi $(X^8 + X^7 + X^4 + X)$, jika ada pangkat lebih dari X^7 maka di mod kan dengan $(X^8 + X^4 + X^3 + X + 1)$,
  $(X^8 + X^7 + X^4 + X)$ mod $(X^8 + X^4 + X^3 + X + 1)$, setiap pangkat yang sama dihilangkan dan sisanya menjadi hasil.$(X^8 + X^7 + X^4 + X)$ mod $(X^8 + X^4 + X^3 + X + 1) = X^7 + X^3 + 1$ dalam bilangan biner menjadi 10001001
- **03 x A4** diubah ke bilangan biner 00000011 x 10100100. Bilangan biner tersebut diubah ke bilangan polinomial (X + 1) x $(X^7 + X^5 + X^2)$.Untuk perkalian 03 setiap pangkat dari bilangan polinomial hasil *ShifRows* ditambah 1, $(X^8 + X^6 + X^3)$ lalu ditambahkan bilangan polinomial hasil *ShifRows* sebelum setiap pangkat ditambah 1,$(X^8 + X^6 + X^3) + (X^7 + X^5 + X^2)$, setiap pangkat yang sama dihilangkan dan sisanya menjadi hasil. $(X^8 + X^6 + X^3) + (X^7 + X^5 + X^2) = (X^8 + X^7 + X^6 + X^5 + X^3 + X^2)$mod $(X^8 + X^4 + X^3 + X + 1) = X^7 + X^6 + X^5 + X^4 + X^2 + X + 1$, dalam bilangan biner menjadi 1111011
  Demikian seterusnya hingga :
- 01 x BE diubah ke bilangan biner 00000001 x 10111110
  = (1) x $(X^7 + X^5 + X^4 + X^3 + X^2 + X)$
  $= X^7 + X^5 + X^4 + X^3 + X^2 + X$ biner: 10111110
  Setiap hasil perkalian di xor kan seperti berikut:
  11001001 xor 01010011 xor 01100001 xor 10111110 = 01000101, hexa: 45

Baris ke 3 dan kolom ke 1

- 01 x C9 diubah ke bilangan biner 00000001 x 11001001
  = (1) x $(X^7 + X^6 + X^3 + 1)$
  $= X^7 + X^6 + X^3 + 1$ biner 11001001
- 01 x A4 diubah ke bilangan biner 00000001 x 10100100
  = (1) x $(X^7 + X^5 + X^2)$
  $= X^7 + X^5 + X^2$biner : 10100100
- 02 x D6 diubah ke bilangan biner 00000010 x 11010110
  = (X) x $(X^7 + X^6 + X^4 + X^2 + X)$
  $= (X^8 + X^7 + X^5 + X^3 + X^2)mod(X^8 + X^4 + X^3 + X + 1)$
  $=X^7 + X^5 + X^4 + X + 1$ biner: 10110011

Sampai selesai proses, Setiap hasil perkalian di xor kan seperti berikut:
11001001 xor 10100100 xor 10110011 xor 11011001 = 00000111, hexa: 07

Baris ke 4 dan kolom ke 1

- 03 x C9 diubah ke bilangan biner 00000011 x 11001001
  = (X+1) x $(X^7 + X^6 + X^3 + 1)$
  $= (X^8 + X^7 + X^4 + X) + (X^7 + X^6 + X^3 + 1)$
  $= (X^8 + X^6 + X^4 + X^3 + X + 1)$mod $(X^8 + X^4 + X^3 + X + 1)$
  $= X^6$biner: 01000000
- 01 x A4 diubah ke bilangan biner 00000001 x 10100100
  = (1) x $(X^7 + X^5 + X^2)$
  $= X^7 + X^5 + X^2$ biner : 10100100

Dan seterusnya.

Setiap hasil perkalian di xor kan seperti berikut:
01000000 xor 10100100 xor 11010110 xor01100111 = 01010101, hexa: 55

Langkah-langkah diatas diulangi sampai semua hasil *ShifRows* terhitung dan diperoleh hasil *MixColumns* sebagai berikut:

16	60	1A	FB
45	A3	72	B0
07	ED	98	70
55	B0	CE	FD

Hasil *MixColumns* di xor kan dengan *Key schedule round* 1
Setelah di xor maka hasilnya (*AddRoundKey*):
Lakukan seperti proses *Round* 1 sampai *Round* 9, maka akan diperoleh hasil sebagai berikut:
Round 2*SubBytes:*
ShiftRows:MixColumns, Hasil *MixColumns* di xor kan dengan *Key schedule round* 2
Setelah di xor maka hasilnya (*AddRoundKey*), Demikian seterusnya sampai round ke 8.
Round 9
SubBytes:

71	62	06	9b
6d	35	bb	f5
5e	58	f5	1b
81	10	93	4c

ShiftRow, MixColumns, Hasil *MixColumns* di xor kandengan *Key schedule round* 9
Setelah di xor maka hasilnya (*AddRoundKey*):

91 10010001	9E 10011110	9B 10011011	45 01000101
5B 01011011	1C 00011100	AA 10101010	EA 11101010
2C 00101100	27 00100111	E9 11101001	53 01010011
D2 11010010	98 10011000	3F 00111111	2F 00101111

Final Round, SubBytes:
Hasil dari *ShiftRows* di xor kan dengan *Key schedule round* 10, Setelah di xor maka hasilnya (*AddRoundKey*) adalah:
Hasil enkripsi yaitu: 52 7E D2 76| C2 F9 71 51 | F5 7D 7B 34 | 9B 13 7D 96. Pesan audio yang telah di enkripsi tidak akan dapat diputar.

Tahap Dekripsi
Ciphertext: 52 7E D2 76| C2 F9 71 51 | F5 7D 7B 34 | 9B 13 7D 96
Kunci : 73 69 73 74 | 65 6D 5F 69 | 6E 66 6F 72 | 6D 61 73 69
Initial Round atau sering disebut *AddRoundKey, Ciphertext* xor *Key schedule round* 10, Hasil *AddRoundKey* adalah:

81 10000001	0B 00001011	14 00010100	6E 01101110
9C 10011100	AC 10101100	87 10000111	39 00111001
1E 00011110	ED 11101101	71 01110001	CC 11001100
15 00010101	B5 10110101	46 01000110	75 01110101

Hasil *AddRoundKey* dilakukan *Inverse ShiftRows,* yaitu hasil *SubBytes* digeser ke kanan mulai dari baris ke 1 dilakukan 0 pergeseran, baris ke 2 dilakukan 1 pergeseran, baris ke 3 dilakukan 2 pergeseran, dan baris ke 4 dilakukan 3 pergeseran.

Tahap selanjutnya adalah *Inverse SubBytes* yaitu mengubah hasil *Inverse ShiftRows* menggunakan tabel *Inverse* S-Box dan akan diperoleh hasil sebagai berikut:

81	0B	14	6E	→	91	9E	9B	45
39	9C	AC	87		5B	1C	AA	EA
71	CC	1E	ED		2C	27	E9	53
B5	46	75	15		D2	98	3F	2F

Round 1, *AddRoundKey* yaitu hasil *Inverse SubBytes* xor *Key schedule round* 9, Tahap selanjutnya adalah *Inverse MixColumns,* Hasilnya *Inverse MixColumns* adalah

71 01110001	62 01100010	06 00000110	9B 10011011
35 00110101	BB 10111011	F5 11110101	6D 1101101
F5 11110101	1B 00011011	5E 01011110	58 01011000
4C 01001100	81 10000001	10 00010000	93 10010011

Inverse ShiftRows
Tahap selanjutnya adalah *Inverse SubBytes*

71	62	06	9B	→	2C	AB	A5	E8
6D	35	BB	F5		B3	D9	FE	77
5E	58	F5	1B		9D	5E	77	44
81	10	93	4C		91	7C	22	5D

Proses pada*Round* 1 diulangi sampai *Round* 9. Setelah semua proses selesai dilakukan maka akan di dapat hasil dekripsi: 61 06 22 1C 44 70 6E 63 2C 4A 25 6C F2 48 A9 33. File audio yang telah di dekripsi akan dapat diputar kembali.

HASIL DAN PEMBAHASAN .

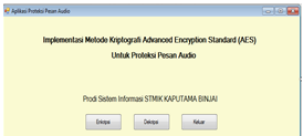
3.1. Implementasi

Hasil setelah dilakukan uji coba terhadap sistem yang telah dibuat dengan menggunakan metode AES untuk proteksi pesan audio.Sistem yang telah dibuat akan dijalankan terlebih dahulu untuk memastikan program dapat berjalan sesuai yang diharapkan atau tidak.

Uji coba pada sistem meliputi dua tahap yaitu uji coba pada proses enkripsi dan uji coba pada proses dekripsi.

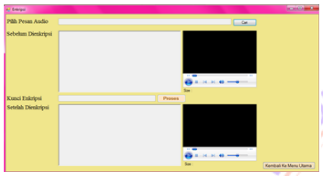
Pada tahap awal proses uji coba, yang dilakukan adalah enkripsi pesan audio. Untuk melakukan enkripsi pada pesan audio, langkah-langkahnya adalah sebagai berikut:

1. Buka sistem proteksi pesan audio, lalu jalankan sistem tersebut.
2. Setelah sistem dijalankan maka akan muncul tampilan menu utama, lalu klik tombol enkripsi.



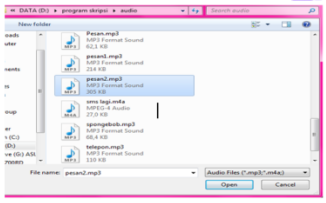
Gambar 2. Tampilan Menu Utama

3. Setelah klik tombol enkripsi, maka akan muncul tampilan menu enkripsi seperti berikut ini:



Gambar 3. Tampilan Menu Enkripsi

4. Masukkan pesan audio yang akan dienkripsi, klik tombol **cari** untuk mencari pesan audio.
5. Pilih pesan audio yang akan dienkripsi kemudian akan muncul tampilan seperti berikut ini:



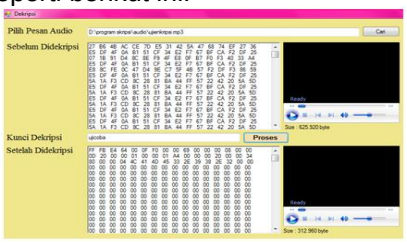
Gambar 4. Tampilan Cari Pesan Audio Untuk Enkripsi

6. Klik open setelah pesan audio dipilih
7. Masukkan kunci enkripsi yang juga berguna sebagai kunci untuk dekripsi.
8. Klik tombol proses, maka enkripsi akan diproses.

3.2. Uji Coba Dekripsi

Pada tahap kedua proses uji coba, yang dilakukan adalah dekripsi pesan audio yang sudah dienkripsi. Untuk dekripsi pesan audio, langkah-langkahnya adalah sebagai berikut:

1. Buka sistem proteksi pesan audio, lalu jalankan sistem tersebut.
2. Setelah sistem dijalankan maka akan muncul tampilan menu utama, lalu klik tombol enkripsi.
3. Setelah klik tombol dekripsi, maka akan muncul tampilan menu dekripsi
4. Masukkan pesan audio yang akan didekripsi, klik tombol **cari** untuk mencari pesan audio yang akan di dekripsi.
5. Pilih pesan audio yang sudah dienkripsi sebelumnya.
6. Klik open setelah pesan audio dipilih.
7. Masukkan kunci dekripsi yang sama seperti kunci untuk enkripsi
8. Klik tombol proses, maka dekripsi akan diproses dan akan muncul tampilan seperti berikut ini:



Gambar 5. Tampilan Proses Dekripsi

9. Setelah enkripsi di proses maka diperoleh hasil enkripsi dan hasil enkripsi akan disimpan.
10. Sebelum disimpan buat dulu nama file hasil enkripsinya. Berikut ini tampilan simpan hasil enkripsi.
11. Setelah hasil dekripsi tersimpan maka akan muncul sebuah pesan pada sistem.

KESIMPULAN

Berdasarkan hasil penelitian dapat disimpulkan bahwa :

1. Sistem yang dibuat menggunakan kunci yang sama untuk melakukan enkripsi dan dekripsi karena metode AES termasuk dalam metode kriptografi jenis kunci simetri.
2. Sistem yang dibuat untuk proteksi pesan audio menggunakan metode AES diperoleh hasil enkripsi berupa pesan audio yang dapat diputar menjadi pesan audio yang tidak dapat diputar dengan ukuran pesan audio hasil enkripsi lebih besar dari pesan audio sebenarnya.
3. Dari hasil uji coba yang telah dilakukan, sistem ini dapat mengenkripsi pesan audio dengan baik dan mengurangi kecurigaan dari pihak lain karena pesan audio yang belum dienkripsi dan sudah dienkripsi memiliki format yang sama dan sistem ini juga dapat melakukan dekripsi pesan audio dengan baik karena pesan audio dapat kembali seperti semula.
4. Sistem yang dibuat hanya bisa melakukan enkripsi dan dekripsi pesan audio yang berformat mp3 dan m4a saja.

DAFTAR PUSTAKA

[1] D. Ariyus, "PENGANTAR ILMU KRIPTOGRAFI Teori Analisis Dan Informasi, FI," *Yogyakarta CV ANDI OFFSET*, 2008.

[2] H. Santoso and M. Fakhriza, "PERANCANGAN APLIKASI KEAMANAN FILE AUDIO FORMAT WAV (WAVE FORM) MENGGUNAKAN ALGORITMA RSA," *J. Ilmu Komput. dan Inform.*, vol. 2, no. 1, pp. 47–54, 2018.

[3] D. Novianto and Y. Setiawan, "Aplikasi Pengamanan Informasi Menggunakan Metode Least Significant Bit (LSB) dan Algoritma Kriptografi Advanced Encryption Standard (AES)," *J. Ilm. Inform. Glob.*, vol. 09, no. 2, pp. 83–89, 2018.

[4] D. Ariyus, "Computer Security," *Penerbit Andi Yogyakarta*, 2006.

[5] "Kriptografi untuk keamanan jaringan dan implementasinya dalam bahasa Jawa / Rifki Sadikin," p. 2012, 2012.

[6] T. Limbong *et al.*, "Optimization of Employee Assignment in Content Management System Making With Hungarian Method," 2018.

[7] R. Munir, "Kriptografi," *Inform. Bandung*, 2006.