Journal of Informatics, Electrical and Electronics Engineering
ISSN 2807-9507 (Media Online)
Vol 2, No 1, September 2022
Hal 18−25
https://djournals.com/jieee

Implementasi Algoritma Quicksort Untuk Pembangkitan Kunci Algoitma RSA Pada Pengamanan Data Audio

Muhammad Rido Hasibuan

Fakultas Ilmu Komputer dan Teknologi Informasi, Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia
Email: ridhohasibuan23@gmail.com
Email Penulis Korespondensi: ridhohasibuan23@gmail.com

Abstrak−Algoritma RSA mempunyai dua kunci, yaitu kunci publik dan kunci rahasia (Private). Algoritma ini memiliki keamanan yang terletak pada kesulitan dalam menghitung alogaritma diskrit. Baik kunci enkripsi maupun dekripsi keduanya merupakan bilangan bulat. Algoritma RSA tipe algoritma kriptografi asimetris terdiri atas dua buah kunci yaitu kunci publik untuk melakukan enkripsi sedangkan kunci pribadi untuk melakukan dekripsi. Dalam algoritma RSA, kunci yang didistribusikan adalah kunci publik yang tidak diperlukan kerahasiannya sedangkan kunci pribadi tetap disimpan atau tidak didistribusikan. Setiap orang yang memiliki kunci publik dapat melakukan proses enkripsi tetapi hasil dari enkripsi tersebut hanya bisa dibaca oleh orang yang memiliki kunci pribadi. Untuk meningkatkan kekuatan dari algoritma tersebut, maka kunci yang digunakan untuk melakukan proses enkripsi dan dekripsi akan dimodifikasi terlebih dahulu menggunakan algoritma pengacakan yaitu Algoritma Quicksort. Algoritma Quicksort merupakan algoritma pengurutan yang dikembangkan oleh Tony Hoare, performa rata-rata pengurutan O (n log n) untuk mengurutkan n item. Tujuan dalam menggunakan algoritma Quicksort ini adalah agar kunci yang dihasilkan lebih sulit ditebak sehingga mempersulit kriptanalis dalam membaca pesan atau informasi tersebut.

Kata Kunci: Kriptografi; Modifikasi; Kunci; Pengurutan; Algoritma Quicksort; Algoritma RSA

Abstract−The RSA algorithm has two keys, namely the public key and the private key. This algorithm has security which lies in the difficulty in calculating discrete algorithms. Both encryption and decryption keys are integers. The RSA algorithm is a type of asymmetric cryptography algorithm, which consists of two keys, namely the public key for encryption and the private key for decryption. In the RSA algorithm, the distributed key is a public key which is not required to be kept secret while the private key is either stored or not distributed. Everyone who has the public key can perform the encryption process but the results of the encryption can only be read by the person who has the private key. To increase the strength of the algorithm, the key used to perform the encryption and decryption process will be modified first using a randomization algorithm, namely the Quicksort Algorithm. The Quicksort algorithm is a sorting algorithm developed by Tony Hoare, the average sorting performance is O (n log n) to sort n items. The purpose of using the Quicksort algorithm is to make the resulting key more difficult to guess, making it difficult for cryptanalysts to read the message or information.

Keywords: Cryptography, Modification, Key, Sorting, Quicksort Algorithm, RSA Algorithm

1. PENDAHULUAN

Saat ini, teknologi komunikasi dan informasi berkembang dengan pesat dan memberikan pengaruh besar bagi kehidupan manusia. Contoh dari perkembangan ini adalah jaringan *internet*, yang pada saat ini telah memungkinkan banyak orang untuk saling bertukar data secara bebas melalui jaringan tersebut. Karena kemudahan yang dimilikinya, *internet* sudah berkembang menjadi salah satu media yang paling populer di dunia. Namun, kemudahan ini juga dimanfaatkan oleh sebagian pihak yang mencoba untuk melakukan kejahatan. Dengan berbagai teknik, banyak yang mencoba untuk mengakses informasi yang bukan haknya. Terdapat beberapa usaha untuk menangani masalah keamanan data rahasia yang dikirimkan melalui *internet*, diantaranya adalah menggunakan teknik kriptografi.

Kriptografi adalah ilmu sekaligus seni untuk menjaga keamanan informasi. Kriptografi terdapat dua proses utama yaitu proses menyandikan pesan yang dapat dibaca (*plaintext*) menjadi pesan yang tidak dapat dibaca atau sudah disandikan (*ciphertext*) yang disebut dengan proses enkripsi (*encryption*) dan proses mengembalikan *ciphertext* menjadi *plaintext* disebut proses dekripsi (*decryption*) [1]. Berdasarkan sejarahnya kriptografi dapat dibagi menjadi kriptografi klasik dan kriptografi modern dan berdasarkan kuncinya dapat dibagi menjadi kriptografi simetris dan kriptografi asimetris.

Adanya kemajuan teknologi semakin memudahkan manusia untuk mendokumentasikan dan menyimpan hal-hal yang sangat penting di dalam kehidupannya. Baik itu berupa *file* dokumen dalam bentuk teks, gambar, suara dan video. *File-file* tersebut terkadang ada yang merupakan suatu rahasia milik pribadi yang tidak ingin diketahui oleh orang lain.

Banyaknya informasi yang bisa didapatkan dengan menggunakan teknologi yang berkembang begitu pesat, menyebabkan penyimpanan suatu data audio haruslah memenuhi aspek keamanan. Agar *file-file* yang bersifat rahasia tersebut tidak dapat dicuri dan digunakan oleh pihak-pihak yang tidak bertanggungjawab. Salah satu cara yang dapat digunakan adalah dengan menyandikan data audio tersebut menjadi sesuatu yang tidak dapat dimengerti dan jika data audio tersebut berhasil dicuri, informasi yang terdapat di dalamnya menjadi tidak berguna.

Teknik kriptografi yang dapat digunakan adalah kriptografi dengan algoritma RSA dan metodenya adalah *Quicksort*. Algoritma RSA adalah sebuah algoritma untuk kriptografi kunci publik. Algoritma ini memiliki keamanan yang terletak pada kesulitan dalam menghitung logaritma diskrit [2]. Algoritma RSA tipe algoritma kriptografi asimetris terdiri atas dua buah kunci yaitu kunci publik untuk melakukan enkripsi sedangkan kunci pribadi untuk melakukan dekripsi. Dalam algoritma RSA, kunci yang didistribusikan adalah kunci publik yang tidak diperlukan kerahasiaannya

sedangkan kunci pribadi tetap disimpan atau tidak didistribusikan. Setiap orang yang memiliki kunci publik dapat melakukan proses enkripsi tetapi hasil dari enkripsi tersebut hanya bisa dibaca oleh orang yang memiliki kunci pribadi. Untuk meningkatkan kekuatan dari algoritma tersebut, maka kunci yang digunakan untuk melakukan proses enkripsi dan dekripsi akan dimodifikasi terlebih dahulu menggunakan algoritma pengacakan. Dan dalam penggunaan metodenya disini penulis menggunakan metode *Quicksort*. Mengapa demikian? Karena sangat memungkinkan untuk menulis algoritma yang lebih cepat untuk beberapa kasus khusus, namun untuk kasus umum, sampai saat ini tidak ada yang lebih cepat dibandingkan algoritma *Quicksort*, performa rata-rata pengurutan O (n log n) untuk mengurutkan item. Algoritma ini digunakan untuk memodifikasi algoritma RSA pada enkripsi yang menggunakan pasangan *plaintext* dengan sebuah kunci rahasia yang diperoleh secara acak. Algoritma *Quicksort* adalah algoritma pengurutan yang dikembangkan oleh Tony Hoare. Algoritma ini juga dikenal sebagai *Partition-Exchange Sort* atau disebut sebagai Sorting Pergantian Pembagi.

Algoritma pengacakan adalah yang menghasilkan permutasi acak dari suatu himpunan terhingga, dengan kata lain untuk mengacak suatu himpunan. Adapun salah satu algoritma pengacakan yang di gunakan adalah Algoritma *Quicksort*. Alasan pemilihan algoritma *Quicksort* untuk pembangkit kunci RSA karena algoritma *Quicksort* salah satu algoritma pengacakan yang dapat membentuk suatu nilai yang baru serta dapat di implemetasikan dalam pembentukan kunci pada algoritma RSA. Untuk meningkatkan kekuatan dari algoritma tersebut, maka kunci yang digunakan untuk melakukan proses enkripsi dan dekripsi akan diubah terlebih dahulu menggunakan pengacakan kunci yaitu algoritma *Quicksort*.

2. METODOLOGI PENELITIAN

2.1 Kriptografi

Kriptografi berasal dari bahasa Yunani, menurut bahasa dibagi menjadi dua, yaitu kripto dan graphia. Kripto berarti secret (rahasia) dan graphia berarti writing (tulisan). Menurut terminologinya, kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat yang lain. Kriptografi pada awalnya dijabarkan sebagai ilmu yang mempelajari bagaimana menyembunyikan pesan. Namun pada pengertian modern kriptografi adalah ilmu yang bersandarkan pada teknik matematika untuk berurusan dengan keamanan informasi seperti kerahasiaan, keutuhan data dan otentikasi entitas. Jadi pengertian kriptografi modern adalah tidak saja berurusan hanya dengan menyembunyikan pesan, namun lebih pada sekumpulan teknik yang menyediakan keamanan informasi [3].

2.2 Algoritma RSA

Algoritma *RSA* merupakan salah satu algoritma kriptografi kunci publik. Algoritma ini adalah algoritma pertama yang cocok dalam melakukan *digital signature*. Saat ini algoritma *RSA* merupakan algoritma yang paling sering dipakai dari algoritma kunci publik lainnya. Algoritma *RSA* dikembangkan pertama kali oleh Ron Rivest, Adi Shamir, dan Len Adleman dari Massachussets Institute of Technology pada tahun 1978. Nama *RSA* sendiri diambil dari nama ketiga peneliti tersebut yaitu, (R)ivest, (S)hamir, dan (A)dleman. *RSA* merupakan algoritma kunci public yang memiliki dua kunci yaitu kunci publik *(public key)* dan kunci pribadi *(private key)* [5].

RSA terbagi menjadi tiga proses, yaitu pembangkitan kunci, enkripsi dan dekripsi. Dasar proses enkripsi dan dekripsi pada algoritma *RSA* yaitu konsep bilangan prima dan aritmatika modulo. Kunci enkripsi tidak dirahasiakan dan diberikan kepada umum (disebut kunci publik), sedangkan kunci untuk dekripsi bersifat rahasia (disebut kunci pribadi).

Untuk menemukan kunci dekripsi, dilakukan dengan cara memfaktorkan bilangan bulat menjadi faktor-faktor primanya. Namun, memfaktorkan bilangan bulat menjadi faktor primanya tidak mudah karena belum ada cara yang efisien untuk melakukan pemfaktoran. Cara yang paling mungkin dilakukan adalah dengan pohon faktor. Namun semakin besar bilangan yang akan difaktorkan maka semakin lama pula waktu yang dibutuhkan untuk menyelesaikannya. Jadi semakin besar bilangan yang akan difaktorkan, semakin sulit pemfaktorannya, semakin kuat pula algoritma *RSA*. Oleh karena itu, dalam menggunakan algoritma *RSA* dianjurkan menggunakan bilangan yang sangat besar agar keamanannya dapat terjamin. Besaran-besaran yang digunakan pada algoritma *RSA* antara lain:

1. p dan q bilangan prima (rahasia)
2. $n = pq$ (tidak rahasia)
3. $p(n) = (p - 1)(p - 1)$ (rahasia)
4. e (kunci enkripsi) (tidak rahasia)
5. d (kunci dekripsi) (rahasia)
6. m (plainteks) (rahasia)
7. c (chiperteks) (tidak rahasia)

2.3 Algoritma Quicksort

Algoritma *quicksort* bekerja menurut prinsip bagi-dan-pecahkan (*divide- and-conquer*). Dengan demikian hal mendasar dalam algoritma *quicksort* adalah pemilhan poros (*pivot*) dan pembuatan partisi sedemikian rupa sehingga elemen dengan nilai kecil ada dibagian kiri poros dan elemen dengan nilai besar ada di bagian kanan poros. Berbagai varian dari *quicksort* pada intinya berupaya untuk mengembangkan teknik-teknik pembuatan partisi yang efektif untuk berbagai macam masukan. Pada tahun 1998 M.D Mcllroy membuat tulisan berjudul "A Killer Adversary for Quicksort" yang menguraikan cara untuk membuat susunan data tertentu (dalam *array*) hingga operasi pengurutan menggunakan *quicksort* mendekati kuadratik O (n^2). Cara ini berlaku untuk setiap varian dari *quicksort* dengan syarat

tertentu.Kebutuhan waktu dari *quicksort* bergantung pada pembuatan partisi, seimbang atau tidak, yang bergantung juga pada elemen yang digunakan sebagai poros. Dalam menghitung kompleksitas ini, perlu dilihat pula perhitungan *recurrence*, karena terdapat fungsi rekursif untuk penyelesaian sub-masalah. Terdapat 3 jenis kompleksitas waktu dari *quicksort* [6].

1. Kasus terburuk (*worst case*), yaitu terjadi bila terbentuk partisi dengan komposisi sub-masalah antara n – 1 elemen dan 0 elemen. Dengan demikian pemanggilan fungsi secara rekursif dengan array berukuran 0 akan langsung kembali, T (0) = O (1), sehingga berlaku: $T(n) = T(n-1) + cn = O(n^2)$.
2. Kasus terbaik *(best case)*, yaitu terjadi bila terbentuk partisi dengan komposisi seimbang, dengan ukuran masing-masing tidak lebih dari n/2. Sehingga didapat: $T(n) = 2T(n/2) + cn = na + cn \log n = O(n \log n)$.
3. Kasus rata-rata *(average case)*, yaitu terjadi dari perimbangan poros antara terbaik dan terburuk, yang dalam prakteknya lebih mendekati kasus terbaik ketimbang terburuk. Sehingga didapat: $T_{avg}(n)=O(n \log n)$.

3. HASIL DAN PEMBAHASAN

3.1 Analisa Masalah

Analisa pada suatu algoritma dapat bertujuan untuk melihat faktor efesiensi dan efektifitas dari algoritma yang sedang dianalisa, juga dapat dilakukan dengan cara melihat sisi waktu tempu dari suatu algoritma, proses, langkah langkah atau satuan waktu yang ditempuh dari suatu algoritma dalam menyelesaikan suatu masalah. Kriptografi merupakan metode dengan menyandikan file *audio* menjadi yang sulit atau bahkan tidak dipahami melalui proses enkripsi, untuk memperoleh kembali informasi yang dapat dengan proses enkripsi, untuk memperoleh kembali informasi yang asli dan dapat dilakukan dengan proses enkripsi yang tentunya dapat digunakan dengan kunci yang benar. Untuk melindungi file *audio* dari pihak-pihak yang tidak berkepentingan tersebut maka diperlukan enkripsi dan dekripsi. Agar dapat dilakukan dengan baik, juga dibutuhkan suatu algoritma untuk enkripsi dan dekripsi salah satunya algoritma RSA.

Algoritma RSA mempunyai dua kunci, yaitu kunci publik dan kunci *private*. Algoritma ini memiliki keamanan yang terletak pada kesulitan dalam menghitung logaritma diskrit. Baik kunci enkripsi maupun dekripsi keduanya merupakan bilangan bulat. RSA adalah salah satu teknik kriptografi dimana kunci untuk melakukan enkripsi berbeda dengan kunci untuk melakukan dekripsi. Kunci untuk melakukan enkripsi disebut sebagai kunci publik, sedangkan kunci untuk melakukan dekripsi disebut sebagai kunci *private*. Orang yang mempunyai kunci publik dapat melakukan enkripsi tetapi yang dalam melakukan dekripsi hanyalah orang yang memiliki kunci privat. Kunci publik dapat dimiliki oleh sembarang orang, tetapi kunci *private* hanya dimiliki oleh orang tertentu saja. Untuk meningkatkan kekuatan dari algoritma tersebut, maka kunci yang digunakan untuk melakukan proses enkripsi dan dekripsi akan diimplementasikan terlebih dahulu menggunakan algoritma pengacakan yaitu Algoritma Quicksort.

3.1.1 Penerapan Algoritma Quicksort

Pembangkit kunci dilakukan dengan menggunakan algoritma yang menggunakan bilangan random dengan memasukkan bilangan random, dianggap dapat menghilangkan kemungkinan penyerang menebak hasil dengan mengetahui algoritmanya. Telah banyak algoritma generator bilangan acak yang diusulkan dan digunakan hingga saat ini. Algoritma-algoritma tersebut menggunakan berbagai pendekatan berbeda untuk menghasilkan bilangan random acak. Salah satu algoritma tersebut adalah Algoritma Quicksort. Algoritma Quicksort merupakan Algoritme pengurutan yang dikembangkan oleh Tony Hoare, performa rata-rata pengurtan O(n log n) untuk mengurutkan n item. Dengan membagi algoritma dasar ini beberapa kali, pemotongan minimum dapat ditemukan dengan probabilitas tinggi. Adapun proses Implementasi kunci Algoritma RSA dengan menggunakan Algoritma Quicksort dijelaskan mengenai rancangan sistem penambahan Implementasi pada metode kriptografi yang diteliti.

Adapun proses modifikasi kunci algoritma RSA dengan menggunakan algoritma *Quicksort* adalah sebagai berikut:

1. Pilih sembarang bilangan prima dengan melakukan 15 kali pengacakan.
2. Pilih dua buah bilangan acak, misal: g untuk nilai minimum dan x untuk nilai maksimum berdasarkan algoritma *Quicksort*.
3. Pembentukan kunci berdasarkan bilangan prima g dan x. Diketahui :

Dimana: B= 1 a_n=41 C= 91 mod= 255

$a_1 = (1 \times (41 + 91)) \ mod\ 255$
$= 132\ mod\ 255 = 132$
$a_2 = (1 \times (132 + 91))\ mod\ 255$
$= 223\ mod\ 255 = 223$
$a_3 = (1 \times (223 + 91))\ mod\ 255$
$= 314\ mod\ 255 = 59$
$a_4 = (1 \times (59 + 91))\ mod\ 255$
$= 150\ mod\ 255 = 150$
$a_5 = (1 \times (150 + 91))\ mod\ 255$
$= 241\ mod\ 255 = 241$

$a_6 = (1 \times (241 + 91)) \ mod \ 255$
$= 332 \ mod \ 255 = 77$
$a_7 = (1 \times (77 + 91)) \ mod \ 255$
$= 168 \ mod \ 255 = 168$
$a_8 = (1 \times (168 + 91)) \ mod \ 255$
$= 259 \ mod \ 255 = 4$
$a_9 = (1 \times (4 + 91)) \ mod \ 255$
$= 95 \ mod \ 255 = 95$
$a_{10} = (1 \times (95 + 91)) \ mod \ 255$
$= 186 \ mod \ 255 = 186$
$a_{11} = (1 \times (186 + 91)) \ mod \ 255$
$= 277 \ mod \ 255 = 22$
$a_{12} = (1 \times (22 + 91)) \ mod \ 255$
$= 113 \ mod \ 255 = 113$
$a_{13} = (1 \times (113 + 91)) \ mod \ 204$
$= 204 \ mod \ 255 = 204$
$a_{14} = (1 \times (204 + 91)) \ mod \ 295$
$= 295 \ mod \ 255 = 40$
$a_{15} = (1 \times (40 + 91)) \ mod \ 131$
$= 131 \ mod \ 255 = 131$

Sehingga didapatkan nilai acak adalah: 132, 223, 59, 150, 241, 77, 168, 4, 95, 186, 22, 113, 204, 40, 131. Maka proses selanjutnya mencari nilai bilangan prima dari nilai acak diatas menggunakan algoritma *Quicksort* dengan cara menyeleksi dan mengurutkan bilangan yang telah diacak sebagai berikut:

132	223	59	150	241	77	168	4	95	186	22	113	204	40	131

1. Langkah pertama yaitu menentukan privotnya, dalam contoh diatas saya memilih angka dari sebelah kanan kolom

132	223	59	150	241	77	168	4	95	186	22	113	204	40	131
													Privot	

2. Kemudian buat partisi disebelah kiri dan kanan dengan pengurutan angka terkecil sebelum angka terbesar

132	223	59	150	241	77	168	4	95	186	22	113	204	40	131
131	132	223	59	150	241	77	168	4	95	186	22	113	204	40
40	131	132	223	59	150	241	77	168	4	95	186	22	113	204
40	131	132	223	59	150	241	77	168	4	95	186	22	113	204
40	59	131	132	223	150	241	77	168	4	95	186	22	113	204
40	59	131	132	223	150	241	77	168	4	95	186	22	113	204
40	59	131	132	150	223	241	77	168	4	95	186	22	113	204
40	59	131	132	150	204	223	241	77	168	4	95	186	22	113
40	59	131	132	113	150	204	223	241	77	168	4	95	186	22
40	59	113	131	132	150	204	223	241	77	168	4	95	186	22
40	59	113	131	132	150	22	204	223	241	77	168	4	95	186
22	40	59	113	131	132	150	204	223	241	77	168	4	95	186
22	40	59	113	131	132	150	186	204	223	241	77	168	4	95
22	40	59	113	131	132	150	186	204	223	95	241	77	168	4
22	40	59	95	113	131	132	150	186	204	223	241	77	168	4
22	40	59	95	113	131	132	150	186	204	223	4	241	77	168
4	22	40	59	95	113	131	132	150	186	204	223	241	77	168
4	22	40	59	95	113	131	132	150	186	204	223	168	241	77
4	22	40	59	95	113	131	132	150	168	186	204	223	241	77
4	22	40	59	95	113	131	132	150	168	186	204	223	77	241
4	22	40	59	77	95	113	131	132	150	168	186	204	223	241

Maka hasil data acak yang telah di urutkan dengan algoritma *Quicksort* adalah sebagai berikut: 4, 22, 40, 59, 77, 95, 113, 131, 132, 150, 168, 186, 204, 223, 241. Maka proses selanjutnya adalah mencari 2 bilangan prima terkecil dengan cara mengurutkan dengan algoritma *Quicksort.* Adapun bilangan prima yang didapatkan dari hasil pengurutan

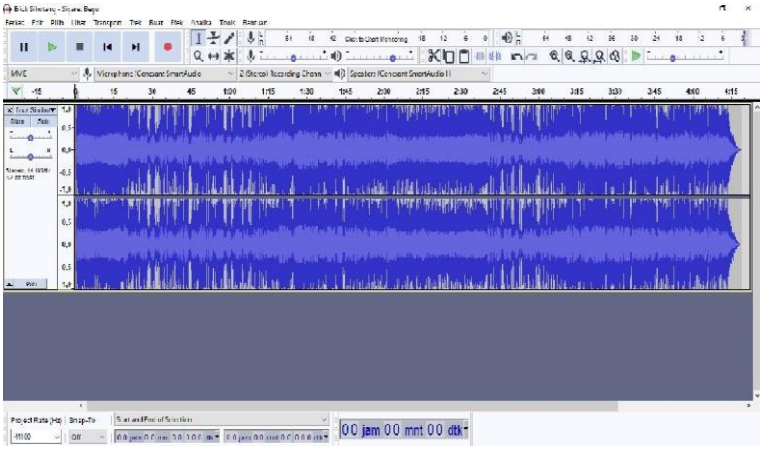
algoritma *Quicksort* adalah: 59, 113. Maka nilai bilangan terkecil 59 dan 113 dimana 59 adalah *e*, 113 adalah *d*. Maka proses selanjutnya menentukan kunci:
n = e*d mod m
n = 59*113 mod 255 n =142
Jadi kunci *public* yang digunakan adalah d= 113, n= 142 dan kunci *private* adalah e = 59, n =142.

3.1.2 Penerapan Algoritma RSA

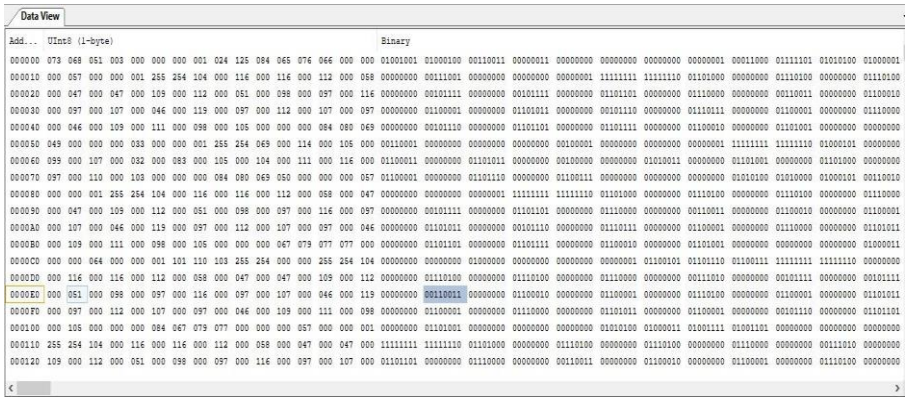
Adapun proses implementasi enkripsi algoritma *RSA* yaitu contoh kasus dalam proses ini adalah *file* audio "Erick Sihotang - SigaretBegu". Data yang diambil hanya sebanyak 12 *bytes* untuk *plainteks*, cara pengambilan nilai text *ASCII* data dokumen menggunakan aplikasi *Binary Viewer*, seperti dibawah ini:

Tabel 1. Sampel File audio yang akan diimplementasi

Nama File	**ErickSihotang-SigaretBegu**
Extension File	MP3

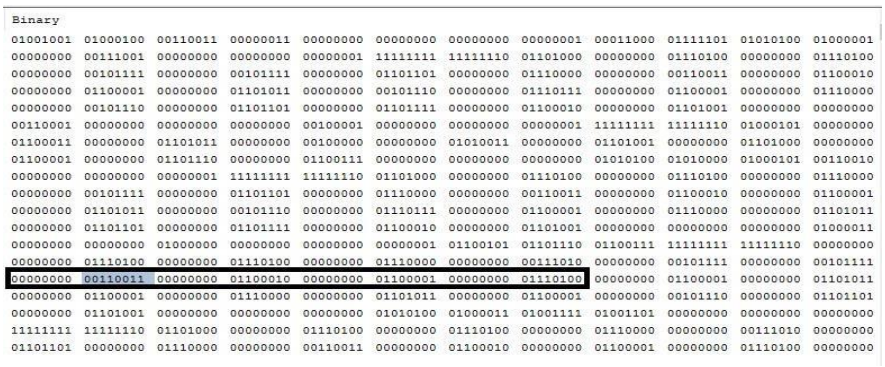


Gambar 1. File Audio Yang Akan Diimplementasikan



Gambar 2. Isi *File Audio* dari hasil *Binary Viewer*

Dari data tersebut diambil sebanyak 8 bytes nilai binary dan dikonversi ke decimal sebagai *sampel metode*, yang berguna untuk mengetahui nilai biner dari bilangan tersebut.



Gambar 3. Nilai Binary Berdasarkan *Binary Viewer*

Dari gambar data *audio* di atas diambil sebanyak 8 bytes untuk plainteks, yaitu :

Teks Binary : **00110011 00000000 01100010 00000000 011000001 00000000 01110100 00000000**

1. Proses Enkripsi dengan menggunakan Algoritma RSA. Enkripsi Metode RSA :
   Kunci PRIVATE (e,n) $K_{private}$ = (59, 142) Kunci PUBLIC (d,n) K_{public} = (113,142)
   Rubah Binary **00110011 00000000 01100010 00000000 011000001 00000000 01110100 00000000** ke bilang decimal.
   Adapun nilainya sebagai berikut : (**00110011**= 51), (**00000000**=00), (**01100010=**98), (**00000000**=00), (**011000001**=97), (**00000000**=00), (**01110100**=116), (**00000000**=00). Susun plainteks menjadi blok-blok m1 , m2 , …, (nilai setiap blok di dalam selang [0, p – 1].
   P_1 = 51
   P_2 = 00
   P_3 = 98
   P_4 = 00
   P_5 = 97
   P_6 = 00
   P_7 = 116
   P_8 = 00
   Adapun rumus enkripsi algoritma RSA yaitu:
   $C = p^e \, Mod \, n$
   Enkripsi P_1 = 51
   $C1 = 51^{59} \, Mod \, 142$
   $C1 = 23$
   Enkripsi P_2 = 00
   $C2 = 00^{59} \, Mod \, 142$
   $C2 = 00$
   Enkripsi P_3 = 98
   $C3 = 98^{59} \, Mod \, 142$
   $C3 = 80$
   Enkripsi P_4 = 00
   $C4 = 00^{59} \, Mod \, 142$
   $C4 = 00$
   Enkripsi P_5 = 97
   $C5 = 97^{59} \, Mod \, 142$
   $C5 = 39$
   Enkripsi P_6 = 00
   $C6 = 00^{59} \, Mod \, 142$
   $C6 = 00$
   Enkripsi P_7 = 116
   $C7 = 116^{59} \, Mod \, 142$
   $C7 = 32$
   Enkripsi P_8 = 00
   $C8 = 00^{59} \, Mod \, 142$
   $C8 = 00$
   Sehingga dapat dihasilkan hasil Enkripsi dari sample data diatas adalah :

Tabel 2. Hasil Enkripsi

23	00	80	00
39	00	32	00

Selanjutnya merubah hasil Enkripsi kebilangan Binary. Adapun nilainya sebagai berikut :

Tabel 3. Hasil Enkripsi ke Binary

Enkripsi	Binary	Enkripsi	Binary
23	00110011	39	00111001
00	00000000	00	00000000
80	01100010	32	00110100

Enkripsi	Binary	Enkripsi	Binary
00	00000000	00	00000000

2. Proses Dekripsi dengan menggunakan Algoritma RSA adalah sebagai berikut.
   Adapun rumus dekripsi algoritma RSA yaitu:
   $P = C^d \, Mod \, n$
   Dekripsi M1= 23
   $P1 = 23^{113} Mod \, 142$
   $P1 = 23$
   Dekripsi M2= 00
   $P2 = 00^{113} Mod \, 142$
   $P2 = 00$
   Dekripsi M3= 80
   $P3 = 80^{113} Mod \, 142$
   $P3 = 60$
   Dekripsi M4= 00
   $P4 = 00^{113} Mod \, 142$
   $P4 = 00$
   Dekripsi M5= 39
   $P5 = 39^{113} Mod \, 142$
   $P5 = 39$
   Dekripsi M6= 00
   $P6 = 00^{113} Mod \, 142$
   $P6 = 00$
   Dekripsi i M7= 32
   $P7 = 32^{113} Mod \, 142$
   $P7 = 32$
   Dekripsi M8= 00
   $P8 = 00^{113} Mod \, 142$
   $P8 = 00$
   Sehingga dapat dihasilkan hasil Deskripsi dari sample data diatas adalah

Tabel 4. Hasil Dekripsi

23	00	60	00
39	00	32	00

Selanjutnya merubah hasil Enkripsi kebilangan Binary. Adapun nilainya sebagai berikut :

Tabel 5. Hasil Dekripsi ke Binary

Dekripsi	Binary	Dekripsi	Binary
23	00110011	39	00111001
00	00000000	00	00000000
60	01100000	32	00110010
00	00000000	00	00000000

Hasil proses data yang sudah didekripsi kembali kedata enkripsi.

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan dapat diambil beberapa kesimpulan Proses penerapan Algoritma Quicksort dengan cara melakukan pengacakan nilai untuk pembentukan kunci berdasarkan metode kriptografi Algoritma RSA. Proses enkripsi dapat dilakukan serta hasil pengambilan enkripsi dengan cara melakukan proses deskripsi dapat dilakukan dengan baik menggunakana Algoritma RSA berdasarkan kunci yang telah di modifikasi dengan algoritma Quicksort.

REFERENCES

[1]. Caroline, Maureen Linda. “Perbandingan Algoritma Kriptografi Kunci Publik RSA, Rabin dan ElGamal”. Program Studi Teknik Informatika Institut Teknologi Bandung, Bandung, 2011.
[2]. Listiyono, Hersatoto. “Implementasi Algoritma Kunci Public Pada Algoritma RSA”. Fakultas Teknologi Informasi, Universitas Stikubank, Semarang, 2009..
[3]. Munir, Rinaldi. Kriptografi, Informatika, Bandung. 2006.
[4]. R. Sadikin, KRIPTOGRAFI UNTUK KEAMANAN JARINGAN. YOGYAKARTA: C.V ANDI OFFSET, 2012

[5]. Rinaldi Munir, M.T. Algoritma RSA dan ElGamal, Bandung, 2004
[6]. R. Sedgewick. Implementing quicksort programs. Comm. ACM, 21:847– 856, 1978.
[7]. Amir Hamzah Suleiman, Media Audio-Visual Untuk Pengajaran, Penerangan dan Penyuluhan. Jakarta : PT Gramedia. 1985
[8]. R. Zendrato and A. U. Hamdani, "Pemodelan Sistem Informasi Pengadaan Alat dan Bahan Praktikum Menggunakan Unified Modeling Language (Studi Kasus: Program Pendidikan Dokter Gigi Spesialis Konservasi Gigi Fakultas Kedokteran Gigi Universitas XYZ)," Konf. Nas. Teknol. Inf. dan Komput., vol. I, no. 1, pp. 86–95, 2017.
[9]. I. Fahmi, Manajemen Pengambilan Keputusan Teori dan Aplikasi. Bandung: PT. Alfabeta, 2016.
[10]. Z. I, "Pemodelan Berbasis UML (Unified Modeling Language) dengan Strategi Teknik Orientasi User Centered Design (UCD) dalam Sistem Administrasi Pendidikan," Univ. Islam Negeri Sumatra Utara Medan, no. January 2013, 201.
[11]. W. Komputer, Membuat Aplikasi Client Server dengan Visual Basic 2008. Yogyakarta: Andi, 2010