

# ENKRIPSI DATA AUDIO MENGGUNAKAN METODE KRIPTOGRAFI RSA

Awang Harsa K<sup>1)</sup>, Andi Yusika R<sup>2)</sup>, Asfami Ansharie<sup>3)</sup>

<sup>1,2,3</sup>Teknik Informatika, STMIK Widya Cipta Dharma

<sup>1,2,3</sup>Jl. M. Yamin No.25, Samarinda, 75123

E-mail : awangkid@gmail.com<sup>1)</sup>, yu5h1k4@gmail.com<sup>2)</sup>, asfami93@gmail.com<sup>3)</sup>

## ABSTRAK

Penerapan Metode RSA pada Enkripsi Data *Audio*, merupakan bentuk penelitian untuk membuktikan bahwa Metode Kriptografi dapat digunakan untuk pencarian solusi, khususnya pada permasalahan kerahasiaan data .

Tujuan dari penelitian ini adalah merancang dan membangun sebuah aplikasi yang dapat menyelesaikan masalah enkripsi data untuk merahasiakan sebuah data dengan menggunakan dua kunci yaitu, proses enkripsi dengan menggunakan kunci *public* dan kunci *Private* digunakan untuk melakukan proses dekripsinya, dengan menggunakan bahasa pemrograman *Visual Basic .NET*. Dalam penelitian ini, teknik pengumpulan data yang digunakan adalah studi pustaka. Metode pengujian yang digunakan adalah metode pengujian *White-Box* yang digunakan untuk menguji *listing Coding* Proses enkripsi dan dekripsinya, *Black Box* digunakan untuk menguji apakah aplikasi berjalan dengan algoritma kunci yang sesuai, menguji daya tahan hasil enkripsi data apakah bisa di enkripsi dengan metode kriptografi lainnya.

Dengan menggunakan tahapan pengembangan *Prototype* yaitu Tahapan Perancangan Antarmuka, *Implementasi*, Pengujian Sistem, agar dalam membangun Aplikasi Enkripsi Data *Audio* menggunakan Kriptografi RSA dengan terstruktur. Aplikasi ini dapat menjadi salah satu media alternatif untuk Keamanan data.

**Kata Kunci:** *Microsoft Visual Studio Ultimate 2012, Kriptografi, Enkripsi Data Audio, Metode RSA..*

---

## 1. PENDAHULUAN

Keamanan pada komputer lebih mengarah kepada keamanan data yang tersimpan di dalam komputer tersebut. Salah satu cara untuk mengamankan data komputer adalah melakukan enkripsi pada sebuah data atau *file* yang kita anggap penting. Teknik enkripsi ini adalah teknik untuk merubah bentuk data, sehingga orang lain tidak mengetahui bentuk asli dari data tersebut.

Untuk melakukan pengiriman data secara *manual*, sangat memungkinkan diketahui oleh orang lain. Untuk mengirimkan data yang bersifat rahasia, maka diperlukan teknik enkripsi untuk merubah bentuk data tersebut agar tidak mudah dibaca atau dilihat oleh orang lain. Setelah melalui teknik enkripsi, data yang telah dirubah tetap dapat terlihat. Tetapi data yang telah melalui proses enkripsi memiliki bentuk yang telah berubah dari bentuk aslinya. Data adalah keterangan tertulis mengenai sesuatu fakta yang masih berdiri sendiri, belum mempunyai arti sebagai kelompok, belum terhubung satu sama lain, dan belum diolah sesuai keperluan tertentu.

Dalam penelitian ini data yang akan dienkripsi adalah data *audio* karena enkripsi data *audio* relatif berukuran kecil jadi proses enkripsi menjadi lebih cepat, enkripsi relatif ringan, *file audio* sering digunakan sebagai pesan pribadi untuk orang-orang tertentu pada saat ini.

Teknik enkripsi yang digunakan untuk mengenkripsi sebuah data adalah dengan metode RSA yang lebih menekankan proses enkripsi dan dekripsi yang memiliki dua kunci yaitu *private-key* kunci yang digunakan untuk

mengdekripsi data pada pemilik data dan hanya pemilik data yang tau kunci ini, sedangkan *public-key* adalah kunci yang dapat disebarluaskan secara luas tanpa harus khawatir karena kunci rahasia tersebut hanya dapat di dekripsi hanya oleh pengirim dan penerima.

## 2. RUANG LINGKUP PENELITIAN

Permasalahan difokuskan pada :

1. Dalam pembuatan aplikasi ini hanya akan membahas mengenai penyandian pada *file audio*.
2. Format *file audio* yang dapat dienkripsi dengan aplikasi ini adalah *file* yang berekstensi mp3,wav,wma,ogg.
3. Maksimal karakter kunci yang digunakan adalah 3 digit, hal ini dilakukan agar proses enkripsi tidak memakan waktu yang terlalu lama.
4. Karakter kunci yang digunakan hanya angka saja.
5. Bilangan Prima yang digunakan untuk kunci p dan q harus menghasilkan nilai 1 menggunakan perhitungan fpb.
6. Proses dekripsi menghasilkan *chiperteks* yang berekstensi *file rsafile*.
7. *File Audio* yang telah dienkripsi hanya dapat didekripsi kembali ke bentuk semula dengan menggunakan perhitungan kunci yang telah ditentukan saat melakukan enkripsi.

## 3. BAHAN DAN METODE

Adapun bahan dan metode yang digunakan dalam membangun game ini yaitu:

### 3.1 Kriptografi

Menurut Dony, (2008). Kriptografi (*Cryptography*) berasal dari bahasa Yunani yaitu dari kata *Crypto* dan *Graphia* yang berarti penulisan rahasia. Kriptografi adalah suatu ilmu yang mempelajari penulisan secara rahasia. Kriptografi merupakan bagian dari suatu cabang ilmu matematika yang disebut *Cryptology*. Kriptografi bertujuan menjaga kerahasiaan informasi yang terkandung dalam data sehingga informasi tersebut tidak dapat diketahui oleh pihak yang tidak sah.

Dalam menjaga kerahasiaan data, kriptografi mentransformasikan data asli (*plaintext*) ke dalam bentuk data sandi (*ciphertext*) yang tidak dapat dikenali. *Ciphertext* inilah yang kemudian dikirimkan oleh pengirim (*sender*) kepada penerima (*receiver*). Setelah sampai di penerima, *ciphertext* tersebut ditransformasikan kembali ke dalam bentuk *plaintext* agar dapat dikenali. Proses transformasi dari *plaintext* menjadi *ciphertext* disebut proses *Encipherment* atau enkripsi (*encryption*), sedangkan proses mentransformasikan kembali *ciphertext* menjadi *plaintext* disebut proses dekripsi (*decryption*). Untuk mengenkripsi dan mendekripsi data. Kriptografi menggunakan suatu algoritma (*cipher*) dan kunci (*key*). *Cipher* adalah fungsi matematika yang digunakan untuk mengenkripsi dan mendekripsi data. Sedangkan kunci merupakan sederetan *bit* yang diperlukan untuk mengenkripsi dan mendekripsi data.

Algoritma kriptografi modern tidak lagi mengandalkan keamanannya pada kerahasiaan algoritma tetapi kerahasiaan kunci. *Plaintext* yang sama bila disandikan dengan kunci yang berbeda akan menghasilkan *ciphertext* yang berbeda pula.

### 3.2 Jenis Algoritma Kriptografi

Algoritma kriptografi dibagi atas dua golongan yaitu di halaman berikut :

#### 1. *Symmetric Algorithms*

Algoritma kriptografi simetris atau disebut juga algoritma kriptografi konvensional adalah algoritma yang menggunakan kunci untuk proses enkripsi sama dengan kunci untuk proses dekripsi. Proses enkripsi-dekripsi algoritma kriptografi simetris dapat dilihat pada gambar dibawah ini :



Gambar 1. Algoritma Kriptografi Simetris

Algoritma kriptografi simetris dibagi menjadi 2 kategori yaitu algoritma aliran (*Stream Ciphers*) dan algoritma blok (*Block Ciphers*). Pada algoritma aliran, proses penyandiannya berorientasi pada satu *bit* atau satu *byte* data. Sedangkan pada algoritma blok, proses penyandiannya berorientasi pada sekumpulan *bit* atau *byte* data (per blok).

#### 2. *Asymmetric Algorithms*

Algoritma kriptografi asimetris adalah algoritma yang menggunakan kunci yang berbeda untuk proses enkripsi dan dekripsinya. Algoritma ini disebut juga

algoritma kunci umum (*public key algorithm*) karena kunci untuk enkripsi dibuat umum (*public key*) atau dapat diketahui oleh setiap orang, tapi kunci untuk dekripsi hanya diketahui oleh orang yang berwenang mengetahui data yang disandikan atau sering disebut kunci pribadi (*private key*). Proses enkripsi-dekripsi algoritma asimetris dapat dilihat pada gambar dihalaman berikutnya :



Gambar 2. Algoritma Kriptografi Asimetris

Pada algoritma *public key* ini, semua orang dapat mengenkripsi data dengan memakai *public key* penerima yang telah diketahui secara umum. Akan tetapi data yang telah terenkripsi tersebut hanya dapat didekripsi dengan menggunakan *private key* yang hanya diketahui oleh penerima.

### 3.3 Dasar Matematika Kriptografi

Dasar-dasar matematika yang digunakan kriptografi adalah Pembagi Bersama Besar, Aritmatika Modulo.

#### 1. Pembagi Bersama Terbesar (PBB)

Menurut Renaldi (2012) Jika  $a$  dan  $b$  adalah dua bilangan bulat tidak nol. Pembagi bersama terbesar (PBB-*Greatest Common Divisor* atau GCD) dari  $a$  dan  $b$  adalah bilangan bulat terbesar  $d$  sedemikian sehingga  $d \mid a$  dan  $d \mid b$ . Dalam hal ini dituliskan dengan  $\text{gcd}(a,b)=d$ . Dimana  $d$  :

- 1.)  $d$  merupakan pembagi persekutuan dari  $a$  dan  $b$ ,
- 2.) Jika  $c$  merupakan pembagi persekutuan dari  $a$  dan  $b$ , maka  $c \leq d$
- 3.) Contoh faktor pembagi dari 30: 1, 2, 3, 5, 6, 10, 15, 30;
- 4.) Faktor pembagi dari 20: 1, 2, 4, 5, 10, 20;
- 5.) Faktor pembagi bersama dari 30 dan 20 adalah 1, 2, 5, dan 10; Sehingga  $\text{gcd}(30,20) = 10$

Dalam beberapa cabang matematika yang lebih abstrak, syarat  $d$  yang kedua diubah dengan  $c \mid d$  menggantikan  $c \leq d$  sehingga  $d$  dan  $-d$  keduanya merupakan  $\text{gcd}(a,b)$ .

#### 2. Aritmetika modular

Menurut Arif (2008), Aritmetika modular merupakan operasi matematika yang banyak diimplementasikan pada metode kriptografi. Pada metode kriptografi *asymetric*, Dalam kriptografi modular, *domain* yang digunakan adalah *subset* dari bilangan bulat dan bersifat *finite* (terbatas, besarnya *domain* merupakan bilangan bulat). *Domain* dari aritmatika modular adalah  $\{0, 1, 2, \dots, n-1\}$  dimana  $n$  adalah besarnya *domain*. Aritmatika disebut dengan aritmatika modulo  $n$ , dengan pertambahan dan perkalian seperti aritmatika biasa menghasilkan bilangan yang termasuk didalam *domain*. Jika hasil merupakan bilangan diluar *domain* maka bilangan harus dikurangi dengan kelipatan  $n$  sampai menghasilkan bilangan didalam *domain*.

Operasi modulo dapat didefinisikan sebagai berikut:  $a \bmod n = r = a - n \cdot q$ , dengan kata lain  $a \bmod n$  merupakan *remainder* atau *residue* dari pembagian  $a$  oleh  $n$ . Contoh lain dari aritmatika modulo adalah:

- 1)  $23 \bmod 5 = 3 = 23 - 5(4)$  dimana  $a = 23, n = 5, r = 3, q = 4$
- 2)  $8 \bmod 5 = 3 = 8 - 5(1)$  dimana  $a = 8, n = 5, r = 3, q = 1$
- 3)  $5 \bmod -4 = 1 = 5 - (-4)(-1)$  dimana  $a = 5, n = -4, r = 1, q = -1$

Aritmatika modulo cocok digunakan dalam kriptografi karena dua alasan yaitu sebagai berikut :

- 1.) Oleh karena nilai-nilai aritmatika modulo berada dalam himpunan berhingga ( $0$  sampai modulus  $m-1$ ), maka kita tidak perlu khawatir hasil perhitungan berada di luar himpunan.
- 2.) Karena kita bekerja dengan bilangan bulat, maka kita tidak khawatir kehilangan informasi akibat pembulatan sebagaimana pada operasi bilangan riil.

### 3. Algoritma Euclidean

Menurut Renaldi (2012) Algoritma euclidean merupakan suatu algoritma yang digunakan untuk mencari *Greatest Common Divisor* (GCD) atau biasa dikenal dengan Pembagi Bersama Terbesar (PBB) dari dua bilangan, khususnya untuk bilangan-bilangan yang sangat besar sehingga tidak perlu mencari faktorisasi prima dari kedua bilangan tersebut. Algoritma euclidean ini biasanya diperkenalkan kepada mahasiswa yang sedang mempelajari mata kuliah teori bilangan tetapi tidak jarang juga soal-soal olimpiade matematika dan ujian universitas membutuhkan cara ini untuk menyelesaikan soal yang diberikan. Oleh karena itu Istana Matematika ingin menjelaskan tentang pengertian dari algoritma yang menarik ini algoritma euclidean.

### 4. Bilangan Prima

Menurut Rinaldi (2012), Bilangan Bulat Postif yang mempunyai aplikasi penting dalam ilmu komputer dan matematika diskrit adalah bilangan prima. Bilangan prima adalah bilangan bulat postif yang lebih besar dari 1 yang habis dibagi oleh 1 dan dirinya sendiri.

Sebagai contoh, 23 adalah bilangan prima karena ia hanya habis dibagi oleh 1 dan 23, karena bilangan prima harus lebih besar dari 1, maka barisan bilangan prima dimulai dari angka 2 selanjutnya 3,5,7,11,13, .... Seluruh bilangan Prima adalah bilangan ganjil kecuali 2 yang merupakan bilangan genap.

Bilangan selain prima disebut bilangan komposit (*composite*). Misalnya 20 adalah bilangan komposit karena dapat dibagi oleh 2, 4,5, dan 10, selain nilai 1 dan 20 itu sendiri.

### 3.4 Algoritma RSA

Menurut Rinaldi (2012), Algoritma ini pertama kali dikenalkan oleh ketiga orang yaitu Ron Rivest, Adi Shamir, dan Len Adleman pada tahun 1976, mereka bertiga adalah peneliti dari MIT (*Massachusetts Institute of Technology*), mereka memperkenalkan algoritma ini untuk mengirimkan kunci rahasia kepada penerima.

Pengiriman kunci rahasia pada saluran publik (telepon, pos, *internet*) sangat tidak aman. Oleh karena itu, kunci harus dikirim melalui saluran kedua yang benar-benar aman. Saluran kedua tersebut umumnya lambat dan mahal.

Algoritma ini tidak berdasarkan pada proses enkripsi dan dekripsi, melainkan lebih kepada proses matematika yang dilakukan untuk menghasilkan kunci rahasia yang dapat disebarkan secara bebas tanpa harus khawatir karena kunci rahasia tersebut hanya dapat didekripsi hanya oleh pengirim dan penerima pesan. Dasar dari algoritma ini memfaktorkan bilangan yang besar menjadi faktor-faktor prima.

Langkah-langkah Metode RSA terdiri dari menentukan kunci/membuat kunci, teknik proses enkripsi dan dekripsi dan Langkah-langkah dalam pertukaran kunci dengan menggunakan algoritma RSA Adalah Sebagai dihalaman Berikut :

1. Pilih dua bilangan prima,  $p$  dan  $q$  adalah bilangan yang dirahasiakan yang harus relatif prima dihitung dengan Pembagi Bersama Terbesar dan menghasilkan nilai 1.
  2. Hitung  $n = p * q$ ,  $n$  adalah modulo pembagi
  3. Hitung  $\phi(n) = (p - 1)(q - 1)$ ,  $\phi(n)$  adalah bilangan modulo pembagi untuk mencari nilai  $d$  yang relatif prima dengan nilai  $e$ , dihitung menggunakan Pembagi Bersama Terbesar (PBB) dan menghasilkan nilai 1
  4. Pilih sebuah bilangan bulat  $e$  untuk kunci publik harus relatif prima dengan nilai  $\phi(n)$ , dihitung menggunakan Pembagi Bersama Terbesar (PBB) dan menghasilkan nilai 1
  5. Hitung kunci dekripsi,  $d$ , dengan persamaan  $e * d \bmod \phi(n) = 1$ ,  
Hasil dari perhitungan algoritma di atas:
    1. Kunci publik adalah pasangan nilai  $(e, n)$
    2. Kunci privat adalah pasangan nilai  $(d, n)$
- Pengertian – Pengertian variabel :
1.  $p$  dan  $q$  adalah nilai bilangan Prima yang di sepakati oleh pengirim dan penerima *file* dan nilai itu di rahasiakan.
  2.  $n$  dan  $\phi(n)$  adalah modulo pembagi untuk mencari kunci *Private*
  3.  $e$  adalah Kunci Publik dan  $d$  adalah Kunci *Private*.

contoh soal yang pertama sebagai berikut:

$$\begin{aligned}
 p &= 5 & \text{Publik Key} &= 7 \\
 q &= 11 \\
 \text{Mencari Modulo pembagi} \\
 n &= P * Q \\
 \text{maka } 5 * 11 &= 55 & n &= 55 \\
 \phi(n) &= (P-1) (Q-1) \\
 \text{maka } (5-1) * (11-1) &= 40 & \phi(n) &= 40
 \end{aligned}$$

Private Key Karena nilai  $d$  tidak diketahui maka dilakukan pencarian nilai  $d$  dengan gcd dari nilai  $e$  (*Public Key*) dengan nilai  $\phi(n)$  menghasilkan nilai 1 .

$$\begin{aligned}
 e &= 7 & \phi(n) &= 40 \\
 \text{nilai } 1 &= 40/7 = 5 & \text{nilai } 2 &= 40 \\
 &= 7 * 5 = 35 & &= 1 * 5 = 5 \\
 &= 40-35 = 5 & &= 40 - 5 = 35
 \end{aligned}$$

Karena nilai 1 tidak mendapatkan hasil akhir 1 maka Nilai yang tadinya dibawah menjadi nilai yang berada di atas jadi dan hasil pengurangan dari nilai dibawah

$$\begin{aligned} \text{nilai 1} &= 7/5 = 1 & \text{nilai 2} &= 1 \\ &= 5 * 1 = 5 & &= 1 * 35 = 35 \\ &= 7 - 5 = 2 & &= 1 - 35 = -34 \\ & & &= -34 \bmod 40 \\ & & &= 6 \\ \text{nilai 1} &= 5/2 = 2 & \text{nilai 2} &= 35 \\ &= 2 * 2 = 4 & &= 2 * 6 = 12 \\ &= 5 - 4 = 1 & &= 35 - 12 = 23 \end{aligned}$$

Maka nilai d adalah 23 karena  $7 * 23 \bmod 40 = 1$

Contoh 2 menggunakan algoritma Euclidean:

$$\begin{aligned} p &= 89 & q &= 107 \\ e &= 3 \\ n &= 89 * 107 = 9523 \\ \phi(n) &= (89 - 1) * (107 - 1) = 9328 \\ \text{nilai 1} &= 9328/3 = 3109 & \text{nilai 2} &= 9328 \\ &= 3109 * 3 = 9327 & &= 3109 * 1 \\ & & &= 3109 \\ &= 9328 - 9327 & &= 9328 - 3109 \\ &= 1 & &= 6219 \end{aligned}$$

Private key = 6219

Perhitungan menggunakan algoritma Euclidean terkadang tidak bisa digunakan karena untuk mencari nilai d terkadang hasil sisa bagi yang di hitung tidak sesuai dengan hasil yang didapat dengan rumus  $e * d \bmod \phi(n) = 1$  dan tingkat kesulitan untuk mencari kunci dengan algoritma ini akan menyulitkan pengguna akan tetapi ada cara kedua untuk mencari nilai d (*Private Key*). Dengan cara menggunakan rumus  $e * d \bmod \phi(n) = 1$  dengan memulai angka d dari angka 1 misalkan kita ambil contoh nilai  $e = 7$  dan nilai  $\phi(n) = 40$  maka hasilnya akan sama dengan algoritma euclidean yaitu  $d = 23$ . Akan tetapi menggunakan metode yang kedua akan membutuhkan proses yang panjang bila menggunakan hitungan manual tergantung dari nilai p dan q yang digunakan semakin besar akan semakin sulit untuk mencari perhitungan nilai d.

Selanjutnya Langkah-langkah proses enkripsi dan dekripsi dalam metode rsa dapat dilakukan bila kunci *public* dan kunci *private* sudah di dapatkan/dihitung, dan tentukan nilai Pi. Rumus yang digunakan untuk proses enkripsi yaitu :

$$C_i = P_i^e \bmod n$$

dan bila nilai Ci didapatkan proses dekripsi bias dilakukan dengan rumus dibawah ini

$$P_i = C_i^d \bmod n$$

Keterangan :

- Pi = Plainteks
- Ci = Chiperteks
- e = Public Key
- d = Private Key
- n = modulo pembagi

Contoh soal yang pertama :

1. Proses Enkripsi
  - Pi = 32
  - n = 55
  - e = 7
  - d = 23

Hitung Ci

$$C_i = 32^7 \bmod 55 = 43$$

2. Proses Dekripsi

Hitung Pi

$$P_i = 43^{23} \bmod 55 = 32$$

Contoh soal yang kedua :

1. Proses Enkripsi

Pi = 17

n = 55

e = 7

d = 23

Hitung Ci

$$C_i = 17^7 \bmod 55 = 17$$

2. Proses Dekripsi

Hitung Pi

$$P_i = 17^{23} \bmod 55 = 18$$

### 3.5 Microsoft Visual Basic .Net 2012

*Microsoft Visual Basic* 2012 salah satu bagian dari produk pemrograman yang di keluarkan oleh *Microsoft*, yaitu *Microsoft Visual Studio* 2012. Sebagai produk pengembangan atau *Integrated Development Environment* (IDE) yang di keluarkan oleh *Microsoft*, *Microsoft Visual Studio* 2012 berisi beberapa IDE seperti *Visual Basic*, *Visual C++*, *Visual Web Developer*, *Visual C#* dan *Visual F#*. Semua IDE tersebut sudah mendukung penuh implementasi *.Net Framework* terbaru, yaitu *.Net Framework* 4.0 yang merupakan pengembangan dari *.Net Framework* 3.5. Adapun *database* standar yang di sertakan adalah *Microsoft SQL Server* 2008 *Express*.

### 3.6 Tahapan Pengembangan Prototype

Menurut Simarmata (2010), metode *prototype* secara formal telah di perkenalkan pada komunitas sistem informasi pada awal tahun 1980-an, model ini di gunakan untuk mengatasi kelemahan model air terjun. Jika dirasa "cukup baik" barulah pengembang akan membuat *program* sesungguhnya berdasarkan model *prototype*.

Menurut Aji (2006), *Prototyping* merupakan salah satu metode pengembangan perangkat lunak yang banyak digunakan. Dengan metode *prototyping* ini pengembang dan pelanggan dapat saling berinteraksi selama proses pembuatan sistem.

Sering terjadi seorang pelanggan hanya mendefinisikan secara umum apa yang dikehendakinya tanpa menyebutkan secara detail *output* apa saja yang dibutuhkan, pemrosesan dan data-data apa saja yang dibutuhkan. Sebaliknya disisi pengembang kurang memperhatikan efisiensi algoritma, kemampuan sistem operasi dan *interface* yang menghubungkan manusia dan komputer.

Harus dibutuhkan kerjasama yang baik diantara keduanya sehingga pengembang akan mengetahui dengan benar apa yang diinginkan pelanggan dengan tidak mengesampingkan segi-segi teknis dan pelanggan akan mengetahui proses-proses dalam menyelesaikan sistem yang diinginkan. Dengan demikian akan menghasilkan sistem sesuai dengan jadwal waktu penyelesaian yang telah ditentukan.

Kunci agar model *prototype* ini berhasil dengan baik adalah dengan mendefinisikan aturan-aturan main pada saat awal, yaitu pelanggan dan pengembang harus setuju bahwa *prototype* dibangun untuk mendefinisikan kebutuhan. *Prototype* akan dihilangkan sebagian atau seluruhnya dan perangkat lunak aktual direayasa dengan kualitas dan implementasi yang sudah ditentukan.

Tahap Tahap *Prototype* Dalam Aplikasi Enkripsi RSA ini menggunakan metode *prototype system*, adapun langkah-langkah atau tahapannya adalah sebagai berikut:

#### 1. Pengumpulan Kebutuhan

Pelanggan dan pengembang bersama-sama mendefinisikan format seluruh perangkat lunak, mengidentifikasi semua kebutuhan, dan garis besar sistem yang akan dibuat.

#### 2. Membangun *Prototyping*

Membangun *prototyping* dengan membuat perancangan sementara yang berfokus pada penyajian kepada pelanggan (misalnya dengan membuat *input* dan *output*).

#### 3. Evaluasi *Prototyping*

Evaluasi ini dilakukan oleh pelanggan apakah *prototyping* yang sudah dibangun sudah sesuai dengan keinginan pelanggan. Jika sudah sesuai maka langkah 4 akan diambil. Jika tidak *prototyping* direvisi dengan mengulangi langkah 1, 2, dan 3.

#### 4. Mengkodekan Sistem

Dalam tahap ini *prototyping* yang sudah di sepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.

#### 5. Menguji Sistem

Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus dites dahulu sebelum digunakan. Pengujian ini dilakukan dengan *Beta Testing*, *Black Box* dan *White Box*.

#### 6. Evaluasi Sistem

Pelanggan mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan. Jika ya, langkah 7 dilakukan; jika tidak, ulangi langkah 4 dan 5.

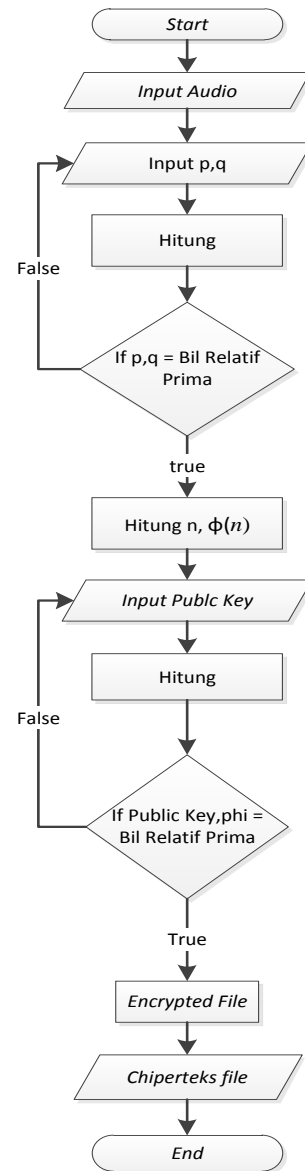
#### 7. Menggunakan Sistem

Perangkat lunak yang telah diuji dan diterima pelanggan siap untuk digunakan.

### 4. RANCANGAN SISTEM ATAU APLIKASI

Perancangan aplikasi Enkripsi Data Audio Menggunakan Metode Kriptografi RSA ini menggunakan *Flowchart* sebagai salah satu cara untuk mempermudah dalam pembuatan aplikasi ini.

1 Di bawah ini adalah tahapan *flowchart* tahapan enkripsi *file audio* :

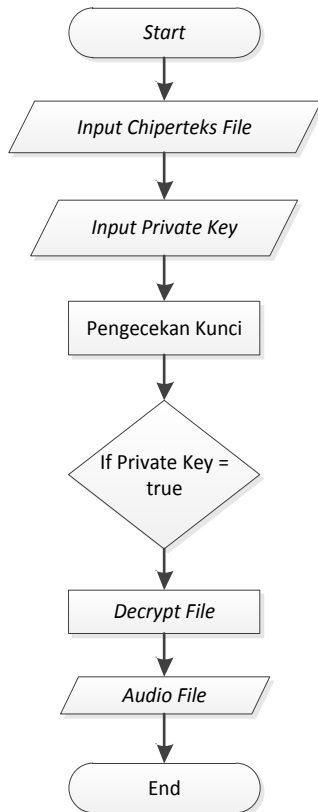


**Gambar 3 Flowchat Enkripsi File Audio**

*Flowchart* Enkripsi *File Audio* Merupakan gambaran keseluruhan diagram alir proses enkripsi yang dimulai dari memasukan *file audio* yang ingin di enkripsi, setelah *file* sudah dipilih di lanjutkan dengan memasukan kunci yang digunakan pada tahapan ini kunci yang dimasukan adalah *p* dan *q* bilang prima yang di rahasiakan selanjutnya nilai *p* dan *q* di hitung untuk mengetahui dua nilai tersebut relatif prima atau tidak bila nilai relatif prima maka proses dilanjutkan akan tetapi bila nilai tidak relatif prima proses akan kembali ke *input p* dan *q*, proses kedua masukan juga *public key*. Setelah memasukan kunci selesai lanjut ke proses hitung nilai *e* (Kunci Publik),  $\phi(n)$  proses ini digunakan untuk menentukan kunci yang akan digunakan bisa dipakai atau tidak tergantung dari nilai  $\phi(n)$  apakah relatif prima untuk bilangan *e* (*Public Key*) jika nilai *e* relatif prima terhadap  $\phi(n)$  maka proses enkripsi bisa dilakukan dan jika tidak relative prima terhadap  $\phi(n)$  maka aplikasi

tidak akan melanjutkan ke proses selanjutnya. Proses yang terakhir proses enkripsi disini nilai *plainteks audio* akan di ubah sesuai metode RSA untuk di jadikan *file chiperteks* setelah proses selesai maka sistem akan menghasilkan *output file chiperteks* dan semua proses sudah selesai dilakukan.

2. Di bawah ini adalah tahapan *flowchart* tahapan Dekripsi *file audio* :



**Gambar 4 Flowchart Tahapan Dekripsi File Audio**

Pada Gambar 4 *Flowchart* Dekripsi *File Audio* Merupakan gambaran keseluruhan diagram alir proses Dekripsi yang dimulai dari memasukan *file* yang telah terenkripsi (*Chiperters File*) setelah *file* sudah dipilih di lanjutkan dengan memasukan kunci yang digunakan pada tahapan ini kunci yang dimasukan adalah *p* dan *q* bilang prima yang di rahasiakan dan diikuti pula dengan *Private Key* pada aplikasi. Selanjutnya proses *Count Private Key* adalah proses menentukan apakah kunci *private* yang di gunakan itu benar atau tidak dengan dilakukannya perhitungan ulang pada nilai *p* dan *q* dan mangasilkan nilai *n* dan *phi n* maka di faktorkan nilai *d* (*Private Key*) untuk mencari nilai *e* (*Public Key*) bila hasilnya sesuai maka proses dekripsi bisa dilakukan. Dan di tahap terakhir proses pendekripsian *file* yang dimana *file chiperteks* dikembalikan seperti *file* awal dan aplikasi menghasilkan *output file audio* yang tadinya terenkripsi dan semua proses selesai.

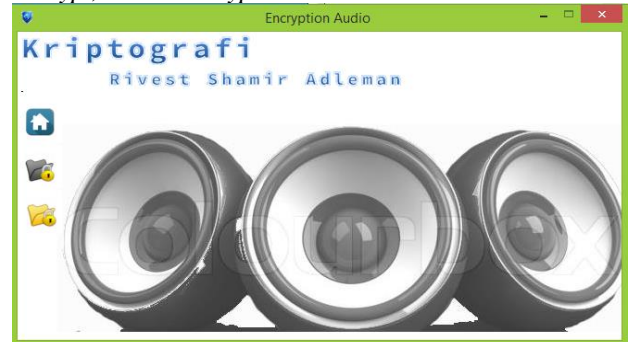
## 5. IMPLEMENTASI

Tahap implementasi atau tahap membangun sistem menjadi nyata dan mengkodekan aplikasi. Adalah Tampilan Antarmuka (*Interface*)

merupakan tahapan yang bertujuan mengubah hasil dari rancangan sistem menjadi bentuk nyata. Pada saat pertama kali aplikasi dijalankan maka akan muncul sebuah tampilan seperti gambar di bawah ini.

### 1. Tampilan Awal *From*

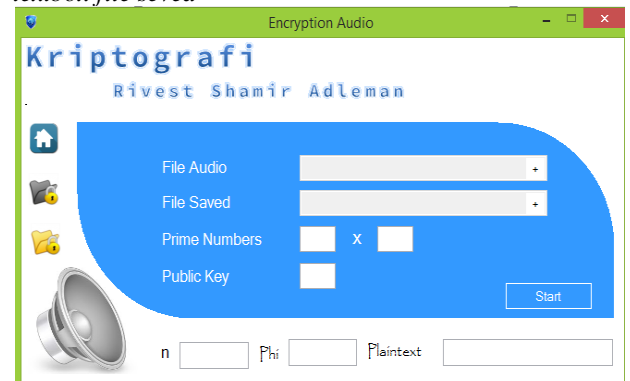
Di bawah ini adalah tampilan awal *from* enkripsi *file audio* yang menampilkan tiga *button* *Home*, *button encrypt*, *button decrypt*.



**Gambar 5 Tampilan Awal *From***

### 2. Tampilan *Frame* Enkripsi

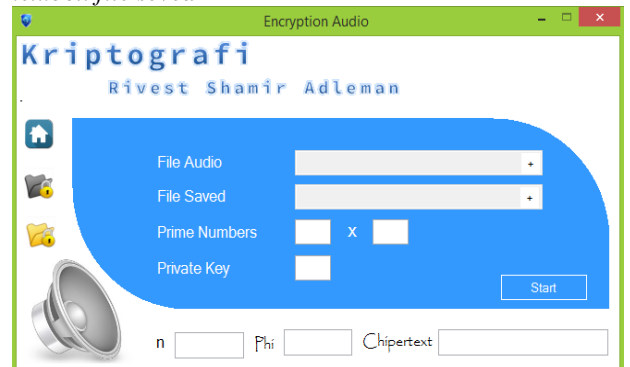
Tampilan *Frame* Enkripsi bila di aktifkan terdapat dua *button* tambahan yaitu *button Browse* yang terdapat dalam *textbox file audio* dan *button change* didalam *textbox file seved*



**Gambar 6 Tampilan *Frame* Enkripsi**

### 3. Tampilan *Frame* Dekripsi

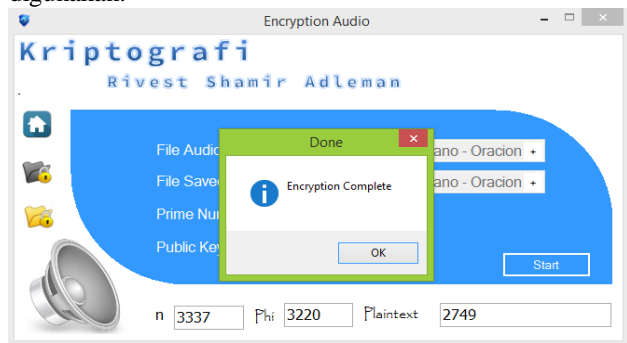
Tampilan *Frame* dekripsi bila di aktifkan terdapat dua *button* tambahan yaitu *button Browse* yang terdapat dalam *textbox file audio* dan *button change* didalam *textbox file seved*



**Gambar 7 Tampilan *Frame* Dekripsi**

#### 4. Percobaan Menggunakan Aplikasi

Pengujian algoritma kunci yang digunakan bertujuan untuk mengetahui apakah aplikasi yang dibangun ini telah sesuai dengan metode dan algoritma yang digunakan.

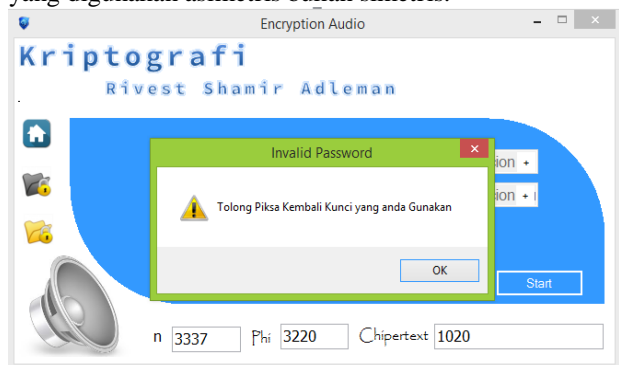


**Gambar 8 Percobaan Enkripsi Menggunakan Kunci Publik**

Proses Enkripsi berhasil dengan menggunakan variabel nilai sebagai berikut :

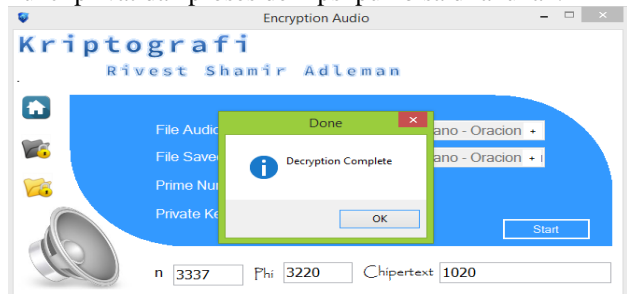
$p = 47$   $q = 71$  Publik Key = 79

Percobaan selanjutnya proses dekripsi yang menggunakan kunci yang sama/kunci publik dan aplikasi tidak mengizinkan kunci tersebut karena algoritma kunci yang digunakan asimetris bukan simetris.



**Gambar 9 Percobaan Dekripsi Menggunakan Kunci Publik**

Percobaan selanjutnya proses dekripsi menggunakan kunci privat dan proses dekripsi pun bisa dilakukan.



**Gambar 10 Proses Dekripsi Menggunakan Kunci Private**

#### 6. KESIMPULAN

Dengan adanya hasil penelitian yang dilakukan dan berdasarkan uraian uraian yang dibahas dalam bab-bab sebelumnya, maka dapat ditarik kesimpulan bahwa :

1. Metode RSA yang diaplikasikan pada aplikasi ini dapat berfungsi dengan baik serta menggunakan kunci publik untuk enkripsi dan kunci privat untuk dekripsi.
2. Waktu yang digunakan untuk melakukan proses enkripsi tergantung dari Ukuran *File* dan digit kunci yang digunakan.
3. Enkripsi tidak dapat berfungsi pada format audio yang terdaftar di aplikasi.
4. Semakin besar ukuran *file audio*, maka semakin lama waktu estimasi proses enkripsi, hal ini dikarenakan lamanya proses perhitungan.
5. Implementasi enkripsi rsa *file audio* hanya menghasilkan *file* yang berbeda yang tak bisa digunakan sama sekali.

#### 7. SARAN

Adapun saran-saran yang dapat dikemukakan yaitu sebagai berikut :

1. Dalam Pemrosesan Enkripsi dan Dekripsi kunci yang digunakan masih tergolong biasa maka lebih baik lagi untuk kunci di enkripsikan menggunakan metode yang berbeda.
2. Aplikasi dapat di kembangkan ke operasi sistem *handphone* seperti android dan *windows phone*.
3. Kunci yang digunakan dalam implementasi masih tergolong lemah karena pendeknya jumlah karakter, maka diharapkan untuk penelitian yang lebih lanjut ada metode pembangkitan karakter.

#### 8. DAFTAR PUSTAKA

- A.A. Ngurah Pardnya Andika, I Ketut Gede Suhartana S.Kom.,M.Kom, I Made Widiartha S.Si.,M.Kom, 2012, *Enkripsi dan Dekripsi Audio Format AMR Dengan Algoritma Kriptografi Loki97*, Jurnal Ilmu Komputer FMIPA, Universitas Udayana
- Aji, Andika Kusuma. 2006. *Perancangan Prototype Knowledge Management System Untuk Pengelolaan Pengetahuan Rumah Sakit*. ITS. Surabaya
- Ariyus, Dony. 2008. *Pengantar Ilmu Kriptografi*. Penerbit : ANDI. Yogyakarta.
- Arif 2008. *Kriptografi visual pada biner dan citra berwarna serta pengembangannya dengan setenografi*
- Avianto Kurdi, Gerry 2015, *Enkripsi Data Keuangan Menggunakan Metode Deffie-Hellmen*, STMIK Widya Cipta Dharma Samarnda
- Harahap , Sofyan Syafri. 2006. *Analisis Kritis Atas Laporan Keuangan, Edisi Satu*, PT Raja Grafindo Persada, Jakarta
- Jogiyanto. 2010. *Analisis dan Desain Sistem Informasi*. Yogyakarta: Penerbit Andi.
- Kristanto, Andri 2007, *Perancangan sistem informasi dan aplikasinya* , Yogyakarta:Gava Media, 2007
- Mulyanto , Agus. 2009. *Sistem Informasi Konsep dan Aplikasi*. Pustaka Pelajar.
- Munir, Rinaldi 2012, *Matematika Diskrit Revisi Kedua*, Informatika, Bandung

- Nugroho Adi. *Rekayasa Perangkat Lunak Berorientasi Objek Dengan Metode USDP*. Yogyakarta: Andi, 2010
- Prayudi, Yudi, Idham Halik. 2005. *Studi Analisis Algoritma Rivest Code 6 (RC6) Dalam Enkripsi/Dekripsi Data*. Seminar Nasional Aplikasi Teknologi Informasi 2005 (SNATI 2005), Yogyakarta.
- Simarmata. Janner. 2010, *Rekayasa Perangkat Lunak*, Andi Offset, Yogyakarta.
- Sadiman, dkk. 2008. *Media Pendidikan, Pengertian, Pengembangan dan Pemanfaatan Manajemen*. Jakarta : Raja Grafindo Persada
- Tri Rahajoeningroem, Muhammad Aria, 2011, *Studi Dan Implementasi Algoritma RSA untuk Pemangamanan Data Transkrip Akademik Mahasiswa*, Jurnal Teknik Elecktro majalah ilmiah unikom Vol 8, Universitas Komputer Indonesia
- Wahana kompuer 2013, shourtcourse : *visual basic 2012 programing*
- Widya, Abdul Kadir. 2009. *Dasar Perancangan dan Implementasi Database Relasional*. Yogyakarta : Andi Offset.
- Yunizar, Muhammad. 2011. *Teknik Uji Coba Black Box dan White Box*. Universitas Gunadarma. Jakarta