VOL. 16 NO. 3 SEPTEMBER 2015

ISSN : 1411-3201

Jurnal Ilmiah DASI

DATA MANAJEMEN DAN TEKNOLOGI INFORMASI

STMIK AMIKOM
YOGYAKARTA

VOL. 16 NO. 3 SEPTEMBER 2015
JURNAL ILMIAH
Data Manajemen Dan Teknologi Informasi

Terbit empat kali setahun pada bulan Maret, Juni, September dan Desember berisi artikel hasil penelitian dan kajian analitis kritis di dalam bidang manajemen informatika dan teknologi informatika. ISSN 1411-3201, diterbitkan pertama kali pada tahun 2000.

PENANGGUNG JAWAB :
Ketua STMIK AMIKOM Yogyakarta, Prof. Dr. M. Suyanto, M.M.

ALAMAT PENYUNTING & TATA USAHA
STMIK AMIKOM Yogyakarta, Jl. Ring Road Utara Condong Catur Yogyakarta, Telp. (0274) 884201 Fax. (0274) 884208, Email : jurnal@amikom.ac.id

BERLANGGANAN
Langganan dapat dilakukan dengan pemesanan untuk minimal 4 edisi (1 tahun) pulau jawa Rp. 50.000 x 4 = Rp. 200.000,00 untuk luar jawa ditambah ongkos kirim.

DAFTAR ISI

ANALISIS DAN IMPLEMENTASI ALGORITMA KRIPTOGRAFI KUNCI PUBLIK RSA DAN LUC UNTUK PENYANDIAN DATA

Bayu Setiaji

Teknik Informatika STMIK AMIKOM Yogyakarta
email : bayusetiaji@amikom.ac.id

Abstract

The use of data communication networks between various computer systems have developed rapidly in various fields, so that the necessary existence of other systems maintain the confidentiality and security of data exchange.

In handling the security and confidentiality of the data used to perform cryptographic algorithms for data encryption, which is to transform the data into a form that can not be understood by people who are not entitled to receive such data. Therefore, in addition to the key confidentiality, reliability cryptographic algorithms used j'uga affect the reliability of handling security and confidentiality of data in the data communication system.

Cryptographic systems, based on the number of key usage, can be divided into two, namely, first, conventional cryptography uses a single key and a second public key cryptography that uses two keys. Conventional cryptography is processing the data using a single key and algorithm based on the substitution and permutation. Public key cryptography is the processing of data by using two separate key and algorithm based on mathematical functions.

Public key cryptography algorithms most widely used today is the algorithm Rivest Shamir Adleman (RSA). In the further development of public-key cryptography algorithm developed called Lucas (LUC), which is based on large integers on a series lucas. LUC is an alternative algorithm other than RSA algorithm.

In this study will be discussing the use of public key cryptography algorithm, which is a comparative study of the use of the RSA algorithm and LUC algorithms for data encryption. Which is expected to get a public RSA algorithm is faster and valid in terms of the RSA key generation algorithm is still superior in data encryption.

***Keywords* :**
Cryptography, Network Security, Encryption, Mathematics, Lucas,.

Pendahuluan

Latar Belakang Masalah

Dengan semakin meningkatnya pemakaian jaringan komputer untuk komunikasi data, maka penanganan keamanan dan kerahasiaan data menjadi hal yang sangat panting dan khusus. Salah satu cara menjaga keamanan dan kerahasian data pada jaringan komunikasi data adalah dengan menggunakan sistem kriptografi.

Pada sistem kriptografi digital pada saat int dikenal dua metode, yaitu konvensional dan kunci publik. Pada sistem kriptografi konvensional, proses untuk mengubah bentuk *plaittlext* ke bentuk *chiperlext* dan proses kebalikannya memerlukan sebuah kunci yang sama, yang mutlak harus dijaga kerahasiannya. Pada sistem kriptografi kunci publik untuk proses tersebut diperlukan dua buah kunci, yaitu satu kunci untuk enkripsi data yang tidak dirahasiakan (disebut kunci publik) , dan satu kunci lagi untuk dekripsi data yang harus dijaga kerahasiaannya (disebut kunci privat).

Untuk meningkatkan kehandalan keamanan data pada Jaringan komputer, banyak dikembangkan sistem kriptografi baru. Salah satunya adalah menggunakan algoritma kunci publik berdasarkan deret *Lucas,* yaitu algoritma *LUC(lucas),* yang kehandalan keamanan datanya dapat dikatakan sebanding dengan algoritma RSA, teknik yang telah banyak digunakan dalam berbagai aplikasi dan diakui kekuatan keamanannya.

Tinjauan Pustaka

Pendahuluan

Kriptografi *(Cryptography)* adalah ilmu yang mcmpelajari teknik pcnyandian data secara rahasia yang bertujuan untuk menghindari manipulasi terhadap data oleh orang yang berhak berhak. Secara singkat dapat dituliskan bahwa kriptografi adalah ilmu orang yang mempelajari penulisan data secara rahasia.[2]

Ilmu kriptografi, termasuk cukup tua usianya, secarah perkcmbangannya dapat dibagi dua periode, yaitu periode klasik dan modem. Pada pcriodc klasik, tcknik sederhana pernah dilakukan dalam perang *Gallic, oleh Julius Caesar,* yaitu mengganti huruf-huruf melalui pergeseran huruf-huruf tersebut. Teknik lainnya pada periode ini antara lain *Polybius squre.* Pada periode modem, awalnya kriptografi

hanya dipcrgunakan oleh pemerintahan untuk keperluan militer.[2][3]

Ketika *National Bureau of Standard* (NBS) pada tahun 1975 memperkenalkan *Data Encryption,Staridard* (DES) kepada publik di Amcrika, maka kriptografi mulai diterapkan secara luas. Adanya pengembangan metode baru pada teknik kriptografi, mcnyebabkan DES sebagai pelopor, digolongkan sebagai mctode konvensional.

Metode kriptografi yang berbeda dcngan DES, ditemukan pada awal tahun 70an oleh *National seuirity Agnicy* (NSA), dinamakan kriptografi kunci publik, pertengahan tahun 70-an metode tcrsebut dibuatkan notasinya dan dipublikasikan oleh *Martin Hellman* dan *Whioeld Diffie.* Sejak saat itu pengembangan kriptografi kunci publik mulai dilakukan. Salah satu perkembangan penting kriptografi kunci publik adalah RSA, yang dibuat tahun 1977 (dipatenkan awal tahun 80-an). RSA yang dibuat olch *Rivest, Shamir dan Adleman,* merupakan metodc kriptografi kunci publik yang paling banyak dipakai untuk komunikasi data sampai saat ini. [4][3]

Elemen Dasar Sistcm Kriptotarfi Untuk Mentransformasikan Data adalah :

1. Algoritma kriptografi
   Merupakan satu set peraturan atau langkah-langkah yang tetap dalam melakukan transformasi. Algoritma berfungsi mclakukan transformasi terhadap data.
2. Kunci kriptografi
   Merupakan satu set variabel yang terdiri dari urutan bit untuk mentransformasikan data. Kunci ini berfungsi sebagai pengontrol dari transformasi.

Bcrbeda dengan algoritma kriptografi yang kebanyakan bersifat tetap, bahkan ada yang distandarkan, dalam pcmakaian kunci ini scringkali diubah-ubah untuk alasan keamanan.

Konsep Dasar Kriptografi

Selain dampak positif dari perkembangan Jaringan komputer, komunikasi data pada medium yang terbuka bagi masyarakat luas *(open system),* sangat memungkinkan terjadinya penyadapan dan pengubahan data oleh pihak yang tidak sah. Untuk menjamin kerahasiaan dan keaslian data, maka digunakan teknik kriptografi yang melakukan transformasi terhadap data sehingga data yang dihasilkan tidak dapat dimengerti oleh pihak ketiga. Ada dua elemen dasar sistem kriptogarfi untuk mentransformasikan deretan bit data asli menjadi deretan bit data sandi. [2][3]

Istilah enkripsi dipakai untuk menyatakan transformasi bersifat algoritma yang dilakukan simbol per simbol atau bit per bit terhadap data, yang dilakukan dari sisi pengirim. Adapun istilah dekripsi adalah transformasi balikan yang dilakukan pada sisi penerima. Masukan untuk algoritma pada proses enkripsi adalah data asli *(plaintext),* sedangkan keluarannya adalah data yang telah ditransformasikan yang disebut data *sandi (ciphertext).*

Proses dekripsi yang dilakukan di sisi penerima bertujuan memperoleh kcmbali data asli dari data sandi. Dikarenakan informasi yang dikirimkan dari pengirim kepada penerima adalah dalam bentuk data sandi, maka orang yang melakukan pcnyadapan pada media transmisi tidak akan memahami isi data yang didapatnya. Usaha untuk memecahkan sistem kriptografi juga dilakukan orang , usaha itu dinamakan kriptanalisis *(cryptanalysis),* orang yang melakukannya disebut kriptanalis *(cryptana-lyst).[3]*[4]

Sistem kriptografi yang berhasil dari segi keamanannya dapat diklasifikasikan menjadi dua bagian. Asumsi dasar yang dipakai dalam merancang sistem kriptografi adalah bahwa seorang kriptanalis mengetahul keseluruhan mekanisme enkripsi kecuali kuncinya.

Adapun Klasifikasi Sistem Kriptografi Yang Berhasil Dari Segi Keamanannya adalah sebaga berikut : [1]

1. Aman yang tidak bergantung pada kondisi *(unconditionally secure)*
   Yaitu jika jumlah informasi yang tersedia bagi kriptanalis selalu tidak cukup untuk memecahkan sistem kriptografi, tidak peduli sebanyak apapun dan sekuat apapun perhitungan yang dilakukan.
2. Aman secara perhitungan *computational secure)*
   Yaitu keadaan sistem kriptografi dengan tingkat keamanan yang lebih rendah. Sistem kriplografi yang dikatakan aman secara perhitungan pada jangka waktu dan kondisi tertentu dapat mcnguntungkan bagi kriptanalis. Sistem kriptografi ini mengandung infomasi yang secara unik cukup untuk mcnemukan data asli dan kunci dalam kurun waktu tertentu (misalnya x tahun), walaupun belum dapat dipecahkan sebelum kurun waktu tersebut (>x tahun).

Biarpun Sistem *unconditionally secure* mcmang tcrbukti lebih aman, tetapi sejumlah kunci yang diperlukannya mcmbuatnya tidak mungkin diterapkan pada sebagian besar aplikasi. Dalam kenyataannya orang lebih scring menggunakan sistem *computational secure* karena lebih mudah diimplementasikan baik secara perangkat keras maupun lunak. Dengan menggunakan kunci sebesar beberapa ratus digit, orang akan mendapatkan suatu sistcm yang hampir mirip dengan *unconditionally secure,* karena sumber daya yang diperlukan oleh kriptanalis untuk memecahkan sistem terscbut akan sangat besar.[4][4]

Sistem Kriptografi

Ada bermacam teknik dalam sistem kriptografi yang dikembangkan untuk mengamankan sistem komunikasi. Tetapi semua itu dapat dikclompokkan menjadi dua kategori dasar, yaitu penyandian analog, yang merupakan kriptografi klasik yang saat ini sudah tidak dipakai lagi dalam komunikasi komputer, dan penyandian digital, yang merupakan kriptografi

modem. Penyandian analog diperoleh dengan melakukan berbagai manipulasi tcrhadap sinyal analog, khususnya dengan operasi yang mempe-ngaruhi waktu dan frequensi darl sinyal. Sedangkan penyandian digital diperoleh dengan perhitungan yang melibatkan sinyal analog yang telah dikon-versikan ke dalam bentuk digital.[3]

Berikut ini akan dibahas hanya sistem kriptografi modern. Sccara umum sistem kriptografi modern berdasarkan manajemen kunci yang dipakai dapat dikelompokan menjadi dua, yaitu :[3][4]

1. Sistem kriptografi konvensional; disebut sistem simetris, contohnya adalah DES, FEAL, LDEA.
2. Sistem kriptografi kunci publik; disebut system asimetris. Contohnya adalah RSA, skema *Diffe-Hellman, Knapsack,* DSS, skcma *EIgamal, LUC.*

Sistem Kriptografi Konvensional

Dalam sistem ini kunci enkripsi dan dekripsi bersifat identik, dan harus dijaga kerahasiaannya. Hanya pemakai yang sah yang dapat mcngggunakan kunci tcrsebut untuk penyandian data. Sebelum komunikasi berlangsung, terlebih dahulu kunci rahasia perlu didistribusikan antara pengirim dan penerima data. Dengan kata lain kedua pihak yang terlibat harus memegang kunci rahasia dari transformasi kriptografi, schingga disebut sistem yang simetrik. Dalam sistem kriptografi konvensional, prosedur E mcmproses data asli P dengan kunci K untuk menghasilkan data sandi C.[1][3]

$$E[K,P] = C$$

Prosedur D dengan kunci K, akan mengubah data sandi C menjadi data asli *P.*

- $D[K,C] = P$

Metode Penelitian

Sistem Kriptografi RSA

Sistem kriptografi RSA adalah salah satu sistem kriptografi kunci publik yang ditemukan oleh Rivest, Shamir dan Adleman dari MIT. Sejak skema sistem ini ditemukan, sistem ini menguasai sebagai satu-satunya sistem yang diterima dan diterapkan secara luas sebagal sistem kriptografi kunci publik.[1]

Sistem ini termasuk sistem enkripsi blok, karena data asli dan data sandi adalah bilangan integer antara 0 sampai (n -1), untuk semua nilal n positif. Keamanan sistem kriptografi ini bergantung kondisi tertentu. Seperti kebanyakan sistem kriptografi kunci publik lainnya, harus ada cara yang mudah diterapkan untuk membangkitkan pasangan kunci enkripsi dan dekripsinya, sehingga setiap pemakai dapat membangkitkan sebuah pasangan tanpa perlu mempersoalkan kemampuan matematiknya. Dalam skema RSA ini, algoritma pembangkitan kunci pertama kali akan memilih dua bilangan prima besar p dan q.[4]

Kondisi Untuk Keamanan Sistem Kriptografi, Yaitu :

1. Sulitnya menarik akar modular (yaitu, $c^{1/4}$ yang dalam hal ini $c = m^e \bmod n$, dengan e diketahui).
2. Sulitnya memfaktorkan bilangan besar n (yaitu, $n = pq$, dan kemudian menemukan $d = e^{-1}$ dengan pcngetahuan tentang $\varphi(n) = (p-1)(q-1)$;
3. Sulitnya menghitung logaritma modular (yaitu. $d = \log_m s$ dengan $s = m^d \bmod n$ adalah signature terhadap suatu data asli m);

Dengan membuat setiap faktor p dan q memiliki panjang ratusan digit, maka pemfaktorannya diperkirakan akan memakan waktu sampai puluhan tahun dengan menggunakan algoritma tercanggih.[2]

Landasan Matematis untuk Algoritma RSA

Sebelum mulai membahas keseluruhan dari algoritma kriptografi, terlebih dahulu akan dibahas landasan matematisnya. Di sini akan dibuktikan kebenaran algoritma dekripsi menggunakan sebuah identitas dari euler dan Thermal untuk sebuah interger data M yang prima relatif terhadap n.[2]

$$M^{\varphi(n)} \cong I \pmod n$$

Di sini j (n) adalah euler tolient function yang menyatakan jumlah bilangan bulat positif kurang dari n yang prima relatif terhadap n. Disini, kita pakai sifat mendasar dari euler's totient function,untuk bilangan prima p:[2]

$$\varphi(p) = P-1$$

$$(n) = \varphi(p) * \varphi(q)\ (p-1)*(q-1) = n-(p+q)+I \quad (3.3)$$

Karena d relatif terhadap j (n), la mempunyai invers multiplikatif e,

$$e.d \cong I \pmod{\varphi(n)}$$

Sekarang dapat dituliskan bahwa persamaan dekripsi data

$$D(E(M)) = M$$

dan enkripsi data

$$E(D(M)) = M$$

dapat dipenuhi (yaitu bahwa dekripsi bekerja benar bila e dan d dipilih dengan cara demikian), maka terbentuk

$$D(E(M,),) = (E(M))^d = (M^e)^d \bmod n = M^{ed} \bmod n$$

$$E(D(M)) = (D(M))^e = (M\ d)^e \bmod n = M^{ed} \bmod n$$

$$M^{e.d} \bmod n = M^{k\,\varphi(n)+1} \quad \text{(untuk suatu integer k)}$$

Dari persamaan diatas bisa dilihat bahwa untuk semua M sedemikian sehingga p tidak membagi M dan karena (p-1) membagi $\varphi(n)$

$$M^{p-1} \cong I9 \bmod p)$$

$$M^{k\varphi(n)+1} \cong M \pmod n$$

ini benar secara trivial ketika M= 0(mod p), sehingga persamaan ini terpenuhi untuk semua M. Dengan cara yang sama, Untuk q dihasilkan

$$M^{k.\varphi(n)+1} \cong M \pmod n$$

Bersama dua persamaan terakhir mengakibatkan bahwa untuk semua

Me.d ≅ Mk. ≅ (n)+1 ≅ M(mod n).

Algoritma Pembangkitan Kunci

Sebelum melakukan apilkasi sistem kriptografi kunci publik, setiap penggunanya harus membangkitkan sepasang kunci.

Sebelum melakukan aplikasi sistem kriptogtafi kunci publik, setiap penggunanya harus membangkilkan sepasang kunci. Pembangkitan kunci tersebut harus berdasarkan kepada permintaan sebagai berikut:

1. Menentukan dua bilangan prima yang besar, *p* dan *q*
2. Memilih kunci publik e terlebih dahulu, kemudian menghitung kunci privat d, atau sebaliknya

Memilih Kunci Private d

Sangat mudah untuk memilih suatu bilangan d yang prima relatif terhadap φ (n). SebagaI contoh, suatu bilangan prima yang lebih besar daripada maksimum antara p dan q akan memenuhi. Adalah penting bahwa d harus dipilih dari suatu himpunan yang cukup besar sehingga seorang kriptanalis tidak dapat menemukannya dengan pencarian secara langsung.[2]

Algoritma Enkripsi dan Dekripsi

Untuk mendekripsi suatu data asli M dengan metode ini, digunakan suatu kunci publik (*e*,n) dengan *e* dan n adalah sepasang integer positif.

Pertama, data asli ditampilkan sebagai suatu bilangan bulat antara 0 dan n-1 (data asli yang panjang dipecah menjadi sederetan blok, yang dinyatakan sebagai sebuah integer). Tujuan proses ini tidak untuk menyandikan data asli tersebut, tapi hanya untuk menampilkan dalam bentuk angka yang diperlukan untuk enkripsi.[2]

Kemudian, data asli disandikan dengan dipangkatkan oleh e dan dimodulo dengan n. Hasilnya (data sandi C) adalah sisa ketika Me dibagi dengan n.

$C = E(M) = M^e \bmod n$, untuk data asli M

Untuk mendekripsikan data sandi tersebut, C dipangkatkan dengan d kemudian di-modulo lagi dengan n.

$M = D(C) = C^d \bmod n$, untuk data sandi C

Jika diperhatikan, enkripsi tidak meningkatkan ukuran sebuah data asli, keduanya, data asli dan data sandi adalah integer dalam Jangkauan 0 s/d n-1.

Kunci enkripsi adalah sepasang integer positif (e,n). Dengan cara yang sama, kunci dekripsi adalah juga sepasang integer positif (d,n). Setiap pemakai mengumumkan kunci enkripsinya, dan merahasiakan kunci dekripsinya. Untuk mendapatkan pasangan bilangan kunci enkripsi dan dekripsi dengan metode ini, maka cara yang harus dilakukan adalah sebagal berikut : [2]

pertama menghitung n sebagai hasil perkalian dua buah bilangan prima random p dan q yang sangat besar,

n = p.q

Meskipun n diumumkan sebagai kunci publik, faktornya p dan q akan secara efektif tersembunyi dari orang lain, karena sulitnya memfaktorkan n, ini juga akan menyembunyikan cara d dapat diturunkan dari e. [3]

Kedua mengaambil integer d secara random dan besar, yang prima relatif terhadap (p-1)(q-1)

fbt(d,(p-1)(q- 1))

Integer e akhirnya dihitung dari p, q, dan d sebagai "mullitiplicative invers" dari d modulo (p-1)(q-1), sehingga didapat

e.d ≅ I(mod(p-1)((q-1))

Pada landasan matematis telah dibuktikan bahwa E dan D adalah permutasi invers. Menghitung Me (mod n) membutuhkan paling banyak 2 $\log_2$ (e) perkalian dan $2\log_2(e)$ pembagian menggunakan prosedur tertentu.

Contoh Sederhana Penyandian Data

Contoh sederhana penyandian data menggunakan algoritma kriptografi kunci publik *RSA,* Sebagai Berikut :

Contoh Sederhana Penyandian Data Mengunakan Algoritma RSA.[2][3]

Misalkan suatu kasus, p = 47, q = 59, n = p.q = 47.59 = 2773, dan d = 157. Maka φ (2773) 46.58

2668, dan e dapat dihitung sebagai berikut :

$x_o = 2668$, $a_o = 1$, $b_o = 0$

$x_1 = 157$, $a_1 = 0$, $b_1 = 1$

$x_2 = 156$, $a_2 = 1$. $b_2 = -16$ (karena 2668 = 157.16 + 156)

$x_3 = 1$ $a_3 = -1$, $b_3 = 17$ (karena 157 = 1. 156 + 1)

Karena itu e = 17, adalah invers multiplikatif dari d. Dengan n = 2773. Sehingga dapat mengkodekan dua huruf per blok, mengganti dengan bilangan dua digit untuk setiap blank = 00, A = 01, B = 02, ..., Z = 26. Misalkan data aslinya, adalah :

ITS ALL GREEK TO ME

Maka data sandinya menjadi :

0920	1900	0112	1200	0718
	0505	1100	2015	0013
	0500			

Karena e = 10001 dalam biner, blok pertama (M 920)dienkripsikan mcnjadi :

$M^{17} \cong (((M^2)^2)^2)^2 . M \cong 948(\text{mod})2773$

Secara keseluruhan, data asli tersebut dienkripsikan menjadi

0948	2342	1084	1444	2663
	2390	0778	0774	0219
	1655			

dapat diuji bahwa hasil dekripsi

$948157 \cong 920(\text{mod}2773)$. dan seterusnya.

Sistem Kriptografi LUC

LUC adalah sistem kliptografi kunci publik yang dikembangkan oleh sekelompok peneliti dari Selandia Baru. Algoritmanya mempunyai kemiripan dengan RSA, dapat digunakan untuk penyandian data, signature, dan pembangkitan kunci. [4]

LUC berdasarkan bilangan bulat besar pada deret lucas ini telah dipelajari secara khusus untuk menguji keprimaan, dan hasilnya digunakan sebagai algoritma yang efisien untuk implementasi LUC.

Landasan Matematis Algoritma LUC

Landasan matematis dari kriptografi kunci publik LUC adalah deret Lucas. Deret Lucas adalah dua urutan integer U_n dan V_n yang dibangkitkan oleh dua bilangan bulat P dan Q. Teori ini secara umum pertama kali dikembangkan oleh Edouard Lucas pada tahun 1878. Fokus utama yang menjadi perhatian dalam pemakaian deret Lucas adalah untuk pengujian bilangan prima.[3]

Hasil dan Pembahasan

Tahapan Umum Perangkat Lunak Sistem Kriptografi *RSA*

Simulasi yang dibuat dalam algoritma kriptografi kunci publik *RSA* ini meliputi dua bagian besar, yaitu :

1. Tahap pembangkitan kunci
2. Tahap enkripsi dan dekripsi

Berikut inai akan dijelaskan masing-masing tahapan sistem kriptografi diatas.

Tahap Pembangkitan Kunci

Untuk pembangkitan kunci, pertama dicari dua buah bilangan prima yang berbeda, yaitu *p* dan *q* yang digunakan untuk mencari n *dan teta(n),* kemudian digunakan fungsi *fluclidean,* fungsi ini digunakan untuk memperoleh pasangan kunci privat dan kunci publik, melalui proses pengulangan sampai kondisi yang diinginkan terpenuhi. Pada algoritma ini, kunci privat yang digunakan adalah *d*. Nilai *d* didapatkan dari nilai fungsi random yang lebih besar dari I dan lebih kecil dai *teta(n) ,* dan *d* adalah bilangan prima yang relatif prima terhadap t*eta(n)* dimana t*eta(n)* adalah hasil perkalian dua bilangan relatif prima *p-1* dan *q-1*. Perhitungan kunci publik didapat dari fungsi invers d yang di-modulo oleh *teta(n).*

Sebenamya penggunaan kunci yang panjang akan memiliki tingkat keamanan yang lebih tinggi, namun pada model simulasi ini pasangan bilangan prima *p* dan *q* untuk pembangkit kunci yang digunakan dibatasi hanya sampal 5 digit. Hal ini dikarenakan keterbatasan bahasa pemrograman yang digunakan, karena penggunaan kunci yang besar akan mengakibatkan nilai kunci publik melebihi dari batas rentang nilai tipe data yang digunakan. Seperti diketahui pada bab sebelumnya, algoritma kriptografi kunci publik *RSA* banyak menerapkan operasi perpangkatan eksponensial. Untuk membatasi hasil perhitungan yang melebihi batas rentang nilai dari tipe data yang digunakan, maka nilai *p* dan *q* dibatasi nilainya.

Hal ini untuk mengantisipasi kunci publik yang dihasilkan memiliki nilai yang besar diluar batas rentang nilai tipe data yang digunakan

Tahap Enkripsi dan Dekripsi

Tahap ini dilakukan setelah masing masing pemakai dalam Jaringan, membangkitkan kunci privat dan kunci publik. Kcmudian dalam simulasinya bila seorang pemakai ingin mengirimkan informasi kepada pemakai lain, maka ia dapat menulilskan input berupa teks ataupun membuka file tipe apa saja yang telah ada sebelumnya, *plaintext* tersebut kemudian akan diproses menggunakan prosedur *blockread dan blockwrite.*

Masing-masing blok-pesan kemudian akan dipangkatkan dengan kunci publik dan di-modulo n, secara berulang, maka menghasilkan kumpulan data sandi yang disebut *chipertext* yang akan disimpan dalam sebuah file untuk dikirimkan kepada pemakai yang dituju. Untuk menghindari pembengkakan yang terlalu besar *pada chipertext,* maka tipe dari file *chipertext* yang dipilih adalah file bertipe *long integer.*

Hal ini disesualkan dengan isi dari *chipertext* yang berupa kumpulan bilangan *long integer.* Sampai tahap ini, selesailah tahapan proses enkripsi yang dilakukan pada sisi pengirim. Selanjutnya pembahasan akan dilakukan pada sisi penerima untuk mengembalikan pesan dengan proses dekripsi, dimana sebelumnya hasil berupa *chipertext* tersebut akan dikirimkan kepada pemakai yang dituju. Pada tahap selanjutnya pada sisi penerima, dilakukan proses pengembalian pesan dengan menggunakan prosedur dekripsi. Pada prosedur dekripsi ini, setelah menerima pesan, pasangan kunci privat yang sebelumnya telah dibuat , digunakan. pembacaan terhadap file *chipertext* tersebut dilakukan dengan prosedur *blockread,,* dengan algoritma dekripsi *RSA,* suatu *chipertext* akan diolah kembali dengan menggunakan kunci privat yaitu d, menjadi nilai ordinal (desimal) dari blok-pesan. Selanjutnya proses dekripsi merupakan kebalikan dari proses enkripsi, dimana setiap dua blok-pesan berurutan diproses. Selanjutnya nilai ordinal tersebut akan diubah menjadi karakter dengan menggunakan fungsi *chr* yang ada pada bahasa pemrograman. Demikian seterusnya setiap blok-pesan dikembalikan, yang akhirnya dituliskan kembali pada suatu file baru.

Tahapan Umum Perangkat Lunak Sistem Kriptografi Modified LUC

Simulasi yang dibuat dalam algoritma kriptografi kunci publik *LUC* ini, serupa dengan simulasi algoritma *RSA*, meliputi dua bagian besar, yaitu tahap pembangkitan kunci dan tahap enkripsi

dan dekripsi. Berikut ini akan dijelaskan masing-masing tahapan sistem kriptografi diatas.

Tahap Pembangkitan Kunci

Tahap pertama, untuk pembangkitan kunci, pertama dicari dua buah bilangan prima yang berbeda, yaitu *p* dan *q* yang digunakan untuk mencarl n dan dengan perhitungan lebih rumit mencari teta(n), kemudian dihitung s(n) yang merupakan *faktor kali terkecil (fkt)* dari *p-1* dan *q-1*. Selanjutnya digunakan fungsi Euclidean, fungsi im' digunakan untuk memperoleh pasangan kunci privat dan kunci publik yang prima relatif, melalui proses pengulangan sampai kondisi yang diinginkan terpenuhi.

Pada algoritma ini, kunci privat yang digunakan adalah d. Nilai d didapatkan dari nilai fungsi random yang lebih besar dari 1 dan lebih kecil dari *leta(n)* , dan d adalah bilangan prima yang relatif prima terhadap *tela(n)* dimana *leta(n)* adalah hasil perkalian dua bilangan relatif prima *p-1, p+4 ,q+1* dan q-1 yang dalam simulasi ini panjangnya maksimal 5 digit. Perhitungan kunci publik didapat dari fungsi *inverv d* yang dimodulo oleh *s(n)* Seperti diketahui pada sub bab sebelumnya, algoritma kriptografi kunci publik LUC sama dengan algoritma kriptografi kunci publik RSA banyak menerapkan operasi perpangkatan eksponensial.

Tahap Enkripsi dan Dekripsi

Pada tahap ini, sama dengan pada tahapan untuk algoritma RSA, dilakukan setelah masing masing pemakai dalam jaringan, membangkitkan kunci, baik kunci privat yang dijaga kerahasiaannya, maupun kunci publik. Kemudian dalam simulasinya bila seorang pemakai ingin mengirimkan informasi kepada pemakai lain, maka ia dapat menuliskan input berupa file teks ataupun dapat membuka file tipe apa saja yang telah ada. *Plaintext* tersebut kemudian akan diproses menggunakan fungsi *blockread.* Pesan kemudian akan dipangkatkan dengan kunci publik dan dimodulo n, secara berulang, maka menghasilkan kumpulan data sandi yang disebut *chipertext* yang akan disimpan dalarn sebuah file untuk dikirimkan kepada pemakai yang dituju. Untuk menghindari pembengkakan yang terlalu besar pada *chipertext,* maka tipe dari file *chipertexi* yang dipilih adalah file bertipe *long integer.* Hal ini disesuaikan dengan isi dari *chipertext* yang berupa kumpulan bilangan *long integer.* Sampai tahap ini, selesailah tahapan proses enkripsi yang dilakukan pada sist pengirim.

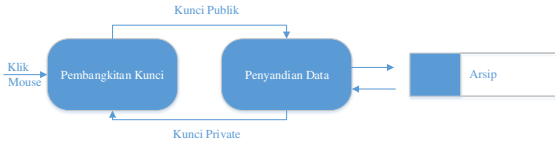
Pada tahap selanjutnya pada sisi penerima,(dimana sebelumnya hasil berupa *chipertext* tersebut akan dikirimkan kepada pemakai yang dituju), dilakukan proses pengembalian pesan dengan menggunakan prosedur dekripsi. Pada prosedur dekripsi ini, setelah menerima pesan, pasangan kunci privat yang sebelumnya telah dibuat , digunakan. pembacaan terhadap file *chipertext* tersebut dilakukan kembali dengan fungsi *blockread,* dengan algoritma dekripsi LUC, *suatu chipertext akan* diolah kembali dengan menggunakan kunci privat yaitu d, menjadi nilai ordinalnya (desimal).

Selanjutnya proses dekripsi merupakan kebalikan dari proses enkripsi, dimana setiap blok-pesan berurutan diproses. Selanjutnya nilai ordinal tersebut akan diubah menjadi karakter dengan menggunakan fungsi *chr* yang ada pada bahasa pemrograman. Demikian seterusnya setiap blok-pesan dikembalikan menjadi karakter, yang akhirnya dituliskan kembali pada suatu file baru.

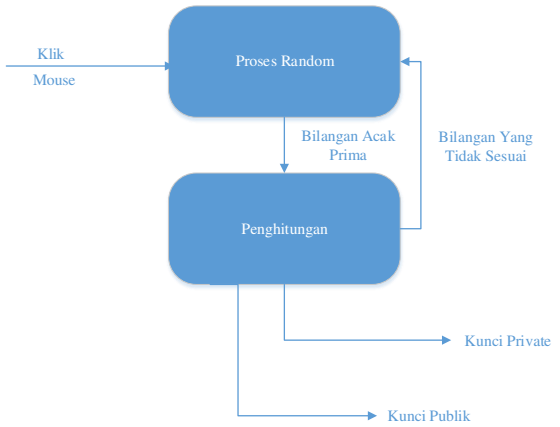
Perancangan Perangkat Lunak

Perancangan perangkat lunak dalam Penelitian ini meliputi perancangan struktur data, perancangan modul, perancangan menu, serta perancangan demo program

Gambar 1. Level 1 Sistem Kriptografi Kunci Publik



Gambar 2. Level 2 Pembankitan Kunci



Gambar 3. Level 2 Penyandian

Perancangan Menu Perangkat Lunak

Pada perangkat lunak yang akan dibuat, menunutama terdiri dari menu Proses RSA, menu Modified LUC, menu Arsip, menu Animasi, menu Pembuat menu Bantuan, dan menu Keluar. Tujuh menu ini dinyatakan dalam bentuk tombol-tombol yang tersedia pada bahasa pemrograman yang dipakai:

1. Menu Animasi terdiri dari empat buah item menu, yaitu item Mulai, item pause, item Stop dan item Keluar. item mulai digunakan untuk memulai demo program, item Stop digunakan untuk menghentikan demo program, item Pause digunakan untuk menghentikan sementara demo

program yang sedang berjalan untuk dapat menjelaskan prosesnya, sedangkan item Keluar digunakan untuk keluar dari demo program.

2. Menu Proves RSA menggambarkan proses kriptografi kunci publik RSA. Pada bagian ini akan diperlihatkan mulai dari proses penghasilan kunci, proses enkripsi dan proses dekripsi. Menu ini memiliki enam buah item menu, yaitu item Arsip, item Kunci, item Enkriptif, item Dekripsi, item Bantuan dan item Keluar.
3. Menu modified LUC merupakan menu yang menggambarkan proses kriptografi kunci publik LUC Pada bagian ini akan diperlihatkan mulai dari proses penghasilan kunci, proses enkripsi dan proses dekripsi. Menu ini memiliki enam buah item menu, yaitu item Arsip, item Kunci, item Fnkripvi, item Dekripsi, item Bantuan dan item Keluar.
4. Menu Arsip terdiri dari delapan item, yaitu ltem Baru, item Buka, item Simpan, item Simpan sebagai, item Cetak, item Aturan Cetak, dan item tutup. Baru digunakan untuk membuat suatu arsip baru, Baru digunakan untuk membuka suatu arsip yang sudah pernah disimpan, Simpan dan Simpan sebagai digunakan untuk menyimpan suatu arsip dengan nama tertentu, Cetak dan Aturan Cetak digunakan untuk pencetakan suatu arsip, dan tutup digunakan untuk menutup arsip.
5. Menu Bantuan adalah item menu sebagai panduan bagi pemakai perangkat lunak ini dalam pengoperasiannya. Menu ini terdiri dari tiga buah pilihan, yaitu item Lihat hal selanjutnya, item Lihat hal sebelumnya, dan item Keluar.
6. Menu Keluar merupakan pilihan menu yang digunakan untuk keluar dari menu Proses yang sedang dijalankan.
7. Menu pembuat berisi tampilan form keterangan perangkat lunak yang dibuat. Pada form ini J'uga terdapat suatu tombol yang digunakan untuk keluar dari menu.

Implementasi Perangkat Lunak

Implementasi yang dibahas pada Penelitian ini, meliputi implementasi setiap modul, batasan iniplementast dan lingkungan implementasi.

Implementasi Modul

Seperti telah dijelaskan pada sub bab perancangan, perangkat lunak yang dibuat pada Penelitian ini memiliki delapan buah modul yaitu : modul ARSIP, modul PROSES RSA, modul PEMBUAT, modul MODIFIED LUC modul KELUAR, modul ANIMASI dan modul BANTUAN Berikut ini akan dijelaskan implementasi darl beberapa modul yang dianggap inti dari perangkat lunak ang dibuat, yaitu modul PROSES RSA dan modul MODIFIED LUC yang masing-masing terdiri atas modul KUNCI, modul ENKRIPSI dan modul DEKRIPSI.

Modul RSA

Modul ini diimplementasikan dalam suatu unit benama RSA, yang terdiri dari beberapa buah prosedur. Perosedur-prosedur yang ada dalam modul ini adalah Prosedur Bangkit Kunci, Enkripsi dan Dekripsi. berikut ini akan diperlihatkan algoritma masing-masing sub-modul :

1. Modul KUNCI

Algoritma dari Prosedur Bangkit, dalam proses eksekusinya memerlukan dua buah prosedur, yaitu prosedur generatekey dan prosedur Relatif Trime.

ALGORITMA PROSEDUR BANGKIT KUNCI RSA

```
==========================================
Procedure Bangkit Kunci ;
------------------------------------------------------------
--
Procedure RelativePrime(no1,no2:LongInt; var
RelPrime:boolean);
Var
   temp : LongInt;
begin
Relprime := false;
While (not RelPrime) and (no2 <> 0 ) do
begin
        temp := no1 mod no2;
        if (temp = 1) then RelPrime := true
        else
        no1 := no2: no2 := temp;
        end;
end;
procedure GenerateKey(prime1,prime2,LongInt;var
e,d,n : longint);
var
        Tetan,i : LongInt;
        found,IsRelPrime : boolena;
        begin
                n := prime1 * prime2;
                Tetan := (Prime1-1) * (Prime2-1);
                repeat
                { chose d value }
                        repeat
                                d          :=
random(1000)
                        until (d>1) and (d<Tetan)
and (d <> Tetan);

        RelativePrime(Tetan,d,IsRelPrime);{Max
word data type}
                until (IsRelPrime);
                i := 0; found := false;
        while (i <= Tetan) and (not found) do
        begin
                if (i * Tetan + 1)div d) > 0 then
                e := (i*Tetan+1) div d;
```

```
                found := true;
        end;
        i := i+1;
end;
end;.
```

Modul LUC

Modul ini diimplementasikan dalam suatu unit bemama LUC, yang terdiri dari beberapa buah prosedur. Perosedur-prosedur yang ada dalam modul ini adalah Prosedur Bangkit Kunci, Enkripsi dan Dekripsi. Berikut ini akan diperlihatkan algoritma masing-masing sub-modul :

1. Modul KUNCI

ALGORITMA PROSEDUR BANGKIT KUNCI LUC

```
=====================================
Procedure Bangkit Kunci ;
-------------------------------------------------------------
--
Procedure RelativePrime(no1,no2:LongInt; var
RelPrime:boolean);
Var
   temp : LongInt;
begin
Relprime := false;
While (not RelPrime) and (no2 <> 0 ) do
begin
        temp := no1 mod no2;
        if (temp = 1) then RelPrime := true
        else
        no1 := no2: no2 := temp;
        end;
end;

Function fkt(var u,v : longint) : longint;
var
a,b,c,x :longint;
begin
a:= u:b:=v;
repeat
x:=u div v; c:=u - (x*v); u:=v; v:=c;
until (c=0);
fkt:=(a*b) div u;
end;

procedure GenerateKey(prime1,prime2,LongInt;var
e,d,n : longint);
var
        Tetan,i,u,v,sn : LongInt; found,ISRelPrime :
boolean;
begin
        n:=prime1*prime2;
        Tetan:=(prime1-1)*(prime2-
1)*(prime1+1)*(prime2+1);
        u:=(prime1-1);v:=(prime2-1); sn:=fkt(u,v);
                repeat                  {choose e value}
                        repeat
                        e:=random(1000);
                until(e>1) and (e<Tetan) and
(e<>Tetan);

        RelativePrime(Tetan,e,IsRelPrime); {Max
Word Data Type}
        until(IsRelPrime);
                i:= found:=False;
        while(i<=sn)and(notfound)do
        begin
                if(i*sn+1)mode e)=0 then
                begin
                        if(i*sn+1)dive e)>0 then
                        d:=(i*sn+1)dive         2;
found:=true;
                end;
                i:=i+1;
        end;
end
```

Pada Prosedur Bangkit Kunci, digunakan prosedur findprime untuk menghasilkan kunci p *dan q* Secara acak yang memenuhi kondisi tertentu, yang berhubungan dengan proses pencarian n dan teta(n). Kemudian dilakukan perhitungan faktor *kali terkecil (fkt*) pada p-I dan *q-1* Untuk menghasikan *s(n)* yang dipakai untuk perhitungan terhadap *teta(n)*. Penghitungan terhadap kunci publik dilakukan dengan menggunakan fungsi invers dan dimodulo dengan *s(n)* untuk menghasilkan kunci privat. Setelah didapatkan pasangan kunci privat d dan kunci publik e, selanjutnya pasangan kunci privat dan kunci publik yang dihasilkan melalui prosedur ini akan digunakan untuk proses enkripsi dan dekiripsi.

2. Modul ENKRIPSI

PROSEDUR ENKRIPSI LUC

```
=====================================
Procedure Enkripsi ;
-------------------------------------------------------------
--
begin
        assign(FilePlain1,filein);
reset(filePlain1);Assign(FileCipher,TMP.CPT);
        rewrite(FileCipher); writeln("Sedang Proses
Enkripsi...");size1:=filesize(Fileplain1);
        ifsize1>-100 then
        begin
                repeat

        blockread(fileplain1,arraytamp,100);size1:
=size1-100;
                        for i:=1 to 100 do

        arraycipher[i]:=Encrypt_Decrypt(e,arrayta
mp[i],n);blockwrite(filecipher,arraycipher,100);
                        until size1<100;

        blockread(fileplain1,arraytamp,size1);
                        for i:=1 to size do
```

```
        arraycipher[i]:=Encrypt_Decrypt(e,arrayta
mp[i],n);blockwrite(filecipher,arraycipher,size1);
                        end
                        else
                        begin

        blockread(fileplain1,arraytamp,size1);
                        for i:=1 to size do

        arraycipher[i]:=Encrypt_Decrypt(e,arrayta
mp[i],n);blockwrite(filechiper,arraychiper,size1);
                        end;
                        CloseFile(FilePlain1);
Reset(FileCipher);
                                writeln;
writeln("Prses Selesai, Tekan Enter Untuk
Keluar...");readln;
                        end.
```

Pada Prosedur Ekripsi LUC mekanismenya sama dcngan prosedur enkripsi RSA hanya berbeda pada algontma Encrypt-Decrypt-nya, proses diawali dengan pembacaan arsip plaintext. Karena untuk simulasi arsip tcrsebut bertipe file, maka digunakan fungsi assign untuk membuka arsip dan menampung-nya. Proses dilanjutkan dengan pengolahan plaintext menjadi chipertext, yang diawaii dengan menggu-nakan prosedur Encrypt-Decrypt, untuk menglutung bilangan-bilangan yang dihasilkan dengan kunci publik dan n. Selanjutnya arsip hasil enkripsi tcrsebut akan dirimkan dan diproses pada bagian dekripsi, disisi penerima.

Analisis Perbandingan Hasil Eksekusi Perangkat Lunak Kriptografl RSA dan *Modified L*UC

Berikut ini akan diberikan tabel yang memper-lihatkan perbandingan performansi dari masing-masing algoritma dilihat dari waktu eksekusi pada saat proses enkripsi dan dekripsi di pada percobaan dibawah ini dilakukan dengan tipe dan ukuran file serta *kombinasi* Pemakaian pasangan kunci yang berbeda-beda.

Tabel 1. Waktu Enkripsi dan Dekripsi Berdasarkan Ukuran dan Tipe File pada Algoritma RSA

No	Kunci	Tipe File	Ukuran File	Waktu Enkripsi	Waktu Dekripsi
	e, d		(byte)	(Detik)	(Detik)
1	34003,107	TXT	64.000	4	2
	691,11		59.000	3	1
	30533,157		30.000	2	1
	853,317		40.000	2	1
2	1157,137	DOC	62.000	3	2
	479,583		60.000	3	3
	223,247		43.000	2	2
	1979,395		48.000	3	2
3	11497,313	PAS	8.000	<1	1
	43,283		13.000	1	1
	4749,309		5.000	<1	<1
	1891,295		6.000	<1	<1
4	3841,193	BMP	64.000	3	2
	119-327		33.000	1	2
	1421,325		38.000	2	2
	1249,73		44.000	2	1
5	2133,157	WAV	24.000	1	<1
	1819,179		55.000	3	2
	641,161		63.000	3	2
	4183,247		38.000	2	2
6	493,229	MID	12.000	1	<1
	619,397		7.000	<1	<1
	2069,317		2.000	<1	<1
	1013,101		11.000	1	1
7	3019,2355	EXE	7.000	1	1
	2315,227		20.000	1	1
	949,349		57.000	3	3
	641,257		20.000	1	1

Tabel 2. Waktu Enkripsi dan Dekripsi Berdasarkan Ukuran dan Tipe File pada Algoritma LUC

No	Kunci	Tipe File	Ukuran File	Waktu Enkripsi	Waktu Dekripsi
	e, d		(byte)	(Detik)	(Detik)
1	311,671	TXT	64.000	4	3
	397,973		59.000	3	3
	337,2399		30.000	2	2
	179,125		40.000	2	2
2	113,425	DOC	62.000	3	2
	127,451		60.000	3	3
	367,139		43.000	2	2
	149,3931		48.000	2	3
3	373,65	PAS	8.000	1	1
	109,1969		13.000	1	1
	149,1217		5.000	<1	<1
	301,8821		6.000	1	<1
4	97,49	BMP	64.000	3	2
	179,59		33.000	2	2
	311,6671		38.000	2	2
	203,1175		44.000	2	1

5	65,257	WAV	24.000	1	1
	169,197		55.000	2	2
	301,1301		63.000	3	3
	203,1175		38.000	2	2
6	131,4651	MID	12.000	1	1
	281,6581		7.000	<1	1
	293,49		2.000	<1	<1
	379,4915		11.000	1	1
7	353,14369	EXE	7.000	1	1
	347,3827		20.000	1	1
	949,349		57.000	3	3
	641,257		20.000	1	1

Kesimpulan

1. Algoritma Publik RSA lebih cepat dan valid.
2. Dilihat dari segi pembangkitan kunci algoritma RSA masih lebih unggul di dalam penyandian data.

Saran

Adapun saran-saran yang dapat dikemukakan di sini sehubungan dengan pengembangan perangkat lunak lebih lanjut adalah:

1. Untuk kedua sistem kriptografi kunci publik ini, jika dlimplementasi dengan algoritma untuk mengatasi perhitungan digit yang besar pada jaringan komputer, maka dapat digunakan kunci dengan digit desimal besar untuk meningkatkan faktor kekuatan keamanan.
2. Untuk program kriptografi *modified LUC* yang dibuat pada Penelitian itu, penulis mengusul-kannya sebagal altematif solusi permasalahan ke-tidak-validan algoritma kunci publik *LUC.*.

Daftar Pustaka

[1] Stallings, William. 1995. Network and Interrietwork Security Principles and Practice,Prentice Hall : USA.

[2] Smith, P. 1993. LUC Public Key Encryption A Secure Alternative to RSA. Dr. Dobb.v Journal, January.

[3] Denning, D. 1982. Criptography and Data Security, Purdue University: USA.

[4] Cantu, Marco. 1995. Mastering Delphi, SYBEY Rhee, Man Young. 1994. Cryptography and,securityre Comunications, McGraw-Hill: Singapore.