ISBN: 9-789-7936-499-93

PENGAMANAN DOKUMEN MENGGUNAKAN METODE RSA *(RIVEST SHAMIR ADLEMAN)*BERBASIS WEB

Ardelia Nidya Agustina[1], Aryanti[2], Nasron[2]
Program Studi Teknik Telekomunikasi, Jurusan Teknik Elektro, Politeknik Negeri Sriwijaya Palembang
Jl Srijaya Negara, Bukit Besar, Ilir Barat 1, Kota Palembang, Sumatera Selatan
Telp. (0711) 712716
E-mail: ardelianidya@gmail.com

ABSTRAK
Perkembangan teknologi masa kini kian berkembang pesat. Semakin banyak pula pengguna yang mengakses internet. Salah satu akibat dari hal ini makin banyak nya penyadapan terhadap suatu dokumen yang bersifat rahasia. Sehingga apabila berbicara mengenai sebuah pengamanan pasti tidak akan jauh dari apa yang disebut kriptografi. Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, serta otentikasi. Metode yang cukup aman untuk penyandiaan yang digunakan saat ini ialah metode RSA(REVEST SHAMIR ADLEMAN). Metode RSA termasuk jenis metode Asimetris. Dimana metode RSA ini mempunyai dua kunci yang berbeda pada proses enkripsi dan dekripsi. Pada penelitian ini, dilakukan suatu analisis dalam perspektif keamanan file yang bertujuan untuk mengamankan file menggunakan metode RSA. Untuk proses penerapan nya suatu data yang dikirim terlebih dahulu dienkripsi oleh pengirim dan menghasilkan data terenkripsi selanjutnya akan dikirim kepada penerima untuk dilakukan proses dekripsi yang menghasilkan suatu data yang sebenarnya. Dari penelitian ini dapat disimpulkan bahwa tingkat keamanan dengan menggunakan metode RSA termasuk dalam kategori metode yang aman dipakai untuk proses pengamanan dokumen.
Kata Kunci: kriptografi, Asimetris,RSA.

1.PENDAHULUAN

Keamanan dan kerahasiaan data merupakan salah satu aspek yang sangat penting dalam sistem informasi pada saat ini. Munculnya teknologi internet dan multimedia telah mendorong berbagai macam usaha untuk melindungi, mengamankan, danmenyembunyikan data pada *file* digital dari pihak-pihak yang tidak mempunyai otoritas untuk mengakses *file-file* tersebut. Salah satu usaha untuk mengamankan data daninformasi diantaranya denganmenggunakan kriptografi. Berbagaimacam algoritma kriptografi dapat diimplementasikan untuk mewujudkan sistem keamanan data. Diantaranya yaitu algoritma kriptografi Rivest Shamir Adleman (RSA) (Rakhman, 2015).

Algoritma RSA dibuat oleh tiga orang peneliti dari MIT *(Massachussets Institute of Technology)* pada tahun 1976, yaitu Ron Rivest, Adi Shamir dan Leonard Adleman. 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, sedangkankunci untuk melakukan dekripsi disebut sebagai kunci privat. (Rahajoeningroem, 2015).

Adapun penelitian terkait sebelumnya membahas tentang Pengujian Sistem Enkripsi-Dekripsi Dengan Metode RSA untuk Pengamanan Dokumen (Supriyono, 2008). Selanjutnya penelitian terkait tentang Aplikasi Pengaman Data Menggunakan Algoritma RSA (Rivest-Shamir-Adleman) dilakukan oleh (Mujiarto, 2012).

Pada penelitian kali ini membahas tentang pengamanan dokumen berbasis web. Dengan menggunakan metode-metode yang ada pada kriptografi. Penelitian kali ini menggunakan metode RSA. Metode RSA termasuk ke dalam jenis algoritma asimetris. Proses enkripsi dokumen dilakukan pada saat dokumen tersebut dikirim untuk mengamankan *file* agar tidak dapat terbaca oleh orang yang tidak berhak mengakses *file* tersebut . RSA yang mempunyai dua kunci yang berbeda, disebut pasangan kunci *(key pair)* untuk proses enkripsi dan dekripsi.Kunci-kunci yang ada pada pasangan kunci mempunyai hubungan secara matematis, tetapi tidak dapat dilihat secara komputasi untuk mendeduksi kunci yang satu ke pasangannya. Algoritma ini disebutkunci publik, karena kunci enkripsi tidak bersifat rahasia. Orang-orang dapat menggunakan kunci publik ini, tapi hanya orang yang mempunyai kunci privat sajalah yang bisa mendekripsi data tersebut (Kurniawan, 2011).

2. TINJAUAN PUSTAKA

Algoritma kriptografi dibagi menjadi 2 macam yaitu Algoritma Simetris dan Algoritma Asimetris. Dimana algoritma simetris menggunakan satu kunci untuk proses enkripsi dan dekripsinya. Sedangkan Algoritma Asimetris menggunakan dua kunci yang berbeda untuk proses enkripsi dan dekripsinya. Dimana algoritma. Lalu algoritma asimetrismenggunakan kunci yang berbeda untuk proses enkripsi dan dekripsinya, yaitu kunci umum *(Public key)* yang digunakan proses enkripsi yaitu perubahan data text asli *(Plaintext)* menjadi *text* rahasia *(Ciphertext)* yang sifatnya tidak rahasia, dan kunci pribadi *(private key*) yang digunakan
Algoritma RSA memiliki besaranseperti berikut :

1. p dan q bilangan prima (rahasia)

ISBN: 9-789-7936-499-93

2. n= p.q (tidak rahasia)
3. Φ(n)=(p-1)(q-1) (rahasia)
4. e= kunci enkripsi (tidak rahasia)
5. d=kunci dekripsi (rahasia)
6. m=plaintext (rahasia)
7. c= ciphertext (tidak rahasia)(dewanto, yanto, 2013)

2.1. KEAMANAN RSA

Keamanan dari sistem kriptografi RSA adalah didasari oleh dua problem matematika yaitu masalah dalam faktorisasi bilangan berjumlah banyak. Dan masalah dari RSA, yaitu mencari modulo akar e n dari sebuah bilangan komposit (yang faktor-faktornya tidak diketahui proses dekripsi penuh dari sebuah *ciphertext* RSA dianggap sesuatu hal yang tidak mudah karena kedua masalah ini diasumsikan sulit. Permasalahan dari RSA didefinisikan sebagai tugas untuk mencari suatu akar modulo e n (e pangkat ke n)dari bilangan komposit (. Mengembalikan suatu nilai m dimana m e=c mod n, (e, n) adalah kunci publik RSA dan c adalah *ciphertext* RSA. Metode pendekatan yang diyakini dapat menyelesaikan masalah RSA saat ini adalah memfaktor dari modulus n. Dengan kemampuan untuk mengembalikan faktor yang merupakan bilangan prima, sebuah serangan dapat menghitung eksponen rahasia dari d dan dari kunci publik (e, n), lalu mendekripsi c menggunakan prosedur standar. Untuk menyelesaikannya, penyerang . Memfaktor nilai n menjadi p dan q, lalu menghitung (p-1)(q-1) yang dapat menghasilkan nilai d dan e.

2.2. ALGORITMA RSA

1. Menentukan 2 bilangan prima, dengan nama p dan q. Misal nilai $p = 51$ dan $q = 5$. (1)
2. Menghitung nilai modulus (n) : (2)
   $\rightarrow n = p \times q$
   $\rightarrow n = 51 \times 5$
   $\rightarrow n = 255$
3. Menghitung nilai totient n :
   $\rightarrow \phi(n) = (p-1) \times (q-1)$ (3)
   $\rightarrow \phi(n) = (51-1) \times (5-1)$
   $\rightarrow \phi(n) = (50 \times 4)$
   $\rightarrow \phi(n) = 200$
4. Menentukan nilai e dengan syarat gcd (e, $\phi(n)$) = 1 (4)
   Dimana e = bilangan prima, dan $1 < e < \phi(n)$.
   Pilih kunci publik e adalah 7 (relatif prima terhadap 200)
5. Mencari nilai *deciphering exponent* (d), maka : (5)
   $\rightarrow d = (1 + (k \text{ x } \phi(n)) / e)$
   $\rightarrow d = (1 + (k \text{ x } 200)) / 7$
   Nilai k merupakan sembarang angka untuk pencarian hingga dihasilkan suatu nilai integer atau bulat. Dengan mencoba nilai k = 1,2,..., hingga diperoleh nilai d yang bulat, yaitu d = 343.
6. Dari langkah-langkah yang sudah diuraikan sebelumnya, maka nilai n, e, dan d telah didapatkan sehingga pasangan kunci telah terbentuk.
   Pasangan kunci publik (n, e) = (255, 7)
   Pasangan kunci rahasia (n, d) = (255, 343)

Plaintext : polsri
p=112
o=111
l=108
s=115
r=114
i=105

Enkripsi RSA

$= 112^7 \bmod 255 = 73$ = pada tabel ascii **I**
$= 111^7 \bmod 255 = 36$ = pada tabel ascii **#**
$= 108^7 \bmod 255 = 252$ = pada tabel ascii **w**

ISBN: 9-789-7936-499-93

$=115^{7}$ mod 255 = 55 = pada tabel ascii***n***
$=114^{7}$ mod 255 = 24= pada tabel ascii **(cancel)**
$=115^{7}$ mod 255 = 45 = pada tabel ascii –

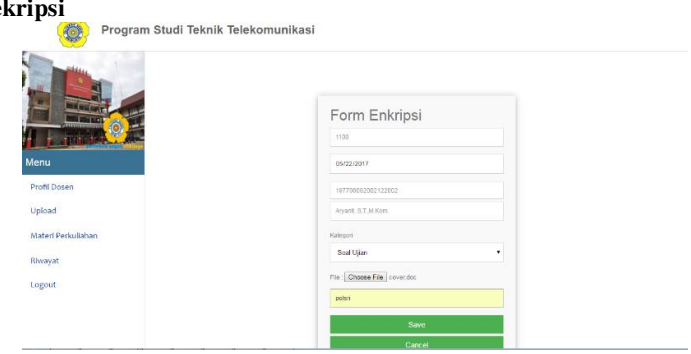
Dekripsi RSA

$=73^{343}$ mod 255 = 112 = pada tabel ascii **p**
$=36^{343}$mod 255 = 36 = pada tabel ascii **o**
$= 252^{343}$ mod 255 = 252 = pada tabel ascii **l**
$=55^{343}$ mod 255 = 55= pada tabel ascii **s**
$=24^{343}$ mod 255 = 24= pada tabel ascii **r**
$=45^{343}$ mod 255 =45= pada tabel ascii **i**

6. HASIL DAN PEMBAHASAN

Pengujian pada aplikasi pengamanan dokumen menggunakan metode RSA. Jenis *file* yang akan di enkripsi ialah doc,docx,pdf,ppt, dan xls. Pengujian sistem dilakukan untuk mengetahui apakah penelitian ini telah memenuhi tujuan untuk mengamankan dokumen menggunakan metode RSA . Pengujian sistem secara menyeluruh yaitu pengujian sistem pada aplikasi yang akan digunakan oleh admin/dosen/mahasiswa, dari segi tampilan dan segi proses yang terjadi di setiap halaman dan selanjutnya melakukan proses enkripsi dan dekripsi *file* dengan menerapkan algoritma kriptografi RSA. Plaintext yang digunakan sebagai bahan unjuk kerja adalah "polsri" dengan kunci Pasangan kunci publik (n, e) = (255, 7)Pasangan kunci rahasia (n, d) = (255, 343).

Proses Enkripsi dan Dekripsi



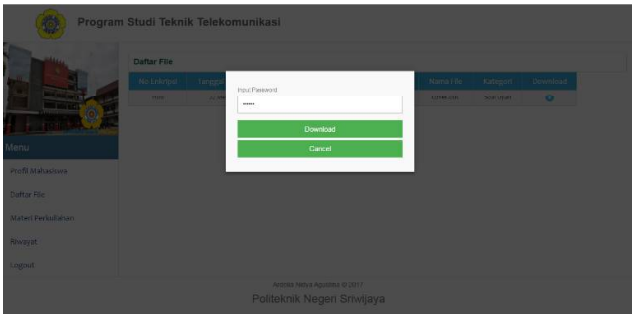
Gambar 1 Tampilan *Form* Enkripsi

Pada gambar 1 merupakan *form* enkripsi dimana pada menu ini proses enkripsi dilakukan

Gambar 2 Tampilan Daftar Enkripsi

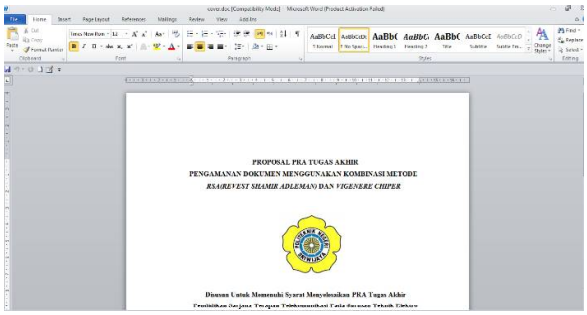
Pada gambar 2 merupakan tampilan daftar enkripsiberisikan *file-file* yang telah di enkripsi.

ISBN: 9-789-7936-499-93



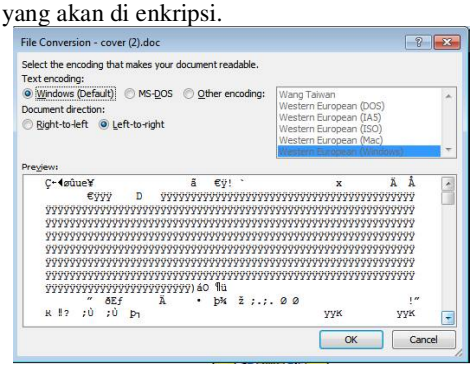
Gambar 3. Tampilan *input password* proses dekripsi

Pada gambar 3 merupakan tampilan *input password*. Saat proses enkripsi mahasiswa terlebih dahulu menginput *password*. Apabila password benar maka *file* yang di download akan terdekripsi. Namun apabila *password* salah *file* yang didownload merupakan *file* enkripsi atau *file* acak yang tidak dapat dibaca.



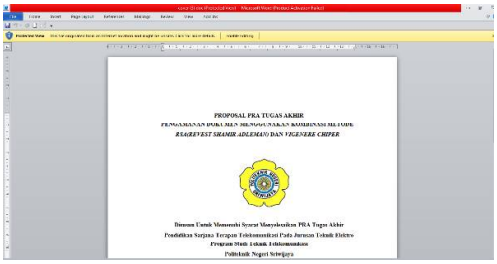
Gambar 4 Tampilan *file* sebenarnya

Pada gambar 4 merupakan isi *file* yang akan di enkripsi.



Gambar 5 tampilan *file* enkripsi

Pada gambar 5 merupakan tampilan *file*enkripsi atau *file* yang tidak dapat terbuka dikarekan *password* yang di *input* saat mendownload tidak sama dengan *password* saat dosen mengupload. Dan mahasiswa tidak dapat membuka informasi sebenarnya pada *file* tersebut.Proses enkripsi dibutuhkan waktu yang cukup lama. Itu dikarenakan proses pengacakan data yang dilakukan pada proses enkripsi. Jika *password* benar maka *file* tersebut akan terdekripsi. Sebaliknya, apabila password salah *file* yang terdownload tidak dapat dibuka atau *file* tersebut merupakan *file* acak. Proses pengamanan data proses enkripsi dilakukan pada bit data pada *file* sehingga struktur pada *file* akan berubah dan tidak akan dapat terbuka sebelum melakukan proses dekripsi kembali. Saat proses dekripsi akan dikembalikan ke struktur bit data dan nilai pada *file* tersebut seperti semula.

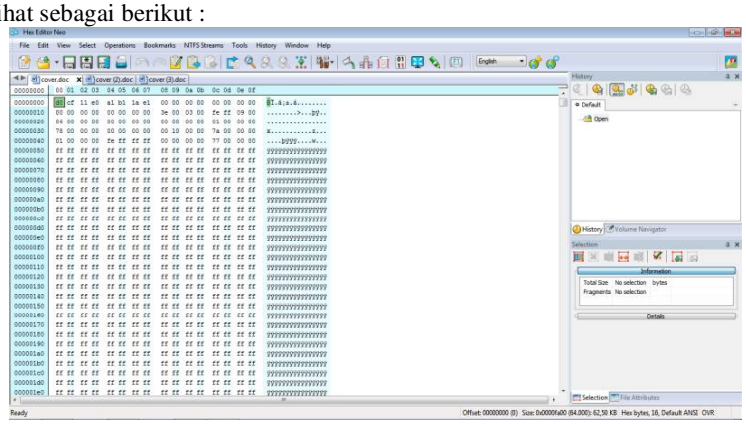


ISBN: 9-789-7936-499-93

Gambar 6 tampilan *file*berhasil dekripsi

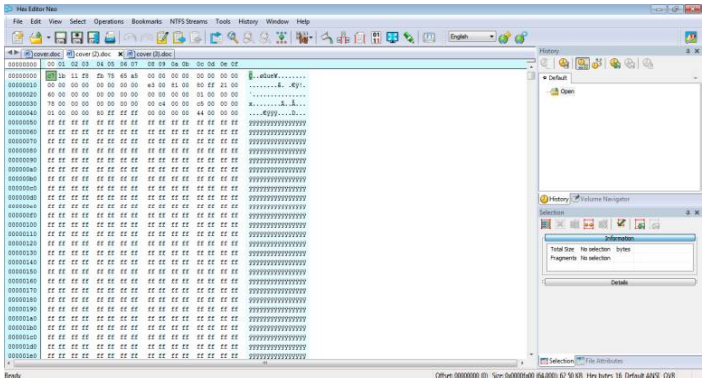
.Gambar 6 merupakan tampilan *file* yang berhasil di dekripsi. Karena pasword yang di *input* saat mendownload atau mendekripsi sama dengan saat dosen mengupload.

Untuk menganalisis keamanan terhadap *file*, maka diperlukanan pengecekan terhadap *file*. Apakah *file* tersebut telah terenkripsi dengan baik dan sempurna sehingga informasi yang ada pada *file* tidak dapat di akses oleh orang yang tidak berhak. Pada pengujian yang telah dilakukan pada file "cover.doc", struktur *file* asli atau *file* yang belum di enkripsi dapat kita lihat sebagai berikut :



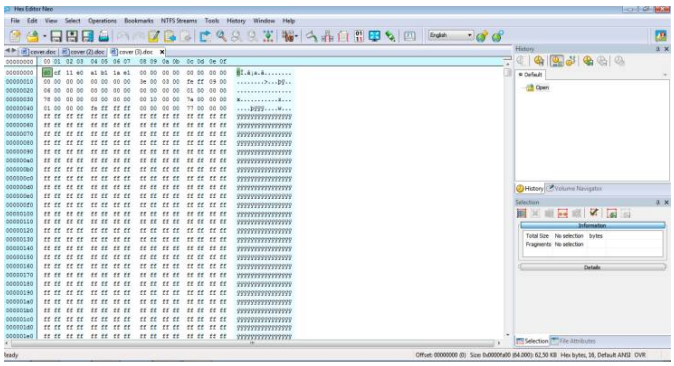
Gambar 7. Tampilan *file* "cover.doc dalam heksadesimal"

File yang di enkripsi diberi nama cover(2). Berikut merupakan struktur *file*yang telah enkripsi dalam bentuk heksadesimal.



Gambar8. Tampilan *file* "cover (2).doc dalam heksadesimal"

Pada gambar 8 merupakan struktur dari *file* "Cover(2).doc" di bagian kiri tampilan menunjukkan struktur *file* dalam bentuk heksadesimal, sedangkan di bagian kanan merupakan bentuk karakter (ASCII) dari heksadesimal. Dapat kita lihat pada gambar di atas. Kedua nya memiliki hasil yang berbeda. Hal ini dapat membuktikan bahwa file "cover(2)" telah berhasil di enkripsi. Untuk membuktikan bahwa aplikasi berjalan dengan baik, dapat kita lakukan proses dekripsi terhadap *file* "cover(2).doc"



Gambar9. Tampilan file "cover (3).doc dalam heksadesimal"

ISBN: 9-789-7936-499-93

Pada gambar 9 merupakan tampilan *file* "cover(3).doc" yang merupakan *file* yang berhasil di dekripsi.

3.2. Analisa waktu dekripsi dan enkripsi

Tabel 3.1. Pengujian Waktu proses enkripsi dan dekripsi

No.	Nama File	Tipe File	Size File (KB)	Waktu Enkripsi(s)	Waktu Dekripsi (s)
1	Tinjauan Pustaka	Txt	34KB	52.3569939	51.1279249
2	Bab2	Docx	559KB	849.663017	829.230204
3	Pengenalan kewirausahaan	Pptx	76KB	112.539223	108.245191
4	Ppt Sempro	Pptx	92KB	135.337074	132.773593
5	Tabel	Xls	49KB	73.6961288	71.5742077
6	Wireshak	Doc	386KB	583.406437	578.870110

Dari pengujian yang telah dilakukan pada tabel 3.1 dapat dianalisa bahwa ukuran *file* mempengaruhi waktu proses enkripsi dan dekripsi. Karena semakin besar ukuran *file*, semakin lama juga proses enkripsi nya.Waktu yang dibutuhkan untuk melakukan enkripsi maupun dekripsi *file*relatif sama. Karena pada proses enkripsi dan dekripsi *size file* tidak berubah. Namun ada beberapa faktor yang dapat mempengaruhi waktu enkripsi dan dekripsi dapat sedikit berbeda. Misalnya spesifikasi kecepatan dari laptop. Terkoneksinya laptop dengan jaringan internet, dan lain sebagainya.

7. PENUTUP

Kesimpulan

Dari penelitian yang dilakukan maka dapat disimpulkan :

1. Aplikasi pengamanan *file*menggunakan bahasa pemrograman PHP.
2. Penerapan program pengamanan dokumen menghasilkan suatu aplikasi yang dapat mengubah *file* asli (plaintext) menjadi *file* terenkripsi*(ciphertext)* yang tidak dapat dibaca informasi dari filenya kemudian mengembalikannya kembali menjadi file *aslinya (ciphertext*) tanpa merubah ataupun merusak isi *file* nya.
3. Proses pengujian aplikasi menggunakan metode RSA dan telah di tes secara aplikasi, hasilnya adalah prosesenkripsi dan deskripsi telah sesuai dengan kaidah algoritma kriptografi RSA.
4. Semakin besar ukuran *file*maka semakin lama proses enkripsi dan dekripsi.
4. Aplikasi yang dihasilkan dapat digunakan untuk dokumen *office* yang berformat doc,docx,txt,xls,ppt,dan pptx.

Saran

Penelitian yang telah dilakukan ialah membuat suatu aplikasi kriptografi RSAmenggunakan bahasa pemrograman *PHP*. Perlu dilakukan penelitian untuk membuat aplikasi berdasarkan algoritma kriptografi RSA menggunakan bahasa pemrograman yang berbeda. Penelitian selanjutnya diharapkan dapat menambahkan format *file* untuk di enkripsi.

PUSTAKA

Dewanto, Yanto. (2013). Pembuatan Aplikasi SMS Kriptografi RSA dengan Android.

Ginting, Isnanto, Windasari .(2015). Implementasi Algoritma Kriptografi RSA untuk Dekripsi dan Enkripsi *Email.*

Kurniawan, Rahmad. (2011). Analisa dan Perancangan Perangkat Lunak Keamanan Data dengan Menggunakan Algoritma RSA.

Rahajoeningroem, Aria. (2011). Studi dan Implementasi Algoritma RSA untuk Pengamanan Data Transkip Akademik Mahasiswa.

Rakhman. (2015). Implentasi Algoritma Kriptografi Rivest Shamir Adleman (RSA) dan *Vigenere Cipher* Pada Gambar Bitmap 8 Bit.

Wicaksono, (2015). Enkripsi Menggunakan Algoritma RSA.