VOLUME 14 NOMOR 2 NOVEMBER 2020

ISSN 1978-3310
E-ISSN 2615-3467

Infoman's

Jurnal Ilmu-ilmu Informatika dan Manajemen

Journal of Informatics Sciences and Management

STMIK SUMEDANG

Infoman's

Jurnal Ilmu-ilmu Informatika dan Manajemen
Journal of informatics Sciences and Management
STMIK SUMEDANG

Articles

ISSN : 1978-3310 | E-ISSN : 2615-3467

Aplikasi Keamanan Data *Multimedia Message Service* (MMS) Pada *Microsoft Office File* Memanfaatkan Algoritma *Rivest-Shamir Adleman* (RSA) Dan *Blowfish* Berbasis Android

Abdul Aziz
Pusat Teknologi Penerbangan – LAPAN
Rumpin, Bogor
email : abdul.aziz@lapan.go.id

ABSTRACT
Use of Multimedia Message Service (MMS) on mobile phone becomes one of the options in the exchange of data. The problem is, the biggest security hole on the exchange of data via MMS is a message that is sent will be stored in the Multimedia Message Service Center (MMSC) and the operator or person who has full access rights can view files sent. For these reasons need a method that can secure data that is not known by the MMS unauthorized person is one with cryptography. Cryptographic algorithms used in this study is a hybrid combining the Blowfish symmetric key algorithm that uses a single key operation has advantages in speed, and the asymmetric key algorithm Rivest-Shamir Adleman (RSA) that uses two keys namely public key and a private key has a security level higher. Estimated time achieved in the encryption, using the Blowfish algorithm and the RSA algorithm did not depend on the size of the plaintext or the key length used.

Keywords **-** Blowfish, Hybrid, MMS, RSA

ABSTRAK
Penggunaan *Multimedia Message Service* (MMS) pada telepon seluler menjadi salah satu pilihan dalam melakukan pertukaran data. Permasalahannya, celah keamanan terbesar pada pertukaran data melalui MMS yaitu pesan yang dikirimkan akan disimpan di *Multimedia Messaging Service Center* (MMSC) dan *operator* atau orang yang mempunyai hak akses penuh dapat melihat *file* yang dikirimkan. Dengan alasan tersebut butuh metode yang dapat mengamankan data MMS agar tidak diketahui oleh orang yang tidak berhak yaitu salah satunya dengan kriptografi. Kriptografi yang dipakai menggunakan algoritma *hybrid* yaitu dalam penelitian ini menggabungkan antara algoritma kunci simetris *Blowfish* yang memakai satu kunci mempunyai kelebihan dalam kecepatan operasi, dan algoritma kunci asimetris *Rivest-Shamir Adleman* (RSA) yang memakai dua kunci yaitu kunci publik dan kunci privat mempunyai tingkat keamanan yang lebih tinggi. Estimasi waktu yang dicapai dalam melakukan enkripsi, baik menggunakan algoritma *Blowfish* maupun algoritma RSA ternyata tidak bergantung pada besar ukuran *plaintext*-nya ataupun panjang kunci yang dipakai.

Kata Kunci **-** *Blowfish*, *Hybrid*, MMS, RSA

1. Introduction

Penggunaan *Multimedia Message Service* (MMS) pada telepon seluler menjadi salah satu pilihan dalam melakukan pertukaran data. MMS merupakan salah satu aplikasi pada telepon seluler yang dapat mengirimkan file berupa gambar, *audio*, *video*, bahkan *Microsoft Office file*. *Microsoft Office* adalah aplikasi perkantoran yang dirilis oleh *Microsoft Corporation* dalam membantu pengoperasian dan penyelesaian pekerjaan sehari-hari serta memberikan hasil yang optimal. *Microsoft Office file*

diantaranya terdiri dari *Microsoft Office Word, Microsoft Office Excel, Microsoft Office Power Point, Microsoft Office Access, Microsoft Office Publisher, Microsoft Office Visio*.

Permasalahannya, celah keamanan terbesar pada pertukaran data melalui MMS yaitu pesan yang dikirimkan akan disimpan di *Multimedia Messaging Service Center* (MMSC) dan operator atau orang yang mempunyai hak akses penuh dapat melihat file yang dikirimkan [3]. Hal ini sangat rentan sekali terhadap kebocoran data. Masalah tersebut menjadi sebuah kekhawatiran bagi pengguna MMS ketika *file* yang dikirimkan adalah *file* yang sangat penting yang hanya boleh diketahui oleh orang-orang tertentu saja.

Dengan alasan tersebut butuh metode yang dapat mengamankan data MMS agar tidak diketahui oleh orang yang tidak berhak yaitu salah satunya dengan kriptografi. Kriptografi merupakan salah satu solusi untuk membatasi wewenang tersebut, dengan aplikasi kriptografi yang ditanamkan pada telepon seluler, maka pengguna dapat memastikan MMS yang ingin dikirim tak dapat dibaca oleh orang lain kecuali pengguna yang diijinkan.

Kriptografi yang dipakai menggunakan algoritma *hybrid* yaitu dalam penelitian ini menggabungkan antara algoritma kunci simetris *Blowfish* yang memakai satu kunci mempunyai kelebihan dalam kecepatan operasi, dan algoritma kunci asimetris *Rivest-Shamir Adleman* (RSA) yang memakai dua kunci yaitu kunci publik dan kunci privat mempunyai tingkat keamanan yang lebih tinggi. Jadi dengan algoritma *Hybrid*, pengguna MMS dapat mengamankan data lebih cepat dan lebih tinggi keamanannya dibandingkan dengan hanya satu algoritma saja.

Secara gambarannya, penelitian ini membuat aplikasi yang dapat digunakan untuk mengamankan data *Microsoft Office File* yang akan dikirimkan melalui MMS. Algoritma RSA digunakan untuk untuk tujuan pertukaran kunci. Sedangkan algoritma *Blowfish* digunakan untuk mengenkripsi dan mendekripsi *file*. *File* yang dikirim melalui MMS menjadi lebih aman setelah diubah ke dalam data terkunci. Karena *file* yang dikirimkan bukan *file* asli melainkan *file* yang sudah didekripsi dan hanya dapat dibaca oleh pihak yang berhak.

2. Research Method

2.1. Aplikasi

Menurut Ali Zaki dan SmitDev Community [8] "Program aplikasi adalah komponen yang berguna melakukan pengolahan data maupun kegiatan-kegiatan seperti pembuatan dokumen atau pengolahan data".

Program aplikasi berjalan diatas sistem operasi, sehingga agar program aplikasi bisa diaktifkan, perlu melakukan instalasi sistem operasi terlebih dahulu.

2.2. Keamanan Data

Menurut Dony Ariyus [1],

> Keamanan data pada lalu-lintas jaringan adalah suatu hal yang diinginkan semua orang untuk menjaga privasi, supaya data yang dikirim aman dari gangguan orang yang tidak bertanggung-jawab, yang disembunyikan menggunakan algoritma kriptografi.

Pada dasarnya komponen kriptografi terdiri dari beberapa komponen, [1]:

1. *Enkripsi*: merupakan hal yang sangat penting dalam kriptografi, merupakan cara pengamanan data yang dikirimkan sehingga terjaga kerahasiaannya. Pesan asli disebut *plaintext* (teks-biasa), yang diubah menjadi kode-kode yang tidak dimengerti. Enkripsi bisa diartikan dengan *cipher* atau kode. Sama halnya dengan tidak mengerti sebuah sebuah kata maka kita akan melihatnya di dalam kamus atau daftar istilah. Beda halnya dengan enkripsi, untuk mengubah teks-biasa ke bentuk teks-kode kita gunakan algoritma yang dapat mengkodekan data yang kita ingini.

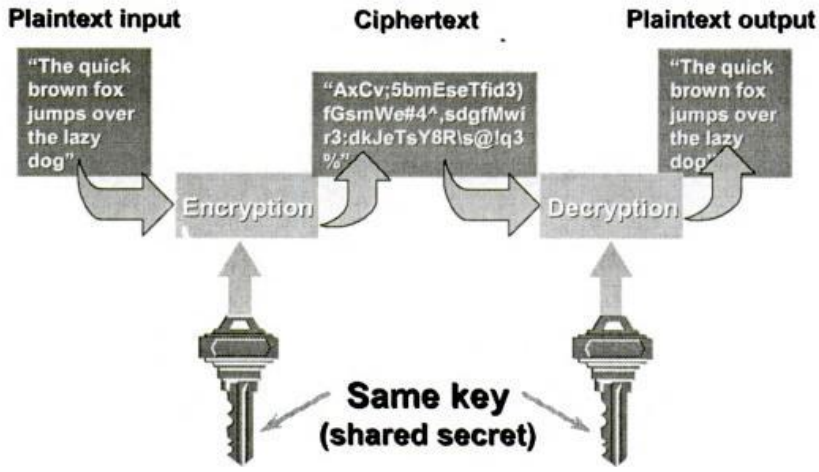
2. *Dekripsi*: merupakan kebalikan dari enkripsi. Pesan yang telah dienkripsi dikembalikan ke bentuk asalnya. Algoritma yang digunakan untuk dekripsi tentu berbeda dengan yang digunakan untuk enkripsi.
3. *Kunci*: adalah kunci yang dipakai untuk melakukan enkripsi dan dekripsi. Kunci terbagi menjadi dua bagian, yaitu kunci rahasia (*private key*) dan kunci umum (*public key*).
4. *Chipertext*: merupakan suatu pesan yang telah melalui proses enkripsi. Pesan yang ada pada teks-kode ini tidak bisa dibaca karena berupa karakter-karakter yang tidak mempunyai makna (arti).
5. *Plaintext*: sering disebut *cleartext*. Teks-asli atau teks-biasa ini merupakan pesan yang ditulis atau diketik yang memiliki makna. Teks-asli inilah yang diproses menggunakan algoritma kriptografi untuk menjadi *chipertext* (teks-kode).
6. *Pesan*: dapat berupa data atau informasi yang dikirim (melalui kurir, saluran komunikasi data) atau yang disimpan di dalam media perekaman (kertas, *storage*).
7. *Cryptanalysis*: kriptanalisis bisa diartikan sebagai analisis kode atau suatu ilmu untuk mendapatkan teks-asli tanpa harus mengetahui kunci yang sah secara wajar. Jika suatu teks-kode berhasil diubah menjadi teks-asli tanpa menggunkan kunci yang sah, proses tersebut dinamakan *breaking code*. Hal ini dilakukan oleh para kriptanalis. Analisis kode juga dapat menemukan kelemahan dari suatu algoritma kriptografi dan akhirnya dapat menemukan kunci atau teks-asli dari teks-kode yang dienkripsi dengan algoritma tertentu.

2.3. Algoritma Kriptografi Modern

Kriptografi modern merupakan suatu perbaikan yang mengacu pada kriptografi klasik. Pada kriptografi modern terdapat berbagai macam algoritma yang dimaksudkan untuk mengamankan informasi yang dikirim melalui jaringan komputer. Algoritma kriptografi modern terdiri dari dua bagian [1]:

2.3.1. Algoritma Simetris

Algoritma simetris adalah algoritma yang menggunakan kunci yang sama untuk enkripsi dan dekripsinya. Contoh: Alice ingin mengirim pesan x dengan aman menggunakan saluran umum kepada Bob. Alice menggunakan kunci $x\sigma$ yang sebelumnya telah disepakati antara Alice dan Bob. Untuk mengirim pesan e $x\sigma$ (x) kepada Bob, dia akan mendekripsi teks-kode yang diterima dengan kunci yang sama dengan yang digunkan untuk memperoleh akses ke pesan yang diterima, begitu sebaliknya. Secara gambaran algoritma simetris terlihat pada Gambar 1.



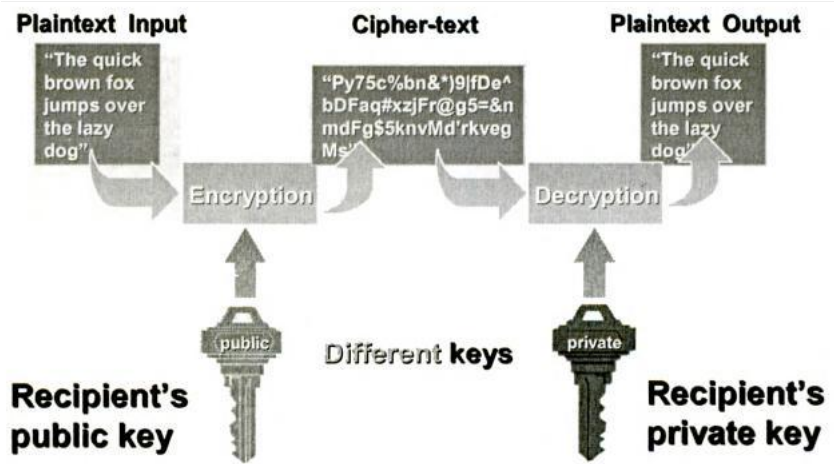
Gambar 1. Algoritma Simetris
Sumber: Buku Pengantar Ilmu Kriptografi, 2008

Aplikasi dari algoritma simetris digunakan oleh beberapa algoritma di bawah ini:

a) *Data Encryption Standard* (DES),
b) *Advance Encryption Standard* (AES),
c) *International Data Encryption Algorithm* (IDEA),
d) A5,
e) RC4.

2.3.2. Algoritma Asimetris

Algoritma asimetris adalah pasangan kunci kriptografi yang salah satunya digunakan untuk proses enkripsi dan yang satu lagi untuk dekripsi. Semua orang yang mendapatkan kunci publik dapat menggunakannya untuk mengenkripsi suatu pesan, sedangkan hanya satu orang saja yang memiliki rahasia itu, yang dalam hal ini kunci rahasia, untuk melakukan pembongkaran terhadap kode yang dikirim untuknya. Contoh algoritma terkenal yang menggunakan kunci asimetris adalah RSA (merupakan singatan dari nama penemunya, yakni Rivest, Shamir dan Adleman). Secara gambarannya algoritma asimetris terlihat pada Gambar 2.



Gambar 2. Algoritma Asimetris
Sumber: Buku Pengantar Ilmu Kriptografi, 2008

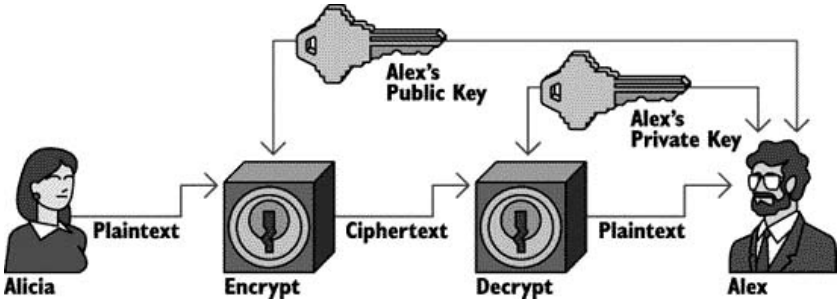
Dari sekian banyak algoritma kriptografi kunci-publik yang pernah dibuat, algoritma yang paling populer adalah algoritma RSA. Algoritma ini melakukan pemfaktoran bilangan yang sangat besar. Oleh karena alasan tersebut RSA dianggap aman. Untuk membangkitkan dua kunci, dipilih dua bilangan prima acak yang besar. Algoritma RSA dibuat oleh 3 orang peneliti dari *Massachussets Institute of Technology* (MIT) pada tahun 1976, yaitu: Ron (R)ivest, Adi (S)hamir, dan Leonard (A)dleman. RSA mengekripsikan teks-asli yang dienkripsi menjadi blok-blok yang mana setiap blok memiliki nilai bilangan biner yang diberi symbol "n", blok teks-asli "M" dan blok teks-kode "C",

2.4. *Rivest-Shamir Adleman* (RSA)

Dari sekian banyak algoritma kriptografi kunci-publik yang pernah dibuat, algoritma yang paling populer adalah algoritma RSA. Algoritma ini melakukan pemfaktoran bilangan yang sangat besar. Oleh karena alasan tersebut RSA dianggap aman. Untuk membangkitkan dua kunci, dipilih dua bilangan prima acak yang besar. Algoritma RSA dibuat oleh 3 orang peneliti dari *Massachussets Institute of Technology* (MIT) pada tahun 1976, yaitu: Ron (R)ivest, Adi (S)hamir, dan Leonard (A)dleman. RSA mengekripsikan teks-asli yang dienkripsi menjadi blok-blok yang mana setiap blok memiliki nilai bilangan biner yang diberi symbol "n", blok teks-asli "M" dan blok teks-kode "C", numeric yang lebih

kecil daripada "n" (data biner dengan pangkat terbesar). Jika bilangan prima yang panjangnya 200 digit, dapat ditambah beberapa bit 0 di kiri bilangan untuk menjaga agar pesan tetap kurang dari nilai "n".

Proses enkripsi dan dekripsi RSA terlihat pada Gambar 3.



Gambar 3. Proses Enkripsi dan Dekripsi RSA
Sumber: http://labsky2012.blogspot.com, 2012

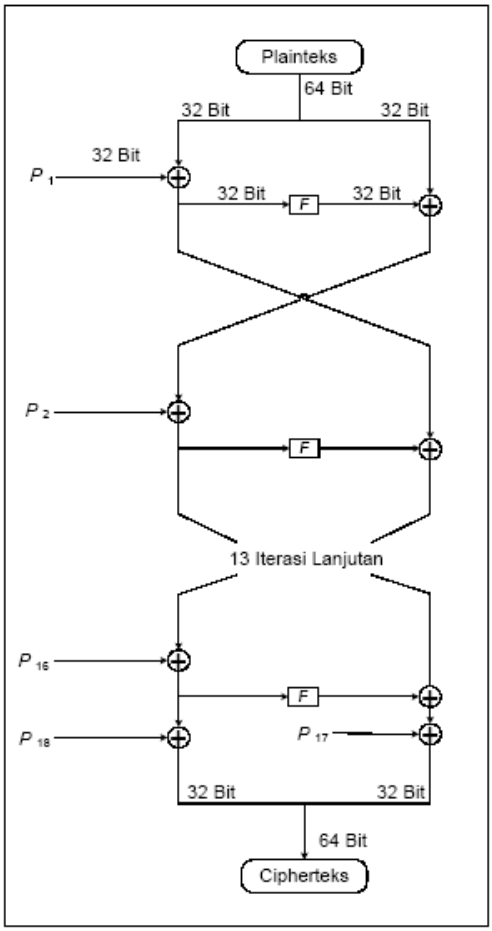
2.5. *Blowfish*

Blowfish merupakan sebuah algoritma kunci simetri blok kode yang dirancang pada tahun 1993 oleh Bruce Chneier untuk mengganti DES. Pada saat itu banyak sekali rancangan algoritma yang ditawarkan, namun hampir semua terhalang oleh paten atau kerahasiaan pemerintah Amerika. Schneier menyatakan bahwa *Blowfish* bebas paten dan akan diletakkan pada domain publik. Dengan pernyataan Schneier tersebut *Blowfish* telah mendapatkan tempat di dunia kriptografi, khususnya bagi masyarakat yang membutuhkan lgoritma kriptografi yang cepat, kuat, dan tidak terhalang oleh lisensi.

Keberhasilan *Blowfish* dalam menembus pasar terbukti dengan diadopsinya Blowfish sebagai *Open Cryptography Interface* (OCI) pada kernel Linux versi 2.5 ke atas. Dengan diadopsinya *Blowfish* berarti dunia *open source* menganggap *Blowfish* adalah salah satu algoritma terbaik. Kesuksesan *Blowfish* mulai memudar setelah kehadiran algoritma dengan ukuran blok yang lebih besar seperti AES. AES sendiri memang dirancang untuk menggatikan DES, sehingga secara keseluruhan AES lebih unggul dari DES dan juga *Blowfish*.

Blowfish adalah algoritma kriptografi kunci simetri blok kode dengan panjang blok tetap 64 bit. *Blowfish* menerapkan teknik kunci berukuran sembarang. Ukuran kunci yang dapat diterima oleh *Blowfish* adalah antara 32 bit hingga 448 bit, dengan ukuran *default* sebesar 128 bit. *Blowfish* memanfaatkan teknik pemanipulasiaan bit dan teknik pemutaran ulang dan pergiliran kunci yang dilakukan sebanyak 16 kali. Algoritma utama terbagi menjadi dua subalgoritma utama, yaitu bagian ekspansi kunci dan bagian enkripsi-dekripsi data.

Pengekspansian kunci dilakukan pada saat awal dengan masukan sebuah kunci dengan panjang 32 bit hingga 448 bit, dan keluaran adalah sebuah larik upa-kunci dengan total 4168 byte. Bagian enkripsi-dekripsi data terjadi dengan memanfaatkan perulangan 16 kali terhadap jaringan *Feistel*. Setiap perulangan terdiri dari permutasi dengan masukan kunci dan substitusi data. Semua operasi dilakukan dengan memanfaatkan *operator* XOR dan *operator* penambahan. Penambahan dilakukan terhadap empat larik *lookup* yang dilakukan setiap putarannya. Proses enkripsi pada *Blowfish* ini terlihat pada Gambar 4.



Gambar 4. Proses Enkripsi pada *Blowfish*
Sumber: Buku Pengantar Ilmu Kriptografi, 2008

3. Result and Analysis

Hasil pengujian ini dimaksudkan untuk dapat mengetahui hasil pengujian yang telah dilakukan sebelumnya dengan melakukan beberapa percobaan lalu membandingkan data dengan informasi dan menampilkannya pada sebuah grafik. Pengujian ini menghasilkan informasi estimasi waktu pengenkripsian dan persentase perubahan.

Estimasi waktu bertujuan untuk mengetahui berapa lama waktu yang dicapai untuk mengenkripsi sebuah *plaintext* dengan bermacam-macam ukurannya dengan algoritma yang digunakan. Estimasi waktu ini diambil dari lama proses pengenkripsian yang dilakukan.

Persentase perubahan bertujuan untuk mengetahui berapa persen perubahan dari ukuran *plaintext* ke *chipertext*. Persentase perubahan ini dihitung dengan rumus:

$$\text{Persentase perubahan } (\%) = \frac{UC-UP}{UC} x\ 100 \qquad (1)$$

Keterangan:

UC = Ukuran *chipertext file* (bytes)

UP = Ukuran *plaintext file* (bytes)

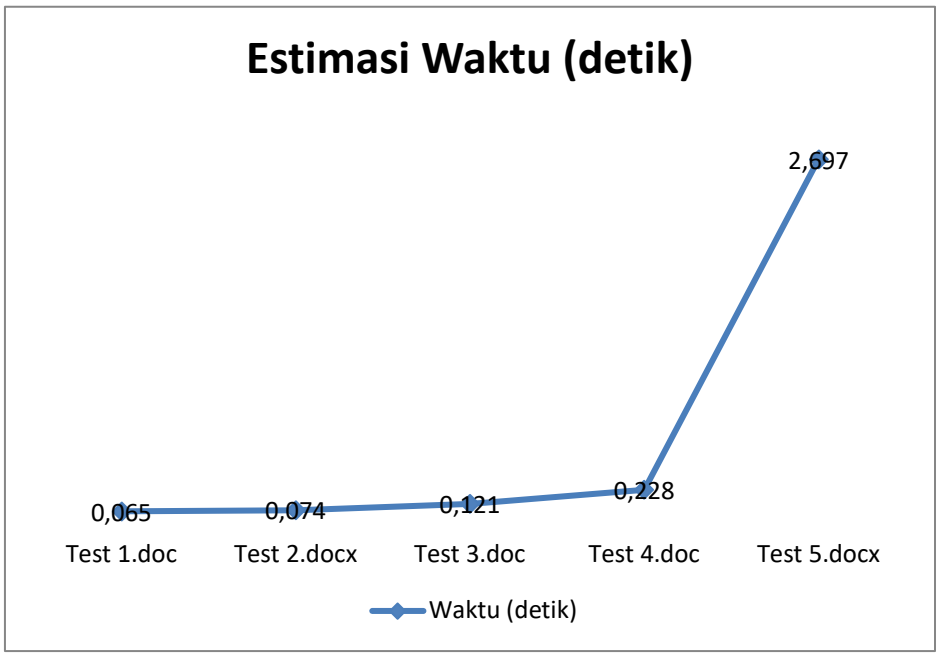
3.1. Hasil Uji Enkripsi pada *File Ms.Word*

Pengujian dilakukan terhadap 5 buah *file Ms. Word* yang berbeda dengan ukuran *file* yang berbeda pula. Pengenkripsian menggunakan algoritma *Blowfish* dengan masukan kunci berupa string "teknik". Hasil analisis ini menghasilkan informasi pada Tabel 1.

Tabel 1. Hasil Uji Enkripsi pada *File MS. Word*

No	Nama File	Ukuran *Plaintext* (bytes)	Estimasi Waktu (detik)	Ukuran *Chipertext* (bytes)	Persentase perubahan (%)
1	Test 1.doc	11,140	0.065	11,144	0.035906643
2	Test 2.docx	24,019	0.074	24,024	0.020816853
3	Test 3.doc	52,224	0.121	52,232	0.015318627
4	Test 4.doc	107,520	0.228	107,528	0.007440476
5	Test 5.docx	1,443,497	2.697	1,443,504	0.000484933

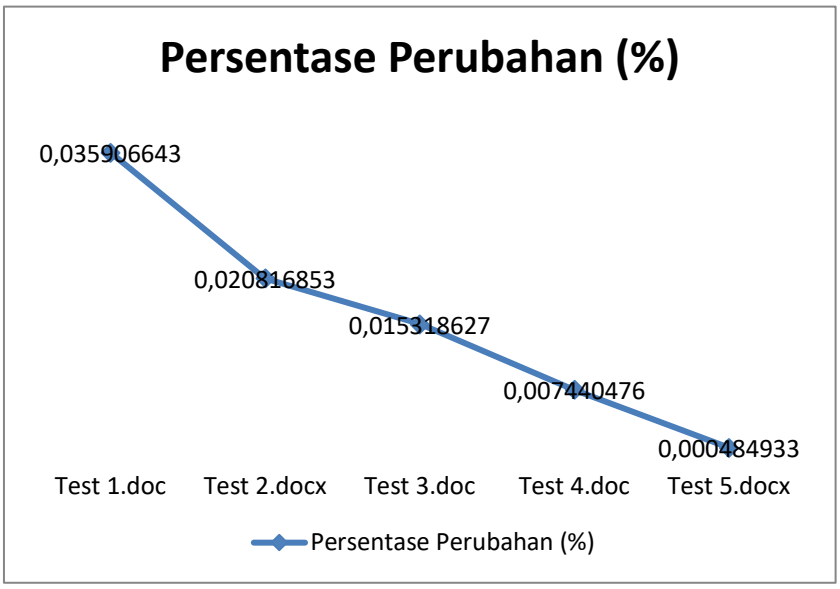
Dari Tabel 1 terdapat informasi mengenai estimasi waktu yang diperlukan dalam melakukan enkripsi pada *file Ms. Word* yang dapat divisualisasikan pada grafik Gambar 5.



Gambar 5. Grafik Estimasi Waktu Proses Enkripsi pada *File Ms. Word*

Grafik tersebut menunjukkan perbandingan waktu yang dicapai untuk mengenkripsi antara beberapa *file Ms. Word* dengan ukuran yang berbeda. Grafik menunjukkan bahwa semakin besar ukuran *file* yang dienkripsi, maka semakin lama pula waktu yang dihabiskan dalam proses pengenkripsian.

Pada Tabel 1 terdapat juga informasi mengenai persentase perubahan setelah proses enkripsi pada *file Ms. Word*. Persentase perubahan ini menunjukkan persentase selisih antara *plaintext* dan *chipertext* lalu membandingkan dengan percobaan-percobaan yang lainnya. Persentase perubahan pada *file Ms. Word* dapat dilihat pada grafik Gambar 6.



Gambar 6. Grafik Persentase Perubahan pada *File Ms. Word*

Grafik tersebut menunjukkan bahwa perbandingan persentase perubahan setelah proses enkripsi pada *file Ms. Word*. Berdasarkan rumus yang dipakai untuk menghitung persentase perubahan, maka grafik menunjukkan semakin besar ukuran *file* yang dienkripsi, maka semakin kecil pula tingkat kenaikannya.

3.2. Hasil Uji Enkripsi pada Kunci Menggunakan Algoritma RSA

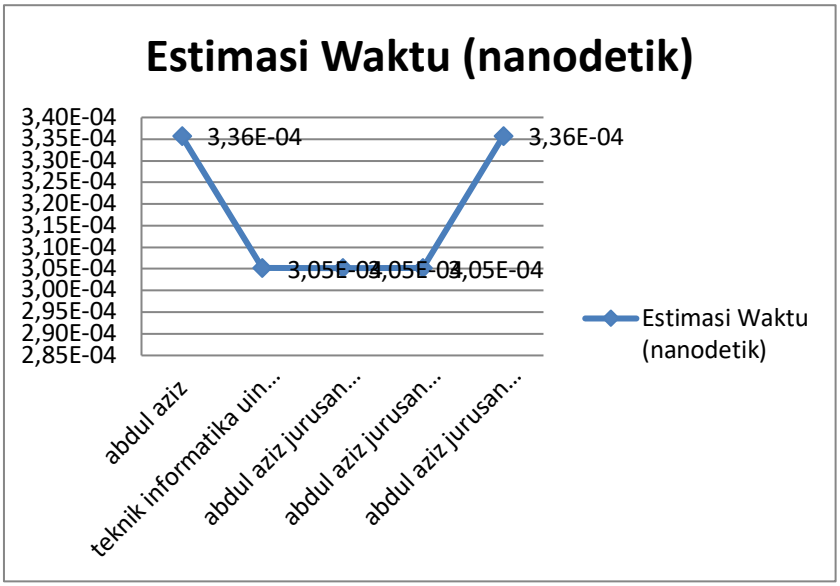
Pengujian dilakukan terhadap beberapa kunci berbeda yang digunakan untuk mengenkripsi *file* dengan kunci *e* (35879) yang sama dan *modulus* (505265315669768807) yang sama untuk mengenkripsi kunci (*plaintext*). Hasil analisis ini menghasilkan informasi pada Tabel 2.

Tabel 2. Hasil Enkripsi Menggunakan Algoritma RSA

No	Plaintext	Chipertext	Estimasi Waktu (nanodetik)
1	abdul aziz	c5b4736f7dcd45066799a3904f1ca7336a7	3.36E-04
2	teknik informatika uin bandung	28701f76dcfd42cbad93c6fe1a62828e3a0b	3.05E-04
3	abdul aziz jurusan teknik informatika	3e560b849424d6514b813b7ae826db5f01b	3.05E-04
4	abdul aziz jurusan teknik informatika uin sunan gunung djati bandung	31185095695dd7e91a8c2bfcaf7524285918	3.05E-04
5	abdul aziz jurusan teknik informatika fakultas sains dan teknologi universitas uin sunan gunung djati bandung	d3ce4b0a99762988691d8385e068b09653c	3.36E-04

Dari Tabel 2 terdapat informasi mengenai panjang *chipertext* yang menunjukkan bahwa semakin panjang *plaintext* yang dienkripsi maka semakin panjang pula *chipertext* yang dihasilkan. Pada tabel

tersebut terdapat juga estimasi waktu yang diperlukan dalam melakukan enkripsi pada beberapa *plaintext* yang dapat divisualisasikan pada grafik Gambar 7.



Gambar 7. Grafik Estimasi Waktu Enkripsi dengan Kunci yang Berbeda-beda

Grafik tersebut menunjukkan perbandingan waktu yang dicapai untuk mengenkripsi sebuah *plaintext* yang berbeda-beda dengan pasangan kunci yang sama. Tidak dapat dikatakan jika setiap *plaintext* yang dienkripsi semakin panjang maka akan semakin lama pula estimasi waktu yang dicapai. Karena grafik menunjukkan bahwa terjadi perubahan estimasi waktu yang tidak beraturan terhadap proses enkripsi.

4. Conclusion

Setelah melakukan penelitian ini, maka dapat ditarik kesimpulan sebagai berikut:

1. Algoritma *Blowfish* baik digunakan sebagai algoritma penyandian untuk *file* (*plaintext*) yang sangat besar, karena semakin besar *plaintext* yang dienkripsi menghasilkan persentase perubahan atau selisih yang semakin kecil dan estimasi waktu yang dicapai pun tidak bergantung pada besar ukuran *plaintext*.
2. Algoritma *Blowfish* baik digunakan sebagai algoritma penyandian untuk *file* yang akan dikirimkan melalui teknologi komunikasi yang terdapat batasan maksimum ukuran *file* yang dikirimkan, seperti contohnya pada teknologi MMS ini, dengan alasan *chipertext* yang dihasilkan memiliki selisih yang sedikit dari *plaintext*-nya.
3. Algoritma RSA baik digunakan sebagai algoritma penyandian untuk pertukaran kunci, dengan alasan ukuran kunci yang kecil dan membutuhkan kerahasiaan yang besar.
4. Estimasi waktu yang dicapai dalam melakukan pengenkripsian, baik menggunakan algoritma *Blowfish* maupun algoritma RSA ternyata tidak bergantung pada besar ukuran *plaintext*-nya ataupun panjang kunci yang dipakai.

Setelah penelitian dengan pembuatan aplikasi keamanan data MMS memanfaatkan algoritma RSA dan *Blowfish* berbasis android ini dilakukan, maka muncul beberapa saran untuk pengembangan kedepannya, sebagai berikut:

1. Dengan kombinasi algoritma RSA dan *Blowfish* dalam mengamankan data ini diharapkan dapat menerapkannya pada komunikasi data yang lain dan sistem operasi yang lain.

2. Pengimplementasian algoritma RSA maupun algoritma *Blowfish* diharapkan dapat diimplementasikan lebih spesifik lagi terhadap sebuah obyek.

Harus diakui bahwa algoritma keamanan data ini hanya dipakai oleh sebagian kalangan saja, kebanyakan masyarakat tidak memperdulikannya. Dengan hal demikian, untuk kedepannya pengembang dapat mengimplementasikan algoritma ini pada teknologi yang banyak dipakai oleh kebanyakan masyarakat.

References

[1] Ariyus, Dony. (2008). Pengantar Ilmu Kriptografi. Yogyakarta: Andi.
[2] Booch, Grady dkk. (1999). The Unified Modeling Language User Guide. Addison-Wesley.
[3] Hius, J. (2013). Implementasi Algoritma Enhanced 1-D Chaotic Key Based Algorithm (ECKBA) untuk sistem Kriptografi pada Aplikasi MMS.
[4] Pressman, R.S. (2002). Rekaya Perangkat Lunak. Penerbit McGrawHill Terjemahan Andi Publisher.
[5] Sitinjak, Suriski dkk. (2010). Aplikasi Kriptografi File Menggunakan Algoritma Blowfish. Yogyakarta.
[6] Tambunan, S.E.A. (2010). Implementasi Algoritma Kriptografi Blowfish untuk Keamanan Dokumen pada Microsoft Office. Yogyakarta.
[7] Tanto, Iwan dan Ricco. Perancangan Perangkat Lunak Enkripsi dan Dekripsi File dengan Algoritma RSA Dan RC4. Medan.
[8] Zaki, Ali dan SmitDev Community. (2007). Cara Mudah Merakit PC. PT Elex Media Komputindo. Jakarta.