BAB VI

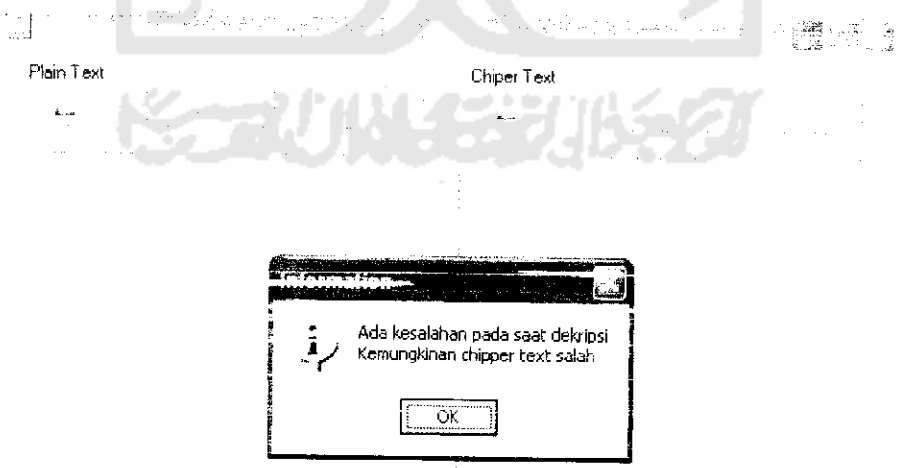
ANALISIS KINERJA PERANGKAT LUNAK

6.1 Penanganan Kesalahan

Dalam tahap ini akan dijelaskan mengenai hasil pengujian terhadap program aplikasi yang telah dibuat. Dengan pengujian ini diharapkan tingkat kesalahan baik dalam pengolahan data maupun sistem itu sendiri manjadi sangat minim bahkan tidak ada.

Pengujian kenerja pada enkripsi/dekripsi data ini dilakukan untuk mengetahui kesalahan tersebut. Penanganan kesalahan pada enkripsi/dekripsi data dilakukan dengan memberikan peringatan dalam bentuk pesan kesalahan yang berisikan informasi kesalahan yang mungkin dilakukan oleh user.

Contoh penanganan kesalahan dapat dilihat pada gambar 6.1.

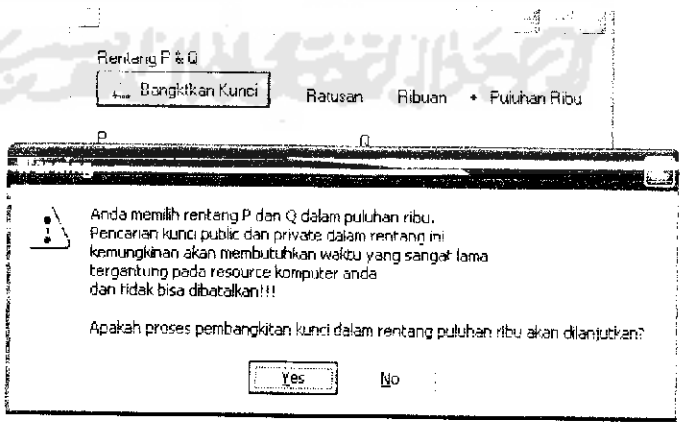


Gambar 6.1. Penanganan kesalahan pada dekripsi teks

Pada gambar di atas diperlihatkan contoh penanganan kesalahan yang terjadi ketika melakukan proses dekripsi teks. Kesalahan terjadi karena data yang dimasukan tidak sesuai dengan tipe data yang seharus menjadi masukan dimana teks yang akan didekripsi harus berupa bilangan bulat desimal.

6.2 Analisis Pembangkitan Kunci

Analsis pembangkitan kunci menjelaskan seluruh proses pembangkitan kunci. Dalam pembangkitan kunci terjadi proses pemilihan bilangan prima secara acak. Dalam program ini diberikan tiga pilihan bentuk bilangan prima yaitu ratusan, ribuan dan puluhan ribu. Semakin besar bilangan prima maka proses pembangkitan kunci akan membutuhkan waktu yang semakin lama, maka ketika user memilih bilangan prima yang besarnya pulahan ribu, program akan memberikan pesan peringatan "bahwa waktu yang dibutuhakan relatif lama, apakah proses akan dilanjutkan?" Lebih jelasnya lihat gambar 6.2.



Gambar 6.2. Pesan peringatan waktu pembangkitan kunci

Pada proses ini jika user memilih "yes" maka program akan dilanjutkan dengan kosekuensi waktu yang dibutuhkan untuk pembangkitan kunci relatif lama, tetapi jika user memilih "no" maka proses akan kembali ke menu pembangkitan kunci.

6.3 Analsis Input

Analsis input adalah menjelaskan keseluruhan masukan program yang berupa informasi masukan enkripsi. Input dari sistem dalam apalikasi ini adalah sebagai berikut:

6.3.1 Input Enkripsi file atau Folder

Dalam enkripsi file, input atau masukan adalah semua jenis tipe file. Dianjurkan untuk menggunakan file yang berukuran dibawah 50Mb hal ini disebabkan keterbatasan memori dan space hardisk pada PC. Dan juga semakin besar ukuran file maka semakin lama pula waktu yang dibutuhkan.

6.3.2 Input Enkripsi Teks

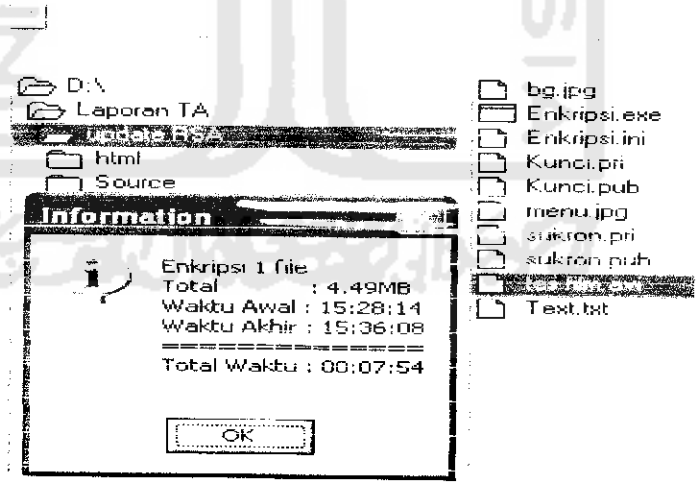
Masukan pada proses enkripsi teks adalah berupa data teks baik yang berupa file ataupun berupa tulisan. Disarankan apabila teks sudah berupa file gunakan enkripsi file karena jika file teks yang berukuran besar dienkripsi melalui enkripsi teks membutuhkan rentang waktu yang relatif lebih lama jika dibanding dengan enkripsi file.

6.4 Analisis Output

Analisis output menjelaskan keseluruhan hasil dari program yang berupa informasi dari hasil enkripsi data, baik berupa file maupun teks. Output dari aplikasi ini adalah sebagai berikut:

6.4.1 Output Enkripsi File

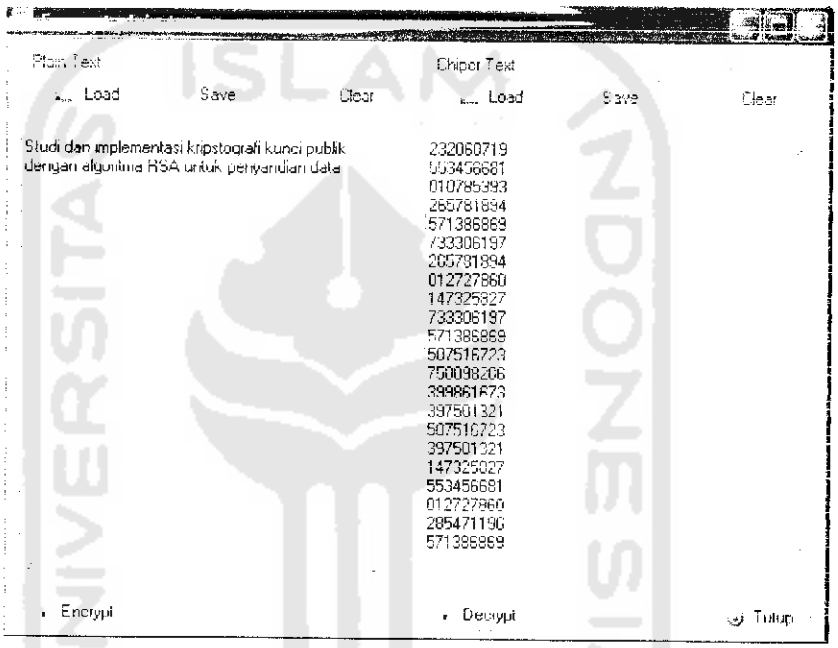
Dalam enkripsi file, output berupa file yang berekstensi *.enc (encrypt). Hal ini untuk memudahkan mengenali data yang telah dienkripsi sehingga ketika akan melakukan dekripsi user tinggal memilih file yang berekstensi *.enc, karena hanya file yang berekstensi *.enc yang bisa didekripsi. Output enkripsi juga berupa informasi ukuran file dan waktu proses yang ditampilakan melalui sebuah pesan. Untuk lebih jelasnya lihat gambar 6.3.



Gambar 6.3. Pesan informasi waktu untuk proses enkripsi file.

6.4.2 Output Enkripsi Teks

Hasil keluaran dari enkripsi teks adalah teks dekripsi yang berupa cipher teks atau data sandi yang berupa bilangan bulat desimal. Susunan bilangan desimalnya tersusun kebawah untuk setiap satu karaketer teks. Lihat gambar 6.4.



Gambar 6.4. Hasil enkripsi teks.

Pada ekripsi RSA, setiap satu karekter plainteks akan berubah menjadi bilangan desimal dengan panjang digit sama dengan digit panjang kunci (n).

6.5 Analsis Perbandingan

Analisis perbandingan digunakan untuk melihat kinerja dari perangkat lunak. Perbandingan disini berupa berbandingan waktu proses enkripsi dan dekripsi file, serta ukuran file setelah dienkripsi. Dalam perbadingan ini

digunakan program aplikasi kriptografi simetris yang lain yang telah jadi (Blowfish, RC5 dan DES). Untuk RSA akan digunakan 3 macam kunci () sedangkan untuk blowfish, RC5 dan DES menggunakan satu macam kunci. Kunci kuci tersebut adalah sebagai berikut:

1. RSA1 : d = 51239, e = 32759, n = 205571 (p dan q dalam rentang ratusan)
2. RSA2 : d = 6393251, e = 1611731 , n = 10990439 (p dan q dalam rentang ribuan)
3. RSA3 : d – 436195813, e – 384942277 , n – 780093779 (p dan q dalam rentang pulah ribu)
4. Blowfish : abcdefghijklmnopqrstuvwxyz
5. RC5 : abcdefghijklmnopqrstuvwxyz
6. DES : abcdefghijklmnopqrstuvwxyz

6.5.1 Analisis Perbandingan Waktu Proses

Waktu proses yang akan dibandingkan adalah waktu proses enkripsi dan dekripsi antara algoritma RSA, Blowfish, RC5 dan DES. Alasan pemilihan algoritma ini adalah karena programnya sudah ada sehingga tinggal dibandingkan dengan aplikasi yang dibuat yaitu RSA. Berdasarakan pengujian terhadap beberapa file diperoleh perbandingan waktu yang dapat dilihat pada tabel 6.1. dan 6.2. serta digambarkan dalam sebuah grafik diagram. Lihat gambar 6.5. dan 6.6.

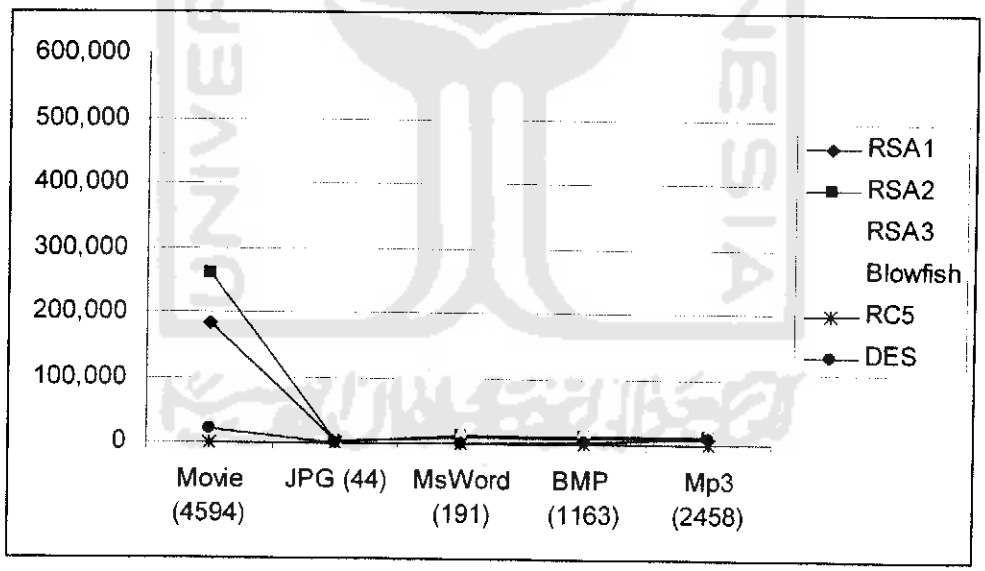
Tebel 6.1. Perbandingan waktu proses enkripsi

Jenis File dan Ukuran File (KB)	*Waktu Enkripsi (detik)*					
	RSA1	RSA2	RSA3	Blowfish	RC5	DES
Movie (4594)	185,006	262,638	510,885	0,625	0,528	20,689
JPG (44)	2,153	3,175	4,897	0,031	0,030	0,221
MsWord (191)	7,781	10,886	18,097	0,050	0,041	0,801
BMP (1163)	8,472	12,388	21,280	0,94	0,140	4,487
Mp3 (2458)	8,713	12,508	21,300	0,416	0,281	9,663

Sumber : Data pengujian

Kunci-Kunci Enkripsi

1. RSA1 : e = 32759, n = 205571 (p dan q dalam rentang ratusan)
2. RSA2 : e = 1611731 , n = 10990439 (p dan q dalam rentang ribuan)
3. RSA3 : e = 384942277 , n = 780093779 (p dan q dalam rentang puluhan ribu)
4. Blowfish : abcdefghijklmnopqrstuvwxyz
5. RC5 : abcdefghijklmnopqrstuvwxyz
6. DES : abcdefghijklmnopqrstuvwxyz



Gambar 6.5. Diagram perbandingan waktu proses enkripsi (dalam detik)

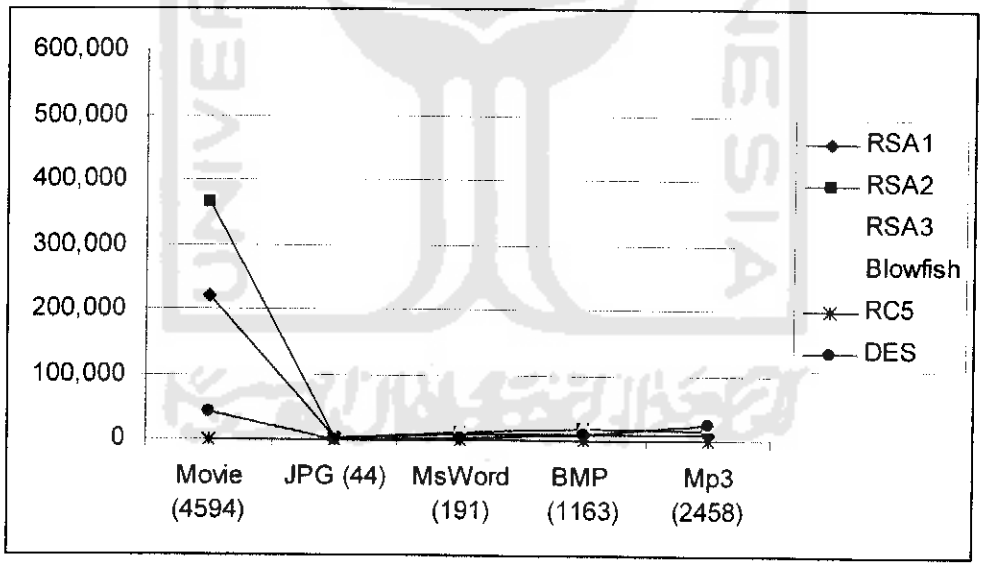
Tabel 6.2. Perbandingan waktu proses dekripsi

Jenis File dan Ukuran File (KB)	*Waktu Dekripsi (detik)*					
	RSA1	RSA2	RSA3	Blowfish	RC5	DES
Movie (4594)	222,269	366,638	510,855	1,121	1,062	43,878
JPG (44)	2,752	3,705	5,558	0,200	0,031	0,596
MsWord (191)	9,233	13,370	20,329	0,111	0,060	1,732
BMP (1163)	10,645	17,756	24,165	0,310	0,281	10,384
Mp3 (2458)	10,545	16,013	24,626	0,631	0,581	23,472

Sumber : Data pengujian

Kunci-Kunci Dekripsi

1. RSA1 : d = 51239, n = 205571 (p dan q dalam rentang ratusan)
2. RSA2 : d = 6393251, n = 10990439 (p dan q dalam rentang ribuan)
3. RSA3 : d = 436195813, n = 780093779 (p dan q dalam rentang puluhan ribu)
4. Blowfish : abcdefghijklmnopqrstuvwxyz
5. RC5 : abcdefghijklmnopqrstuvwxyz
6. DES : abcdefghijklmnopqrstuvwxyz



Gambar 6.6. Diagram waktu proses dekripsi (dalam detik)

Dari data yang diperoleh ternyata RSA membutuhkan waktu lebih lama dibandingkan dengan Blowfish, RC5 dan DES. Karena perbedaan yang cukup

jauh dalam waktu proses yang terjadi pada aplikasi RSA, hal ini menjadi kelemahan RSA, maka dari itu sebaiknya RSA digunakan untuk file yang berukuran kecil sehingga waktu yang dibutuhkan juga tidak terlalu lama.

6.5.2 Analisis Perbandingan Penggunaan Memori

Salah satu kriteria keunggulan suatu sistem aplikasi adalah dapat dilihat dari kebutuhan memori, maka dari itu perlu dibandingkan dalam penggunaan memori antara aplikasi yang telah dibuat yaitu RSA dengan sistem enkripsi lain yaitu Blowfish, RC5 dan DES untuk penggunaan file yang sama. Dari pengamatan kebutuhan RSA pada saat inisialisai bilangan prima tercatat 4,824 KB dan pada saat program *stand by* tercatat 8,593 KB. Untuk melihat perbandingan kebutuhan memori ketika program sendang menjalankan proses enkripsi dapat dilihat pada tabel 6.3. serta digambarkan dalam sebuah grafik diagram. Lihat gambar 6.7.

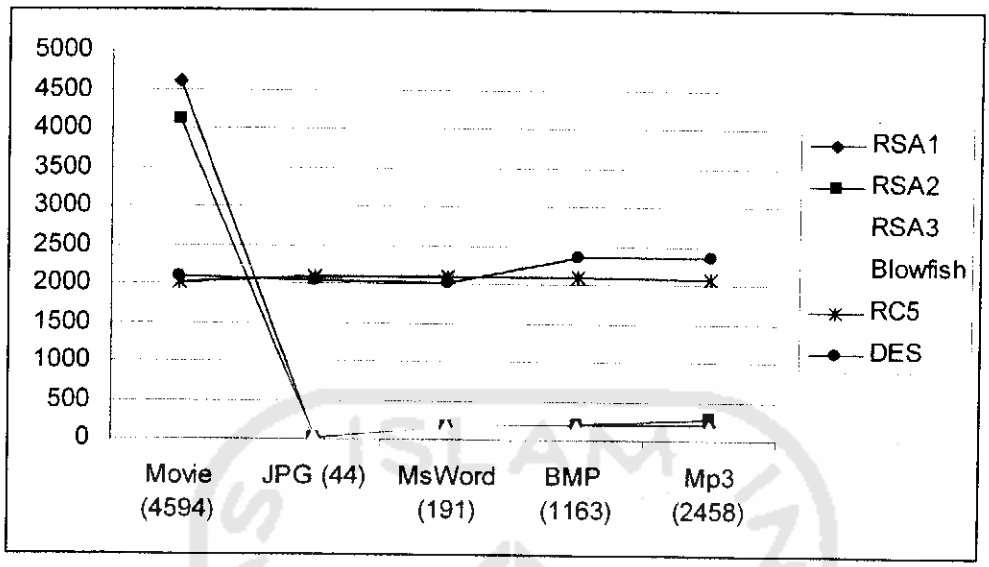
Tabel 6.3. Perbandingan pemakain memori

Jenis File dan Ukuran File (KB)	*Kebutuhan Memori Enkrpis (KB)*					
	RSA1	RSA2	RSA3	Blowfish	RC5	DES
Movie (4594)	4608	4132	3772	2396	2026	2100
JPG (44)	64	32	48	2076	2104	2044
MsWord (191)	192	188	192	2120	2092	2020
BMP (1163)	204	211	192	2312	2108	2368
Mp3 (2458)	204	274	192	3924	2076	2368

Sumber : Data pengujian

Kunci-Kunci Enkripsi

1. RSA1 : c = 32759, n = 205571 (p dan q dalam rentang ratusan)
2. RSA2 : e = 1611731 , n = 10990439 (p dan q dalam rentang ribuan)
3. RSA3 : e = 384942277 , n = 780093779 (p dan q dalam rentang puluhan ribu)
4. Blowfish : abcdefghijklmnopqrstuvwxyz
5. RC5 : abcdefghijklmnopqrstuvwxyz
6. DES : abcdefghijklmnopqrstuvwxyz



Gambar 6.7. Diagram perbandingan pemakaian memori (dalam KB)

Dari data yang ada kebutuhan memori untuk proses aplikasi RSA relatif lebih kecil dibanding dengan 3 program pembanding lainnya. Kebutuhan memori terbesar pada aplikasi RSA bisa terjadi pada proses enkripsi file movie (berekstensi *.avi). maka dari itu untuk melakukan enkripsi pada file selain movie RSA dapat digunakan sebagai aplikasinya, namun yang perlu diketahui adalah program aplikasi yang dibuat memang membutuhkan memori yang cukup besar yaitu kurang lebih 8 MB pada saat aplikasi dalam keadaan *stand by* hal ini mungkin bisa digunakan sebagi pertimbangan penggunaan RSA.

6.5.3 Analisis Perbandingan Ukuran File

Dalam setiap perubahan file terutama pada proses enkripsi akan menghasilkan ukuran file yang berbeda dengan file asal. Demikian pula yang terjadi pada aplikasi ini. Untuk perbandingan ukuran file hasil enkripsi, maka akan

dibandingkan dengan beberapa algoritma kriptografi simetris (Blowfish, RC5 dan DES). Berdasarkan pengujian terhadap beberapa jenis dan ukuran file diperoleh perbandingan perubahan ukuran file yang dapat dilihat pada tabel 6.4.

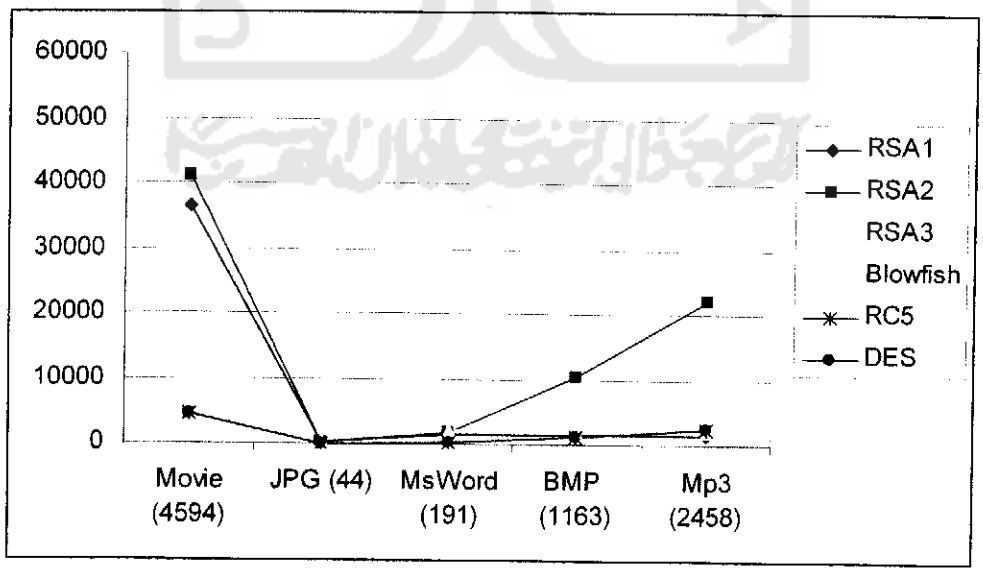
Tabel 6.4. Perbandingan perubahan ukuran file hasil enkripsi (dalam KB)

Jenis File dan Ukuran File (KB)	*Hasil File Enkripsi (KB)*					
	RSA1	RSA2	RSA3	Blowfish	RC5	DES
Movie (4594)	36740	41342	50528	4594	4594	4594
JPG (44)	352	396	483	44	44	44
MsWord (191)	1528	1719	2101	191	191	191
BMP (1163)	1589	10464	2184	1163	1163	1163
Mp3 (2458)	1620	22119	2272	2458	2458	2458

Sumber : Data pengujian

Kunci-Kunci Enkripsi

1. RSA1 : e = 32759, n = 205571 (p dan q dalam rentang ratusan)
2. RSA2 : c = 1611731 , n = 10990439 (p dan q dalam rentang ribuan)
3. RSA3 : e = 384942277 , n = 780093779 (p dan q dalam rentang puluhan ribu)
4. Blowfish : abcdefghijklmnopqrstuvwxyz
5. RC5 : abcdefghijklmnopqrstuvwxyz
6. DES : abcdefghijklmnopqrstuvwxyz



Gambar 6.8. Diagram perbandingan perubahan ukuran file hasil enkripsi.

Dari data yang ada terjadi perubahan ukuran file menjadi lebih besar hanya terjadi pada aplikasi RSA. Sementara 3 program pembading lainnya tetap (tidak terjadi perubahan ukuran file). Ini menjadi salah satu kelemahan RSA. Perubahan file yang terjadi sangat besar rata-rata adalah besar file asal kali panjang digit kunci. Maka untuk file-file yang besar sebaiknya tidak menggunakan RSA.

6.6 Kelebihan dan Kekurangan RSA

Dari analisis di atas dapat diketahui bahwa algoritma RSA mempunyai kelebihan tingkat keamanan yang yang baik ini terbukti bahwa RSA menjadi standar de facto dunia bagi kriptografi asimetrik sampai saat ini. Selain itu RSA juga mempunyai kelebihan keamanan kunci yang lebih terjamin karena kunci untuk dekripsi berbeda dengan kunci enkripsi dan kunci dekripsi sama sekali tidak diberikan kepada orang lain, hal ini juga berakibat penggunaan kunci yang sangat sedikit. Hal ini berbeda dengan 3 program pembanding (Blowfish, RC5 dan DES) yang merupakan kriptografi simetris dimana kunci enkripsi dan dekripsi adalah sama hal ini yang menyebabkan keamanan kunci yang kurang terjamin dan kebutuhan kunci yang lebih banyak dibanding dengan kriptografi asimetris. Kalau kunci yang dibutuhkan oleh kriptografi asimetris adalah n pasang kunci (n adalah jumlah komputer yang tersambung dalam jaringan), sedangkan kalau kriptografi simetris dibutuhkan jumlah n^2 (n adalah jumlah komputer tersambung dalam jaringan)

Namun disamping beberapa kelebihan tadi, RSA juga mempunyai kekurangan yaitu hasil file enkripsi yang besar, hal ini ini menyebabkan

kubutuhan space hardisk yang besar. Disamping itu, waktu proses yang lama menjadi kelemahan RSA.

6.7 Analisis Keamanan Algoritma RSA

Kekuatan keamanan kriptosistem RSA terletak pada susahnya memfaktorkan bilangan-bilangan yang besar. Dengan asumsi bahwa seorang penyerang memiliki akses penuh terhadap kunci publik (e, n) dan ciphertext, maka untuk mendekripsi ciphertext penyerang harus mampu memfaktorkan n ($n = p\,q$). Jika nilai p dan q telah diketahui, penyerang dapat menghitung kunci pribadi d dari e dengan menggunakan algoritma extended Euclidian. Bila d sudah diperoleh, maka ciphertext dapat didekripsi menjadi plaintext semula.

Saat ini teknik pemfaktoran bilangan 120 digit sudah tidak membutuhkan waktu yang begitu lama jika dikerjakan dengan komputer. Oleh karena itu agar tingkat keamanannya tinggi, kriptografi RSA minimum harus menggunakan modulus 512 bit (154 digit). Kebanyakan implementasi kriptografi RSA saat ini menggunakan modulus 1024 bit. Walau demikian kriptografi RSA 2048 bit sudah diimplementasikan dan mulai digunakan.