Model Pengamanan Berkas Menggunakan Kriptografi Asimetris RSA Dan Algoritma Kompresi PPM Pada File Curriculum Vitae (CV)

Siti Annisa[1], Henki Bayu Seta[2], Noor Falih[3]
Program Studi Informatika / Fakultas Ilmu KomputerUniversitas Pembangunan Nasional Veteran Jakarta
JL. RS Fatmawati No. 1, Pondok Labu, Jakarta Selatan, DKI Jakarta 12450
[1]sitia@upnvj.ac.id, [2]henkiseta@upnvj.ac.id, [3]falih@upnvj.ac.id

Abstrak. Model pengamanan berkas adalah suatu model yang mengimplementasikan satu atau lebih algoritma keamanan untuk melindungi keamanan berkas. Sudah banyak penelitian yang mengusung model pengamanan berkas. Namun, berdasarkan penelitian sebelumnya, model pengamanan berkas masih dapat dikembangkan. Peneliti mengusung model pengamanan berkas baru menggunakan kriptografi asimetris RSA dan algoritma kompresi PPM. Kriptografi asimetri RSA dipilih karena kuatnya algoritma dalam mengamankan berkas. Algoritma kompresi PPM dipilih sebagai penyokong kelemahan RSA dengan mengurangi besarnya ukuran *file ciphertext* yang dihasilkan. Penelitian ini bertujuan untuk mengukur performa dari model pengamanan berkas menggunakan kriptografi asimetris RSA dan algoritma kompresi PPM dari segi keamanan, waktu, dan ukuran berkas yang dihasilkan. Dari 5 berkas CV yang digunakan didapat hasil waktu komputasi sebesar 0,37 detik dengan rasio kompresi sebesar 70,7611 %. Dengan demikian hasil dari model yang menggunakan RSA dan PPM memiliki hasil rasio kompresi yang lebih baik dibandingkan model pengamanan berkas terhadulu yang menggunakan Blowfish dan LZW.

Kata Kunci: Model pengamanan berkas, RSA, PPM

1 Pendahuluan

Model pengamanan berkas adalah suatu model yang mengimplementasikan satu atau lebih algoritma keamanan, dalam hal ini kriptografi, untuk melindungi keabsahan, integritas, dan keamanan berkas. Sudah banyak penelitian yang mengusung model pengamanan berkas guna memenuhi kebutuhan akan perlindungan keamanan. Terdapat sekelompok peneliti yang melakukan sebuah penelitian berjudul *Analisis Performa Kriptografi Hybrid Algoritma Blowfish dan Algoritma RSA,* penelitian [1] menghasilkan suatu aplikasi yang menerapkan algoritma Blowfish dan RSA dalam mengamankan data. Dari penelitian ini didapat suatu model pengamana yang lebih aman karena gabungan kekuatan dari algoritma RSA dan algoritma *blowfish* untuk untuk menanggulangani kelemahan RSA dalam waktu pemrosesan. Namun, kecepatan algoritma gabungan dalam mengenkripsi dan mendekripsi data tidak lebih cepat dibanding algoritma pembentuknya, yaitu RSA atau Blowfish.

Lalu pada tahun 2018 terdapat suatu penelitian yang berjudul *Implementasi Algoritma Prediction by Partial Matching (PPM) Pada Kompresi File Teks Terenkripsi Elgamal*. Dari penelitian [2] didapatkan kesimpulan berupa algoritma PPM baik digunakan untuk mengompresi *file* berbentuk teks, dengan hasil akhiran kompresi lebih kecil dibanding *file* awal dan algoritma ElGamal dapat membantu mengamankan informasi yang berada dalam *file* teks. Selain dari dua penelitian tersebut, terdapat penelitian yang membandingkan beberapa algoritma kriptografi asimetris berdasarkan performanya. Dari membandingkan performa tiga algoritmaasimetris paling terkenal yaitu RSA, ECC, dan ElGamal, penelitian [3] mampu membuktikan bahwa RSA terbukti unggul dalam penyandian data, keamanan, serta lebih cepat dan *valid*.

Berdasarkan referensi penelitian yang telah peneliti temukan, peneliti mengusung untuk menggunakan kriptografi asimetris RSA, dimana RSA merupakan kriptografi asimetris yang paling aman dengan tingkat kesulitan pemecahan kunci yang tinggi. Lalu peneliti mengusung untuk menggunakan algoritma kompresi Prediction by Partial Matching atau PPM, dimana algoritma efektif untuk mengurangi ukuran berkas, sehingga dapat meningkatkan performa algoritma RSA.

Peneliti mengusung untuk menggunakan berkas Curriculum Vitae (CV) sebagai objek dari penelitian ini

sebagaimana CV merupakan berkas yang berisi data diri pribadi yang bersifat privasi yang membutuhkan pengamanan, yang mana hal ini dibahas dalam Permenkominfo No. 20 tahun 2016 dimana tiap warga negara berhak atas perlindungan data pribadi dalam sistem elektronik. Berdasarkan paparan yang sudah diuraikan, maka penulis akan mengusulkan penggunaan algoritma RSA dan algoritma kompresi PPM dalam melindungi berkas dan mengukur performanya dari segi keamanan, waktu pemrosesan, dan ukuran berkas yang dihasilkan. Sehingga penelitian ini akan mengangkat judul "Model Pengamanan Berkas Menggunakan Kriptografi Asimetris RSA dan Algoritma Kompresi PPM Pada File Curriculum Vitae (CV)".

2 Tinjauan Pustaka

2.1 Algoritma PPM

Algoritma PPM atau *Prediction by Partial Matching* (PPM) adalah algoritma kompresi data lossless dimana hasil kompresi tidak akan menghilangkan atau mengurangi suatu informasi[4].

2.1.1 Proses *Encode*

Tahapan proses *encoding* dengan algoritma PPM adalah sebagai berikut:

1. Berikan nilai probabilitas terhadap simbol yang akan di-*encode*
2. Berikan rentang nilai dari 0-1 sesuai dengan probabilitas yang ada. Tidak adaketentuan urutan, asal *encoder* dan *decoder* melakukan hal yang sama.
3. Tetapkan nilai *low* sebagai 0, dan *high* sebagai 1
4. Selama simbol masukkan masih ada, lakukan:
5. Ambil simbol *input*
6. CR adalah *high* dikurang *low*
7. *High* adalah *low* ditambah CR yang dikali *range* tertinggi simbol
8. *Low* adalah *low* ditambah CR yang dikali *range* terkecil simbol
9. Bila sudah tidak ada simbol yang tersisa cetak *Low* dimana *Low* adalah *outputencode* atau hasil *encode*

2.1.2 Proses *Decode*

Tahapan proses *decoding* dengan algoritma PPM adalah sebagai berikut:

1. Ambil hasil *output encode* sebagai *Encoded Symbols* (ES)

Lakukan :

a. Cari *range* dari simbol yang merupakan ES
b. Cetak simbol tersebut
c. CR adalah *range* tertinggi simbol dikurang *range* terendah simbol
d. Lakukan suatu proses pada ES dimana ES yang baru adalah ES dikurang *range* terendah simbol
e. Lakukan suatu proses pada ES dimana ES yang baru adalah ES dibagidengan CR
f. Lakukan hingga simbol habis

2.2 RSA

Algoritma RSA merupkan algoritma kunci publik yang paling aman dan banyak digunakan.Algoritma ini dinamai dari tiga orang penemunya; (R)ivest, (S)hamir, (A)dleman. Keamanan algoritma RSA terletak pada mudahnya mengalikan dua bilangan prima besar, dan sulitnya memfaktorkan hasil perkalian dua bilangan prima besar tersebut. Nilai faktor kedua bilangan tersebut atau modulus bilangan tersebut memengaruhi kekuatan algoritma ini, semakin besar nilainya maka semakin kuat *cipher* kita[5].

3 Metodologi Penelitian

3.1 Identifikasi Masalah

Pada tahapan ini, penulis mengidentifikasi beberapa penelitian terdahulu yang dapat dikembangkan. Peneliti pun membuat hipotesa tentang model pengamanan berkas menggunakan kriptografi asimetris RSA dan algoritma kompresi PPM, apakah penggunaan kedua algoritma tersebut sebagai alat untuk mengamankan berkas dapat menghasilkan performa yang lebih baik dibandingkan dengan model pengamanan yang sudah ada pada penelitian sebelumnya.

3.2 Tinjauan Pustaka

Penulis melakukan tinjauan pustaka pada jurnal dan buku yang berkaitan dan sejenis dengan tulisan ini yang menunjang dasar penulisan.

3.3 Analisa dan Perancangan Sistem

Penulis akan menganalisa kebutuhan sistem dan langkah yang akan dilakukan oleh sistem.

3.4 Implementasi

Penulis akan merealisasikan sistem yang dirancang menjadi sebuah *website* yang dapat melakukan pekerjaan sesuai dengan tujuan awal yang diinginkan

3.5 Pengujian Sistem

Penulis akan mencari tahu estimasi waktu yang dibutuhkan mulai dari kompresi, enkripsi, dekripsi, hingga dekompresi *file* serta membandingkan waktu yang dibutuhkan dari *file* yang tidak di kompresi dan dekompresi, serta mengevaluasi perubahan ukuran *file* sebelum dan sesudah proses pengamanan selesai.

3.6 Simpulan dan Dokumentasi

Tahap akhir yang ditempuh adalah penarikan kesimpulan, apakah sistem dapat memberikan hasil yang menjawab permasalahan yang ada atau tidak, serta melakukan dokumentasi penulisan.

4 Hasil dan Pembahasan

Website diciptakan dengan tujuan menguji performa kombinasi model pengamanan berkas yang menggunakan teknik kompresi PPM dan kriptografi asimetris RSA pada *file* curriculum vitae (CV). *Website* menggunakan teknik kompresi PPM untuk mengurangi ukuran *file* yang akan dienkripsi menggunakan kriptografi asimetris RSA untuk mengamankan *file,* sehingga isi *file* tidakdapat dimaknai bagi orang yang tidak memiliki kunci privat untuk menerjemahkannya.

4.1 Analisis Proses Kompresi dan Enkripsi

Pada proses kompresi dan enkripsi tahapan-tahapan yang akan dijalankan berupa sebagai berikut:

1. Memasukkan *file* dengan format .doc, .docx, .txt, atau .pdf dan kunci publik yangdibutuhkan pada proses enkripsi.
2. *File* akan dikompresi terlebih dahulu sebelum dienkripsi.
3. Tahapan dari kompresi yang akan dilakukan adalah sebagai berikut:

Terdapat file sitiannisa.txt berisi string “SITI_ANNISA” yang akan di-*encode*, didapat tabel probabilitas dan range-nya sebagai berikut:

Tabel 1. Range dan Probabilitas Tiap Karakter

Karakter	Probabilitas	Range
S	2/10	0,8 – 1,0
I	3/10	0,5 – 0,8

ISSN : 2655-139X (ONLINE)
ISSN : 0216-4221 (PRINT)

T	1/10	0,4 – 0,5
_	1/10	0,3 – 0,4
A	1/10	0,2 – 0,3
N	2/10	0 – 0,2

1. S
   CR = 1 – 0 = 1, range tertinggi(S) = 1, range terendah(S) = 0,8
   High = 0 + CR * 1 = 1
   Low = 0 + CR * 0,8 = 0,8
2. I
   CR = 1 – 0,8 = 0,2, range tertinggi(I) = 0,8, range terendah(I) = 0,5
   High = 0,8 + CR * 0,8 = 0,96
   Low = 0,8 + CR * 0,5 = 0,9
3. S
   CR = 0,927576576 – 0,92757636= 0,000000216, range tertinggi(T) = 1, range terendah(T) = 0,8
   High = 0,92757636+ CR * 1 = 0,927576576
   Low = 0,92757636+ CR * 0,8 = 0,9275765328

Tabel 2. Hasil Proses Kompresi

Karakter	LOW	HIGH	CR
	0	1	1
S	0,8	1	0,2
I	0,9	0,96	0,06
T	0,924	0,93	0,006
I	0,927	0,9288	0,0018
_	0,92754	0,92772	0,00018
A	0,927576	0,927594	0,000018
N	0,927576	0,9275796	0,0000036
N	0,927576	0,92757672	0,00000072
I	0,92757636	0,927576576	0,000000216
S	0,9275765328	0,927576576	0,0000000432

Dari keseluruhan proses akan didapat Low = 0,9275765328, bila diubah dalam bentuk kode ASCII-nya yaitu 484457505553555453515056, nilai inilah yang akan digunakan untuk menggantikan string “SITI_ANNIS”.

a. Hasil kompresi kemudian akan dienkripsi dengan menggunakan kunci publik.Tahapan dari pembangkitan kunci publik sendiri adalah sebagai berikut:
b. Pilih bilangan bulat prima p dan q, p = 98689 dan q = 98389.
c. Hitung r, dimana r merupakan hasil kali p dan q,r = 98689 *98389 = 9709912021
d. Hitung φ (r) = (p-1)(q-1)
   φ (r) = (98689-1)(98389-1) = 9709714944
e. Pilih kunci publik yang prima terhadap φ (r), yang berarti faktor pembagiterbesarnya adalah 1, atau GCD(e,m) = 1.
f. GCD(e, 9709714944) = 1
g. e = 3221
h. kunci publiknya {e,r} = {3221, 9709912021}

Tahapan enkripsi adalah sebagai berikut:
a. Hasil kompresi m = 0,9275765328 atau bila diubah dalam bentuk kode ASCII-nya yaitu 484457505553555453515056
b. Pecah m atau kode ASCII menjadi blok-blok lebih kecil yaitu 12 blok dengan ukuran 2 digit, nilai m haruslah masih terletak dalam rentang [0, 9709912021-1] agar dapat dilakukan transformasi satu-ke-satu
   $m_1 = 48$ $m_4 = 50$ $m_7 = 55$ $m_{10} = 51$
   $m_2 = 44$ $m_5 = 55$ $m_8 = 54$ $m_{11} = 50$

$m_3 = 57$ $m_6 = 53$ $m_9 = 53$ $m_{12} = 56$

c. Kunci publik B berupa {e, r} = {3221, 9709912021}maka tiap blok-blok yang akandienkripsi sebagai berikut:
   $C_1 = 48^{3221} \bmod 9709912021 = 18185970$
   $C_2 = 44^{3221} \bmod 9709912021 = 6052028637$
   $C_3 = 57^{3221} \bmod 9709912021 = 6591183325$
   $C_{12} = 56^{3221} \bmod 9709912021 = 1832697024$

d. Sehingga *ciphertext* atau hasil enkripsi sebagai berikut :
C = 18185970605202863765911833251388688514327207665151877032373272076651590734595751877032377886389029138868851418326970245159073459575187703237788638902913886885141832697024

4.2 Analisa Proses Dekripsi dan Dekompresi

Pada proses dekripsi dan dekompresi tahapan-tahapan yang akan dijalankan berupa sebagaiberikut:

1. Memasukkan *file* yang telah terenkripsi dan kunci privat yang dibutuhkan pada prosesdekripsi.
2. *File* selanjutnya akan didekripsi dengan menggunakan kunci privat. Tahapan daripembangkitan kunci privat sendiri adalah sebagai berikut:
3. Bangkitkan kunci privat dengan rumus berikut $d = e^{-1} \bmod \varphi(r)$ atau $ed = 1 \bmod \varphi(r)$atau $d = ((\varphi(r)*i)+1)/e$
   d = (9709714944+1)/3221 = 3.014.503,2427 (i = 1)
   d = (19419429888+1)/3221 = 6.029.006,4852 (i = 2)
   d = (29.129.144.832+1)/3221 = 9.043.509,7277 (i = 3)
   d = (18.700.910.982.144+1)/3221 = 5.805.933.245,0000 (i = 1926)
4. Sehingga didapat kunci privat = {d, r}, d = 5.805.933.245 dan r = 9.709.912.021.

Tahapan dari proses dekripsi *file* adalah sebagai berikut:

1. Tiap blok yi akan dikembalikan dengan rumus sebagai berikut:xi = yiSK mod r
   Ciphertext = 18185970605202863765911833251388688514327207665151877032373272076651590734595751877032377886389029138868851418326970245, d = 5.805.933.245.
   C1 = 18185970 C4 = 1388688514
   C2 = 6052028637 C5 = 3272076651
   C3 = 6591183325 C6 = 5187703237
   C7 = 3272076651 C10 = 7886389029
   C8 = 5907345957 C11 = 1388688514
   C9 = 5187703237 C12 = 1832697024
2. Kunci publik B berupa {d, r} = {5.805.933.245, 9.709.912.021}maka tiap blok-blok yangakan dienkripsi sebagai berikut:
   $m1 = 18185970^{5.805.933.245} \bmod 9709912021 = 48$
   $m2 = 6052028637^{5.805.933.245} \bmod 9709912021 = 44$
   $m3 = 6591183325^{5.805.933.245} \bmod 9709912021 = 57$
   $m12 = 1832697024^{5.805.933.245} \bmod 9709912021 = 56$
3. Lalu blok m1, m2, m3 akan diubah kembali dari bilangan menjadi huruf denganmenggunakan kode ASCII hasil dekripsi.
4. Hasil dekripsinya adalah 484457505553555453515056, kode ASCII berarti 0,9275765328.

File yang telah didekripsi kemudian akan dikembalikan ke ukuran semula dengan prosesdekompresi.

Tahapan dari proses dekompresi adalah sebagai berikut:

1. *Decode* dari output proses *encode* sebelumnya 0,92757653281.
   a. “S”, range 0,8 - 1

   CR = 1 – 0,8 = 0,2

ES = 0,9275765328 – 0,8 = 0,1275765328
ES = 0,1275765328 / 0,2 = 0,637882664

b. "I", range 0,5 – 0,8

CR = 0,8 – 0,5 = 0,3
ES = 0,637882664 – 0,5 = 0,137882664
ES = 0,137882664 / 0,3 = 0,45960888

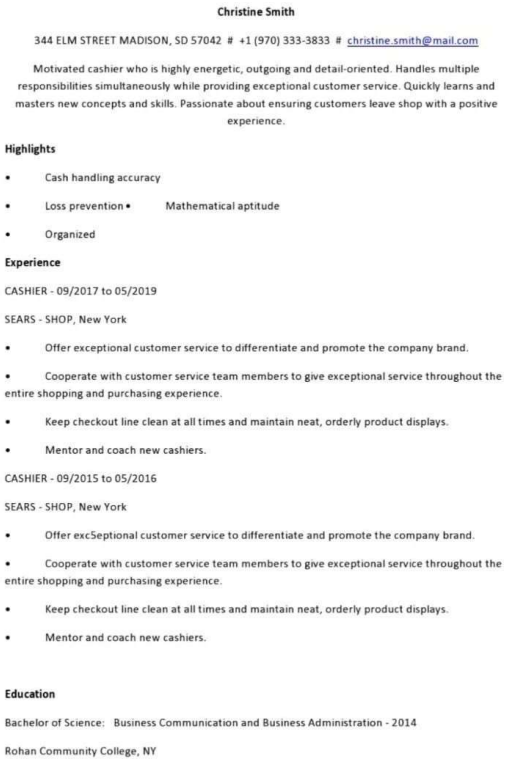
c. "S", range 0,8 – 1

CR = 1 – 0,8 = 0,2
ES = 0,8 – 0,8 = 0
ES = 0 / 0,2 = 0

2. Hasil yang didapatkan dari proses dekripsi dan dekompresi adalah *file* asli yang memilikiukuran awal sebelum terkompresi dan tidak terenkripsi.

5 Implementasi

Tahap ini merupakan pengimplementasian kode yang dikembangkan menjadi sebuah *website*. Berikut ini merupakan hasil implementasi kode dengan menggunakan berkas 'Christine Smith CV.doc' yang memiliki isi sebagai berikut :



Christine Smith

344 ELM STREET MADISON, SD 57042 # +1 (970) 333-3833 # christine.smith@mail.com

Motivated cashier who is highly energetic, outgoing and detail-oriented. Handles multiple responsibilities simultaneously while providing exceptional customer service. Quickly learns and masters new concepts and skills. Passionate about ensuring customers leave shop with a positive experience.

Highlights

- Cash handling accuracy
- Loss prevention • Mathematical aptitude
- Organized

Experience

CASHIER - 09/2017 to 05/2019

SEARS - SHOP, New York

- Offer exceptional customer service to differentiate and promote the company brand.
- Cooperate with customer service team members to give exceptional service throughout the entire shopping and purchasing experience.
- Keep checkout line clean at all times and maintain neat, orderly product displays.
- Mentor and coach new cashiers.

CASHIER - 09/2015 to 05/2016

SEARS - SHOP, New York

- Offer exc5eptional customer service to differentiate and promote the company brand.
- Cooperate with customer service team members to give exceptional service throughout the entire shopping and purchasing experience.
- Keep checkout line clean at all times and maintain neat, orderly product displays.
- Mentor and coach new cashiers.

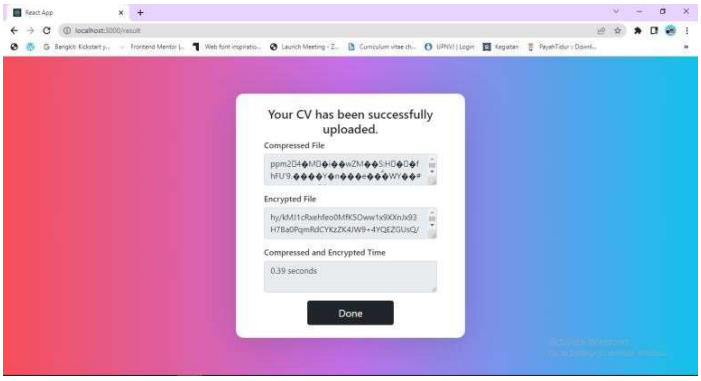
Education

Bachelor of Science: Business Communication and Business Administration - 2014

Rohan Community College, NY

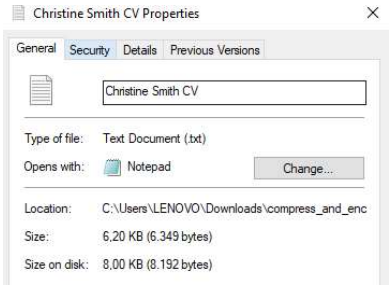
Gambar. 1. Gambar ini menunjukan contoh berkas Curriculum Vitae (CV) yang digunakan dalam pengaplikasian model pengamanan berkas menggunakan RSA dan PPM.

Proses kompresi dan enkripsi pada website menghasilkan output sebagai berikut:



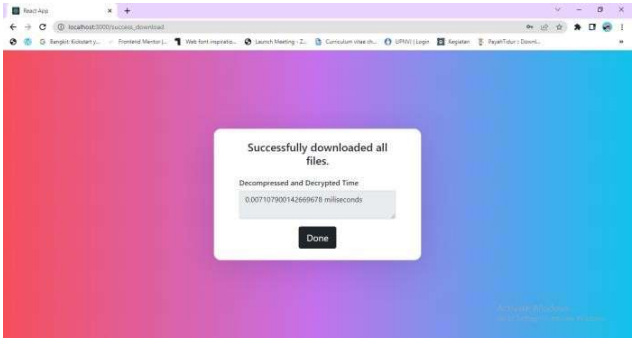
Gambar. 2. Gambar ini menunjukan proses Kompresi dan Enkripsi Berkas CV menggunakan model pengamanan berkas RSA dan PPM yang telah dibuat peneliti. Dalam gambar ini ditunjukan hasil kompresi, enkripsi, dan waktu komputasi.

Berkas yang telah dikompresi dan dienkripsi berubah ukurannya menjadi 6 kb dimana ukuran awalnya adalah 23 kb, dimana hasil berkas yang telah terkompresi dan terenkripsi akan tersimpan dalam format .txt.



Gambar. 3. Gambar ini menunjukan ukuran berkas setelah proses encode, dimana berkas berkurang ukurannya menjadi 6,2 KB.

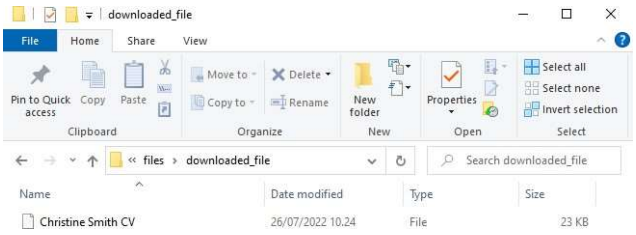
Proses dekripsi dan dekompresi pada berkas yang telah ter-encode pada website dilakukan dalam proses berikut :



Gambar. 4. Gambar ini menunjukan waktu komputasi dari proses dekripsi dan dekompresi berkas CV yang kita masukkan, yaitu berkas pada Gambar.1.

Berkas yang telah de-decode berubah ukurannya menjadi ukuran semula yaitu 23 kb, dimana berkas dapat dilihat pada direktori komputer lokal Anda dalam folder downloaded_files.

ISSN : 2655-139X (ONLINE)
ISSN : 0216-4221 (PRINT)



Gambar. 5. Gambar ini menunjukan ukuran berkas setelah di-decode, dimana berkas kembali memiliki ukuran yang sama seperti sebelum dilakukan proses encode.

5.1 Pengujian

5.1.1 Pengujian Rasio Kompresi & Waktu Komputasi

Pengujian ini dilakukan dengan tujuan melihat rasio kompresi serta waktu proses *encode* dan *decode*.

Tabel 3. Hasil Proses *Encode*

No	Nama file	Ukuran file asli (Kb)	Ukuran setelah *encode* (Kb)	Waktu proses *encode* (detik)	Rasio Kompresi (%)
1	Christine Smith CV.doc	23	7	0.45	69,5652
2	Christoper Morgan CV.doc	24	7	0.34	70,8333
3	Claudia Alves CV.doc	23	7	0.31	69,5652
4	Damien Le Grand CV.doc	29	8	0.36	72,4137
5	Dara Farhan.doc	28	8	0.39	71,4285

Pada tabel di atas proses *encode* pada 5 berkas menghasilkan ukuran kompresi, waktu proses *encode*, dan rasio kompresi yang berbeda-beda.

Tabel 4. Hasil Proses *Decode*

No	Nama file	Ukuran setelah *encode* (Kb)	Ukuran setelah *decode* (Kb)	Waktu proses *decode* (milisecond)
1	Christine Smith CV.doc	7	23	0,05
2	Christoper Morgan CV.doc	7	24	0,049
3	Claudia Alves CV.doc	7	23	0,05
4	Damien Le Grand CV.doc	8	29	0,051
5	Dara Farhan.doc	8	28	0,048

Pada tabel diatas menunjukan ukuran berkas setelah *decode* sama dengan ukuran berkas asli.

5.1.2 Pengujian Keterkaitan Spesifikasi Perangkat

Pengujian ini dilakukan dengan tujuan melihat pengaruh spesifikasi perangkat yan digunakandalam menjalankan *website*.

Pada penelitian yang menggunakan perangkat laptop merk Lenovo tipe K2450 yang menggunakanintel core i5 GEN 4, RAM sebesar 8 GB, Windows 10, dan HDD 500 GB menghasilkan rata-rata rasio kompresi sebesar 71,8065% dan rata-rata waktu pemrosesan *encode* sebesar 0,33 detik. Hasildapat dilihat pada tabel dibawah ini.

Tabel 5. Hasil Proses *Encode* di Perangkat 1

No	Nama file	Ukuran file asli (Kb)	Ukuran setelah *encode* (Kb)	Waktu proses *encode* (detik)	Rasio Kompresi (%)
1	Christine Smith CV.doc	23	7	0.45	69,5652
2	Christoper Morgan CV.doc	24	7	0.34	70,8333
3	Claudia Alves CV.doc	23	7	0.31	69,5652
4	Damien Le Grand CV.doc	29	8	0.36	72,4137
5	Dara Farhan.doc	28	8	0.39	71,4285

Pada penelitian yang menggunakan perangkat yang menggunakan intel core i5, RAM sebesar 16 GB, Windows 10, menggunakan HDD sebesar 1000 GB dan SSD sebesar 240 GB menghasilkan rata-rata rasio kompresi sebesar 71,8065% dan rata-rata waktu pemrosesan *encode* sebesar 0,1246 detik. Hasil dapat dilihat pada tabel dibawah ini.

Tabel 6. Tabel Hasil Proses *Encode* di Perangkat 2

No	Nama file	Ukuran file asli (Kb)	Ukuran setelah *encode* (Kb)	Waktu proses *encode* (detik)	Rasio Kompresi (%)
1	Christine Smith CV.doc	23	7	0.16	69,5652
2	Christoper Morgan CV.doc	24	7	0.14	70,8333
3	Claudia Alves CV.doc	23	7	0.12	69,5652
4	Damien Le Grand CV.doc	29	8	0.17	72,4137
5	Dara Farhan.doc	28	8	0.14	71,4285

Pada penelitian yang menggunakan perangkat laptop merk Dell tipe latitude E6410 yang menggunakan intel core i5 GEN 1, RAM sebesar 2 GB, Windows 10, dan HDD 320GB menghasilkan rata-rata rasio kompresi sebesar 71,8065% dan rata-rata waktu pemrosesan *encode* sebesar 0,98 detik. Hasil dapat dilihat pada tabel dibawah ini.

Tabel 7. Tabel Hasil Proses *Encode* di Perangkat 3

No	Nama file	Ukuran file asli (Kb)	Ukuran setelah *Encode* (Kb)	Waktu proses *encode* (detik)	Rasio Kompresi (%)
1	Christine Smith CV.doc	23	7	3,88	69,5652
2	Christoper Morgan CV.doc	24	7	1,24	70,8333
3	Claudia Alves CV.doc	23	7	0,66	69,5652
4	Damien Le Grand CV.doc	29	8	0,71	72,4137
5	Dara Farhan.doc	28	8	0,66	71,4285

5.1.3 Pengujian Frequency Test

Pengujian ini dilakukan untuk mengetahui apakah *ciphertext* yang dihasilkan acak atau tidak.

Langkah-langkah untuk melakukan pengujian ini diantaranya:

1. Menghitung panjang bit dari *ciphertext*, serta menghitung bit-0, dan bit-1nya. Lalumenghitung total dari bit yang ada.
   n = panjang deret bit
   bit-0 = jumlah bit-0 x (-1)bit-1 = jumlah bit-1 x (1)
   |Sn| = total penjumlahan bit = bit-0 + bit-1
2. Menghitung nilai observasi uji statistik dengan rumus
3. $S_{obs} = |Sn| / \sqrt{n}$ (**1**)
4. Menghitung nilai-P (probabilitas) dengan rumus

Nilai-P $= \text{erfc}(\frac{Sobs}{\sqrt{2}})$ (**2**)

5. Dimana erfc merupakan uji statistik untuk mendapat nilai-P dengan melihat

kedekatan pecahan bit 1 dengan peluang kemunculan ½.

6. Bila nilai probabilitas kurang dari 0,01 maka *ciphertext* yang dihasilkan tidak acak, dan juga sebaliknya bila lebih besar dari 0,01, maka *ciphertext* yang dihasilkan bersifat acak.

Berikut merupakan hasil kalkulasi frequency test dengan menggunakan 50 berkas:

Tabel 8. Tabel Hasil Frequency Test

No	Nama file	N	Bit-0	Bit-1	|Sn|	*Sobs*	$\frac{Sobs(-)}{\sqrt{2}}$	Nilai-P	Keputusan
1	Christine Smith CV.doc	47000	23308	23692	384	1.7712	1.2524	0,0771	Acak
2	Christoper Morgan CV.doc	50712	25082	25630	548	2.4334	1.7206	0,0149	Acak
3	Claudia Alves CV.doc	50424	25106	25318	212	0.9440	0.6675	0,3506	Acak
4	Damien LeGrand CV.doc	62128	30747	31381	634	2.5435	1.7981	0,0113	Acak
5	Dara Farhan.doc	59792	29773	30019	246	1.006	0.7113	0,3153	Acak

5.1.4 Pengujian Pengaruh Kompresi PPM

Pengujian ini dilakukan untuk mengetahui pengaruh kompresi PPM pada model pengamanan berkas, apakah penggunaan kompresi PPM sebelum berkas dienkripsi memengaruhi waktu pemrosesan berkas.

Tabel 9. Hasil Proses *Encode* dan *Decode* Menggunakan Kompresi PPM dan Kriptografi RSA

No	Nama file	Waktu proses *encode* (detik)	Waktu proses *decode* (detik)
1	Christine Smith CV.doc	0.45	0,05
2	Christoper Morgan CV.doc	0.34	0,049
3	Claudia Alves CV.doc	0.31	0,05
4	Damien Le Grand CV.doc	0.36	0,051
5	Dara Farhan.doc	0.39	0,048

Dari pengujian menggunakan 5 berkas berformat .doc rata-rata waktu pemrosesan *encode* yang didapatkan bila menggunakan model pengamanan berkas yang menggunakan kompresi PPM dan kriptografi RSA adalah 0,0496 detik, sedangkan *decode*nya adalah ,0496 detik.

Tabel 10. Hasil Proses *Encode* Menggunakan Kriptografi RSA

No	Nama file	Waktu proses *encode* (detik)	Waktu proses *decode* (detik)
1	Christine Smith CV.doc	2,99	4,95
2	Christoper Morgan CV.doc	2,32	4,90
3	Claudia Alves CV.doc	4,04	4,98
4	Damien Le Grand CV.doc	2,63	3,62
5	Dara Farhan.doc	2,4	3,73

Dari pengujian menggunakan 5 berkas berformat .doc rata-rata waktu pemrosesan yang didapatkan bila menggunakan model pengamanan berkas yang tidak menggunakan kompresi PPM dan kriptografi RSA adalah 2,876 detik, sedangkan *decode*nya adalah 4,436 detik.

5.1.5 Pengujian Pengaruh Server

Pengujian ini dilakukan untuk mengetahui pengaruh server yang digunakan untuk meng-hosting berkas pada waktu pemrosesan berkas.

Tabel 11. Hasil Proses *Encode* Menggunakan Server Pertama

No	Nama file	Waktu proses *encode* (detik)
1	Christine Smith CV.doc	0,24
2	Christoper Morgan CV.doc	0,25
3	Claudia Alves CV.doc	0,23
4	Damien Le Grand CV.doc	0,25
5	Dara Farhan.doc	0,25

Percobaan pertama dilakukan pada *website* model pengamanan berkas yang menggunakan server dimana RAM yang diberikan sebesar 1 GB, SSD 20 GB, dan *bandwidth* tak terbatas. Dari pengujian yang menggunakan server ini didapat rata-rata waktu proses *encode* sebesar 0,244 detik.

Tabel 12. Hasil Proses *Encode* Menggunakan Server Kedua

No	Nama file	Waktu proses *encode* (detik)
1	Christine Smith CV.doc	0,26
2	Christoper Morgan CV.doc	0,28
3	Claudia Alves CV.doc	0,26
4	Damien Le Grand CV.doc	0,29
5	Dara Farhan.doc	0,29

Percobaan kedua dilakukan dengan menggunakan hosting bersama dimana dalam satu server terdapat lebih dari 1 *website* dengan RAM sebesar 1GB, SSD 25GB, dan *bandwidth* tak terbatas. Dari pengujian yang menggunakan server ini didapat rata-rata waktu proses *encode* sebesar 0,276 detik.

4.4.7 Pengujian Pengaruh Koneksi Internet

Pengujian ini dilakukan untuk mengetahui pengaruh koneksi internet yang digunakan dengan kecepatan waktu pemrosesan berkas.

Tabel 13. Hasil Proses *Encode* Pada Percobaan Pertama

No	Nama file	Waktu proses *encode* (detik)
1	Christine Smith CV.doc	0,26
2	Christoper Morgan CV.doc	0,28
3	Claudia Alves CV.doc	0,26
4	Damien Le Grand CV.doc	0,29
5	Dara Farhan.doc	0,29
6	David Eliot.doc	0,24
7	David Garcia.doc	0,15
8	David Richardson CV.doc	0,13
9	Diana Richardson CV.doc	0,26
10	Elita Fransiskus CV.doc	0,25
11	Elizabeth Holmes CV.doc	0,27
12	Erika Sanusi CV.doc	0,25
13	Fransiska Tita CV.doc	0,28

Percobaan pertama dilakukan dengan menggunakan koneksi internet dengan kecepatan mengunggah berkas sebesar 15,98 Mbps dan kecepatan mengunduh berkas sebesar 31,20 Mbps. Dari percobaan pertama didapat rata-rata waktu proses *encode* sebesar 0,2469 detik.

Tabel 14. Hasil Proses *Encode* Pada Percobaan Kedua

No	Nama file	Waktu proses *encode* (detik)
1	Christine Smith CV.doc	0,28
2	Christoper Morgan CV.doc	0,30
3	Claudia Alves CV.doc	0,28
4	Damien Le Grand CV.doc	0,28
5	Dara Farhan.doc	0,32
6	David Eliot.doc	0,45
7	David Garcia.doc	0,13
8	David Richardson CV.doc	0,13
9	Diana Richardson CV.doc	0,26
10	Elita Fransiskus CV.doc	0,32
11	Elizabeth Holmes CV.doc	0,29
12	Erika Sanusi CV.doc	0,21
13	Fransiska Tita CV.doc	0,23

Percobaan kedua dilakukan dengan menggunakan koneksi internet dengan kecepatan menggunggah berkas sebesar 2,98 Mbps dan kecepatan mengunduh berkas sebesar 8,72 Mbps. Dari percobaan kedua didapat rata-rata waktu proses *encode* sebesar 0,2676 detik.

6 Kesimpulan dan Saran

6.1 Kesimpulan

Berdasarkan hasil dari penelitian yang telah dilakukan, penulis dapat menarik kesimpulan sebagaiberikut:

1. Model pengamanan berkas menggunakan kriptografi asimetri RSA dan algoritma kompresi PPM mampu mengamankan berkas karena memiliki hasil *ciphertext* yang dihasilkan bersifat acak.
2. Waktu komputasi proses *encode* dan *decode* berbanding lurus dengan ukuran berkas yang digunakan. Semakin besar ukuran berkas yang digunakan, maka semakin banyak waktu komputasi yang digunakan.
3. Hasil komputasi dengan menggunakan perangkat dengan spesifikasi yang berbeda juga memengaruhi waktu komputasi. Spesifikasi perangkat yang memiliki RAM, danSSD atau HDD yang berbeda memiliki peran terhadap perbedaan hasil komputasi. Semakin besar spesifikasinya maka semakin cepat waktu komputasinya.
4. Kompresi PPM mampu mengurangi waktu pemrosesan berkas. Dari hasil pengujian model pengamanan berkas yang menggunakan kompresi PPM dan kriptografi RSA dan model pengamanan berkas yang hanya menggunakan RSA saja memiliki waktu pemrosesan yang berbeda dengan waktu pemrosesan yang tidak menggunakan kompresi PPM lebih besar dibandingkan dengan yang menggunakan PPM.
5. *Server* yang digunakan berpengaruh dalam waktu pemrosesan berkas. Dari hasil pengujian model pengamanan berkas yang menggunakan *server* dimana satu *server* terdapat lebih dari satu *website* dengan spesifikasi yang kurang lebih sama dengan *server* lain yang hanya terdapat satu *website*, waktu pemrosesan yang dihasilkan berbeda, dimana pada *server* yang hanya terdapat satu *website* memiliki waktu pemrosesan yang lebih cepat.
6. Koneksi internet yang digunakan berpengaruh dalam waktu pemrosesan berkas. Dari hasil pengujian model pengamanan berkas dengan menggunakan tiga koneksi internet yang berbeda waktu pemrosesan yang dihasilkan juga berbeda. Semakin besar kecepatan koneksi internet yang digunakan, maka semakin cepat waktu pemrosesannya.

6.2 Saran

Berdasarkan hasil dari penelitian yang telah dilakukan, penulis memiliki beberapa saran yangdapat diberikan untuk penelitian selanjutnya, diantaranya:

1. Penggunaan algoritma kompresi selain PPM untuk memberikan hasil kompresi yang berbeda yang memiliki performa lebih baik, misalnya bzip2, *Associative Coder of Buyanovsky* (ACB), *Dynamic Markov Compression*, dan lain sebagainya.
2. Model pengamanan berkas ini hanya bisa digunakan dengan mengakses *website*, diharapkan kedepannya dapat dikembangkan menjadi *mobile application*.
3. Model pengamanan berkas ini hanya menerima berkas dengan format string, diharapkan kedepannya dapat dikembangkan untuk menerima format gambar, *video*, atau *audio*.

Referensi

[1] Suhandinata, S., Rizal, R.A., Wijaya, D.O., Warren, P., dan Srinjiwi. 2019. *Analisis PerformaKriptografi Hybrid Algoritma Blowfish dan Algoritma RSA*. Medan: Universitas Prima Indonesia.

[2] Pahrizal, dan Pratama, D.. 2016. *Implementasi Algoritma RSA untuk Pengamanan Data Berbentuk Teks.* Bengkulu: Universitas Muhammadiyah Bengkulu.

[3] Saputro, T.H., Hidayanti, N., dan Ujianto, E.I.H. 2020. *Survei Tentang Algoritma Kriptografi Asimetris* . Yogyakarta: Universitas Teknologi Yogyakarta.

[4] Mayo, Francisco Dominguez, dkk. (2020). Web Information Systems and Technologies: 14th International Conference, WEBIST 2018, Seville, Spain, September 18–20, 2018, Revised Selected Papers. Jerman: Springer International Publishing.

[5] Munir, Rinaldi. (2019). *Kriptografi*. Bandung: Informatika Bandung