ANALISIS PERFORMA KRIPTOGRAFI HYBRID ALGORITMA BLOWFISH DAN ALGORITMA RSA

Article · December 2019

DOI: 10.33330/jurteksi.v6i1.395

5 authors, including:

Reyhan achmad Rizal
Universitas Prima Indonesia (UNPRI)
7 PUBLICATIONS **1** CITATION

SEE PROFILE

JURTEKSI (Jurnal Teknologi dan Sistem Informasi) ISSN 2407-1811 (Print)
Vol. VI No. 1, Des 2019, hlm. 1 - 10 ISSN 2550-0201 (Online)
DOI: https://doi.org/10.33330/jurteksi.v6i1.395
Available online at http://jurnal.stmikroyal.ac.id/index.php/jurteksi

ANALISIS PERFORMA KRIPTOGRAFI HYBRID ALGORITMA BLOWFISH DAN ALGORITMA RSA

Sebastian Suhandinata[1*],Reyhan Achmad Rizal[1], Dedy OngkyWijaya[1], Prabhu Warren[1], Srinjiwi[1]
[1]Prodi Teknik Informatika,Fakultas Teknologi dan Ilmu Komputer,Universitas Prima Indonesia, Indonesia
email: *abazcobuzet7@hotmail.com

Abstract: Computer data security relies on preventing data theft from irresponsible parties by using cryptography method. Some cryptography algorithms have good and poor performance in encrypting and decrypting data depending on the *key* types. Therefore the purpose of this research is to measure the performance of the hybrid algorithm, consisting a symmetric *key*Blowfish algorithm and an asymmetric *key* RSA algorithm, in encrypting and decrypting multiple types of data such as documents, photos, audios, and videos. The result is the performance of the hybrid algorithm is almost on par with Blowfish and provides a more secure data encryption and decryption by taking advantage of RSA algorithm. The average encryption performance of hybrid algorithm is 0.85s on document, 1.06s on photo, 3.38s on audio, and 15.56s on video. While the average decryption performance of hybrid algorithm is 1.01s on document, 1.38s on photo, 4.3s on audio, and 27.56s on video.

Keywords: *Hybrid* cryptography, Data security, Performance, Blowfish, RSA

Abstrak: Keamanan data komputer berhubungan dengan pencegahan dari pencurian data oleh pihak yang tidak bertanggung jawab, salah satu cara pengamanan data komputer yaitu dengan teknik kriptografi. Beberapa metode kriptografi memiliki performa yang baik dan buruk tergantung dengan tipe kuncinya. Maka dari itu, tujuan dari penelitian ini adalah mengukur tingkat kecepatan kriptografi *hybrid*, terdiri dari algoritma simetris *Blowfish* dan algoritma asimetris RSA, dengan beberapa tipe data seperti dokumen, foto, audio dan video. Hasil dari penelitian ini adalah algoritma *hybrid* memiliki performa yang tidak jauh berbeda dari algoritma *Blowfish* dan membuat proses enkripsi dan dekripsi data lebih aman dengan keunggulan dari algoritma RSA. Rata-rata kecepatan enkripsi algoritma *hybrid* untuk dokumen 0,85 detik, gambar 1,06 detik, audio 3,38 detik, dan video 15,56 detik. Sedangkan rata-rata kecepatan dekripsi algoritma *hybrid* untuk dokumen 1,01 detik, gambar 1,38 detik, audio 4,3 detik, dan video 27,56 detik.

Kata kunci: Kriptografi*hybrid*, Keamanan data, Performa, *Blowfish*, RSA

JURTEKSI (Jurnal Teknologi dan Sistem Informasi) ISSN 2407-1811 (Print)
Vol. VI No. 1, Des 2019, hlm. 1 - 10 ISSN 2550-0201 (Online)
DOI: https://doi.org/10.33330/jurteksi.v7i1.xxx
Available online at http://jurnal.stmikroyal.ac.id/index.php/jurteksi

PENDAHULUAN

Keamanan informasi merupa-kan salah satu masalah penting, seiring dengan perkembangan software dan pengguna internet. Keamanan komputer berhubungan dengan pencegahan dari pencurian data atau informasi dari orang yang tidak bertanggung jawab, baik itu mengakses dan memodifikasi informasi. Pengamanan komputer berfungsi untuk melindungi informasi agar tidak dapat diakses bagi orang yang tidak berhak. Banyak cara yang dapat digunakan dalam pengamanan komputer, salah satunya dengan menggunakan kriptografi [1].

Pengamanan data dengan metode kriptografi merupakan salah satu teknik yang digunakan untuk menyembunyikan pesan menjadi suatu bentuk lain sehingga tidak dapat dipahami dan diterapkan untuk mengamankan *file* seperti dokumen, gambar, audio, dan video. Metode kriptografi dapat diklasifikasikan menjadi 3 jenis yaitu kriptografi simetris, asimetris dan *hybrid* [2].

Kriptografi simetris menggu-nakan kunci yang sama disebut kunci privat, terdiri dari metode-metode diantaranya *Data Encryption Standard* (DES), *Rivest Cipher 4* (RC4), *Advanced Encryption Standard* (AES), *One Time Pad* (OTP), *Blowfish*, dan sebagainya, sedangkan kriptografi asimetris menggunakan kunci privat dan kunci publik dalam mengamankan data misalnya algoritma RSA (*Rivest Shamir Adleman*),*El Gamal*, *Elliptic Curve*, *Hill Cipher*, *Diffie-Hellman* dan sebagainya [3]. Kriptografi *hybrid* memanfaatkan dua tingkatan kunci, yaitu kunci rahasia (simetri) – yang disebut juga *session key*, untuk enkripsi data dan pasangan kunci privat-kunci publik untuk melindungi kunci simetri [2].

Algoritma *Blowfish* adalah algoritma simetris yang memiliki kecepatan proses enkripsi data dengan *rate* 26 *clock cycles per byte* dan hanya menggunakan operasi-operasi sederhana seperti penambahan, XOR, dan *lookup table* pada operan 32-bit. Tingkat keamanan bervariasi tergantung panjang kunci yang digunakan oleh algoritma *Blowfish*, bisa sampai sepanjang 448-bit [4].

Algoritma RSA menggunakan kunci publik dan kunci rahasia dalam mengenkripsi data, dimana kunci publik boleh diketahui oleh siapa saja sedangkan kunci rahasia hanya boleh diketahui oleh pihak tertentu guna mendekripsi data [5].

Megah Mulya membuat perbandingan kecepatan algoritma asimetris yang terdiri dari RSA, *El Gamal* dan *Elliptic Curve*, algoritma RSA memiliki waktu enkripsi dan dekripsi yang paling lama dibandingkan dengan algoritma *El Gamal* dan *Elliptic Curve* dan kecepatan ketiga algoritma menurun signifikan terhadap ukuran data [6].

Tri Andriyanto dan Crispina Pardede membandingkan dua algoritma simetris, *Blowfish* dan IDEA mengenai performa enkripsi dan dekripsi beberapa tipe data. Penelitian tersebut mendapatkan hasil bahwa tingkat kecepatan enkripsi dan dekripsi data algoritma IDEA lebih cepat dari algoritma *Blowfish* meskipun penggunaan memori dalam prosesnya relatif sama [7].

JURTEKSI (Jurnal Teknologi dan Sistem Informasi) ISSN 2407-1811 (Print)
Vol. VI No. 1, Des 2019, hlm. 1 - 10 ISSN 2550-0201 (Online)
DOI: https://doi.org/10.33330/jurteksi.v7i1.xxx
Available online at http://jurnal.stmikroyal.ac.id/index.php/jurteksi

Penelitian yang dikembangkan oleh Ritu Tripathi dan Sanjay Agrawal dalam membandingkan performa algoritma simetris DES, 3DES, AES, *Blowfish* dan algoritma asimetris RSA dan *Diffie-Hellman*, dapat disimpulkan bahwa penggunaan memori dan daya komputasi algoritma *Blowfish* yang rendah jadi proses enkripsi sangat cepat, tetapi algoritma RSA memiliki tingkat reliabilitas keamanan yang tinggi karena menggunakan pemfaktoran bilangan prima yang besar dalam membangkitkan kuncinya [8].

Kriptografi *hybrid* antara algoritma simetris dan algoritma asimetris dibutuhkan karena masalah keamanan kunci simetris, tetapi proses enkripsi dan dekripsi pesan besar atau kecil lebih cepat. Kriptografi asimetris mempunyai tingkat keamanan kunci lebih tinggi, tetapi kecepatan enkripsi maupun dekripsi yang dilakukan kriptografi asimetris lebih lama. Sehingga kedua algoritma kriptografi simetris dan asimetris digabung untuk memberikan perlindungan untuk kunci simetris serta meningkatkan kecepatan kriptografi kunci asimetris [9].

Menurut beberapa penelitian yang telah dilakukan, dikarenakan waktu enkripsi dan dekripsi RSA lebih lama dibanding dengan *Blowfish*, maka kami akan menganalisis performa kriptografi *hybrid*, algoritma *Blowfish* dan algoritma RSA untuk mengetahui tingkat kecepatan enkripsi dan dekripsi data jika algoritma RSA dikombinasikan dengan algoritma *Blowfish*.

Algoritma Kriptografi

Kriptografi berasal dari Bahasa Yunani, yaitu *crypto* yang berarti rahasia, dan *graphia* yang berarti tulisan. Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan yang akan dikirim dari satu tempat ke tempat lain.

Algoritma kriptografi adalah urutan langkah-langkah logis untuk untuk merahasiakan informasi dari orang-orang yang tidak berhak.

Menurut Amita Pandey, dasar konsep kriptografi terdiri dari:

1. *Plain text*, adalah pesan asli yang ingin dikirim.
2. *Cipher text*, adalah pesan yang tidak dapat dimengerti oleh siapapun yang awalny amerupakan *plain text*.
3. *Encryption*,mengkonversi *plain text* menjadi *cipher text*, membutuhkan 2 proses, algoritma enkripsi dan kunci.
4. *Decryption*, mengkonversi *cipher text* menjadi *plain text*, membutuhkan 2 proses, algoritma dekripsi dan kunci.
5. *Key*, merupakan kombinasi dari angka atau huruf atau symbol spesial yang digunakan dalam enkripsi dan dekripsi dan memiliki peran penting dalam kriptografi karena algoritma bergantung kepadanya [10].

Algoritma kriptografi dapat diklasifikasikan menjadi 3 berdasarkan jenis kuncinya, kriptografi kunci simetris, kriptografi kunci asimetris dan kriptografi *hybrid*[2].

Kriptografi Kunci Simetris

Dalam proses enkripsi dan dekripsi kriptografi kunci simetris, kunci yang sama digunakan sehingga kerahasiaan kunci dapat dijamin dan tersembunyi. Algoritma simetris memiliki kelebihan mengonsumsi

JURTEKSI (Jurnal Teknologi dan Sistem Informasi) ISSN 2407-1811 (Print)
Vol. VI No. 1, Des 2019, hlm. 1 - 10 ISSN 2550-0201 (Online)
DOI: https://doi.org/10.33330/jurteksi.v7i1.xxx
Available online at http://jurnal.stmikroyal.ac.id/index.php/jurteksi

daya komputasi komputer yang kecil dan bekerja secara cepat pada pengenkripsian data.

Kriptografi kunci simetris memiliki dua mode sebagai block cipher atau sebagai stream cipher. Dalam block cipher semua data dibagi menjadi beberapa blok dan kunci akan diberikan tergantung dari panjang blok, sedangkan dalam stream cipher data dibagi menjadi -bits kecil yang diacak lalu dienkripsi. Beberapa contoh dari algoritma simetris adalah algoritma AES, algoritma DES, algoritma *Blowfish*, algoritma Triple DES dan algoritma RC4 [11].

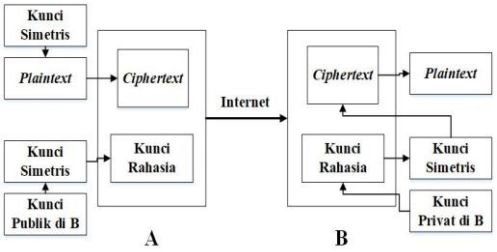
Kriptografi Asimetris

Kriptografi kunci asimetris memiliki kunci yang berbeda dalam proses enkripsi dan dekripsi, dikenal dengan nekripsi kunci publik. Salah satu kunci disebarkan (kunci publik) dan yang lainnya dirahasiakan (kunci privat). Apabila kunci enkripsi bersifat publik maka sistem membuka koneksi privat dari publik untuk membuka kunci pengguna. Apabila kunci dekripsi bersifat publik maka sistem berperan sebagai pemverifikasi data yang dikunci oleh pemilik kunci privat.

Metode algoritma asimetris penting karena dapat digunakan untuk membagikan kunci enkripsi atau data lainnya secara aman meskipun kedua belah pihak tidak memiliki kesempatan untuk menyetujui kunci privat. Kunci yang digunakan pada algoritma asimetris umumnya panjang dimana meningkatkan keamanan data yang dikirim, setidaknya berukuran 3000-bit atau lebih untuk mencapai tingkat keamanan algoritma simetris 128-bit. Contoh dari algoritma asimetris adalah algoritma RSA dan *Diffie-Hellman* [11].

Kriptografi *Hybrid*

Kriptografi *hybrid* merupakan protokol yang memanfaatkan beberapa sandi dari algoritma berbeda secara bersamaan dengan keunggulan tiap algoritma tersebut. Salah satu cara yang sering diterapkan adalah membangkitkan kunci simetris dan mengenkripsi kunci ini dengan kunci asimetris dari kunci publik penerima. Data dienkripsi dengan kunci simetris dan kunci rahasia ini dikirim ke penerima kemudian penerima mendekripsi kunci rahasia terlebih dahulu menggunakan kunci privat miliknya, lalu mendekripsi data dengan kunci yang telah didekripsi tersebut [12].



Gambar 1. Skema Kriptografi *Hybrid*[12]

Dalam gambar 1 menunjukkan diagram sistem kriptografi *hybrid* dimana menggabungkan keuntungan dari kecepatan enkripsi algoritma simetris dan kemampuan algoritma asimetris mengamankan proses pertukaran kunci.

Algoritma *Blowfish*

Bruce Schneier merancang algoritma *Blowfish* pada tahun 1993 sebagai alternatif enkripsi data yang cepat dan terbuka (*open-source*). Sejak dicetus, algoritma ini telah dianalisa terus menerus, dan perlahan

JURTEKSI (Jurnal Teknologi dan Sistem Informasi) ISSN 2407-1811 (Print)
Vol. VI No. 1, Des 2019, hlm. 1 - 10 ISSN 2550-0201 (Online)
DOI: https://doi.org/10.33330/jurteksi.v7i1.xxx
Available online at http://jurnal.stmikroyal.ac.id/index.php/jurteksi

diakui sebagai algoritma enkripsi yang handal. Banyak kelebihan dari algoritma *Blowfish* seperti kompati-bilitas dan efisiensi dalam penerapannya dan tidak ada lisensi yang diperlukan. Dasar operasi *Blowfish* mencakup *lookup table*, penambahan dan XOR. *Lookup table*terdiri dari empat *S-boxes* dan sebuah *P-array*. *Blowfish* adalah blok cipher 64-bit yang disebut menggantikan algoritma DES, dengan operasi algoritma yang cepat dan mampu mengenkripsi data pada mikroprosesor berukuran 32-bit[13].

Blowfish adalah algoritma simetris 64-bit yang menggunakan panjang kunci bervariasi dari 32-bit sampai 448-bit (14 bytes). *Blowfish* dirancang untuk mengenkripsi *plain text* 64-bit ke *cipher text* 64-bit secara efisien dan aman. Operasi yang digunakan dalam prosesnya berupa *lookup table*, modulus, penambahan dan XOR untuk meminimalisir waktu yang dibutuhkan dalam mengenkripsi dan mendekripsi data pada prosesor 32-bit[13].

Blowfish menggunakan *sub-key* besar yang harus dihitung sebelum enkripsi dan dekripsi data. Algoritma *Blowfish* menerapkan jaringan Feistel yang terdiri dari 16 putaran. Input adalah elemen 64-bit, X untuk alur algoritma enkripsi dengan metode *Blowfish* dijelaskan sebagai berikut:

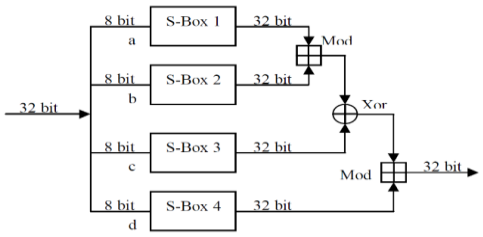
1. Inisialisasi *P-array* diikuti dengan empat *S-boxes* dengan string terdiri dari Pi hexadecimal.
2. P1 di-XOR dengan kunci 32-bit pertama, P2 di-XOR dengan kunci 32-bit kedua, proses diulang sampai semua *P-array* telah selesai di-XOR.
3. Algoritma lalu digunakan untuk mengenkripsi string kosong yang diisi dengan *sub-key* pada tahap 1 dan 2.
4. P1 dan P2 diganti dengan output tahap 3.
5. Enkripsi output tahap 3 dengan algoritma *Blowfish* menggunakan *sub-key* yang telah dimodifikasi.
6. Output dari tahap 5 digunakan untuk menggantikan P3 dan P4.
7. Proses akan terus diulang sampai semua *P-array* telah tergantikan, dilanjut dengan semua 4 *S-boxes*, dengan output yang terus berubah.

Enkripsi data dimulai dengan blok elemen *plain text* 64-bit diubah ke *cipher text* 64-bit. Pertama, segmen *plain text* dibagi menjadi 2 bagian sama yang menjadi dasar daripada *Blowfish*. Langkah selanjutnya adalah implementasi operasi XOR yang dilakukan antara segmen blok 32-bit pertama (L) dan *P-array* pertama. Data berukuran 32-bit didapatkan dari langkah kedua dipindahkan ke fungsi F dimana ditukar ke dalam segmen data blok 32-bit, yang kemudian di-XOR dengan segmen blok 32-bit kedua (R) dari 64-bit*plain text* tadi. Setelah proses XOR selesai, segmen L dan R digunakan untuk iterasi selanjutnya dari algoritma *Blowfish* tersebut. Proses dekripsi data sama dengan enkripsi, hanya *P-array*digunakan dalam urutan terbalik [12].

Fungsi F dari *Blowfish* mungkin adalah hal yang paling rumit dari algoritma ini karena hanya bagian ini yang memanfaatkan *S-boxes*. Fungsi F menerima data 32-bit dan membaginya menjadi 4 bagian 8-bit. Tiap bagian tersebut diubah menjadi 32-bit menggunakan *S-box* yang berhubungan dengan bagiannya.

JURTEKSI (Jurnal Teknologi dan Sistem Informasi) ISSN 2407-1811 (Print)
Vol. VI No. 1, Des 2019, hlm. 1 - 10 ISSN 2550-0201 (Online)
DOI: https://doi.org/10.33330/jurteksi.v7i1.xxx
Available online at http://jurnal.stmikroyal.ac.id/index.php/jurteksi

Kemudian data 32-bit yang diterima di-XOR ataupun digabung untuk menghasilkan data akhir 32-bit untuk permutasi *Blowfish*[12].



Gambar 2. Fungsi F [12]

Algoritma RSA

Ditemukan oleh Ron Rivest, Adi Shamir dan Len Adleman dari MIT (*Massachusetts Institute of Technology*) pada tahun 1977, algoritma RSA merupakan penemuan besar dalam kriptografi kunci publik dan masih populer digunakan sampai saat ini, karena kunci-kunci yang panjang dan penerapannya makin disempurnakan [14].

Algoritma RSA merupakan blok cipher dimana semua informasi dipetakan ke sebuah integer. Algoritma RSA terdiri dari kunci publik dan kunci privat dimana kunci publik dapat diketahui oleh semua orang sedangkan kunci privat hanya diketahui oleh pemilik data. Proses enkripsi menggunakan kunci publik dan proses dekripsi menggunakan kunci privat pemilik data [15].

Algoritma pembangkitan kunci RSA:

1. Tentukan p dan q bernilai dua bilangan prima besar, acak dan dirahasiakan, $p \neq q$, p dan q memiliki ukuran yang sama.
2. Hitung $n = p \times q$, dan hitung $\phi(n) = (p - 1) \times (q - 1)$, bilangan integer n disebut (RSA) modulus.
3. Tentukan e bilangan prima acak yang memiliki syarat: $1 < e < 1(n)$, GCD(e, i(n)) = 1, disebut e relatif prima terhadap i(n), bilangan integer n disebut (RSA) *enciphering component*, sehingga menghasilkan $Dd\ (Ee(m)) = Ee(Dd(c)) \equiv md \bmod n$.

METODE

Dalam bagian ini akan di jelaskan tahap proses dari dua algoritma yaitu *Blowfish* dan RSA dalam enkripsi dan dekripsi dari suatu *file*, serta proses enkripsi dan dekripsi jika kedua algoritma tersebut di gabungkan.

Algoritma*Blowfish*

Blowfish merupakan algoritma yang menerapkan teknik kunci yang berukuran sembarang. Ukuran kunci yang dapat diterima oleh *Blowfish* adalah antara 32-bit hingga 448-bit, dengan ukuran kunci sebesar 128-bit. Berikut adalah proses enkripsi algoritma *Blowfish*:

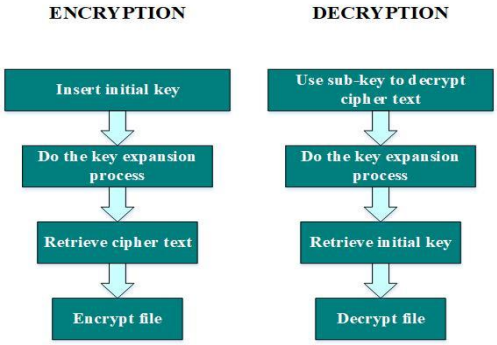
1. Melakukan proses *keyexpansion* dengan memasukan kunci berukuran hingga 488-bit.
2. Membangkitkan kunci dari proses *keyexpansion* menjadi sebuah *sub-key* dengan ukuran keseluruhan 4168-bit.
3. Proses mengenkripsikan data dengan menggunakan sistem operasi XOR.

Berikut adalah proses dekripsi algoritma *Blowfish*:

1. Menggunakan *sub-key* secara terbalik untuk mendekripsikan *ciphertext*.
2. Melakukan proses *keyexpansion* untuk mengembalikan kunci
3. Awal.Dekripsi dari hasil proses

JURTEKSI (Jurnal Teknologi dan Sistem Informasi) ISSN 2407-1811 (Print)
Vol. VI No. 1, Des 2019, hlm. 1 - 10 ISSN 2550-0201 (Online)
DOI: https://doi.org/10.33330/jurteksi.v7i1.xxx
Available online at http://jurnal.stmikroyal.ac.id/index.php/jurteksi

mendekripsian *cipher text* dengan *sub-key*, menggunakan kunci awal.



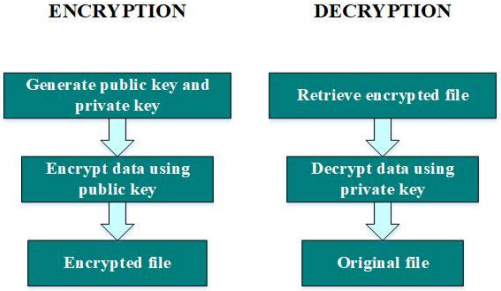
Gambar 3. Proses Enkripsi dan Dekripsi Algoritma *Blow fish*[7]

AlgoritmaRSA

RSA merupakan algoritma yang menggunakan bilangan faktorisasi terbesar dan metode distribusi yang sulit di pecahkan. Kunci RSA pada umumnya berukuran 1024-2048 bit, namun pada penelitian ini kami menggunakan kunci dengan ukuran 1024-bit.

Berikut adalah proses enkripsi dan dekripsi pada algoritma RSA:

1. Bangkitkan kunci publik untuk enkripsi dan kunci privat untuk dekripsi.
2. Mengenkripsi data dengan kunci publik yang dapat diakses oleh siapapun.
3. Mendekripsi data dengan kunci privat yang hanya dapat diakses oleh penerima data.



Gambar 4. Proses Enkripsi dan Dekripsi AlgoritmaRSA [14]

Algoritma*Hybrid*

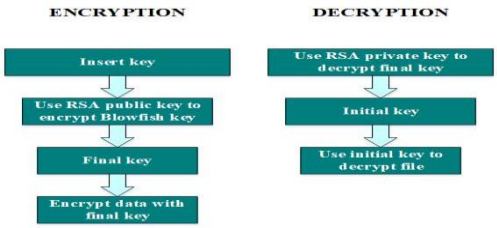
Selain melakukan penelitian dari kedua algoritma tersebut dalam enkripsi dan dekripsi *file*, peneliti juga melakukan penelitian dengan menggabungkan kedua algoritma simetris *Blowfish* dan algoritm aasimetris RSA tersebut sehingga menghasilkan sebuah kriptografi *hybrid*.

Berikut adalah proses enkripsi pada algoritma *hybrid*:

1. Membangkitkan kunci simetris pada algoritma simetris *Blowfish*.
2. Menggunakan kunci privat RSA untuk melakukan proses enkripsi kunci simetris *Blowfish*, sehingga menghasilkan kunci akhir.
3. Gunakan kunci akhir untuk melakukan proses enkripsi *file*.

Berikut adalah proses dekripsi pada algoritma *hybrid*:

1. Menggunakan kunci publik untuk mendekripsikan kunci akhir sehingga kembali menjadi kunci awal.
2. Gunakan kunci awal yang telah di dekripsi untuk mendekripsi *file*.



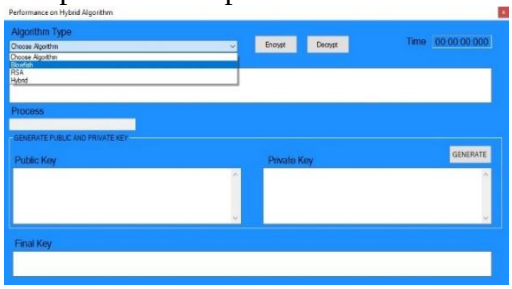
Gambar 5. Proses Enkripsi dan Dekripsi Algoritma *Hybrid*[12]

HASIL DAN PEMBAHASAN

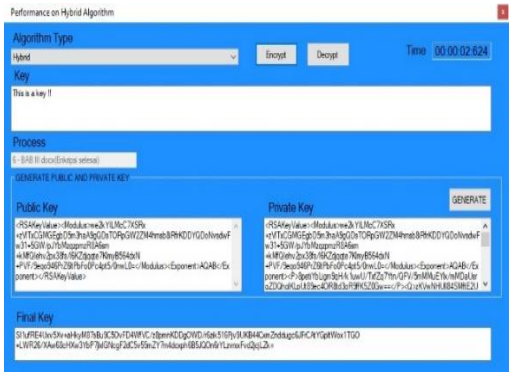
Sistem yang dirancang untuk menunjang penelitian perbandingan performa kriptografi *hybrid* algoritma *Blowfish* dan RSA menggunakan bahasa pemrograman Visual Basic dan

JURTEKSI (Jurnal Teknologi dan Sistem Informasi) ISSN 2407-1811 (Print)
Vol. VI No. 1, Des 2019, hlm. 1 - 10 ISSN 2550-0201 (Online)
DOI: https://doi.org/10.33330/jurteksi.v7i1.xxx
Available online at http://jurnal.stmikroyal.ac.id/index.php/jurteksi

pengujian dilakukan dalam sistem operasi Windows 10. Data yang diambil dari aplikasi adalah pengamatan waktu proses enkripsi dan dekripsi masing-masing algoritma. Spesifikasi perangkat keras komputer dapat mempengaruhi hasil yang didapatkan dari aplikasi.



Gambar 6. Interface Aplikasi



Gambar 7. Proses Enkripsi dan Dekripsi padaAplikasi

Dalam pengujian sistem, ada beberapa langkah yang diambil guna memudahkan dalam analisa performa masing-masing kriptografi. Beberapa langkah tersebut adalah:

1. Menggunakan 3 sampel data per jenis *file* dengan ukuran berbeda.
2. Mengukur jumlah rata-rata hasil waktu enkripsi dan dekripsi masing-masing algoritma.

Hasil waktu enkripsi (E) dan dekripsi (D) yang didapat dari penelitian ini dapat dilihat pada tabel berikut:

Tabel 1. Hasil Waktu Enkripsi dan Dekripsi File *.docx*

Algo\ File	Waktu Proses (s)		
	54 kb	**142 kb**	**492 kb**
Blowfish (E)	0.23	0.54	1.26
Hybrid (E)	0.3	0.84	1.41
RSA (E)	10.96	18.14	64.82
Blowfish (D)	0.39	0.62	1.47
Hybrid (D)	0.51	0.99	1.55
RSA (D)	58.22	117.14	363.9

Tabel 2. Hasil Waktu Enkripsi dan Dekripsi File *.jpeg*

Algo\ File	Waktu Proses (s)		
	256 kb	**474 kb**	**1437 kb**
Blowfish (E)	0.49	0.66	1.49
Hybrid (E)	0.57	1.01	1.61
RSA (E)	41.44	94.05	231.82
Blowfish (D)	0.83	1.02	1.62
Hybrid (D)	0.97	1.25	1.91
RSA (D)	641.78	852.88	1869.19

Tabel 3. Hasil Waktu Enkripsi dan Dekripsi File *.mp3*

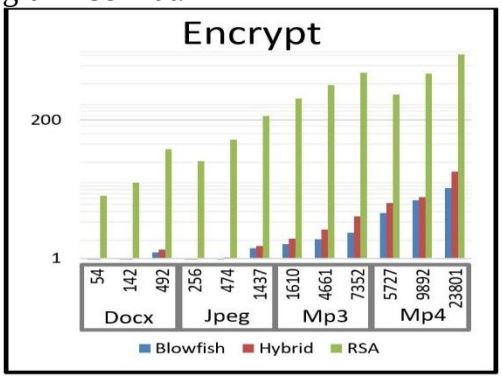
Algo\ File	Waktu Proses (s)		
	1610 kb	**4661 kb**	**7352 kb**
Blowfish (E)	1.73	2.09	2.68
Hybrid (E)	2.14	3.02	4.99
RSA (E)	452.6	764.44	1233.27
Blowfish (D)	2.14	2.82	3.65
Hybrid (D)	3.1	4.18	5.61
RSA (D)	1625.41	2749.55	6923.01

Tabel 4. Hasil Waktu Enkripsi dan Dekripsi File *.mp4*

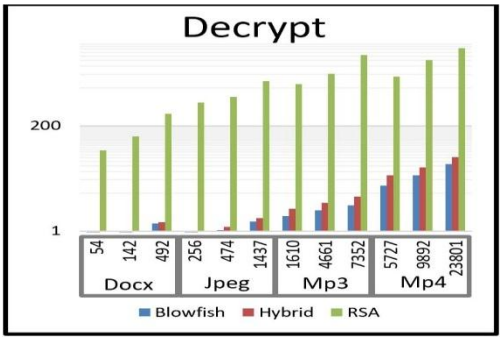
Algo\ File	Waktu Proses		
	5727 kb	**9892 kb**	**23801 kb**
Blowfish (E)	5.69	9.24	14.82
Hybrid (E)	8.31	10.46	27.91
RSA (E)	533.96	1192.86	2462.82
Blowfish (D)	9.9	16.72	29.07
Hybrid (D)	16.65	24.41	41.63
RSA (D)	2347.22	5394.46	9963.48

JURTEKSI (Jurnal Teknologi dan Sistem Informasi) ISSN 2407-1811 (Print)
Vol. VI No. 1, Des 2019, hlm. 1 - 10 ISSN 2550-0201 (Online)
DOI: https://doi.org/10.33330/jurteksi.v7i1.xxx
Available online at http://jurnal.stmikroyal.ac.id/index.php/jurteksi

Dari hasil tabel berikut, proses enkripsi dan dekripsi RSA membutuhkan waktu yang sangat lama dibanding *Blowfish* maupun *hybrid*, tapi waktu enkripsi dan dekripsi *hybrid* hanya berbeda beberapa detik dengan *Blowfish*. Perbandingan hasil dari masing-masing algoritma dapat dilihat pada grafik berikut.



Gambar 8. Grafik Enkripsi



Gambar 9. Grafik Dekripsi

SIMPULAN

Dari analisis yang dilakukan terhadap algoritma hybrid dari algoritma simetris Blowfish dan algoritma asimetris RSA maka penulis dapat mengambil kesimpulan bahwa algoritma hybrid memiliki kecepatan proses enkripsi dan dekripsi yang relatif cepat namun tingkat kecepatan tersebut tidak secepat algoritma pembentuknya.

DAFTAR PUSTAKA

[1] R. Hardi and Z. , "Implementasi Sistem Keamanan Komputer Menggunakan Sistem Terintegrasi Client Server Metode Service Oriented Architecture (SOA)," *Jurnal Teknologi Terpadu Vol. 6 No. 1*, pp. 1-7, 2018.

[2] D. Ariyus, Pengantar Ilmu Kriptografi: Teori, Analisis & Implementasi, Yogyakarta: C.V Andi Offset, 2008.

[3] C. A. Sari, E. H. Rachmawanto, D. W. Utomo and R. R. Sani, "Penyembunyian Data Untuk Seluruh Ekstensi File Menggunakan Kriptografi Vernam Cipher Dan Bit Shiffting," *Journal of Applied Intelligent System Vol. 1 No. 3*, pp. 179-190, 2016.

[4] S. Sitinjak, Y. Fauziah and J. , "Aplikasi Kriptografi File Menggunakan Algoritma Blowfish," *Seminar Nasional Informatika 2010*, pp. 78-86, 2010.

[5] R. Y. Rifai, Y. Chirstyono and I. Santoso, "Implementasi Algoritma Kriptografi Rivest Code 4, Rivest Shamir Adleman Dan Metode Steganografi Untuk Pengamanan Pesan Rahasia Pada Berkas Teks Digital," *Transient Vol. 5 No. 1*, pp. 86-91, 2016.

[6] B. Prasetiyo, R. Gernowo and B. Noranita, "Kombinasi Steganografi Berbasis Bit Matching Dan Kriptografi DES Untuk Pengamanan Data,"

Scientific Journal of Informatics Vol. 1 No. 1, pp. 79-94, 2014

M. Mulya, "Perbandingan Kecepatan Algoritma Kriptografi Asimetri," *Journal of Research in Computer Science and Applications Vol. 1 No. 2,* pp. 7-12, 2013.

[7] T. Andriyanto and D. L. C. Pardede, "Studi Dan Perbandingan Algoritma IDEA Dan Algoritma Blowfish," *Seminar Ilmiah Nasional Komputer dan Sistem Intelijen 2008,* pp. 182-189, 2008.

[8] R. Tripathi and S. Agrawal, "Comparative Study Of Symmetric And Asymmetric Cryptography Techniques," *International Journal of Advance Foundation and Research in Computer Vol. 1 Issue 6,* pp. 68-76, 2014.

[9] S. Singh and A. Singh, "An Information Security Technique Using DES-RSA Hybrid And LSB," *International Journal of Emerging Technologies in Computational and Applied Sciences,* pp. 187-192, 2014.

[10] A. Pandey and P. Bonde, "Performance Evaluation Of Various Cryptography Algorithms Along With LSB Substitution Technique," *International Journal of Engineering Research & Technology Vol. 2 Issue 6,* pp. 866-871, 2013.

[11] A. L. Jeeva, V. Palanisamy and K. Kanagaram, "Comparative Analysis Of Performance Efficiency And Security Measures Of Some Encryption Algorithms," *International Journal of Engineering Research and Applications Vol. 2 Issue 3,* pp. 3033-3037, 2012.

[12] R. P. S and V. Paul, "A Hybrid Crypto System Based On A New Cicle-Symmetric Key Algorithm And RSA With CRT Asymmetric Key Algorithm For E-commerce Application," *International Journal of Computer Applications,* pp. 14-18, 2011.

[13] A. Alabaichi, F. Ahmad and R. Mahmod, "Security Analysis Of Blowfish Algorithm," *International Conference on Informatics & Applications,* pp. 12-18, 2013.

[14] Z. Arifin, "Studi Kasus Penggunaan Algoritma RSA Sebagai Algoritma Kriptografi Yang Aman," *Jurnal Informatika Mulawarman Vol. 4 No. 3,* pp. 7-14, 2009.

[15] P. Kalpana and S. Singaraju, "Data Security in Cloud Computing Using RSA Algorithm," *International Journal of Research in Computer & Communication Technology Vol. 1 Issue 4,* pp. 143-146, 2012.