Kriptografi untuk Keamanan Pengiriman Email Menggunakan Blowfish dan Rivest Shamir Adleman (RSA)

Muhammad Iqbal Zulfikar*, Gunawan Abdillah, Agus Komarudin
Jurusan Informatika, Fakultas Sains dan Informatika
Universitas Jenderal Achmad Yani
Cimahi, Indonesia
*iqbaloid@gmail.com

Abstrak*—Email merupakan media komunikasi jarak jauh melalui internet yang saat ini sering digunakan untuk kebutuhan pertukaran pesan. Tetapi penggunaan email memiliki permasalahan dalam aspek keamanan. Salah satu upaya meningkatkan keamanan pada data dan informasi adalah penerapan teknik dan metode Kriptografi. Kriptografi adalah ilmu untuk mengurangi resiko ancaman keamanan dengan melakukan proses enkripsi dan dekripsi pada data dan informasi. Penerapan satu teknik kriptografi memiliki tingkat resiko kebocoran data yang lebih tinggi dari keamanan yang menerapkan lebih dari satu teknik kriptografi. Oleh karena itu, diperlukan penerapan dua teknik kriptografi yang mampu mengamankan pesan email sebelum proses pengiriman. Kriptografi Blowfish yang memiliki tingkat kecepatan dekripsi tinggi dan penggunaan memory yang kecil digunakan untuk mengamankan isi email. Kriptografi RSA yang populer dengan penggunaan kunci publiknya digunakan untuk mengamankan salah satu komponen dari Blowfish yaitu kunci simetris. Penelitian ini bertujuan membuat sistem yang dapat mengamankan pesan email dan kunci simetris sebelum dilakukan proses pengiriman dengan mengkombinasikan kriptografi Blowfish dan RSA diharapkan dapat meningkatkan keamanan secara lebih. Tahap uji serangan brute force yang dilakukan sebanyak tiga kali menghasilkan plaintext yang tidak utuh dan tahap uji ukuran data setelah dienkripsi membengkak sebesar 0,09KB dari hal tersebut maka kombinasi teknik kriptografi yang digunakan aman dan efisien.

***Kata kunci*—**Kriptografi, Enkripsi, Dekripsi, RSA, Blowfish, Keamanan Email.

I. PENDAHULUAN

Perkembangan teknologi mempermudah keterbukaan terhadap akses data dan informasi, salah satunya adalah media komunikasi sebagai perantara distribusi dan penyampaian pesan jarak jauh. Salah satu implementasinya adalah *email* yang saat ini sering digunakan sebagai media komunikasi jarak jauh [1]. Tetapi akhir-akhir ini sering terjadi permasalahan kasus cyber, di antaranya adalah pemalsuan *email* (*spoofing*), disalahgunakan untuk menyebarkan *spam*, digunakan para peretas sebagai media menyebarkan *malware*, pembobolan *email* (*man in the middle attack*), dan beberapa kasus lainnya [2]. Maka dari itu pada saat melakukan pengiriman atau penerimaan *email* diperlukan perhatian terhadap aspek kerahasiaan (*confidentiality*), integritas (*integrity*), memastikan (*authentication*), dan penyangkalan (*non-repudiation*) [3].

Sudah seharusnya informasi yang disampaikan melalui perantara *email* terjamin keamanannya pada saat didistribusikan dan disimpan di media penyimpanan baik publik maupun privat. Terlebih lagi isi dari *email* tersebut bersifat penting dan rahasia seperti hasil pemilihan umum atau perintah kerja kepolisian. Beberapa *email service* seperti Google dan Yahoo pernah mengalami kasus keamanan, sehingga protokol yang digunakan saat itu diubah dari HTTP menjadi HTTPs, tetapi proses memuat data menjadi lebih lama dikarenakan HTTPs memerlukan proses lebih seperti autentikasi dan semacamnya untuk perizinan koneksi [2]. Dengan kata lain *email service* lebih memperhatikan keamanan akun pengguna untuk mempertingkat keamanan dari *email*. Akan tetapi masih terdapat celah penyadapan terhadap isi dari *email* itu sendiri, peretas mungkin saja akan meretas *mail server* dan melakukan penyadapan pada *email* yang menjadi sasaran.

Dari permasalahan yang ada maka diperlukannya langkah antisipasi berupa meningkatkan keamanan pada data dan informasi. Salah satunya adalah dengan menerapkan metode kriptografi untuk menjaga keaslian dan kerahasiaan informasi dalam bentuk *email*.

Kriptografi adalah studi yang bertujuan untuk mengamankan dan merahasiakan dengan melakukan proses enkripsi dan dekripsi pada data yang akan diamankan [4]. Enkripsi merupakan proses pengubahan data menjadi bentuk sandi yang tidak dipahami dan dibaca, sedangkan dekripsi merupakan proses pengembalian data dalam bentuk sandi ke dalam bentuk semula yang dapat dipahami dan memiliki makna [5]. Dalam kriptografi terdapat beberapa teknik penyandian data yaitu simetris dan asimetris. Kriptografi simetris menggunakan kunci yang sama (kunci simetris) untuk melakukan proses enkripsi dan dekripsi. Kriptografi asimetris menggunakan kunci yang berbeda untuk proses enkripsi (menggunakan kunci publik) dan dekripsi (menggunakan kunci privat) [6][7]. Kedua teknik tersebut memiliki keunggulan dan kekurangan masing-masing yang diukur berdasarkan durasi enkripsi, durasi dekripsi, tingkat perubahan yang dihasilkan, entropi, dan jumlah bit yang dibutuhkan untuk pengkodean secara optimal [8]. Kriptografi asimetris dapat dikatakan lebih aman dari simetris, dikarenakan menggunakan dua kunci yang berbeda dan ukuran kunci yang

ditawarkan relatif lebih panjang bisa mencapai 1024 bit. Disisi lain, untuk kecepatan proses enkripsi dan dekripsi lebih unggul kriptografi simetris dan penggunaan memorinya pun relatif lebih rendah dengan nilai terendah 9,38 KB [9]. Salah satu metode kriptografi simetris yaitu Blowfish. Blowfish termasuk metode yang menerapkan *cipherblock* pada proses enkripsi data dalam 8 byte (64 bit) blok dengan ukuran kunci 32 bit sampai dengan 448 bit dan cocok digunakan untuk menyandikan *file* berukuran besar, tetapi terbilang kurang aman karena kunci yang digunakan hanya satu [10]. Dan salah satu metode kriptografi asimetris yaitu RSA. RSA adalah Kriptosistem kunci publik pertama dan digunakan secara luas untuk mengamankan transmisi data dan cocok digunakan untuk mengenkripsi data berukuran kecil, karena proses enkripsi dan dekripsi RSA terbilang lama [11].

Pada penelitian sebelumnya menerapkan kriptografi simetris untuk enkripsi pesan melalui *command prompt* dengan data biner string dan 8 bit dengan hasil bahwa kriptografi jenis simetris memberikan hasil *ciphertext* yang rumit dan sulit diserang menggunakan serangan *Ciphertext-only attack* [12], mengamankan *email* menggunakan Blowfish dengan waktu enkripsi dan dekripsi terbilang cepat yaitu dengan input sebesar 69 KB lama proses enkripsi sebesar 85 ms dan proses dekripsi sebesar 51 ms [13], dan mengamankan email sebelum dilakukan proses pengiriman dengan mengkombinasikan Kriptografi Hill Cipher dan RSA dengan hasil proses enkripsi RSA lebih cepat dikarenakan data yang dienkripsi berukuran 9 byte dari matriks yang telah ditentukan [14].

Penelitian ini bertujuan membangun sistem yang dapat mengamankan *email* sebelum dilakukannya proses pengiriman, dengan cara dua tahap. Pertama melakukan enkripsi pada *email* tersebut menggunakan metode kriptografi Blowfish. Kedua mengenkripsi kunci simetris menggunakan kriptografi RSA. Dengan mengkombinasikan dua metode kriptografi yang berbeda menjadi jalan keluar untuk menjadikan pengamanan pada informasi menjadi lebih kuat.

II. TINJAUAN PUSTAKA

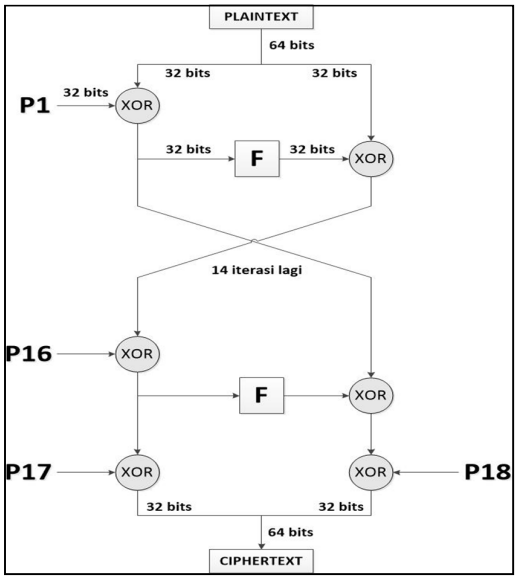
A. Blowfish

Blowfish merupakan kriptografi simetris dengan ukuran kunci beragam dengan rentang antara 32 bit sampai 448 bit yang menerapkan *cipherblock* 64 bit. Blowfish terbilang memiliki proses yang cepat dengan 26 *clock cycle* per byte dan dapat kompak dengan berjalan pada memori kurang dari 5KB. Blowfish dikenal sebagai kriptografi yang memiliki tingkat kecepatan dekripsi yang tinggi, 33% lebih cepat dari proses enkripsi [15]. Blowfish terdiri dari dua bagian proses utama yaitu ekspansi kunci yang berfungsi untuk mengubah kunci ke dalam subkunci dalam bentuk *array* dengan toal 4168 byte dan enkripsi data yang terdiri dari iterasi fungsi sederhana (Feistel Network) sebanyak 16 kali putaran [16].

Tahapan algoritma enkripsi dengan kriptografi Blowfish dijelaskan sebagai berikut:

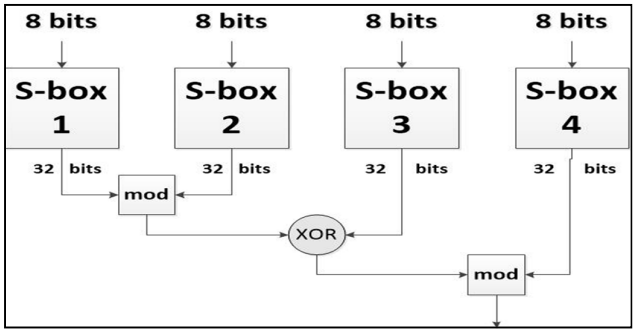
1) Inisialisasi P-array sebanyak 18 buah dengan masing-masing memiliki nilai sebesar 32 bit subkunci.
2) Inisialisasi S-box sebanyak 4 buah masing-masing bernilai 32 bit dengan maksimal tamping sebanyak 256 (0 sampai 255).
3) Masukan *plaintext* yang akan dienkripsi. Dilakukan proses segmentasi terhadap *Plaintext* sebesar per 64 bit, dan apabila kurang dari 64 bit maka akan ditambahkan bitnya, agar ukuran data sesuai dengan operasi selanjutnya.
4) Hasil dari langkah (3) dibagi menjadi 2 bagian, 32 bit pertama disebut XL dan 32 bit kedua disebut XR.
5) Terapkan operasi XL = XL XOR Pi dan XR = F(XL) XOR XR pada hasil dari langkah sebelumnya.
6) Dilakukan proses pertukaran dari XL menjadi XR dan XR menjadi XL terhadap hasil dari langkah (5) ditukar.
7) Lakukan langkah-langkah sebelumnya sebanyak 16 putaran. Pada iterasi ke-16 lakukan kembali langkah (6).
8) Pada iterasi ke-17 lakukan operasi untuk XR = XR XOR P17 dan XL = XL XOR P18.
9) Tahap akhir lakukan proses penyatuan dua bagian yaitu XL dan XR sehingga menjadi 64-bit kembali.

Blowfish menggunakan jaringan Feistel untuk penerapan proses enkripsi dan dekripsinya. Skema jaringan Feistel ditampilkan pada Gambar 1.



Gambar 1. Skema Jaringan Feistel

Dalam algoritma Blowfish juga terdapat fungsi F. Skema dari fungsi F dalam algoritma Blowfish dapat dilihat pada Gambar 2.



Gambar 2. Skema F dalam algoritma Blowfish

Bagi XL ke dalam empat bagian yang dideklarasi sebagai a, b, c, dan d yang masing-masing memiliki ukuran 8 bit. Apabila secara matematis dinyatakan seperti Persamaan (1).

$$F(XL)=((S1,a+S2,b \bmod 2^{32}) \text{xor } S3,c)+S4,d \bmod 2^{32} \quad (1)$$

Pada tahap (1) pada proses enkripsi algoritma Blowfish terdapat P-array yang terdiri dari 18 subkunci. Sub kunci dihitung pada proses ekspansi kunci, untuk alur pembentukan ekspansi kunci dengan kriptografi Blowfish dijabarkan sebagai berikut:

1) Inisialisasi P-array sebanyak 18 buah dan S-box sebanyak 4 buah secara beruntun dengan string tetap. String terdiri atas nilai Pi dibelakang angka 3 dalam bentuk hexadesimal, misal: P1=0x243f6a88 dan seterusnya.
2) P1 XOR 32-bit pertama kunci, XOR P2 dengan 32-bit kedua dari kunci dan seterusnya sampai P18. Lakukan operasi tersebut terhadap bit kunci sampai seluruh P-array telah diopreasikan terhadap bit kunci.
3) Enkripsi *plaintext* yang seluruhnya bernilai nol, dengan masukan subkunci yang telah dideskripsikan pada langkah (1) dan (2).
4) Tukar nilai P1 dan P2 dengan hasil dari langkah (3).
5) Enkripsi hasil langkah (3) dengan subkunci yang telah diubah pada langkah (4).
6) Tukar nilai P3 dan P4 dengan hasil dari langkah (5).
7) Lakukan semua langkah sebelumnya, agar seluruh elemen P-array dan kemudian S-box tertukar secara beruntun.

Diperlukan 521 iterasi untuk menghasilkan subkunci yang diperlukan. Agar proses enkripsi dan dekripsi terbilang efisien, diperlukan ruang penyimpanan untuk menyimpan subkunci-subkunci yang dihasilkan dan tidak membutuhkan proses penurunan pada operasi ekspansi kunci berulang kali, terkecuali terdapat perubahan pada kunci yang digunakan.

Pada proses dekripsi menggunakan algoritma Blowfish tahap yang dioprasikan hamper sama persis dengan proses enkripsi, hanya saja yang membedakan nya yaitu pada inisialisasi P-array (P1,P2,……..P18) diurutkan dengan urutan terbalik atau diinverskan.

B. Rivest Shamir Adleman (RSA)

RSA adalah kriptografi asimetris dengan teknik kunci public yang populer. RSA memiliki keamanan yang tinggi dikarenakan penggunaan dua kunci yang berbeda pada proses enkripsi dan dekripsinya dan sulitnya memfaktorkan bilangan menjadi faktor prima dengan tujuan mendapat kunci untuk proses dekripsi. Namun salah satu ancaman yang sering dialami RSA adalah *Brute Force Attack* [17]. Terdapat tiga proses utama dalam metode RSA yaitu pembentukan dua pasang kunci yang menghasilkan kunci untuk enkripsi dan dekripsi, proses enkripsi, dan proses dekripsi.

Proses pertama yaitu pembentukan dua pasang kunci (kunci publik dan kunci privat). Kunci publik bersifat umum digunakan untuk enkripsi sedangkan untuk kunci privat bersifat rahasia digunakan untuk proses dekripsi. Pengirim maupun penerima pesan harus memiliki kunci publik dan privat, jadi pembentukan kunci publik dan privat dilakukan oleh pengirim dan penerima. Berikut alur proses pembentukan kunci pada RSA:

1) Menentukan nilai dua bilangan prima (p dan q) dengan ketentuan kedua bilangan tersebut tidak boleh sama.
2) Menghitung nilai n dengan Persamaan (2). Nilai dari n digunakan untuk proses enkripsi dan dekripsi.

$$n = p*q \quad (2)$$

3) Menghitung nilai φ dari n (kunci publik) dengan Persamaan (3). Nilai φ(n) digunakan untuk mencari kunci *privat.*

$$\varphi(n) = (p-1)*(q-1) \quad (3)$$

4) Menentukan nilai e. Ketentuan nilai tidak boleh nilai factorial dari nilai φ(n) yang adalah bilangan prima dan $1 < e < \varphi(n)$. Mengecek nilai e menggunakan Persamaan (4).

$$(e*d) \bmod \varphi(n) = 1 \quad (4)$$

5) Menghitung nilai d dengan Persamaan (5). Nilai d digunakan sebagai kunci *privat.*

$$d = \frac{1+k\,\varphi(n)}{e} d = \frac{1+k\,\varphi(n)}{e}$$

(5)

6) Kunci publik (n, e) dan Kunci *privat* (d, n).

Setelah kedua kunci terbentuk, proses selanjutnya yaitu proses enkripsi dan dekripsi. Proses enkripsi dalam RSA menggunakan Persamaan (6) dengan parameter masukan adalah *plaintext* dan kunci publik. Proses dekripsi menggunakan Persamaan (7) dengan parameter masukan adalah *ciphertext* dan kunci *privat*.

$$C = P^e \bmod n \quad (6)$$

$$P = C^d \bmod n \quad (7)$$

Di mana:
C = *ciphertext* / pesan tersandikan.
P = *plaintext* / pesan
e = kunci *publik*
d = kunci privat
n = modulo pembagi

C. Electronic Mail

Electronic Mail atau *email* merupakan sarana pengiriman pesan melalui internet yang paling populer, baik dalam ruang lingkup organisasi maupun pribadi. *Email* digunakan sebagai media komunikasi jarak jauh yang saat ini populer. Pendefinisian

email meliputi empat komponen utama yaitu alamat *email* pengirim, alamat *email* penerima, judul *email,* dan isi *email*.

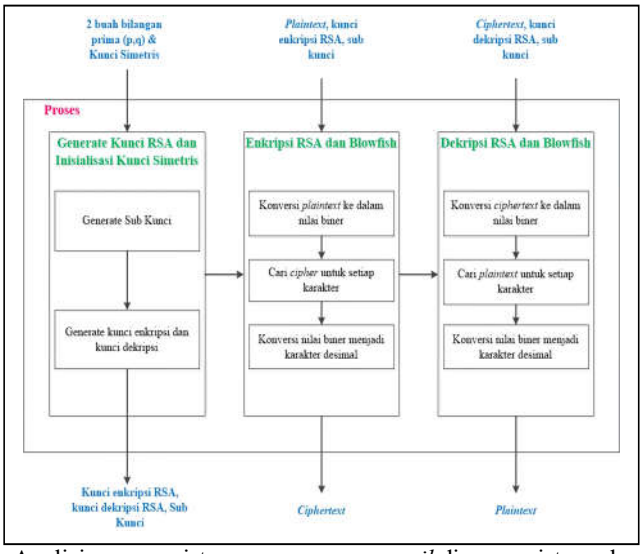
Email memiliki dua sistem dalam pengirimannya yaitu Mail User Agent (MUA) yang fungsinya untuk membuat dan melihat *email* masuk dan Mail Tranfer Agent (MTA) yang fungsinya untuk mengatur segala proses pengiriman dan penerimaan *email,* contohnya adalah menentukan protokol, *header* email, dan konektivitas pada server.

Dalam proses procedural dalam mengirim dan menerima *email* tentu memiliki standarisasi digunakan pada aplikasi *email client* di antaranya adalah Simple Mail Transport Protocol (SMTP) digunakan untuk berkomunikasi dengan server agar *email* yang dikirim dapat diterima oleh penerima *email*, Post Office Protocol versi tiga (POP3) digunakan untuk menerima email dan menyimpannnya di *local mail*, dan Internet Mail Access Protocol (IMAP) berfungsi untuk mengirim email ke webmail dengan komunikasi dua arah sebagai perubahan yang diproses pada *local mail* sebelum dikirim kembali ke server [18].

III. METODE

Sistem yang dibangun mengkombinasikan dua metode kriptografi untuk mengamankan *email.* Rancangan skema sistem pengamanan *email* ini dapat dilihat pada Gambar 3.

Gambar 3. Rancangan skema Sistem Pengamanan *Email*



Analisis proses sistem pengamanan *email* di mana sistem akan melakukan perubahan isi *email* dan kunci simetris dengan kata lain menyamarkan kedua *plaintext* tersebut tanpa mengubah makna dari isi *email* tersebut, hal ini dilakukan guna meningkatkan keamanan dan tidak diketahui oleh orang yang tidak berhak mengakses isi *email* dan kunci simetris tersebut. Pengamanan dilakukan dengan cara menyandikan data masukan menggunakan dua metode kriptografi yaitu Blowfish dan RSA. Blowfish digunakan untuk mengamankan isi *email* yang akan didistribuskan ke penerima *email,* metode Blowfish digunakan karena pada algoritma Blowfish ini terbilang cepat dan sederhana hal tersebut dikarenakan operasi yang digunakan hanya XOR dan penambahan (*addition*) serta menerapkan konsep Fiestel Network dan Block Cipher untuk setiap operasinya dengan begitu Blowfish dapat berjalan pada perangkat yang memiliki memory rendah. Tak hanya itu, Blowfish sangat aman dikarenakan jangkauan ukuran kunci yang digunakan terbilang bervariasi yaitu 32 bit sampai 448 bit. Sedangkan untuk RSA digunakan untuk mengamankan kunci simetris pada Blowfish yang digunakan untuk proses enkripsi dan dekripsi *email* dengan memanfaatkan konsep kunci asimetris agar memberikan kemanan lebih pada saat pertukaran simetris.

D. Rancangan Tahap Enkripsi (Pengirim Email)

1) Memasukan *email* dan kunci simetris. Isi *email* digunakan sebagai *plaintext.*
2) Lakukan proses enkripsi Blowfish dengan masukan dari tahap (1).
3) Kirim *email* terenkripsi ke penerima *email* Hasil dari tahap (2) berupa *ciphertext.*
4) Masukan bilangan prima p dan q.
5) Lakukan pembentukan kunci. Menghasilkan kunci privat dan kunci publik
6) Lakukan proses enkripsi RSA dengan masukan kunci private dan kunci simetris
7) Kirim kunci simetris terenkripsi ke penerima.

E. Rancangan Tahap Enkripsi (Penerima Email)

1) Menerima *email* terenkripsi dan kunci simetris terenkripsi dari pengirim.
2) Memasukan *kunci simetris terenkripsi.*
3) Lakukan proses dekripsi RSA dengan masukan dari kunci privat.
4) Masukan kunci simetris dan *email* terenkripsi
5) Lakukan proses dekripsi Blowfish dengan masukan tahap (3).
6) Email kembali seperti semula dan dapat terbaca.

F. Perhitungan Manual Kriptografi Blowfish

Untuk lebih memahami alur proses dari kriptografi Blowfish maka dibuat contoh perhitungan manual dari proses yang terdapat dari kriptografi Blowfish. Pada proses enkripsi penulis terdapat parameter yaitu *plaintext* = "IBOYIBOY" dan kunci simetris = "1903". Langkah pertama adalah inisialisasi P-array masing-masing 32 bit sebanyak 18 buah. Nilai P-array dapat dilihat pada Tabel 1.

TABEL I. INISIALISASI P-ARRAY

P-array	Hexadesimal	Biner (32 bit)
P1	387A8C6A	00111000 01111010 10001100 01101010
P2	7B932764	01111011 10010011 00100111 01100100
….	….	….
P18	2A9F3763	00101010 10011111 00110111 01100011

Inisilasi S-box sebanyak 4 buah dengan masing berjumlah 255 buah dalam bentuk biner yang sebelumnya dikonversi dari hexadesimal. Nilai S-box dapat dilihat pada Tabel 2.

TABEL II. INISIALISASI S-BOX

P-array	Hexadesimal	Biner (32 bit)
S1,0	D2324F87	11010010 00110010 01001111 10000111
….	….	….
S1,255	B837122F	10111000 00110111 00010010 00101111
….	….	….
S4,0	B1233F72	10110001 00100011 00111111 01110010
….	….	….
S4,255	873FF312	10000111 00111111 11110011 00010010

Konversikan *plaintext=”*IBOYIBOY*”* kedalam biner maka hasil yang didapat adalah 01001001 01000010 01001111 01011001. Kemudian pemecahan pada *plaintext* menjadi dua bagian yaitu XL dan XR. XL = 01001001 01000010 01001111 01011001 dan XR = 01001001 01000010 01001111 01011001.

Sebelum melakukan proses ekspansi kunci pada kunci simetris yang digunakan, lakukan konversi pada kunci simetris = “1903” ke bilangan biner maka hasil yang didapat adalah 00110001 00111001 00110000 00110011. Setelah mendapat nilai biner dari kunci simetris, langkah selanjutnya adalah proses ekspansi kunci parameter masukan yang digunakan adalah P-array dan kunci simetris dalam bentuk bilangan biner. P1 = P1 XOR Kunci maka P1 = 00111000 01111010 10001100 01101010 XOR 00110001 00111001 00110000 00110011 = 00001001 01000011 10111100 01011001. Lakukan operasi XOR sampai seluruh P-array termodifikasi.

Dalam contoh hal ini iterasi yang dilakukan hanya satu iterasi, dikarenakan total iterasi proses enkripsi sebanyak 16 kali putaran. Lakukan operasi XL = XL XOR P1 (subkunci) maka menjadi XL = 01001001 01000010 01001111 01011001 XOR 00001001 01000011 10111100 01011001 = 01000000 00000001 11110010 00000000. Kemudian bagi XL menjadi empat bagian (a,b,c,d) dengan masing-masing 8 bit. a = 01000000, b = 00000001, c = 11110010, dan d = 00000000.

Kemudian lakukan persamaan (1) dengan parameter input adalah S-box dan XL yang telah dibagi menjadi empat bagian. Maka hasil yang didapat adalah dari persamaan (1) adalah F(XL) = 11100110010011010001000011111010111.

Lakukan operasi XR = F(XL) XOR XR. Maka hasil dari operasi tersebut adalah XR = 11000010111011000011011101010111 0. Lalu tukar nilai XR menjadi XL dan XL menjadi XR.

Setelah melakukan 16 iterasi, maka akan menghasilkan nilai dari XL dan XR baru. Tukar kembali XL dan XR. Setelah itu lakukan operasi XR = XR XOR P17 dan XL = XL XOR P18. Kemudian gabungkan kembali nilai XL dan XR sehingga menjadi 64 bit. Tahap akhir yaitu konversi nilai biner kedalam Base64 sehingga menghasilkan *ciphertext* = ″PjwvwoVLCwYw”.

Untuk operasi dekripsi pada Blowfish hamper sama dengan proses enkripsi hanya saja urutan inisialisasi P-array dibuat terbalik.

G. Perhitungan Manual Kriptografi RSA

Pembentukan kunci *public* dipergunakan untuk proses enkripsi sedangkan untuk kunci *private* digunakan untuk proses dekripsi. Pengirim maupun penerima pesan harus memiliki kunci *public* dan *private*, jadi pembentukan kunci *public* dan *private* dilakukan oleh pengirim dan penerima. Pembentukan kunci dengan langkah pertama menentukan dua buah bilangan prima (p dan q) secara acak, misal p = 13 dan q = 31. Selanjutnya pembentukan kunci *public* (n) yang didapat dari hasil 13 (p) dikalikan dengan 31 (q), maka kunci *public* (n) yang didapat adalah 403. Hitung nilai φ dari kunci *public* dengan persamaan (3).

Maka φ = (13-1)*(31-1) = 360. Tentukan nilai K_P dengan ketentuan nilai tidak boleh merupakan nilai factorial dari hasil persamaan (3) yang bersifat bilangan prima, maka sebelumnya dicari nilai factorial dari hasil persamaan (3) dengan kesimpulan tidak boleh bernilai 2, 3, atau 5. Maka disini ditentukan e = 7. Lalu hitung nilai d, dengan persamaan (4)

Maka (7*d) mod 360. Hasil dari d adalah bilangan bulat dengan mencoba nilai-nilai m (bilangan integer) sehingga diperoleh d adalah 103 dengan m = 2. Maka didapat dua pasang kunci adalah Kunci *public* (n, e) = Kunci *public* (403,7) dan Kunci *private* (d, n) = Kunci *private* (103,403)

Dalam enkripsi pada algoritma RSA kunci yang digunakan adalah kunci *public* = (403,7) yang sudah didapat dari proses sebelumnya. Misal *plaintext* adalah IQBAL. Sebelum dilakukan proses enkripsi, konversikan *Plaintext* ke dalam nilai desimal. Setelah *plaintext* dikonversi ke dalam nilai desimal berdasarkan kode ASCII, langkah selanjutnya adalah proses enkripsi dengan persamaan (6)

Maka apabila dijabarkan dari hasil persamaan (6) yaitu $I = 73^7 \bmod 403 = 44$, $Q = 81^7 \bmod 403 = 224$, $B = 66^7 \bmod 403 = 326$, $A = 65^7 \bmod 403 = 234$, $L = 76^7 \bmod 403 = 236$.

Jadi enkripsi yang dihasilkan dari *plaintext* IQBAL merupakan *ciphertext* dalam bentuk nilai desimal yaitu 44224326234236 dan apabila dikonversikan kedalam.

Untuk proses dekripsi menggunakan kunci *private* dan *ciphertext* sebagai masukkannya. *Ciphertext* = 44224326234236 dan Kunci *Private* = (103,403). Proses dekripsi dengan menggunakan persamaan (7).

Maka apabila dijabarkan yaitu $44 = 44^{103} \bmod 403 = 73$, $224 = 224^{103} \bmod 403 = 81$, $326 = 326^{103} \bmod 403 = 66$, $234 = 234^{103} \bmod 403 = 65$, $236 = 236^{103} \bmod 403 = 76$.

Hasil yang didapat adalah 7381666576. Kemudian lakukan konversi dari nilai desimal ke dalam karakter berdasarkan kode ASCII, maka hasil yang didapat adalah “IQBAL”. Jadi hasil

proses dekripsi yaitu *plaintext* memiliki nilai dan makna yang sama dari *plaintext* yang sebelumnya dilakukan proses enkripsi.

Pada Sistem Pengamanan *Email* terdapat satu aktor yang terlibat dalam sistem yaitu pengirim atau penerima pesan.

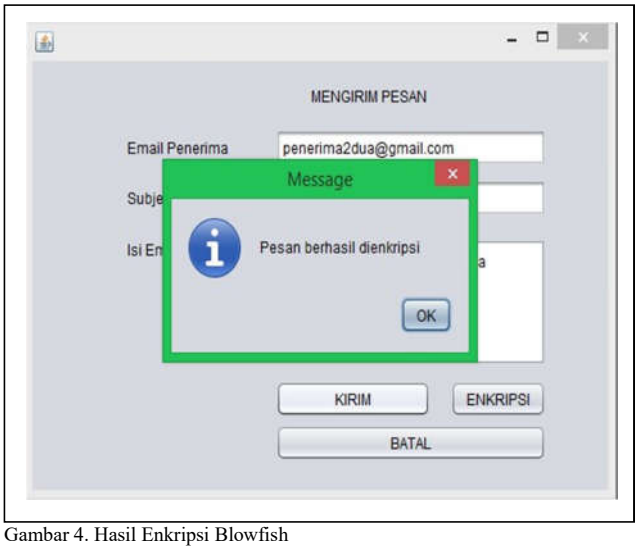
IV. HASIL DAN DISKUSI

Implementasi dan percobaan sistem pada penelitian ini menggunakan Bahasa pemrograman JAVA dengan IDE Netbeans 8.2. Dibutuhkan dua alamat *email* yang digunakan sebagai pengirim dan penerima. Nilai bilangan prima yang dipakai di dalam percobaan adalah p=13 dan q=31, serta kunci Simetris = "IQBAL".

Pengujian keamanan dan analisis dari penelitian ini dilakukan dalam dua tahapan yaitu pada masing-masing algoritma enkripsi dan dekripsi secara paralel. Sedangkan pada tahapan selanjutnya, analisis keamanan pada *ciphertext* kunci simetris dengan cara melihat ketersedian dan kemungkinan kunci yang dihasilkan jika dilakukan penyerangan menggunakan metode brute-force.

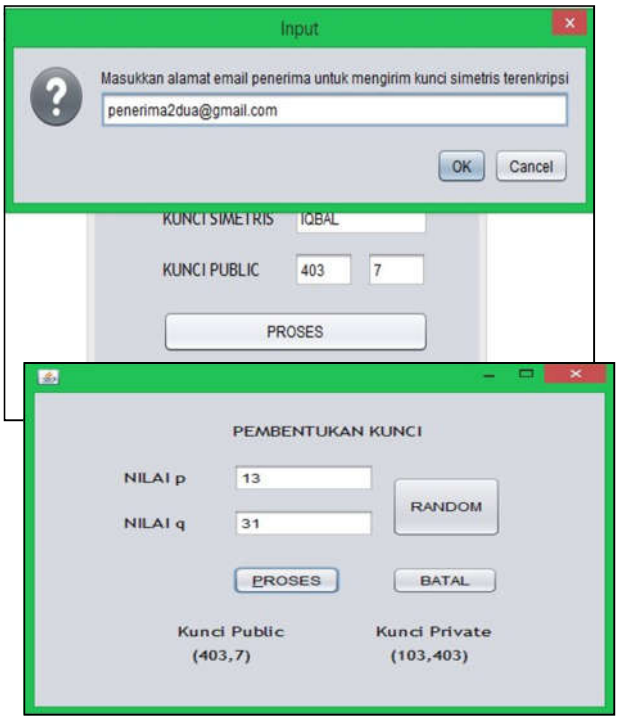
H. Hasil Enkripsi

Hasil enkripsi dari algoritma yang digunakan berupa *ciphertext*. Pada tahapan ini pengujian enkripsi dilakukan pada setiap algoritma kriptografi yang digunakan. Pesan *email hasil* enkripsi menggunakan algoritma Blowfish yaitu *ciphertext* dalam bentuk diperlihatkan pada Gambar 5. Masukan adalah pesan email dengan isi "Mencoba mengirim pesan pada penerima" dengan kunci simetris adalah "mizuka". Hasil dari enkripsi adalah "w4MSQWd7FhJ1EiFPwrMQUMO1P8K1wq zDksOSEMKwKxJ+w7nDmsO+woXChV7Dh1/Dv8OlJ3/Chc K7eg==", yaitu *ciphertext* dalam bentuk base64.



Gambar 4. Hasil Enkripsi Blowfish

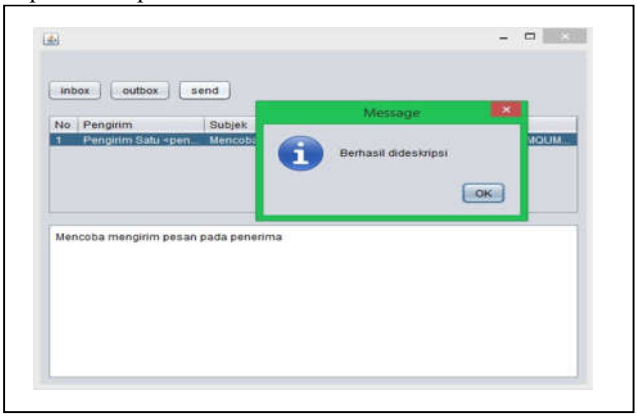
Sedangkan hasil dari algoritma RSA yaitu berupa *ciphertext* dalam bentuk nilai desimal diperlihatkan pada Gambar 5. Masukan adalah kunci simetris dengan isi bertipe text yaitu "IQBAL" dengan kunci *publik* "(403,7)". Hasil dari enkripsi menggunakan algoritma RSA adalah "44224326234236". Kunci *publik* dihasilkan dari fitur pembentukan kunci yang terdapat pada sistem. Kunci *publik* dapat langsung dikirimkan ke penerima dengan memasukan alamat *email* yang dituju Implementasi proses pembentukan kunci dapat dilihat pada Gambar 6.



Gambar 6. Pembentukan Kunci

I. Hasil Dekripsi

Pada tahapan ini, dekripsi pada isi pesan *email* dan kunci simetris disatukan ke dalam satu tampilan. Yang membedakkannya adalah pada saat memasukan kunci yang nantinya dibutuhkan untuk proses dekripsi. Proses dekripsi dapat dilihat pada Gambar 7.



Gambar 7. Hasil Deskripsi

J. Hasil Pengujian terhadap ukuran Data

Hasil pengujian pada ukuran *email* sebelum dan sesudah dienkripsi dapat dilihat pada Tabel 3.

TABEL III. PERBEDAAN UKURAN DATA

Subjek *Email*	Ukuran sebelum dienkripsi	Ukuran sesudah dienkripsi	Selisih
Mencoba Kirim Pesan	4,68 KB	4,77 KB	0.09 KB

Perbedaan ukuran setelah dienkripsi tidak terlalu signifikan dari ukuran sebelumnya. Akibat dari pembesaran ukuran tersebut mempengaruhi waktu proses dekripsi yang sedikit lebih besar dibanding dengan proses enkripsi.

K. Serangan Brute Force

Serangan *Brute Force* sebuah teknik yang digunakan untuk meretas sebuah sistem komputer dengan melakukan beberapa percobaan kemungkinan untuk mendapatkan nilai yang dibutuhkan, biasanya digunakan untuk mengetahui nilai dalam kunci privat. Salah satu nilai pada kunci privat yaitu n rentan dan mudah untuk diketahui karena bersifat umum dan bernilai sama dengan kunci publik. Pada pengujian ini dilakukan sebanyak tiga kali percobaan dengan mencoba nilai dari e dan menkombinasikan nya pada nilai n.

Pada Tabel 4 menunjukan kombinasi pasangan kunci *privat* untuk serangan *brute force.*

TABEL IV. PERCOBAAN SERANGAN BRUTE FORCE

Percobaan	Kunci Publik	Kunci Pivate	Ciphertext
1	(403,7)	(83,403)	44224326234236
2	(403,7)	(89,403)	44224326234236
3	(403,7)	(97,403)	44224326234236

Percobaan selanjutnya dilakukan proses deskripsi pada *ciphertext* dengan menggunakan kombinasi kunci yang tersedia. Hasil deskripsi ditunjukan pada Tabel 5.

TABEL V. PERCOBAAN DEKRIPSI BRUTE FORCE

Percobaan	Kunci Publik	Kunci Pivate	Plaintext
1	(403,7)	(83,403)	V¥B[null][null]
2	(403,7)	(89,403)	[null][null]□h[null]
3	(403,7)	(97,403)	Ï[null][null][null][null]

Dapat ditampilkan pada Tabel 5 hasil dari pengujian serangan *brute force* tidak terdapat kesesuaian pada *plaintext* target serang, yaitu "IQBAL". Jumlah dari karakter kunci d menentukan banyaknya jumlah kemungkinan yang harus dilakukan percobaan untuk mendapatkan *plaintext.*

Dapat dilihat pada Tabel 4 jumlah bit kunci adalah 2 karakter (16 bit). Maka dapat dikatakan kemungkinan kunci yang tersedia adalah 2^{16} = 65.536. Hal tersebut memerlukan waktu yang cukup lama untuk proses perhitungan serangan brute force.

V. KESIMPULAN

Penelitian ini telah melakukan pengamanan *email* dengan mengkombinasikan dua algoritma kriptografi yaitu Blowfish dan RSA untuk proses enkripsi dan dekripsi. Blowfish digunakan untuk mengamankan isi pesan yang dikirim sedangkan RSA digunakan untuk mengamankan kunci simetris yang digunakan untuk proses enkripsi dan dekripsi isi pesan. Dari hasil percobaan hasil enkripsi pada data berukuran 4,68 KB berhasil terenkripsi dan mengalami perubahan ukuran yaitu 0,09 KB lebih besar dari sebelumnya, akan tetapi perubahan tersebut tidak terlalu signifikan untuk berpengaruh dalam waktu proses pengiriman *email.* Dari hasil serangan *brute force* dapat disimpulkan bahwa algoritma kriptografi yang digunakan terbilang aman, panjang karakter kunci sangat berpengaruh terhadap tingkat kesulitan untuk memecahkan *plaintext* menggunakan serangan *brute force.*

REFERENSI

[1] J. Klensin, "Simple Mail Transfer Protocol," *Animal Genetics*, vol. 39, no. 5, pp. 561–563, 2008.

[2] F. Callegati, W. Cerroni, and M. Ramilli, "Man-in-the-middle attack to the HTTPS protocol," *IEEE Security and Privacy*, vol. 7, no. 1, pp. 78–81, 2009.

[3] R. K. Gupta and P. Singh, "A New Way to Design and Implementation of Hybrid Crypto System for Security of the Information in Public Network," *International Journal of Emerging Technology and Advanced Engineering*, vol. 3, no. 8, pp. 108–115, 2013.

[4] G. Devika, M. Ilayaraja, and K. Shankar, "A Modified Symmetric Key Cryptography Method for Secure Data Transmission," *International Journal of Pure and Applied Mathematics*, vol. 116, no. 10, pp. 301–308, 2017.

[5] P. Singh and K. Singh, "Image Encryption and Decryption Using Blowfish Algorithm in Matlab," *International Journal of Scientific & Engineering Research*, vol. 4, no. 7, pp. 150–154, 2013.

[6] R. Vasantha and R. S. Prasad, "An Advanced Security Analysis by Using Blowfish Algorithm," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 2, no. 6, pp. 1031–1036, 2018.

[7] S. Anandakumar, "Image Cryptography Using RSA Algorithm in Network Security," *International Journal of Computer Science & Engineering Technology*, vol. 5, no. 9, pp. 326–330, 2015.

[8] P. Patil, P. Narayankar, D. G. Narayan, and S. M. Meena, "A Comprehensive Evaluation of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish," in *Procedia Computer Science*, 2016, vol. 78, pp. 617–624.

[9] R. Tripathi and S. Agrawal, "Comparative Study of Symmetric and Asymmetric Cryptography Techniques," *International Journal of Advance Foundation and Research in Computer*, vol. 1, no. 6, pp. 68–76, 2014.

[10] A. Gore, S. . Meena, and P. Purohit, "Hybrid Cryptosystem using Modified Blowfish Algorithm and SHA Algorithm on Public Cloud," *International Journal of Computer Applications*, vol. 155, no. 3, pp. 6–10, 2016.

[11] D. Bhole, A. Mote, and P. R. Patil, "A New Security Protocol Using Hybrid Cryptography Algorithms," *International Journal of Computer Sciences and Engineering*, vol. 4, no. 2, pp. 18–22, 2016.

[12] S. Chandra, B. Mandal, S. S. Alam, and S. Bhattacharyya, "Content Based Double Encryption Algorithm Using Symmetric Key Cryptography," in *Procedia Computer Science*, 2015, vol. 57, pp. 1228–1234.

[13] S. Wibowo and Suprayogi, "Aplikasi Enkripsi Email Dengan Menggunakan Metode Blowfish Berbasis J2SE," *Techno.COM*, vol. 13, no. 2, pp. 75–83, 2014.

[14] L. J. Pangaribuan and F. H. Simbolon, "Kriptografi Hybrida Menggunakan Algoritma Hill Cipher dan Algoritma RSA Untuk Keamanan Pengiriman Informasi Pada Email," in *Konferensi Nasional Teknologi Informasi dan Komputer*, 2017, vol. 1, pp. 1–11.

[15] B. Prasetiyo, M. A. Muslim, and H. Susanto, "Penerapan Kriptografi

Algoritma Blowfish pada Pengamanan Pesan Data Teks," *Techno.COM*, vol. 16, no. 4, pp. 358–366, 2017.

[16] F. Zuli and A. Irawan, "Implementasi Kriptografi Dengan Algoritma Blowfish dan Riverst Shamir Adleman (RSA) Untuk Proteksi File," *Jurnal Format*, vol. 6, no. 2, pp. 27–38, 2016.

[17] A. P. Wahyadyatmika, R. R. Isnanto, and M. Somantri, "Implementasi Algoritma Kriptografi RSA pada Surat Elektronik (E-Mail)," *TRANSIENT*, vol. 3, no. 4, pp. 443–450, 2014.

[18] D. Email, A. Ginting, R. R. Isnanto, and I. P. Windasari, "Implementasi Algoritma Kriptografi RSA untuk Enkripsi dan Dekripsi Email," *Jurnal Teknologi dan Sistem Komputer*, vol. 3, no. 2, pp. 253–258, 2015.