Analisis Penggunaan Algoritma RSA untuk Enkripsi Gambar dalam Aplikasi *Social Messaging*

Agus Gunawan / 13515143
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13515143@std.stei.itb.ac.id or *agus.gunawan19@outlook.com*

***Abstract* — Setiap saat tentunya kita tidak pernah lepas dari yang namanya *smartphone*. Apalagi di zaman modern sekarang ini dimana hampir setiap orang sudah memiliki yang namanya *smartphone*. Penggunaan *smartphone* tentunya terfokus untuk berhubungan sosial dengan orang lain. Penggunaan *Social Messaging* merupakan salah satu bentuk dari hubungan sosial tersebut. Terkadang kita ingin mengirim sebuah gambar yang mengandung privasi kepada orang lain. Namun, saat ini *Social Messaging* yang ada belum menyediakan fitur untuk melakukan enkripsi gambar yang dikirim sehingga dapat menjamin privasi dari gambar yang ingin dikirim. Padahal fitur ini sangatlah penting apalagi di zaman sekarang ini dimana teknologi sudah semakin maju dan para peretas tentunya sudah semakin ahli dan mungkin dapat membobol suatu sistem *Social Messaging* sehingga dapat melihat data – data yang ada di *Social Messaging* tersebut. Dengan menggunakan algoritma RSA yang sedikit dimodifikasi, penulis membuat suatu algoritma yang dapat digunakan untuk enkripsi gambar yang akan dikirim melalui aplikasi *Social Messaging* dengan waktu yang tidak menghilangkan kenyamanan saat mengirim gambar yang ingin dienkripsi.**

***Keywords* — RSA, *Social Messaging*, Gambar, Enkripsi Gambar.**

I. PENDAHULUAN

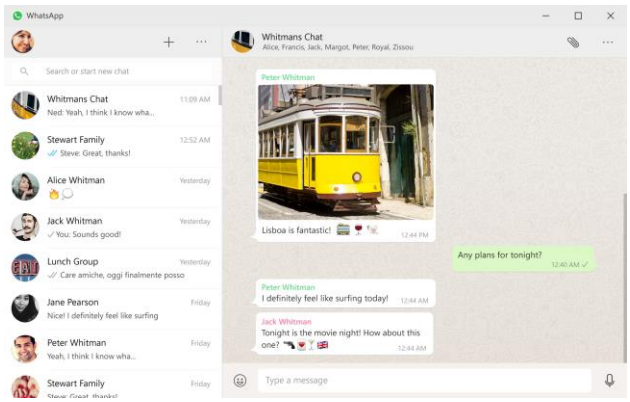
Di zaman sekarang ini perkembangan teknologi sudah sangatlah cepat. Informasi sudah dapat diakses dimana – mana dan sangatlah mudah tersebar dimanapun kita berada. Salah satu teknologi yang berperan penting dalam hal ini adalah Internet.

Internet adalah keseluruhan jaringan komputer/perangkat lainnya yang saling terhubung menggunakan sebuah protokol yaitu Internet Protokol. Pengguna Internet di Indonesia sendiri sangatlah banyak dan menurut data mencapai 82 juta orang yang sudah menggunakan Internet. Salah satu penggunaan Internet adalah mencari informasi dan menyebarkan informasi.

Penggunaan Internet memiliki hubungan dengan makin banyaknya pengguna *smartphone* di Indonesia. *Smartphone* merupakan sebuah perangkat yang memiliki tujuan utama sebagai telepon namun memiliki kemampuan seperti komputer. Kemampuan inilah yang menyebabkan *smartphone* juga dapat menggunakan Internet.

Salah satu penggunaan *smartphone* dan Internet adalah untuk berhubungan sosial dengan orang lain. Salah satu cara untuk berhubungan sosial dengan orang lain adalah dengan menggunakan sebuah aplikasi yang dinamakan *social messaging*. Aplikasi ini merupakan aplikasi yang harus berhubungan dengan Internet dan dapat digunakan untuk melakukan obrolan di *smartphone* baik itu obrolan secara teks maupun verbal.

Penggunaan *social messaging* ini tentunya harus menjaga privasi dari tiap orang yang menggunakannya. Obrolan tiap orang dengan orang lain harus tidak dapat dilihat oleh publik ataupun disadap oleh orang lain sehingga tiap aplikasi *social messaging* harus memiliki keamanan yang baik agar dapat menjaga privasi tiap orang.



Gambar 1 Salah satu aplikasi social messaging (Sumber : https://blog.whatsapp.com/img/faq/id/blog/79a77be6e2d3985e0a2bc8dd7d4a1f8086fc6334.jpg).

Seiring berkembangnya teknologi para peretas juga menjadi semakin ahli dan kemungkinan dapat dengan mudah menerebos masuk ke server dari aplikasi *social messaging* yang memiliki keamanan yang lemah. Setiap orang yang menggunakan aplikasi yang sudah diretas tersebut tentunya akan khawatir dengan obrolan yang dimilikinya dapat tersebar di Internet.

Biasanya aplikasi *social messaging* sudah menyediakan sebuah enkripsi untuk dapat mengamankan obrolan dari penggunanya. Namun, aplikasi – aplikasi yang ada belum

menyediakan pengamanan terhadap kontennya yang salah satunya adalah saat pengguna menggunakan aplikasi *social messaging* untuk berbagi gambar dengan temannya, dimana kebanyakan *social messaging* sekarang belum memiliki kemampuan untuk menjaga privasi dari gambar yang dikirimkan tersebut. Padahal ada kemungkinan bahwa peretas dapat membobol server dari *social messaging* dan menyebarkan gambar – gambar privasi yang ada ataupun mungkin gambar – gambar yang tergolong rahasia diri sendiri.

Oleh karena itu penulis berharap dengan adanya makalah ini aplikasi *social messaging* dapat lebih mengembangkan aplikasinya untuk dapat menyediakan fitur enkripsi gambar yang salah satunya dapat dilakukan dengan menggunakan algoritma RSA dalam pengenkripsiannya karena algoritma ini merupakan algoritma yang termasuk aman dan membutuhkan waktu yang cukup cepat baik untuk mengenkripsi maupun mendekripsi suatu gambar sehingga privasi dari masing – masing pengguna *social messaging* tersebut dapat lebih terjaga.

II. LANDASAN TEORI

2.1. Kriptografi

Kriptografi adalah sebuah ilmu yang digunakan untuk menjaga kerahasiaan sebuah pesan, data ataupun informasi dengan cara merubahnya menjadi bentuk tersandi yang tidak mempunyai makna.

Berdasarkan kunci yang digunakan untuk proses enkripsi dan dekripsi, kriptografi dapat dibedakan menjadi dua macam yaitu :

1. Kriptografi Simetri

Pada kriptografi simetri, kunci untuk proses enkripsi sama dengan kunci untuk proses dekripsi, dan pengirim dan penerima pesan atau data akan memiliki kunci rahasia yang sama. Pengirim pesan atau data akan mengenkripsi sebuah pesan tersebut menggunakan sebuah kunci rahasia. Penerima pesan juga akan mendekripsi pesan tersebut dengan menggunakan kunci rahasia yang sama dengan pengirim pesan sehingga kunci rahasia memiliki peranan yang penting dalam kriptografi ini.

2. Kriptografi Asimetri

Pada kriptografi asimetri, kunci untuk proses enkripsi tidak sama dengan kunci untuk proses dekripsi. Kriptografi ini juga sering disebut dengan istilah kriptografi kunci publik karena kunci untuk enkripsi tidak dirahasiakan dan dapat diketahui oleh siapapun, sedangkan kunci untuk dekripsi hanya diketahui oleh pengirim dan penerima pesan yang sesuai. Biasanya penerapan kriptografi ini menggunakan sebuah algoritma yang dapat membuat sebuah kunci publik (kunci enkripsi) yang dapat dipasangkan dengan sebuah kunci pribadi (kunci dekripsi) dan dapat menyamarkan sebuah pesan menjadi bentuk tersandi yang dapat didekripsi dan menghasilkan hasil yang sama saat sebelum melakukan enkripsi dengan menggunakan kunci pribadi yang telah dibuat tersebut.

2.2. Gambar

Gambar merupakan sebuah kumpulan piksel – piksel yang dikomposisikan untuk mencitrakan sesuatu. Piksel merupakan titik terkecil dari sebuah gambar. Tiap foto tentunya memiliki masing – masing piksel yang tiap piksel diarepresentasikan dengan paduan tiga warna dasar yaitu *RGB (Red – Green – Blue).* Tiap piksel tersebut memiliki angka – angka tertentu untuk masing – masing warna. Angka – angka yang merepresentasikan piksel tersebut tentunya dapat dienkripsi menggunakan algoritma enkripsi yang sudah ada namun dengan sedikit pemodifikasian agar dapat menghasilkan hasil yang sesuai saat didekripsi.

2.3. Algoritma RSA

Algoritma RSA dianamakan berdasarkan penemunya yaitu Ron Rivest, Adi Shamir, dan Len Adleman yang menemukan algoritma ini pada tahun 1976. Algoritma RSA ini merupakan algoritma yang menitikberatkan pada teori bilangan terutama pada konsep bilangan prima dan aritmetika modulo sehingga kunci publik dan kunci pribadi dari algoritma ini merupakan bilangan bulat. Algortima RSA merupakan algoritma kunci publik yang penggunaannya sudah sangat luas dan sangat sering digunakan oleh orang – orang karena algoritma ini dapat digunakan tanpa mempertukarkan kunci rahasia secara terpisah.

Penerapan algoritma RSA terbagi menjadi tiga proses utama yaitu :

1. Pembangkitan Kunci

   Kunci dapat dibuat dengan menggunakan algoritma berikut :

   1. Pilih dua bilangan prima berbeda misalnya a dan b. Kedua bilangan ini harus dirahasiakan.
   2. Hitung n = a x b. Bilangan n tidak dirahasiakan.
   3. Hitung m = (a – 1) x (b – 1).
   4. Pilih sebuah bilangan e dimana bilangan ini mewakilkan kunci publik. Bilangan ini harus relatif prima terhadap bilangan m dimana PBB(e,m) = 1.
   5. Hitung kunci pribadi d dimana $ed \equiv 1 \pmod{m}$.

2. Enkripsi

   Proses enkripsi ini merupakan proses dimana seseorang sebut saja A mengirimkan kunci publiknya (n, e) kepada B. B akan membuat sebuah pesan untuk A yang setiap pesan tersebut akan dirubah ke bentuk sandi dengan rumus $C(i) = P(i)^e \bmod n$. Proses ini dapat diselesaikan secara efisien karena proses ini menggunakan eksponen modulo. Setelah itu B akan mengirim pesan yang telah berbentuk sebagai sandi tersebut kepada A.

3. Dekripsi

   Proses dekripsi ini merupakan proses dimana seseorang yang memiliki kunci pribadi mendapatkan pesan yang sudah disandikan menggunakan kunci publik yang dikirim dari orang yang memiliki kunci pribadi tersebut. Misalnya

orang A yang sudah mengirim kunci publik (n, e) kepada B menerima pesan dalam bentuk sandi dari B. A akan dengan mudah merubah bentuk tersandi tersebut menjadi sebuah teks atau informasi yang dapat dibaca dengan baik dengan menggunakan rumus $P(i) = C(i)^d \bmod n$ dimana d adalah kunci dekripsi.

III. Analisis dan Pembahasan

3.1. Implementasi RSA dalam Gambar yang Dirubah ke Teks Terenkripsi

Teks enkripsi dari suatu gambar diperoleh dari enkripsi sebuah teks yang berisi informasi perpaduan warna dasar RGB dari tiap piksel gambar tersebut. Misalnya saja gambar berdimensi 1 x 4 berikut :

Gambar 2 Contoh piksel dari sebuah gambar yang akan dirubah ke teks

Gambar tersebut akan menghasilkan teks representasi warna RGB dari tiap piksel yang dibaca dengan urutan per-baris terlebih dahulu, hasil dari pembacaan gambar 2 di atas dapat ditabulasi dalam tabel di bawah ini :

R	G	B
255	86	4
0	106	213
0	255	64
255	255	91

Tabel 1 Representasi warna RGB dari gambar 1.

Setelah melakukan pembacaan dari tiap piksel maka akan dilakukan proses enkripsi untuk setiap data yang didapatkan pada tabel tersebut Setelah tabel 1 tersebut dienkripsi untuk setiap warna menggunakan algoritma RSA dengan dua bilangan prima yang dipilih adalah 23 dan 29 sehingga hasil n nya adalah 667. Kemudian kunci publiknya adalah e = 25 dan kunci pribadi d = 345. Pengenkripsian dari data yang didapat pada tabel 1 akan menjadi tabel berikut :

R	G	B
C(255) = 629	C(86) = 405	C(4) = 179
C(0) = 0	C(106) = 582	C(213) = 607
C(0) = 0	C(255) = 629	C(64) = 473
C(255) = 629	C(255) = 629	C(91) = 643

Tabel 2 Teks enkripsi yang diperoleh dari gambar 1

Teks enkripsi ini nantinya akan dikirim oleh orang yang ingin mengirim sebuah gambar dan memiliki kunci publik ke orang yang memiliki kunci pribadi yang bersesuaian dengan kunci publik yang dimilikinya. Setelah teks enkripsi diterima oleh pemilik kunci pribadi sekaligus pengirim kunci publik maka pemilik kunci pribadi tersebut akan melakukan proses dekripsi dan kemudian akan membangun kembali gambar yang sama dengan gambar yang ingin dikirim seperti pada contoh ini gambar yang didekripsi dari representasi tabel pada tabel 2 akan sama dengan gambar 1. Perlu diperhatikan bahwa diperlukan informasi tambahan berupa dimensi dari gambar tersebut. Jadi misal gambar yang ingin dikirim berdimensi 1 x 4 maka di teks yang ingin dikirim ke penerima gambar enkripsi perlu dicantumkan dimensinya sehingga teks yang dikirim ke pemilik kunci pribadi berupa dimensi dari gambar yang dienkripsi dan representasi warna RGB dari setiap piksel tersebut yang telah dienkripsi sebelumnya. Implementasi ini merupakan implementasi paling sederhana dari pengenkripsian gambar menggunakan RSA. Implementasi ini dapat menghasilkan gambar yang sesuai dengan gambar yang ingin dienkripsi oleh pengirim, namun implementasi ini tidak dapat membangkitkan sebuah gambar hasil dari enkripsi dari gambar yang ingin dikirim karena maksimal angka yang dapat direpresentasikan oleh warna dasar RGB adalah 255-255-255. Jika dilakukan enkripsi gambar dan kemudian terdapat sebuah piksel yang melebihi angka 255 tersebut maka akan dapat terbentuk sebuah gambar enkripsi dengan baik, namun saat dilakukan proses dekripsi tidak akan menghasilkan hasil yang sama. Jadi, implementasi ini hanya dapat diterapkan jika orang yang ingin mendapatkan gambar yang ingin dienkripsi tidak bermasalah dengan dikirim dalam bentuk teks enkripsi ataupun jika orang tersebut hanya ingin enkripsi dilakukan pada *server-side* saja.

Penerapan enkripsi ini dalam *social messaging* dapat dilakukan dengan mengirim sebuah gambar yang sudah dienkripsi terlebih dahulu dengan sebuah kunci publik yang kemudian akan dikirim ke pemilik kunci pribadi yang bersesuaian dengan kunci publik tersebut. Saat pengiriman gambar ke server dari *social messaging* terbentuk teks enkripsi yang nantinya *social messaging* akan meminta penerima pesan untuk memasukan kunci pribadinya dan saat kunci pribadi sudah dimasukan maka server akan mendekripsi gambar yang dienkripsi oleh pengirim tersebut. Implementasi ini bekerja secara *server-side* dan pengguna hanya perlu memasukan kunci publik dan gambar saat mengirim gambar, dimana nantinya gambar tersebut akan langsung dihapus ketika sudah dienkripsi.

Alur enkripsi dan dekripsi = Gambar -> Teks Terenkripsi -> Gambar

3.2. Implementasi RSA dalam Gambar yang Dirubah ke Gambar Terenkripsi

Implementasi enkripsi RSA dalam gambar berwarna memerlukan sedikit penyesuaian pada algoritmanya. Penyesuaian yang dilakukan adalah sebagai berikut :

1. Hitung representasi warna RGB dari tiap piksel maksimal pada gambar yang ingin dienkripsi. Misal gambar A memiliki representasi warna RGB dari tiap pikselnya : 88-254-31, 22-31-25. Angka maksimal yang mewakilkan gambar tersebut adalah 254 sehingga cari dua bilangan prima yang menghasilkan perkalian lebih dari sama dengan

bilangan 254 tersebut namun kurang dari bilangan 255.

Pemodifikasian tersebut dilakukan jika kita hanya ingin mengirim gambar yang telah terenkripsi ke orang yang kita ingin kirim. Penyesuaian tersebut dilakukan karena perepresentasian warna dasar RGB memiliki angka maksimal 255 untuk setiap warna sehingga jika kita memilih bilangan a dan b yang melebihi 255 maka akan dihasilkan gambar enkripsi yang sesuai dengan perhitungan. Namun, saat gambar tersebut ingin didekripsi oleh penerima, penerima tidak menghasilkan gambar yang sama dengan gambar yang sebenarnya ingin dikirim oleh pengirim gambar. Contoh kasusnya adalah :

Tabel 2 merupakan hasil enkripsi yang diperoleh dari gambar 1. Saat tabel 2 ini dirubah menjadi gambar kembali maka akan terbentuk gambar berikut :

Gambar 3 Hasil gambar yang telah dienkripsi menggunakan algoritma RSA

Gambar di atas terbentuk karena pembulatan semua bilangan yang berada di atas angka 255 menjadi angka 255 sehingga teks enkripsi yang terbentuk pada tabel 2 berubah menjadi :

R	G	B
255	255	179
0	255	255
0	255	255
255	255	255

Tabel 3 Tabel yang merepresentasikan tiap piksel dari gambar yang terenkripsi

Gambar 3 hasil enkripsi tersebut saat akan didekripsi kembali akan menghasilkan gambar dari dekripsi teks pada tabel 3. Dekripsi dari gambar 3 akan menghasilkan tabel berikut :

R	G	B
P(255) = 384 = 255	P(255) = 384 = 255	P(179) = 4
P(0) = 0	P(255) = 384 = 255	P(255) = 384 = 255
P(0) = 0	P(255) = 384 = 255	P(255) = 384 = 255
P(255) = 384 = 255	P(255) = 384 = 255	P(255) = 384 = 255

Tabel 4 Tabel hasil dekripsi gambar 3

Gambar yang dihasilkan dari tabel 4 adalah gambar berikut :

Gambar 4 Gambar hasil dekripsi dari enkripsi gambar pada gambar 3

Gambar tersebut tidak sama dengan gambar 1 dimana gambar 1 ini merupakan gambar yang seharusnya didapat saat melakukan dekripsi. Sehingga terlihat bahwa algoritma RSA ini tidak dapat menghasilkan gambar yang sesuai jika kita memilih dua angka yang perkaliannya menghasilkan angka yang melebihi batas representasi warna RGB yaitu 255.

Contoh kasus dimana gambar terenkripsi dapat didekripsi kembali sehingga menghasilkan hasil yang sama dengan gambar sebelum dienkripsi adalah jika kita memilih 2 angka prima yaitu 23 dan 11 dan menghasilkan nilai n = 253 dan m = 22 * 10 = 220. Dimana nilai n ini tidak melebihi angka 255. Salah satu kunci publiknya adalah e = 7 dan kunci pribadinya adalah :

$7d \equiv 1 \pmod{220}$ dimana akan diperoleh k adalah 2 agar dapat menghasilkan bilangan bulat sehingga diambil d = 63.

Gambar berikut adalah gambar yang akan dienkripsi menggunakan kunci yang telah dibangkitkan sebelumnya. Gambar harus memenuhi prasyarat bahwa tiap piksel memiliki representasi warna RGB dimana angka dari tiap warna harus kurang dari nilai n yang dipilih. Sehingga gambar yang dapat dienkripsi sangatlah terbatas. Salah satu gambar yang dapat dienkripsi.

Gambar 5 Gambar yang dapat dienkripsi dengan n = 253

Gambar di atas akan terenkripsi menjadi sebuah gambar yang sebelumnya dapat terlebih dahulu di konversi menjadi sebuah teks representasi warna dasar RGB yang representasinya ada pada tabel berikut :

R	G	B
138	77	15
82	82	70
138	77	15
82	82	70

Tabel 5 Tabel representasi dari tiap piksel pada gambar 5

Tabel 5 tersebut akan dienkripsi menggunakan kunci publik e = 7 dan nilai n = 253 sehingga akan didapat representasi warna RGB dalam tabel berikut :

R	G	B
C(138) = 184	C(77) = 242	C(15) = 126
C(82) = 124	C(82) = 124	C(70) = 93
C(138) = 184	C(77) = 242	C(15) = 126
C(82) = 124	C(82) = 124	C(70) = 93

Tabel 6 Tabel enkripsi dari tabel 5

Tabel 6 di atas saat dirubah menjadi bentuk gambar terenkripsi akan menjadi gambar berikut :

Gambar 6 Gambar hasil enkripsi yang akan dikirim ke pemilik kunci pribadi

Saat gambar 6 di atas sampai ke penerima gambar 6 tersebut akan didekripsi dan kembali menjadi gambar 5. Berikut adalah proses pendekripsiannya.

1. Untuk tiap piksel, ambil representasi warna dasar RGBnya
2. Untuk setiap warna tersebut gunakan kunci pribadi untuk merubah representasi warna tersebut misal untuk paduan warna RGB pada piksel 1 yaitu 184-242-126 dengan mendekripsi paduan warna tersebut akan didapatkan paduan warna 138-77-15 yang menghasilkan paduan warna yang sama dengan piksel 1 pada gambar 5.
3. Ubah representasi dari tiap piksel tersebut untuk membangun sebuah gambar yang sama dengan gambar yang ingin dikirim.

Implementasi gambar terenkripsi ini dimana gambar terenkripsi ini akan dikirim ke pemilik kunci pribadi merupakan salah satu bentuk enkripsi pada *client-side* hal ini dikarenakan gambar yang terenkripsi tersebut dirubah terlebih dahulu kemudian baru dikirim melalui aplikasi *social messaging*. Setelah sampai ke penerima, gambar terenkripsi tersebut dirubah kembali menjadi gambar yang sama dengan yang ingin dikirim dengan melakukan proses dekripsi. Namun, implementasi ini memiliki banyak kekurangan yaitu bilangan yang dipilih harus menghasilkan perkalian yang kurang dari 255 dimana angka prima yang menghasilkan perkalian tersebut sangat terbatas sehingga keamanan dari implementasi ini sangat kurang karena dapat dilakukan pencarian secara menyeluruh dengan mencoba – coba menggunakan dua buah bilangan prima yang menghasilkan perkalian kurang dari 255 tersebut. Waktu yang dibutuhkan untuk mendekripsi gambar ini tanpa mengetahui kuncinya juga cepat dan hanya butuh waktu tidak lebih dari satu jam untuk mendapatkan kuncinya dengan benar.

Implementasi ini tidak mempunyai kelebihan yang signifikan hanya dapat mengurangi beban kerja dari *server-side*. Namun, hal itu tidak berpengaruh karena tidak ada artinya jika melakukan enkripsi gambar namun enkripsi itu dapat didekripsi dengan mudah tanpa mengetahui kunci rahasianya apalagi algoritma RSA memiliki kelemehan jika kunci yang dipilihnya tidak terlalu besar maka algoritma RSA ini dapat diterobos. Implementasi ini tidak cocok dilakukan dalam aplikasi *social messaging*.

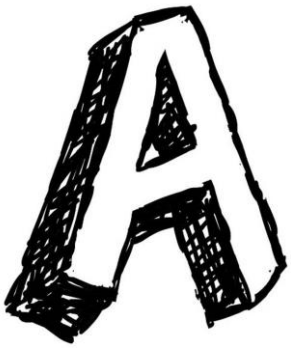
Alur enkripsi dan dekripsi = Gambar -> Gambar Terenkripsi -> Gambar

3.3. Perhatian Khusus

Waktu yang dibutuhkan untuk melakukan proses enkripsi dan dekripsi tergantung dari besar kunci yang digunakan. Jika diinginkan keamanan ekstra dalam pengiriman gambar tersebut tetapi tidak masalah dengan waktu penerimaan dari gambar tersebut maka tidak masalah untuk memilih lebih dari 3 digit kunci keamanan namun tidak melebihi 4 digit karena kompleksitas dari algoritma RSA untuk proses enkripsi dan dekripsinya adalah $O(N^3)$. Namun, jika pengguna hanya ingin mengamankannya dengan cukup, dapat dipilih 2 – 3 digit kunci dimana waktu mengenkripsi dan waktu untuk dekripsi tidak akan mengganggu kenyamanan dari pengguna.

IV. HASIL

Hasil dari penerapan implementasi pertama yaitu pengenkripsian gambar yang dirubah menjadi teks terenkripsi adalah sebagai berikut :



Gambar 7 Gambar yang ingin dienkripsi menggunakan Algoritma RSA dengan n = 667, e = 25, dan d = 345.

Gambar 8 Potongan gambar hasil enkripsi dari tiap representasi warna RGB tiap piksel

Gambar hasil dekripsi yang dihasilkan sama dengan Gambar 7.
Waktu yang dibutuhkan dalam proses enkripsi adalah : 24.4 detik
Waktu yang dibutuhkan dalam proses dekripsi adalah : 6.6 detik
Besar file gambar : 0.6 MB.
Analisa :
Perbedaan waktu tersebut dikarenakan untuk proses enkripsi dimana kompleksitasnya $O(N^3)$ diperlukan tambahan waktu untuk mengubah tiap piksel menjadi representasi angka yang

membutuhkan waktu O(X*Y) sehingga kompleksitas total dapat ditambahkan menjadi $O(N^3) + O(X*Y)$. Untuk proses dekripsi hanya dibutuhkan waktu $O(N^3)$ untuk N adalah perkalian dari dua bilangan prima yang dipilih dan X dan Y merupakan dimensi dari gambar tersebut. Selain itu ada juga faktor dimana pembacaan dari tiap piksel gambar tidak hanya membutuhkan waktu O(X*Y) tersebut karena ada faktor sistem yang mempengaruhi pembacaan tiap piksel dari gambar yang dibaca sehingga waktu yang dibutuhkan terkadang menjadi lebih tinggi.

V. KESIMPULAN

Penggunaan Algoritma RSA untuk mengenkripsi gambar pada *social messaging* merupakan fitur yang bagus dan tidak mudah untuk dibobol. Namun, implementasi enkripsi tersebut harus menggunakan implementasi berbasis teks agar dapat menghasilkan keamanan yang tidak lemah sehingga gambar yang dikirim tidak mudah untuk di dekripsi oleh peretas yang tidak mengetahui kunci rahasianya. Implementasi algoritma RSA ini memiliki sedikit kelemahan karena dapat memberatkan server jika banyak orang yang menggunakan fitur enkripsi tersebut secara bersamaan dengan kunci yang besar. Fitur enkripsi ini juga memerlukan kerjasama bagi penggunanya karena pengirim harus segera menghapus foto setelah mengirim foto tersebut dengan menggunakan kunci publik yang diterimanya dan penerima juga harus segera menghapus foto tersebut saat sudah menerima gambar dan sudah mengunduh ataupun mengamankan gambar tersebut.

VI. UCAPAN TERIMA KASIH

Saya mengucapkan terima kasih kepada Tuhan Yang Maha Esa, karena berkat rahmat-Nya makalah ini dapat diselesaikan dengan baik. Saya juga berterimakasih kepada Bapak Rinaldi Munir dan Ibu Harlili sebagai dosen mata kuliah IF2120 Matematika Diskrit yang sudah membimbing murid – muridnya. Saya juga mengucapkan terima kasih kepada keluarga saya yang sudah mendukung saya saat dalam kesulitan.

REFERENCES

[1] Rinaldi Munir, *Diktat Kuliah IF2120: Matematika Diskrit*. Bandung: Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung, 2006.

[2] Complexity of RSA Algorithm.
https://www.quora.com/What-is-the-complexity-of-RSA-cryptographic-algorithm
Waktu akses : 8 Desember 2016 pukul 22.07 WIB.

[3] RSA Algorithm.
http://www.di-mgt.com.au/rsa_alg.html.
Waktu akses : 8 Desember 2016 pukul 21.15 WIB.

[4] Kriptografi.
https://wendrydesyaputra.wordpress.com/2012/09/03/belajar-kriptografi-bag-1/
Waktu akses : 8 Desember 2016 pukul 20.15 WIB.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 8 Desember 2016

Agus Gunawan / 13515143