Aplikasi Kirim Pesan Berbasis Jaringan Lokal Dengan Menerapkan Algoritma RSA Sebagai Teknik Dalam Menjaga Kerahasiaan

Dony Catur Dermawan[1)], Triawan Adi Cahyanto[2)]

[1,2)]Jurusan Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Jember
Email : [1)] donycad2212@gmail.com, [2)] triawanac@unmuhjember.ac.id

ABSTRAK

Aplikasi kirim pesan bermanfaat untuk berkomunikasi dengan pengguna di area yang terbatas dengan koneksi internet. Permasalahan muncul ketika penggunaan aplikasi kirim pesan yang berjalan di jaringan lokal akan rentan terhadap serangan *Man In The Middle seperti sniffing*. Algoritma RSA merupakan salah satu algoritma kriptografi yang bersifat asimetris, yakni mempunyai kunci publik dan kunci pribadi. Penggunaan dua kunci ini merupakan teknik untuk menjaga kerahasiaan pesan yang berjalan di aplikasi kirim pesan. Pengujian tingkat keamanan terhadap aplikasi kirim pesan dilakukan dengan teknik *sniffing*. Hasil pengujian keamanan aplikasi, diperoleh data *ciphertext* dari proses enkripsi *plaintext* dengan algoritma RSA, namun kunci publik yang digunakan untuk melakukan enkripsi tidak berhasil dipecahkan sehingga tidak dapat melihat pesan yang dikirimkan pada aplikasi tersebut.

Kata Kunci : Kriptografi, *Local Area Network*, RSA, *Chatting*

1. PENDAHULUAN

Pertukaran pesan atau *chatting* merupakan layanan murah, cepat dan inovatif dalam rangka berbagi informasi antar pengguna di suatu tempat. Banyak aplikasi kirim pesan seperti *WhatsApp*, *Telegram*, dan *BBM* yang memberikan kemudahan dalam berinteraksi. Aplikasi kirim pesan tersebut tidak hanya bertukar pesan namun juga bertukar suara dan video. Mayoritas aplikasi kirim pesan membutuhkan koneksi internet untuk saling terhubung antar pengguna (Cahyanto, 2018). Untuk area yang tidak terjangkau akses internet, tentunya akan menyebabkan aplikasi tersebut menjadi tidak berguna (Cahyanto, Oktavianto, & Royan, 2013). Aplikasi kirim pesan berbasis jaringan lokal dibuat dengan tujuan sebagai layanan alternatif kirim pesan di area yang sulit terhadap akses internet. Aplikasi ini berjalan di jaringan lokal seperti kantor, kampus, kafe, dan tempat lainnya. Penggunaan aplikasi di jaringan tentunya akan menimbulkan dampak terhadap aplikasi tersebut, terutama dari sisi keamanan proses pengiriman pesan (Cahyanto & Prayudi, 2014). Oleh karena itu, supaya pesan yang dikirimkan oleh pengirim dapat terjaga kerahasiaannya, maka perlu dikembangkan aplikasi pesan dengan algoritma RSA sebagai mekanisme keamanan dalam pengiriman data. Algoritma RSA merupakan algoritma kriptografi yang bersifat asimetrik (Munir, 2010). Asimetrik yang dimaksud adalah terdapat dua kunci, yaitu kunci publik dan kunci pribadi. Kunci publik digunakan dalam proses enkripsi pesan, dan kunci publik ini umumnya dapat diketahui oleh publik beserta dengan data *ciphertext*. Sedangkan kunci pribadi digunakan dalam membuka *ciphertext* (pesan yang terenkripsi) dan kunci ini hanya dimiliki oleh pengirim dan penerima pesan (Xin

Zhou, Xiaofei Tang, Zhou, & Tang, 2011). Penggunaan faktorisasi bilangan prima sebagai proses pembentukan kunci dengan mode sistem *block cipher* membuat algoritma RSA sulit untuk dipecahkan dalam waktu yang singkat (Vikas, Agrawal, & Deshmukh, 2014).

2. TINJAUAN PUSTAKA

2.1 Jaringan Komputer

Jaringan komputer adalah kumpulan dari perangkat yang saling terhubung dalam satu kesatuan yang bekerja bersama-sama sehingga tercapai tujuan yang diharapkan (Cahyanto, 2011). Berdasarkan ruang lingkup, jaringan komputer dibagi menjadi tiga jenis sebagai berikut.

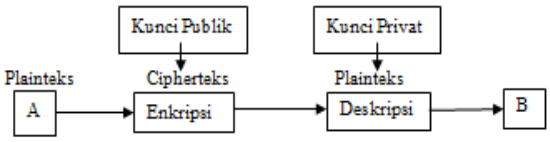
1) *Local Area Network (LAN)*
   LAN merupakan jaringan yang mempunyai ruang lingkup terbatas di suatu area tertentu, seperti kantor, kampus, dan pabrik.
2) *Metropolitan Area Network (MAN)*
   MAN merupakan jaringan yang ruang lingkupnya lebih jauh dibandingkan dengan LAN. Model jaringan seperti ini umumnya digunakan di perkotaan, seperti pada penerapan *smart city* di kota besar.
3) *Wide Area Network (WAN)*
   WAN merupakan jaringan yang ruang lingkupnya mencakup daerah geografis seperti negara atau benua.

2.2 Kriptografi

Kriptografi adalah suatu bidang ilmu atau seni yang mempelajari tentang mekanisme dalam menjaga kerahasiaan pesan (Nagar & Alshamma, 2012). Dalam menjaga kerahasiaan pesan, pesan tersebut akan diubah menjadi data acak atau data yang disandi sehingga hanya pengguna yang memiliki akses terhadap pesan tersebut yang dapat membaca isi pesan (Munir, 2008).

2.3 Algoritma RSA

Algoritma RSA ditemukan oleh peneliti dari MIT (*Massachusetts Institute of Technology*) yaitu Ronald L. Rivest, Adi Shamir, Leonard Adleman. Sebagai algoritma kunci publik, RSA mempunyai dua kunci, yaitu kunci publik dan kunci pribadi (Preetha & Nithya, 2013). Kunci publik boleh diketahui oleh siapa saja dan digunakan untuk proses enkripsi. Sedangkan kunci pribadi hanya pihak-pihak tertentu saja yang boleh mengetahuinya dan digunakan untuk proses dekripsi. Keamanan sandi RSA (*Rivest-Shamir-Adleman*) terletak pada sulitnya memfaktorkan bilangan yang besar. Sampai saat ini RSA (Rivest-Shamir-Adleman) masih handal dan digunakan secara luas (Jaiswal, 2014).



Gambar 1 Skema Algoritma Kunci Publik

Besaran-besaran yang digunakan pada algoritma RSA (*Rivest-Shamir-Adleman*) :

1. p dan q bilangan prima (rahasia)
2. n = p.q (tidak rahasia)
3. ɸ(n) = (p-1)(q-1) (rahasia)
4. e (kunci enkripsi) (tidak rahasia)
5. d (kunci deskripsi) (rahasia)
6. m (plainteks) (rahasia)
7. c (cipherteks) (tidak rahasia)

Secara garis besar, proses kriptografi pada algoritma RSA (*Rivest-Shamir-Adleman*) terdiri dari tiga tahapan (Padmavathi & Kumari, 2013).

1) Pembangkitan Kunci
   Untuk membangkitkan kedua kunci, dipilih dua buah bilangan prima yang besar, misalkan p dan q. Untuk mendapatkan keamanan maksimum, pilih dua bilangan p dan q yang besar.

p-ISSN : 2502-5724; e-ISSN : 2541-5735

Kemudian dihitung :

$n = p*q$ (1)

$\Phi(n) = (p-1)(q-1)$ (2)

Lalu dipilih kunci enkripsi secara acak, sedemikian sehingga e dan (p-1) (q-1) relatif prima. Artinya e dan φ tidak memiliki faktor persekutuan bersama. Kemudian dengan algoritma *Euclidean* yang diperluas lalu dihitung kunci dekripsi d, sedemikian sehingga :

$ed = 1 \bmod (p-1)(q-1)$ (3)

atau

$ed - 1 = k(p-1)(q-1)$ (4)

Dimana k adalah konstanta integer. Perhatikan bahwa d dan n juga relatif prima. Bilangan e dan n merupakan kunci publik, sedangkan d merupakan kunci pribadi. Dua bilangan prima p dan q tidak diperlukan lagi. Namun p dan q kadang diperlukan untuk mempercepat perhitungan dekripsi.

2) Proses Enkripsi

Untuk mengenkripsi pesan m, terlebih dahulu pesan dibagi kedalam blok-blok numerik yang lebih kecil dari n (dengan data *biner*, dipilih pangkat terbesar dari 2 yang kurang dari n). Jadi jika p dan q bilangan prima 100 digit, maka n akan memiliki sekitar 200 buah digit dari setiap blok pesan m, seharusnya kurang dari 200 digit panjangnya. Pesan yang terenkripsi (c), akan tersusun dari blok-blok (c) yang hampir sama panjangnya. Rumus enkripsinya adalah :

$c = m^e \bmod n$ (5)

Dimana :

m = pesan asli
e = proses enkripsi
c = pesan dalam bahasa sandi
n = modulus

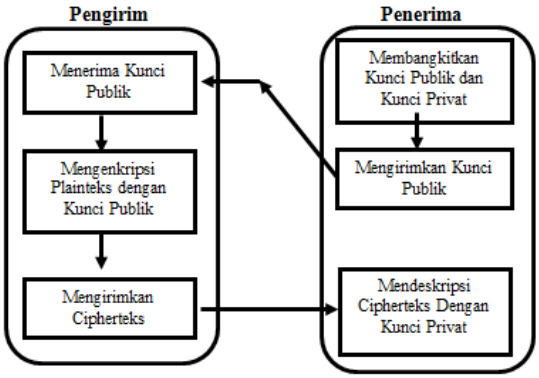
3) Proses Dekripsi

Setelah menerima pesan yang sudah terenkripsi, maka penerima pesan akan melakukan proses dekripsi pesan dengan cara :

$m = c^d \bmod n$ (6)

Dimana :

m = pesan asli
d = proses dekripsi
c = pesan dalam bahasa sandi
n = modulus



Gambar 2 Proses Algoritma RSA

Gambar 2 diatas menjelaskan tentang proses pembentukan algoritma RSA (*Rivest-Shamir-Adleman*). Algoritma RSA (*Rivest-Shamir-Adleman*) tersebut dijalankan dengan membangkitkan kunci publik dan kunci pribadi. Penerima kemudian mengirimkan kunci publik kepada pengirim untuk mengenkripsi pesan. Setelah pesan terenkripsi dalam bentuk *ciphertext* maka dapat dikirimkan kembali ke penerima. Setelah pihak penerima mendapatkan pesan tersebut maka dapat kembali didekripsi dengan kunci pribadinya.

Contoh proses Algoritma RSA (*Rivest-Shamir-Adleman*) dapat dijalankan pada tahapan-tahapan sebagai berikut ini.

plaintext = *m* = CINTA MATI

m diubah ke dalam bentuk ASCII = 67737884653277658473

Pembentukan kunci :

i) Misalkan p = 47 dan q= 71 (p dan q harus bilangan prima) dimana p dan q dipilih secara acak.

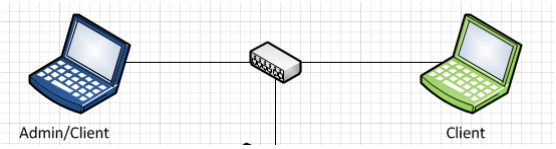
ii) Hitung: n = p*q = 47 * 71 = 3337
iii) Maka Φ (n) = (p-1) (q-1) = 3220
iv) Pilih kunci publik e = 79 (relatif prima terhadap 3220 karena pembagi terbesar bersamanya adalah 1).
v) Didapat kunci pribadi sebagai berikut :
e.d = 1 (mod ф (n))
d = 1 (mod ф (n)) = 1019
vi) Maka kunci publik dan kunci pribadi adalah
Kunci publik = (e, n) = (79,3337)
Kunci pribadi = (d, n) = (1019,3337)

Ubah *plainteks* menjadi *ciphertext* dengan kunci publik :
$c_1 = m_1^e$ *mod* n = 67779 *mod* 3337 = 231
$c_2 = m_2^e$ *mod* n = 37879 *mod* 3337= 3092
$c_3 = m_3^e$ *mod* n = 84679 *mod* 3337 = 470
$c_4 = m_4^e$ *mod* n = 53279 *mod* 3337 = 407
$c_5 = m_5^e$ *mod* n = 77679 *mod* 3337 = 14
$c_6 = m_6^e$ *mod* n = 58479 *mod* 3337= 2842
$c_7 = m_7^e$ *mod* n = 07379 *mod* 3337 = 725
Jadi *ciphertext* yang dihasilkan adalah
231 3092 470 407 14 2842 725

Ubahlah *ciphertext* menggunakan kunci pribadi :
$m_1 = c_1^d$ *mod* n=2311019 *mod* 3337=677
$m_2 = c_2^d$ *mod* n=30921019 *mod* 3337=378
$m_3 = c_3^d$ *mod* n=4701019 *mod* 3337=846
$m_4 = c_4^d$ *mod* n=4071019 *mod* 3337=532
$m_5 = c_5^d$ *mod* n=141019 *mod* 3337=776
$m_6 = c_6^d$ *mod* n=28421019 *mod* 3337=584
$m_7 = c_7^d$ *mod* n= 28421019 *mod* 3337=73
Jadi *plaintext* yang dihasilkan :
677 378 846 532 776 584 73 = CINTA MATI

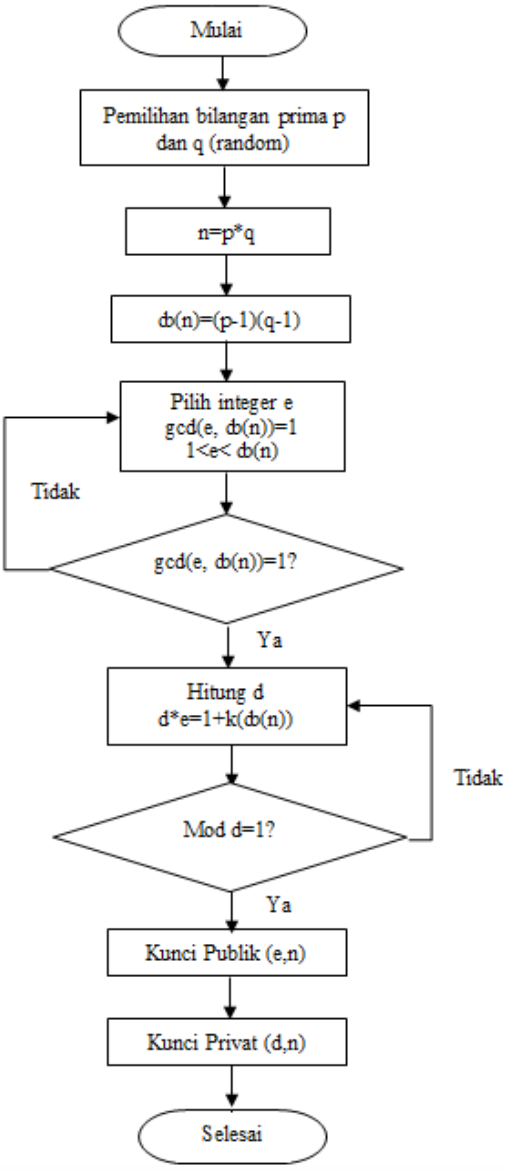
3. METODE PENELITIAN

Sistem menggunakan satu *admin* dan dua *client* yang dibangun dalam jaringan *localhost*, dan kemudian akan diberikan sistem keamanan dengan menerapkan algoritma RSA (*Rivest-Shamir-Adleman*) dalam proses enkripsi dan deskripsi pesan.



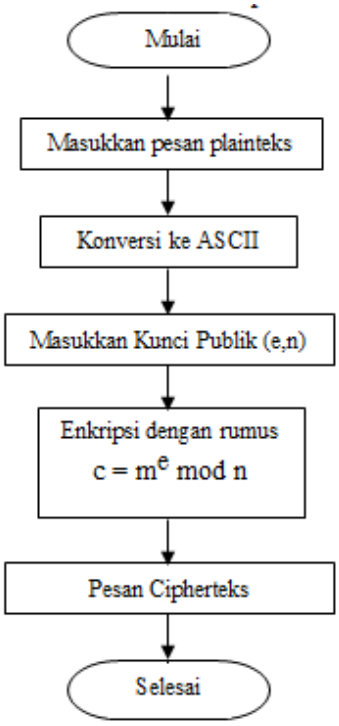
Gambar 3 Jaringan lokal

Flowchart pada Algoritma RSA (*Rivest-Shamir-Adleman*) disusun sesuai tahapan proses terdiri dari Setup Key, Proses Enkripsi, dan Proses Dekripsi ditunjukkan dalam Gambar 4, 5, dan 6.

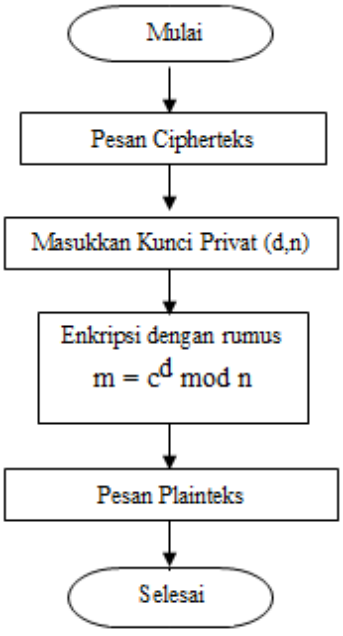


Gambar 4 *Flowchart Setup Key*

p-ISSN : 2502-5724; e-ISSN : 2541-5735



Gambar 5 Flowchart Proses Enkripsi

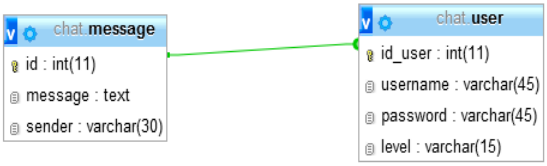


Gambar 6 Flowchart Proses Deskripsi

4. HASIL DAN PEMBAHASAN

Algoritma RSA diterapkan pada perangkat lunak yang dapat melakukan proses enkripsi dan dekripsi pesan sehingga dapat mengamankan data "*chat*" antar pengguna.

Dalam proses pengiriman pesan dimasukkan dalam *database mysql* yang ditampilkan dalam desain ERD (Entity Relationship Diagram) di bawah ini.



Gambar 7 ERD Pengiriman Pesan

Perangkat lunak yang dihasilkan terdiri dari lima halaman yaitu halaman utama, halaman *admin*, halaman penerima, halaman pengirim, dan halaman penyerang yang masing-masing halaman dapat dilihat pada Gambar 8-13.

4.1 Halaman Utama

Gambar 8 halaman utama terdapat *form login* dengan tiga jenis hak akses pengguna yaitu admin, pengirim, dan penerima. Untuk menampilkan halaman penerima maka masukkan *username* penerima sehingga menuju ke halaman penerima.php. Dengan cara sama untuk menampilkan halaman admin, pengirim.

Gambar 8 Halaman Utama

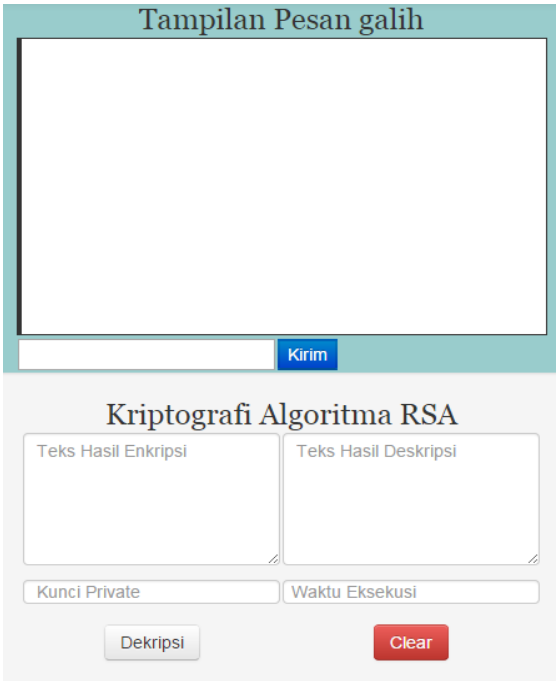
4.2 Halaman Utama

Halaman admin adalah halaman yang digunakan untuk melakukan semua proses dalam aplikasi. Termasuk melakukan pengujian jalannya aplikasi yang akan disimulasikan. Halaman ini dapat dilihat pada Gambar 9.

Gambar 9 Halaman Admin

4.3 Halaman Penerima

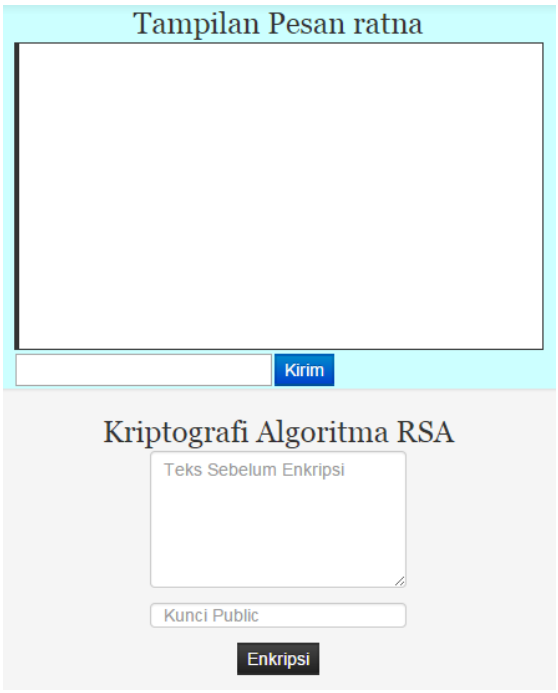
Pada halaman ini terdapat tampilan pesan, dan hasil pesan yang diterima akan di dekripsi menjadi pesan asli. Penerima juga bisa mengirimkan pesan asli sehingga dapat diketahui oleh pengirim dan penyerang. Halaman penerima ini dilengkapi waktu eksekusi untuk mendeskripsikan karakter. Halaman ini dapat dilihat pada Gambar 10.



Gambar 10 Halaman Penerima

4.4 Halaman Pengirim

Halaman ini dapat mengirimkan pesan asli dan pesan yang telah di enkripsi (*ciphertext*). Pengirim merupakan pengirim pesan yang ditujukan pada penerima. Oleh karena itu pengirim hanya bisa mengenkripsi data menjadi *ciphertext* dan hanya penerima yang akan mendeskripsikannya menjadi pesan asli kembali. Halaman ini dapat dilihat pada Gambar 11.



Gambar 11 Halaman Pengirim

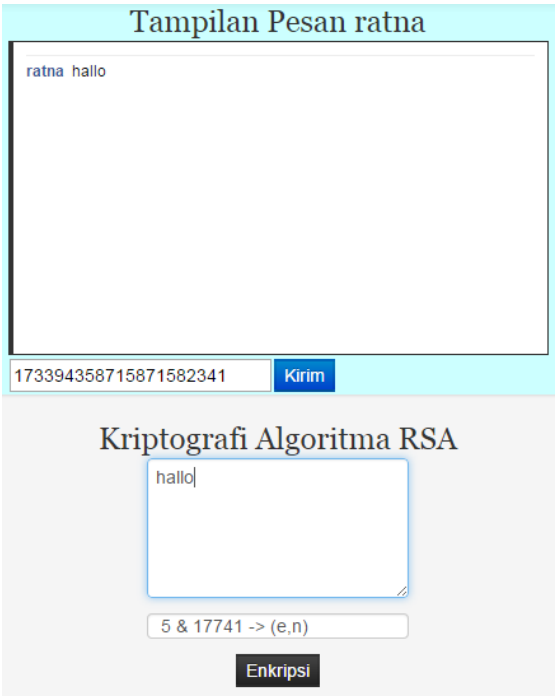
4.5 Pengiriman Pesan Terenkripsi

Setelah melakukan pengiriman pesan yang dapat disadap oleh penyerang (*attacker*), kemudian dibuat pengiriman pesan yang terenkripsi. Simulasi pesan ini akan dijalankan pada proses enkripsi pesan yang akan dikirimkan. Saat pesan diterima, maka pihak penerima dapat mendeskripsikan pesan tersebut sehingga pesan tetap terjamin keasliannya. Dalam aplikasi ini juga diasumsikan bahwa *attacker* dapat menyerang aplikasi tetapi hal tersebut dapat diatasi setelah proses

p-ISSN : 2502-5724; e-ISSN : 2541-5735

dekripsi sehingga pesan yang dikirimkan *attacker* tidak akan terbaca.

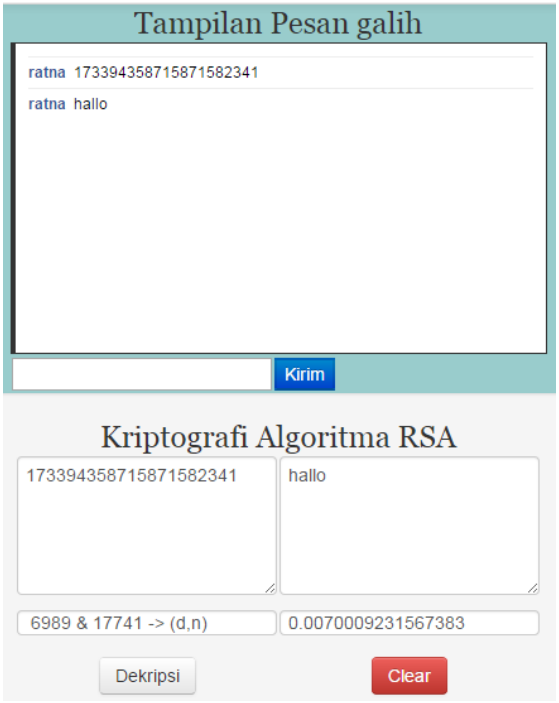
Pada Gambar 12 terjadi proses enkripsi pesan oleh pengirim "ratna" dan kemudian dikirimkan kepada penerima "galih". Pesan yang terenkripsi akan berupa angka yang telah dibentuk menggunakan algoritma RSA (*Rivest-Shamir-Adleman*).



Gambar 12 Pengiriman Pesan Terenkripsi

Pada Gambar 13 ditampilkan bahwa pesan sudah terkirim dalam bentuk enkripsi. Selanjutnya "galih" dapat mendeskripsikan menggunakan kunci pribadinya. Pesan yang terenkripsi dapat dideskripsikan dengan kunci pribadi sehingga pesan *ciphertext* dapat didefinisikan kembali dan pesan asli tidak dapat diketahui *attacker*. Pesan *ciphertext* didekripsi juga dengan algoritma RSA.

Pada saat pengirimanan pesan ada proses verifikasi pesan *ciphertext* dan pesan asli sehingga pesan yang dikirimkan kepada "galih" benar-benar asli dari "ratna". Saat *attacker* menyerang dengan data asli pada pesan galih maka hal tersebut tidak dapat didekripsi.



Gambar 13 Mendeskripsikan Pesan Terkirim

5. KESIMPULAN DAN SARAN

Sistem keamanan membantu dalam mengamankan jaringan, mengantisipasi saat jaringan berhasil ditembus. Pada proses *spoofing* yaitu *man the middle attack* sangat mungkin untuk menjebol informasi pesan yang dikirimkan lewat jaringan. Dalam sistem enkripsi kunci publik-pribadi, yang memegang peranan dalam menjebol kunci pribadi adalah kesulitan mencari faktor prima bilangan yang besar. Pesan yang dikirimkan melalui jaringan lokal sulit didekripsi jika tidak mengetahui kunci pribadi. Pada proses enkripsi meskipun kunci publik diketahui orang lain, tetapi pihak *attacker* tidak bisa mengenkripsi pesan aslinya.

Penerapan protokol *ssl* memperkuat keamanan di level layer 3 dan 4 sehingga proses pengiriman data dapat terenkripsi, dan modifikasi dengan improvisasi random karakter pada algoritma RSA.

DAFTAR PUSTAKA

Cahyanto, T. A. (2011). Analisis deteksi penyusupan pada jaringan komputer menggunakan snort (studi kasus pada Dinas Pariwisata Propinsi Daerah Istimewa Yogyakarta). Yogyakarta. Retrieved from http://search.jogjalib.com/Record/uinsukalib-073461#details

Cahyanto, T. A. (2018). Implementasi Smart Router Berbasis OpenWRT Sebagai Media Untuk File Sharing dan Chatting Pada Laboratorium Terpadu Unmuh Jember. https://doi.org/10.17605/OSF.IO/P6BWS

Cahyanto, T. A., Oktavianto, H., & Royan, A. W. (2013). Analisis dan Implementasi Honeypot Menggunakan Dionaea Sebagai Penunjang Keamanan Jaringan. JUSTINDO (Jurnal Sistem Dan Teknologi Informasi Indonesia), 1(2), 86–92.

Cahyanto, T. A., & Prayudi, Y. (2014). Investigasi Forensika Pada Log Web Server untuk Menemukan Bukti Digital Terkait dengan Serangan Menggunakan Metode Hidden Markov Models. SNATi, 15–19.

Jaiswal, R. J. (2014). Reformed RSA algorithm based on Prime Number. International Journal of Computer Applications, 975–8887.

Munir, I. R. (2008). Pengantar Kriptografi, 16. Retrieved from http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/Pengantar Kriptografi.pdf

Munir, I. R. (2010). Algoritma RSA dan ElGamal, 13. Retrieved from http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/Algoritma RSA.pdf

Nagar, S. A., & Alshamma, S. (2012). High speed implementation of RSA algorithm with modified keys exchange. In 2012 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications, SETIT 2012 (pp. 639–642). https://doi.org/10.1109/SETIT.2012.6481987

Padmavathi, B., & Kumari, S. R. (2013). A Survey on Performance Analysis of DES , AES and RSA Algorithm along with LSB Substitution Technique. International Journal of Science and Research (IJSR), 2(4), 170–174. Retrieved from https://www.ijsr.net/archive/v2i4/IJSRON120134.pdf

Preetha, M., & Nithya, M. (2013). A STUDY AND PERFORMANCE ANALYSIS OF RSA ALGORITHM. IJCSMC, 2(6), 126–139. Retrieved from http://www.ijcsmc.com/docs/papers/June2013/V2I6201330.pdf

Vikas, A., Agrawal, S., & Deshmukh, R. (2014). Analysis and Review of Encryption and Decryption for Secure Communication. International Journal of Scientific Engineering and Research (IJSER), 2(2), 2–4. Retrieved from http://www.ijser.in/archives/v2i2/SjIwMTMxMTU=.pdf

Xin Zhou, Xiaofei Tang, Zhou, X., & Tang, X. (2011). Research and Implementation of RSA Algorithm for Encryption and Decryption. Proceedings of 2011 6th International Forum on Strategic Technology, 1118–1121. https://doi.org/10.1109/IFOST.2011.6021216