Studi dan Implementasi *Secure Chatting* Menggunakan Algoritma RSA

Panji Andhika Pratomo[1)], Taufik[2)], Keni Puspita Sari[3)], Rina Wati[4)]
STMIK Pringsewu Lampung
Jl. Wisma Rini No. 9, Pringsewu, Lampung Telp.(0729) 22240
e-mail: taufiktsani@gmail.com[1)], panjiandhikap@gmail.com[2)],
keypuspita13@gmail.com[3)], rinastmik12@gmail.com[4)]

Abstrak

Perkembangan teknologi jaringan komputer yang kian pesat mengakibatkan munculnya perubahan cara-cara berkomunikasi konvensional. Terdapat banyak aplikasi yang berguna untuk berkomunikasi antar satu orang dan orang lain seperti instant messaging, interactive games, video conference dsb. Jaringan internet adalah suatu sistem jaringan komunikasi berskala besar antar komputer yang sifatnya terbuka, dimana jaringan internet dapat menghubungkan segala platform sistem operasi agar dapat berkomunikasi dan informasi yang lewat didalamnya dapat mudah disadap serta diawasi. Terutama penggunaan aplikasi messenger yang membuat komunikasi antara dua pihak dapat dilakukan secara langsung menggunakan teks yang diketik (chatting). Aplikasi messenger bekerja dengan mengirimkan teks melalui perangkat yang terhubung dengan suatu jaringan. Penggunaan teknologi ini memiliki kelebihan dibandingkan surat elektronik, yaitu komunikasi dapat terjalin real-time. Hal tersebut merupakan salah satu penyebab pertumbuhan yang pesat pada jumlah pemakai aplikasi messenger untuk berkomunikasi. Umumnya aplikasi messenger yang tersedia saat ini mengabaikan faktor keamanan di dalam komunikasi data. Oleh karena itu, user dihadapkan pada resiko terancamnya hak privacy atas informasi yang mereka miliki. Maka diperlukan suatu cara untuk mengamankan data informasi yang akan melewati suatu jaringan komputer yaitu dengan menerapkan enkripsi. Enkripsi adalah proses mengamankan suatu informasi dengan membuat informasi tersebut tidak dapat dibaca tanpa bantuan pengetahuan khusus. RSA adalah sebuah algoritma enkripsi kunci-publik dengan keamanan yang tinggi. Keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor prima. Keamanan dari algoritma RSA adalah didasarkan pada sulitnya memfaktorkan bilangan besar menjadi fakor-faktor primanya.

***Kata kunci**: Chat, RSA, Algoritma, Enkripsi*

1. Pendahuluan

Perkembangan teknologi jaringan komputer yang kian pesat di dalam dunia telekomunikasi mengakibatkan munculnya perubahan cara-cara berkomunikasi konvensional secara langsung antar individu. Diantara pengguna fasilitas internet terdapat banyak aplikasi yang berguna untuk berkomunikasi antar satu orang dan orang lain seperti *instant messaging*, *interactive games, video confrence* dan masih banyak lainnya. Jaringan internet adalah suatu sistem jaringan komunikasi berskala besar antar komputer yang sifatnya terbuka, dimana jaringan internet dapat menghubungkan segala *platform* sistem operasi agar dapat berkomunikasi dan informasi yang lewatdidalamnya dapat mudah disadap serta diawasi. Terutama penggunaan aplikasi *messenger* yang membuat komunikasi antara dua pihak dapat dilakukan secara langsung menggunakan teks yang diketik (*chatting*). Aplikasi *messenger* bekerja dengan mengirimkan teks melalui perangkat yang terhubung dengan suatu jaringan. Penggunaan teknologi ini memiliki kelebihan dibandingkan surat elektronik (*e-mail*), yaitu komunikasi dapat terjalin secara langsung atau *real- time* [4]. Hal tersebut merupakan salah satu penyebab pertumbuhan yang pesat pada jumlah pemakai aplikasi *messenger* untuk berkomunikasi. Umumnya aplikasi *messenger* yang tersedia saat ini mengabaikan faktorkeamanan di dalam komunikasi data. Oleh karena itu, *user* dihadapkan pada resiko terancamnya hak *privacy* atas informasi yang mereka miliki. Maka diperlukan suatu cara untuk

mengamankan data informasi yang akan melewati suatu jaringan komputer yaitu dengan menerapkan enkripsi.

Pada tahun 2011 masyarakat Indonesia dihebohkan dengan kasus tindak pidana korupsi yang dilakukan salah satu oknum artis yang beralih profesi menjadi politisi disuatu partai besar. Kasus itu mulai terbongkar akibat peran aktif penegak hukum dalam mengawasi serta menyadap *gadget* tersangka melalui sebuah *messenger*. Kini riwayat percakapan melalui *messenger* tersebut menjadi salah satu bukti untuk menuntaskan kasus mega korupsi yang ada di Indonesia. Hal ini merupakan salah satu bentuk penyadapan yang bersifat positif, akan tetapi dengan adanya peristiwa ini juga menunjukan bahwa tingkat keamanan *messenger* yang lemah membuat ancaman terhadap hak privasi dari seseorang.

Enkripsi adalah proses mengamankan suatu informasi dengan membuat informasi tersebut tidak dapat dibaca tanpa bantuan pengetahuan khusus dengan tujuan kerahasiaan, integritas data, autentikasi, non-repudiasi. Penerapan enkripsi pada data informasi diharapkan dapat meningkatkan keamanan dalam berkomunikasi antar dua user di dalam jaringan internet. Sehingga pihak yang berkeinginan untuk memonitor dan mengawasi data informasi tidak mampu membaca ataupun menterjemahkan isi dari informasi yang di dapatkan.

RSA adalah sebuah algoritma enkripsi kunci-publik yang dikembangkan oleh 3 orang peneliti MIT pada tahun 1976. Keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor prima . Keamanan dari algoritma RSA adalah didasari oleh dua problem matematika yaitu problem dalam faktorisasi bilangan berjumlah banyak dan mencari modulo akar akan en dari sebuah bilangan komposit *N* yang faktornya tidak diketahui. Setelah menganalisa permasalahan diatas, maka penulis merumuskan permasalahan bagaimana mengimplementasikan algoritma RSA pada Instant Messenger untuk mengenkripsi pesan yang dikirim.dan menguji performa sistem enkripsi yang telah diimplementasikan pada instant messenger. Tujuan dari penelitian ini adalah untuk mengimplementasikan algoritma RSA pada perangkat lunak yang dibuat agar dapat menambah keamanan dalam berkomunikasi dan Untuk menunjukan performa instant messenger dalam proteksi pengiriman pesan..

A Secured Chat System with Authentication as RSA Digital Signature. Banyaknya sistem instant messaging dari berbagai macam vendor menyebabkan kurangnya perhatian akan jaminan keamanan yang pada akhirnya mengarahkan pengguna kepada ancaman sekuritas. Penelitian ini menggunakan tanda tangan digital dengan menerapkan algoritma RSA yang digunakan untuk mengamankan jendela percakapan, serta memastikan keamanan pada saat pengiriman pesan pribadi ke pengguna lain dari sistem *instant messaging* [1].

Aplikasi online *secure chatting* dengan menerapkan enkripsi data menggunakan algoritma *discrete chaotic map* dimana sebuah kunci digunakan untuk proses enkripsi dan dekripsi pesan. Algoritma *discrete chatoic map* termasuk dalam kategori algoritma simetrik, yaitu algoritma yang menggunakan kunci yang sama untuk proses enkripsi dan dekripsi pesan. Penggunaan Algoritma RSA untuk Keamanan Transaksi Online Berbasis Aplikasi Mobile. Algoritma RSA memungkinkan untuk diimplementasikan pada data transaksi dalam aplikasi mobile. RSA membuktikan bahwa semakin besar data terenkripsi, maka semakain lama juga waktu yang diperlukan untuk mendekripsinya. Metode ini juga membuktikan semakin panjang public key dan private key semakin lama prosesnya. Berdasarkan penelitian-penelitian sebelumnya menujukan bahwa algoritma RSA banyak digunakan dalam berbagai macam perangkat karena keamanannya, untuk itu penulis akan membuat sistem berupa instant messenger yang memiliki proteksi pengiriman pesan menggunakan algoritma RSA, dimana sistem akan dibuat dengan metode pengembangan berfase [3] [2].

Kriptografi adalah cara-cara unik setiap pelaku kriptografi untuk menulis pesan rahasia yang mempunyai nilai estetika tersendiri yang lebih dari sekedar privasi. Ada beberapa tujuan dari kriptografi antara lain [5]:

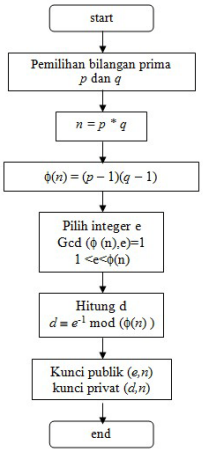
1. Kerahasiaan (*confidentiality*), adalah layanan yang ditujukan untuk menjaga agar pesan tidak dapat dibaca oleh pihak-pihak yang tidak berhak.

2. Integritas data (*data integrity*), adalah layanan yang menjamin bahwa pesan masih asli atau utuh belum pernah dimanipulasi selama pengiriman.
3. Otentikasi (*authentication*), adalah layanan yang berhubungan dengan identifikasi baik mengindentifikasi kebenaran pihak-pihak yang berkomunikasi (*user authentication* atau *entity authentication*) maupun mengindentifikasi kebenaran sumber pesan (*data origin authentication*).
4. Nirpenyangkalan (*non-repudition*), adalah layanan untuk mencegah entitas yang berkomunikasi melakukan penyangkalan, yaitu pengirim pesan menyangkal melakukan pengiriman atau menrima pesan menyangkal telah menerima pesan

Kunci pada RSA mencakup dua buah kunci, yaitu public key dan private key. Public key digunakan untuk melakukan enkripsi yang dapat diketahui oleh orang lain. Sedangkan private key tetap dirahasiakan dan digunakan untuk melakukan dekripsi.
Pembangkitan kunci atau *key generation* dari RSA adalah sebagai berikut :

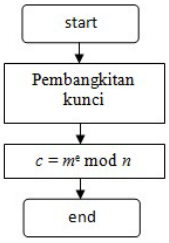
1. Pilih dua bilangan prima, p dan q (rahasia)
2. Hitung $n = p*q$. Besaran n tidak perlu dirahasiakan.
3. Hitung $(n) = (p-1)(q-1)$.
4. Pilih sebuah bilangan bulat untuk kunci publik(e), yang relatif prima terhadap (n) .
5. Hitung kunci dekripsi, d, melalui $ed \equiv 1 \pmod{m}$ atau $d \equiv e^{-1} \bmod ((n))$
6. Kunci publik (e,n) dan kunci privat (d,n)

Gambar 1 di bawah ini adalah *flowchart* untuk membangkitkan kunci publik dan kunci privat pada algoritma RSA:



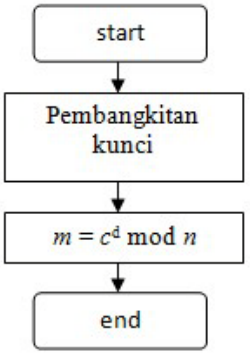
Gambar 1. Pembangkitan Kunci RSA

Gambar 2 berikut ini adalah proses enkripsi RSA yang dilakukan oleh pihak pengirim:



Gambar 2. Pembangkitan Kunci RSA

Gambar 3 berikut ini adalah proses dekripsi RSA yang dilakukan oleh pihak penerima pesan:



Gambar 3. Pembangkitan Kunci RSA

Keamanan dari algoritma RSA adalah didasari oleh dua *problem* matematika yaitu:

1. Problem dalam faktorisasi bilangan berjumlah banyak
2. Mencari modulo akar akan e^n dari sebuah bilangan komposit *N* yang faktornya tidak diketahui.

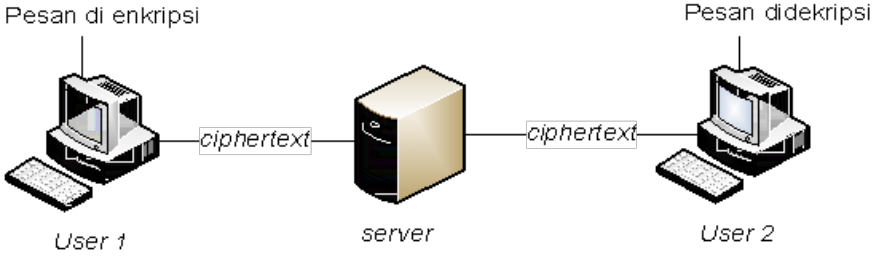
2. Metode Penelitian

2.1. Objek Penelitian

Objek penelitian ini adalah beberapa *instant messenger* terutama dalam penggunaan fasilitas *chatting*. Penggunaan fasilitas *chatting* yang bersifat *real-time* menyebabkan *instant messenger* berkembang dengan cepat. Sebagaimana dijelaskan dilatar belakang masalah jaringan komunikasi dapat menghubungkan berbagai macam *platform* yang menyebabkan suatu informasi dapat disadap dan diawasi, karena hal tersebut bebarapa provider penyedia layanan *instant messenger* kurang memperhatikan proteksi pengamanan data teks dalam komunikasi antar *user*. Untuk itu pada penelitian ini akan dibangun sebuah perangkat lunak dalam bahasa pemrograman *Java* yang memiliki proteksi pengiriman pesan menggunakan algoritma RSA agar dapat menambah keamanan dalam berkomunikasi.

2.2. Arsitektur Sistem

Perangkat lunak yang dibuat menggunakan Bahasa *Java*, yang terdiri atas *server* dan *client*, dimana *server* berguna untuk menanggapi *request* dari *client* sedangkan *client* berguna untuk merequest data dari *server*. Proses pengiriman pesan yang terproteksi pada sistem yang akan dibuat dapat dilihat pada gambar 4 berikut ini:



Gambar 4. Arsitektur Sistem

2.3. Format Pesan

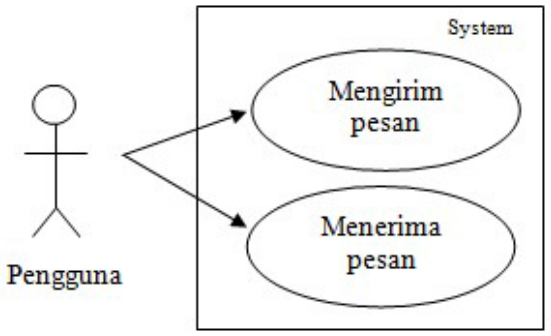
Tabel 1 berikut adalah tabel format pesan yang menunjukkan standar pengiriman sebuah teks dari satu *user* ke *user* lain:

Tabel 1. Format Pesan

No	Format	Event	Keterangan
1	HP | Message	-	Format standar dalam pengiriman pesan
2	Login | *Username* | Password	*Login*	Format pesan untuk *login.*
3	Logout | Username	*Logout*	Format pesan untuk *logout.*
4	Messageto | Tujuan | Pesan	Kirim Pesan	Format pesan ketika pesan dikirim dalam bentuk *chiphertext*
5	messageFrom | Tujuan | Pesan	Pesan diterima	Format pesan ketika pesan telah sampai di tujuan.
6	*Handshaking* | Tujuan | Tukar *Key*	*Handshaking*	Format pesan ketika *user* akan melakukan *chatting.*
7	UserExist | username	-	Format pesan untuk memberitahukan bahwa username sudah tersedia
8	UserNo | username	-	Format pesan untuk memberitahukan bahwa username belum ada
9	Welcome | Pesan	-	Format pesan ketika login berhasil
10	NewUser | Username	-	Format pesan untuk user yang baru masuk
11	UserList | UsernameList	-	Format pesan untuk memberitahukan daftar teman

2.4. Use Case

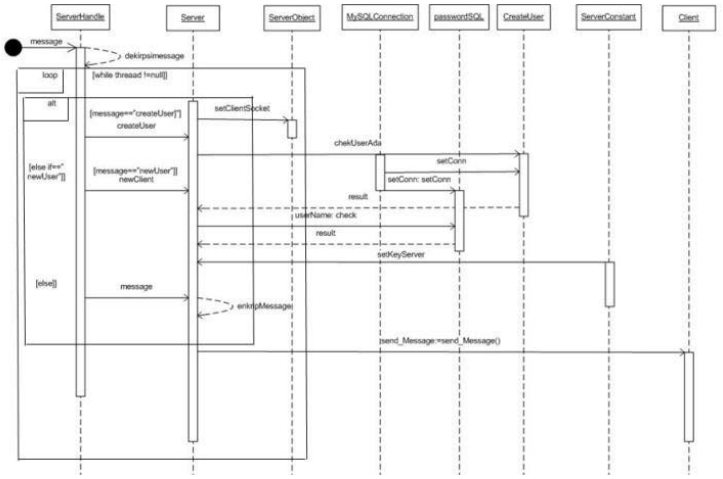
Diagram use case pada sistem ini terdiri dari satu actor yaitu pengguna messenger yang dapat mengirim pesan maupun menerima pesan dan dua use case yaitu use case mengirim pesan yang berguna untuk menangani aksi yang berhubungan dengan mengirim pesan, dimana pesan yang akan dikirim sebelumnya akan dienkripsi dan use case menerima pesan yang berguna untuk mengani aksi yang berhubungan dengan menerima pesan, dimana sebelum pesan diterima maka akan dilakukan proses dekrispi pesan, seperti tampilan gambar 5 berikut:



Gambar 5. *Use Case* Sistem

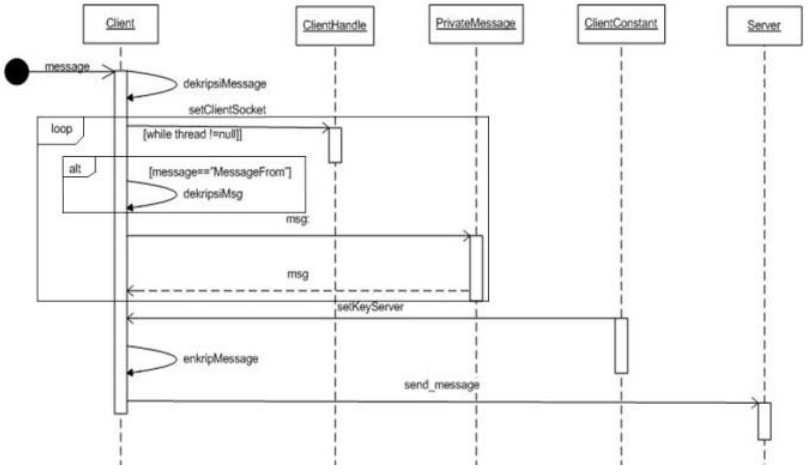
2.5. *Sequence Diagram*

Gambar 6 di bawah ini adalah sequence diagram server, saat server menerima pesan dari obyek client.Pesan akan didekripsi terlebih dahulu. Kemudian pesan diperiksa apakah pesan berupa create User, new User atau yang lainnya. Apabila pesan berupa create User maka server akan mengirimkan pesan cek Userada ke class create User, dengan balasan berupa message result. Apabila pesan berupa new User maka server akan mengirimkan pesan ke class password SQL, dengan balasan berupa *message result* seperti gambar 6 berikut:



Gambar 6. *Sequence Diagram* Mengirim Pesan

Gambar 7 di bawah ini adalah sequence client, saat client "Menerima Pesan". Sebuah message yang telah dikirim ke obyek server kemudian didekripsi. Selanjutnya bila pesan berupa "messageFrom" maka pesan akan dikirimkan ke user interface client. Apabila client akan mengirimkan pesan maka sebelumnya pesan akan dienkripsi oleh key server kemudian dikirimkan ke server.

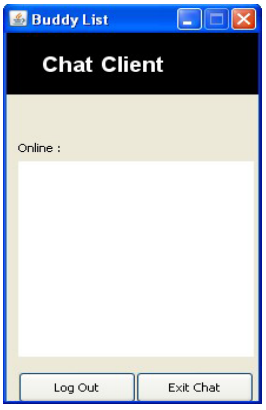


Gambar 7. *Sequence Diagram* Menerima Pesan

3. Hasil Dan Pembahasan

3.1. Implementasi

Implementasi sistem *secure chatting* ini dapat dilihat pada gambar 8 berikut ini:



Gambar 8. Implementasi *Buddy List*

Gambar 9 berikut merupakan implementasi *Private Message*:

Gambar 9. Implementasi *Private Message*

3.2. Pengujian

Pengujian perangkat lunak dilakukan secara bertahap pada lingkungan yang sama dengan tujuan sebagai berikut :

1. Menguji pengaruh sistem enkripsi terhadap performa perangkat lunak. Performa diukur dari penggunaan waktu dan memori perangkat lunak.
2. Menguji kebenaran proses enkripsi dan dekripsi teks pesan pada perangkat lunak.
3. Menguji apakah proses penyadapan akan menghasilkan *cipher text*.

3.3. Kasus Uji

Pengujian terbagi menjadi tiga tahap, yaitu pengujian performa, enkripsi-dekripsi, dan penyadapan. Kasus uji pada pengujian performa dilakukan dengan menganalisis pemakaian waktu dan memori ketika perangkat lunak dijalankan. Untuk menganalisis pemakaian waktu digunakan kode *system.nanoTime()*. Sementara untuk mengetahui penggunaan memori diselipkan kode *Runtime.getRuntime().freeMemory()*. Sementara itu pengujian enkripsi dan dekripsi secara umum dilakukan dengan dua cara, yaitu pengujian pengiriman dan penerimaan teks. Pengujian yang terakhir dutujukan untuk memastikan bahwa sistem enkripsi pada perangkat lunak dapat digunakan untuk mempersulit proses penyadapan. Untuk itu pengujian tersebut dilakukan dengan mencoba menyadap pesan yang dikirim melalui perangkat lunak menggunakan aplikasi *wireshark*.

3.4. Pengujian Performa

Pengujian performa dilakukan dengan membandingkan penggunaan waktu dan memori ketika perangkat lunak dijalankan.

3.5. Pelaksanan dan Hasil Pengujian Waktu

Pengujian dilakukan dengan memasukan fungsi *System.nanoTime()* pada *source code*. Fungsi ini akan menghasilkan angka berisi waktu dalam satuan nanodetik (10^{-9} detik). Berikut ini adalah tabel 2 yang berisi hasil pengujian tersebut:

Tabel 2. Hasil Pengujian Waktu

No	Panjang pesan	Waktu yang digunakan (nanodetik)	
		Pengiriman pesan	Penerimaan pesan
1	1	6092045	63876275
2	3	14220832	55033740
3	8	14413913	51190096
4	12	14852446	49236007
Rata-rata		12394809	54834029,5

3.6. Pelaksanan dan Hasil Pengujian Memori

Pengujian dilakukan dengan menyisipkan fungsi *Runtime.getRuntime().freeMemory()* pada *source code* perangkat lunak. Hasil pengujian dapat dilihat pada tabel 3 dan tabel 4 di bawah ini:

Tabel 3. Hasil Pengujian Waktu

No	Panjang pesan	Total Memori (Byte)	Memori Kosong (Byte)	Memori setelah pemakaian (Byte)	Pemakaian memori (Byte)
1	1	16252928	14220560	14067152	153408
2	3	16318464	14193416	14111816	81600
3	8	16318464	14181648	14100048	81600
4	12	16318464	14479008	14375080	103928
Rata-rata					105134

Tabel 4. Hasil Pengujian Waktu

No	Panjang pesan	Total Memori (Byte)	Memori Kosong (Byte)	Memori setelah pemakaian (Byte)	Pemakaian memori (Byte)
1	1	16252928	14334856	14213232	121624
2	3	16318464	14297440	14182632	114808
3	8	16318464	14301992	14088800	213192
4	12	16318464	14553168	14431712	121456
Rata-rata					142770

3.7. Pengujian Enkripsi-dekripsi

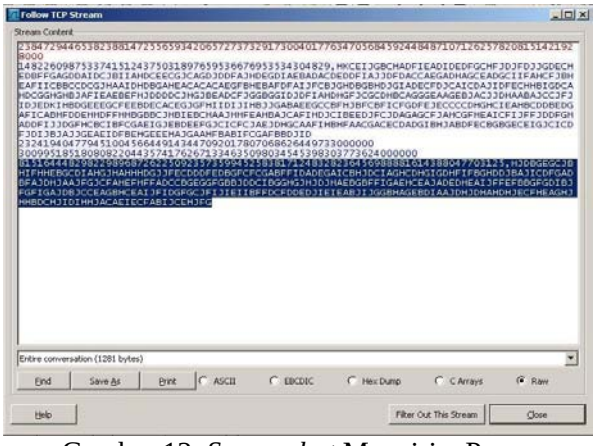
Pengujian dilakukan dengan percobaan mengirimkan pesan antara dua buah client. *Screenshot* pengujian tersebut dapat dilihat pada gambar 10 dan 11 berikut ini:

Gambar 10. *Screenshot* Mengirim Pesan

Gambar 11. *Screenshot* Menerima Pesan

3.8. Pengujian Penyadapan

Pada pengujian ini, pesan dicoba dikirimkan oleh perangkat lunak pengirim pesan ketika *WireShark* sedang berjalan. Pesan yang sama dicoba dikirimkan namun kali ini dalam keadaan terenkripsi. Ketika pesan sedang ditransmisikan ke server, perangkat lunak penyadapan tetap menangkap isi teks pesan yang dikirim, namun teks pesan tersebut adalah teks pesan yang telah dienkripsi. *Screenshot* pengujian tersebut dapat dilihat pada gambar 12 di bawah ini:



Gambar 12. *Screenshot* Mengirim Pesan

4. **Simpulan**

Berdasarkan hasil dan pembahasan diatas, maka didapat simpulan sebagai berikut:

1. Impementasi algoritma RSA dalam pengiriman pesan akan menambah kemanan dalam berkomunikasi. Dengan adanya sistem enkripsi penyadapan yang dilakukan ketika pesan sedang ditransmisikan menjadi semakin sulit karena pesan yang ditangkap berupa ciphertext.
2. Implementasi algoritma RSA tidak menurunkan performa perangkat lunak dengan berarti. Hal ini disebabkan pengaruh enkripsi yang tergolong kecil terhadap penggunaan memori dan waktu pemrosesan pada perangkat lunak.

Daftar Pustaka

[1] Elohor, Emanuzo. *A Secured Chat System with Authentication as RSA Digital Signature*. Department of computer Science, Owo Rufus Giwa Polythenic, Owo, Ondo, Nigeria. 2008.

[2] Firasyan, Winarno, Setiowati. Penggunaan Algoritma RSA untuk Keamanan Transaksi Online Berbasis Aplikasi Mobile. Institut Teknologi Sepuluh Nopember Surabaya; 2011.

[3] Kwok, Wallace, Man. *Online secure chatting*. Departement of electronic Engineering, City University of Hong Kong; 2002.

[4] Menezes, Alfred J., Paul C van Oorschot, and Scott A. Vanstone. *Handbook of applied Cryptography*, CRC Press. 1996.

[5] Schneizer, Bruce. *Apliend Cryptography 2*nd. John Wiley & Sons. 1996.