Security Chatting Berbasis *Desktop* dengan Enkripsi *Caesar Cipher Key Random*

Gratia Vintana[#1], Mardi Hardjianto[#2]

[#]*Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur*

Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260

Telp. (021) 5853753, Fax. (021) 5866369

[1]gratiavintana@gmail.com

[2]mardi.hardjianto@budiluhur.ac.id

Abstraksi - Perkembangan dunia teknologi khususnya pada bidang komunikasi saat ini telah berkembang dengan pesat. Terbukti dengan adanya aplikasi chatting sebagai media komunikasi yang kerap digunakan oleh masyarakat banyak. Dengan menggunakan aplikasi chatting, user dapat berkomunikasi dengan mudah dan cepat. Beberapa aplikasi-aplikasi chatting yang ada saat ini, memiliki kekurangan pada tingkat keamanan pengiriman pesan. Hal ini dapat menjadi masalah bagi pengguna aplikasi chatting, khususnya bila aplikasi tersebut digunakan untuk berkomunikasi membahas kepentingan yang rahasia. Karena tidak sedikit hal yang rahasia perusahaan dibicarakan, maka dibutuhkannya aplikasi chatting yang mampu menjaga kerahasiaan dari isi pesan yang disampaikan. Hal ini dapat dilakukan dengan cara menggunakan metode enkripsi yang dimasukkan pada proses pengiriman pesan pada aplikasi chatting. Dari sekian banyak metode enkripsi yang ada, enkripsi yang digunakan adalah Caesar Cipher dengan key random. Enkripsi Caesar Cipher bekerja dengan merotasi urutan karakter menurut jumlah key yang diberikan. Pesan yang dikirimkan tidak akan tampil dalam bentuk sebenarnya kecuali penerima pesan. Dengan demikian diharapkan pengguna aplikasi chatting dapat berkomunikasi dengan nyaman dan aman tanpa perlu kuatir bila isi pesan disadap oleh orang lain. Hal ini disebabkan karena isi pesan yang dibahas dan dikirim tidak akan dapat dimengerti oleh pihak lain selain penerima tujuan pesan.

Kata kunci : chatting, algoritma enkripsi, caesar cipher

I. PENDAHULUAN

Kebutuhan dasar manusia adalah komunikasi. Tanpa komunikasi manusia tidak dapat bersosialisasi satu dengan yang lainnya. Seiring dengan berkembangnya teknologi informasi dunia, berkembang pula teknologi komunikasi. Mulai dari surat, telepon, hingga sekarang yang paling banyak digunakan adalah internet. Internet semakin banyak diminati karena mudah digunakan, dan dapat diakses setiap orang dari berbagai kalangan. Bukti dari perkembangan teknologi informasi pada bidang komunikasi yaitu dengan adanya e-mail. Dengan menggunakan e-mail, kita dapat mengirimkan pesan kepada orang lain secara cepat. Namun kita sering mengeluh atas lamanya respon/balasan pesan yang kita kirim, dan proses balas pesan yang tidak praktis. Atas dasar itulah dibuatnya aplikasi instant messenger atau yang biasa disebut aplikasi *chatting*. Aplikasi *chatting* merupakan aplikasi yang mengijinkan penggunanya dapat mengirimkan pesan secara satu waktu atau real-time yang membuat jarak sebenarnya seolah-olah tidak berarti di dunia internet. Oleh karena itu dengan memanfaatkan layanan internet, terlebih jika internet tersebut menggunakan jaringan, dimana aplikasi dapat berjalan pada komputer tanpa menggunakan kabel LAN, aplikasi dapat berjalan dengan cepat, mudah, tanpa perlu menunggu lama balasan dari orang yang dituju, dan cara bertukar pesan sangat praktis.

Kini teknologi *chatting* sudah berkembang dengan cepat, banyak sekali aplikasi *chatting* yang sudah ada seperti Yahoo Messenger, mlRC, Google Talk, Windows Live Messenger, dan lainnya yang memiliki kelebihan masing-masing. Namun aplikasi-aplikasi tersebut memiliki kekurangan pada keamanan pengiriman pesan. Kekurangannya adalah pesan yang dikirim tidak diacak sehingga beresiko informasi yang ada dalam pesan tersebut dapat dicuri dan dimengerti maknanya oleh pihak yang tidak berwenang. Hal ini akan sangat merugikan bagi pengguna, bila pesan yang dikirimkan berisikan informasi yang bersifat rahasia, khususnya bagi perusahaan.

PT. Quantum Integrated Services adalah perusahaan yang bergerak dibidang marketing dan *public relations consulting*, dimana membutuhkan komunikasi yang lancar dan aman dalam perusahaan. Oleh sebab itu dalam penelitian ini, akan dibuat aplikasi chatting dimana informasi yang terkandung dalam pesan yang disampaikan dapat terjaga kerahasiaanya. Pesan yang dikirim akan diacak dengan menggunakan metode enkripsi Caesar Cipher dengan *key random*. Aplikasi *chatting*, yang dibuat berbasis *desktop* dan menggunakan jaringan ini , diharapkan dapat memudahkan karyawan PT.Quantum Integrated Services untuk berkomunikasi, tanpa harus kuatir pesan yang disampaikan

diketahui pihak yang tidak berkepentingan.

II. LANDASAN TEORI

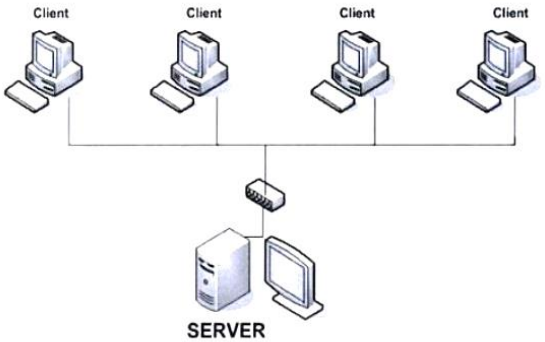
Beberapa konsep yang melatarbelakangi pembuatan aplikasi ini sebagai berikut:

2.1. Jaringan Komputer

Jaringan menghubungkan kelompok yang memiliki kesamaan menjadi satu. Misalnya sistem telepon suatu negara berhubungan dengan sistem telepon dari negara lain untuk membentuk jaringan telepon internasional. Jaringan tersebut akan salain terhubung atau saling terkait. Jaringan komputer adalah sebuah sistem yang terdiri atas sejumlah komputer yang saling terhubung satu dengan yang lain. Internet saat ini merupakan bentuk jaringan komputer raksasa yang menghubungkan semua jaringan komputer yang ada di dunia ini. Secara teknis yang dimaksud dengan komunikasi adalah proses perpindahan data dari sumber menuju ke tujuan dengan menggunakan *transmitter* dan *receiver*. *Transmitter* adalah alat yang mengubah data menjadi sinyal elektronis dan mengirimkan sinyal tersebut sedangkan *receiver* adalah alat yang menerima sinyal dan mengkonversikan sinyal tersebut menjadi data [1].

2.2. Model Jaringan Komputer Client-Server

Pada model ini diperlukan satu atau lebih komputer yang berlaku sebagai *server* untuk mengatur lalu-lintas data dan informasi dalam jaringan komputer. Hal ini menyebabkan setiap komputer yang tergabung dalam jaringan dan ingin berkomunikasi harus berhubungan dengan *server* terlebih dahulu. Komputer yang memanfaatkan *server* pada jaringan tersebut biasa disebut sebagai *client*. Komputer yang berlaku sebagai *server* biasanya menunggu berbagai permintaan *client* untuk kemudian melayani permintaan tersebut. Komputer yang berlaku sebagai *client*, biasanya bersifat aktif untuk mengirim permintaan ke *server* serta menerima layanan dari *server*. Arsitektur *client-server* ditunjukkan seperti pada Gambar 1.



Gbr. 1 Arstitektur Client-Server

Keunggulan menggunakan sebuah tipe jaringan komputer *Client-Server*, diantaranya:

- Kecepatan akses relatif cepat karena penyediaan fasilitas jaringan dan pengelolaannya dilakukan secara khusus oleh server.
- Sistem keamanan jaringan lebih terjamin.
- Administrasi dan pengelolaan jaringan lebih baik karena dikelola oleh administrator jaringan yang bertanggung jawab terhadap keseluruhan jaringan komputer yang terkait.

Kelemahan menggunakan sebuah tipe jaringan komputer *Client-Server*, diantaranya:

- Diperlukan satu komputer khusus yang mempunyai konfigurasi tinggi yang akan berfungsi sebagai *server*.
- Berjalannya sistem jaringan komputer sangat bergantung pada *server* [1].

2.3. Jenis Komunikasi dalam Jaringan

Komunikasi dalam jaringan menurut jenisnya dibagi menjadi dua, yaitu komunikasi dalam jaringan sinkron dan komunikasi dalam jaringan asinkron. Penjelasan masing-masing jenis tersebut sebagai berikut:

1) Komunikasi dalam Jaringan Sinkron

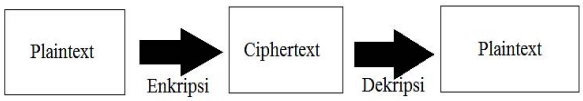
   Komunikasi dalam jaringan sinkron merupakan komunikasi dalam jaringan secara *real-time* menggunakan komputer sebagai media. Komunikasi tersebut dilakukan juga secara serempak atau bersamaan. Contoh komunikasi sinkron misalkan aplikasi *chatting* (Yahoo Messenger, Google Talk, MIRc dll), video *chat* (Skype, Line, Facetime, Google+ Hangout, dan lainnya).

2) Komunikasi dalam Jaringan Asinkron

   Komunikasi dalam jaringan asinkron merupakan komunikasi dalam jaringan secara tunda menggunakan komputer sebagai media. Komunikasi tersebut juga dilakukan secara tidak serempak. Contoh komunikasi asinkron misalnya aplikasi *e-mail*, video *streaming*, dan lainnya[2].

2.4. Kriptografi

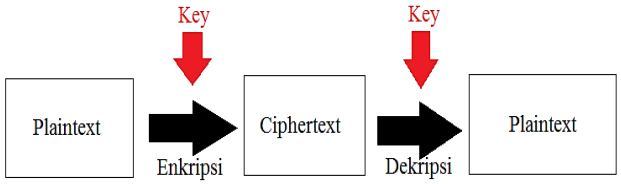
Kriptografi merupakan ilmu merumuskan metode yang memungkinkan informasi yang akan dikirimkan dibuat sedemikian rupa sehingga menjadi dalam kondisi atau bentuk yang aman. Informasi ini hanya mampu diterima dan dimengerti maknanya oleh penerima yang ditujukan informasi tersebut dikirimkan. Fungsi-fungsi yang mendasar dalam kriptografi adalah enkripsi dan dekripsi. Gambaran alur enkrpsi dan dekripsi dapat dilihat pada Gambar 2.



Gbr. 2 Alur Enkripsi Dekripsi pada Kriptografi

Enkripsi adalah proses mengubah suatu pesan asli (*plaintext*) menjadi suatu pesan dalam bahasa sandi (*ciphertext*). Dekripsi adalah proses mengubah pesan dalam suatu bahasa sandi menjadi pesan asli kembali. Seringkali fungsi enkripsi dan dekripsi tersebut diberi parameter tambahan yang disebut dengan istilah kunci. Gambar 3 merupakan gambaran hubungan antara enkripsi, kunci dan dekripsi. Kekuatan dari algoritma kriptografi umumnya

bergantung kepada kuncinya, oleh sebab itu kunci yang lemah tidak boleh digunakan. Panjang kunci yang digunakan juga menentukan kekuatan dari dari algoritma kriptografi [3].

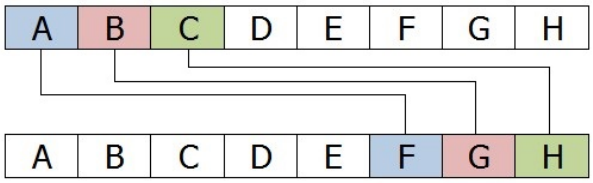


Gbr. 3 Alur Enkripsi Dekripsi dengan Kunci

2.5. Caesar Cipher

Caesar Cipher merupakan salah satu jenis *cipher* substitusi yang membentuk *cipher* dengan cara melakukan penukaran satu karakter diganti dengan karakter yang berada di sejumlah digit sebelah kanan atau kirinya, tergantung arah pergeserannya. Teknik seperti ini disebut juga sebagai cipher abjad tunggal. *Caesar Cipher* adalah dasar enkripsi yang sangat baik untuk dipahami sebelum membahas enkripsi berbasis karakter lainnya yang lebih rumit [4].

Inti dari algoritma kriptografi ini adalah melakukan pergeseran terhadap semua karakter pada plainteks dengan nilai pergeseran yang sama. Contohnya jika diketahui bahwa pergeseran sebanyak lima, maka huruf A akan digantikan oleh F, huruf B menjadi huruf G, dan seterusnya seperti yang ditunjukkan pada Gambar 4.



Gbr. 4 Caesar Cipher dengan Pergeseran Lima

Contoh jika menyandikan kalimat "PESAN INI" makan dengan pergeseran lima, maka kalimat cipherteks yang terbentuk adalah "UJXFS NSN", seperti pada Tabel 1 berikut:

TABEL I
CONTOH KALIMAT DENGAN ENKRIPSI CAESAR CIPHER

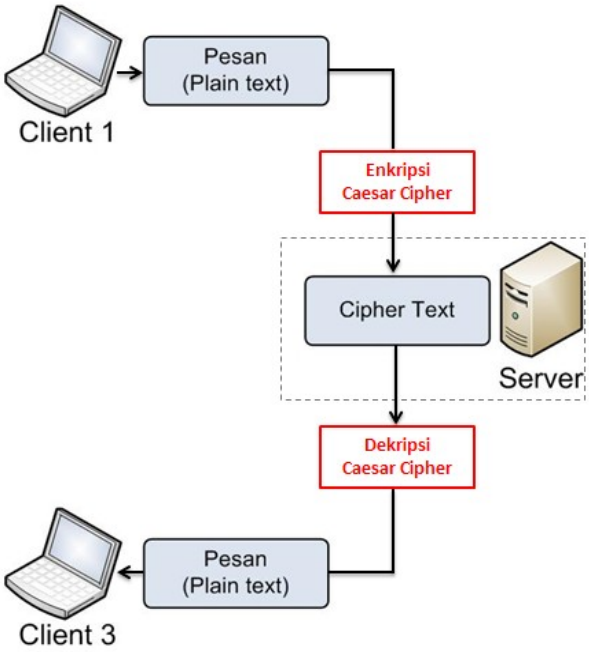
Plaintext								
P	E	S	A	N		I	N	I
Ciphertext								
U	J	X	F	S		N	S	N

Proses penyandian (enkripsi) dapat secara matematis menggunakan operasi modulus dengan mengubah huruf-huruf menjadi angka, A = 0, B = 1,…, Z = 25.
Sandi (*En*) dari huruf yang disimbolkan x dengan jumlah geseran sebanyak n secara matematis dituliskan dengan:
$En(x) = (x+n) \bmod 26$

Sedangkan pada proses pemecahan kode (dekripsi), hasil dekripsi (*Dn*) ditulis dengan:
$Dn(x) = (x-n) \bmod 26$ [5].

III. RANCANGAN SISTEM DAN APLIKASI

Adapun metode kerja penggunaan enkripsi dan dekripsi yang digunakan pada aplikasi ini seperti yang ditunjukkan pada Gambar 5.

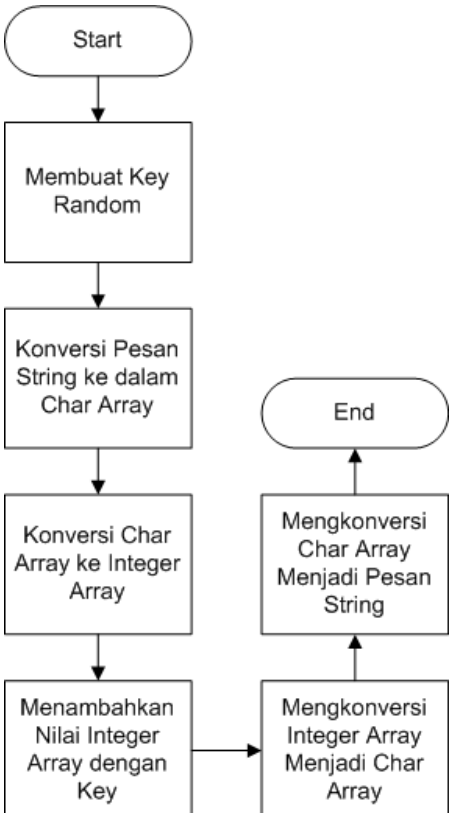


Gbr. 5 Skema Metode Kerja Aplikasi

Rancangan proses tahapan enkripsi dan dekripsi pesan digambarkan melalui *flowchart* sebagai berikut:

3.1. Flowchart *Enkripsi Pesan*

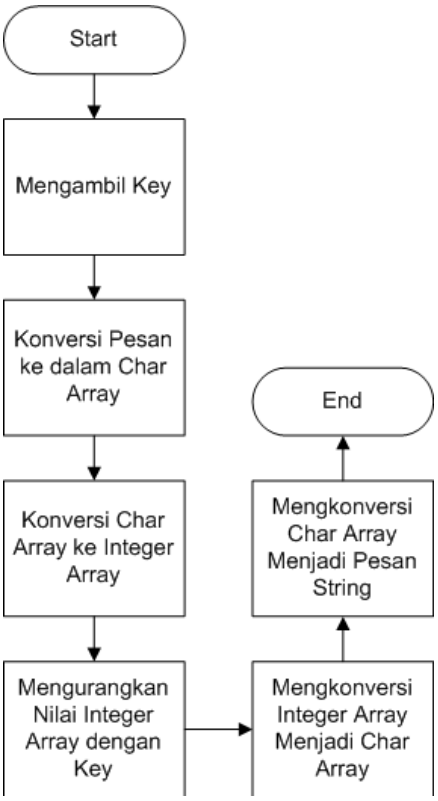
Pada *flowchart* enkripsi pesan, tahap pertama yang dilakukan adalah membuat *key random* (kunci acak). *Key random* yang dibuat dengan batas maksimal 94. Setelah itu mengkonversikan pesan yang masuk ke dalam *char array*, karena pesan yang masuk masih berupa *string*. Pesan yang masuk juga disebut sebagai *plaintext*. *Char array* yang terbentuk dikonversikan menjadi *integer array* untuk mengetahui nilai ASCII setiap elemen. Nilai setiap elemen pada *integer array* ditambahkan dengan nilai *key random* yang sebelumnya telah dibuat. Nilai *integer array* ini dikonversikan kembali menjadi *char array*, dan dilanjuti dengan konversi *char array* menjadi pesan *string*. Hasil dari proses ini didapat pesan *string* yang terbentuk telah berubah menjadi *ciphertext*. Gambar *flowchart* enkripsi pesan dapat dilihat pada Gambar 6:



Gbr. 6 Flowchart Enkripsi Pesan

3.2. Flowchart *Dekripsi Pesan*

Pada *flowchart* dekripsi pesan, hampir sama dengan *flowchart* enkripsi pesan. Pada dekripsi pesan, tahap yang pertama dilakukan adalah mengambil *key*. Kemudian pesan yang didapat dalam bentuk *ciphertext* dikonversi kedalam *char array*. *Char array* yang terbentuk dikonversikan ke dalam bentuk integer array untuk mendapat nilai ASCII setiap elemen. Nilai setiap elemen pada *integer array* dikurangi dengan nilai *key* yang telah diambil. *Integer array* dikonversikan kembali menjadi *char array,* dan *char array* dikonversikan ke dalam bentuk pesan *string*. Maka pesan yang semula *ciphertext* setelah proses dekripsi dapat menjadi *plaintext*. Gambar *flowchart* dekripsi pesan dapat dilihat pada Gambar 7.



Gbr. 7 Flowchart Dekripsi Pesan

IV. HASIL DAN PEMBAHASAN

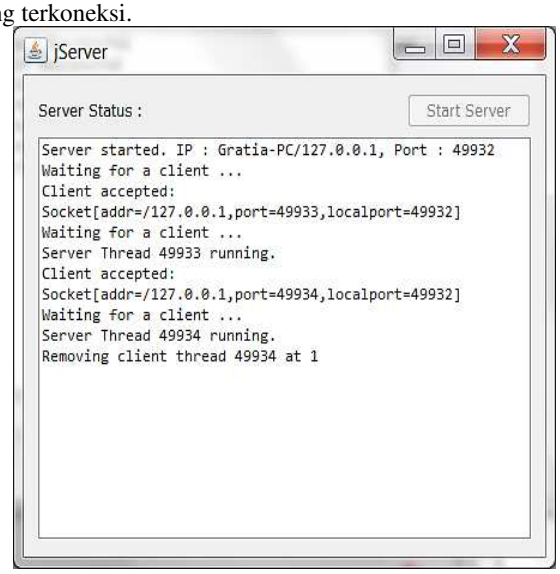
Tampilan layar aplikasi yang dibuat dibagi menjadi tiga menurut jumlah halaman, yaitu:

4.1. Server

Tampilan awal halaman *server* dapat terjadi pada saat aplikasi *server* pertama kali dijalankan, seperti pada Gambar 8.

Gbr. 8 Tampilan Awal Server

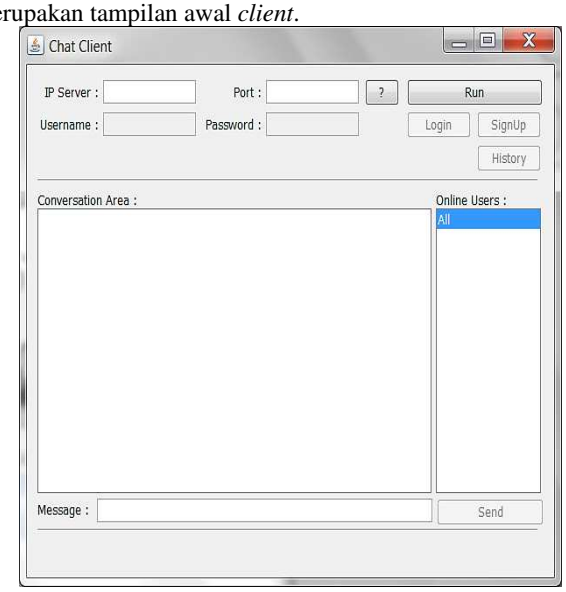
Tampilan halaman *server* saat berhasil terkoneksi dan mendapat dua *client* yang terhubung, serta ada *client* yang keluar. Berikut pada Gambar 9 merupakan tampilan *server* yang terkoneksi.



Gbr. 9 Tampilan Server Terkoneksi

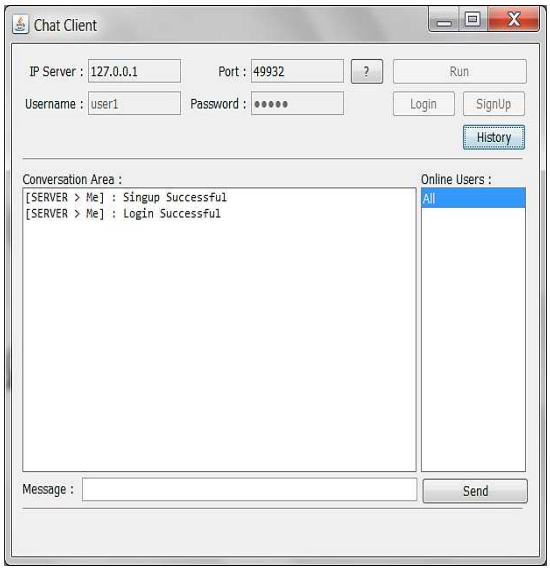
4.2. *Client*

Tampilan awal halaman *client* dapat terjadi pada saat aplikasi *client* pertama kali dijalankan. Berikut Gambar 10 merupakan tampilan awal *client*.



Gbr. 10 Tampilan Awal Client

Pada halaman *client*, *user* dapat *login* atau *signup* untuk pengiriman pesan. Berikut pada Gambar 11 merupakan *user* yang berhasil melakukan *signup* dan *login*.



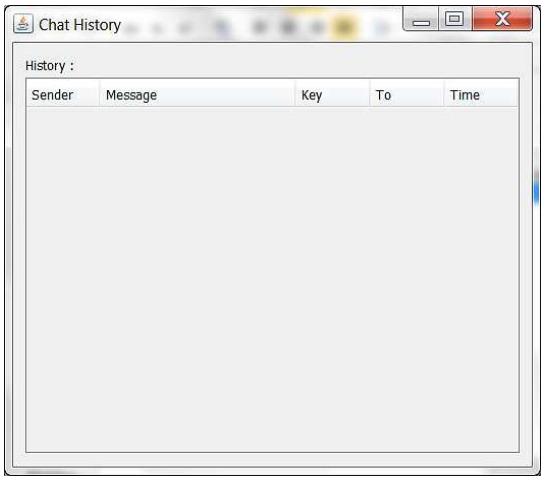
Gbr. 11 Tampilan Client Signup dan Login

User yang telah berhasil login dapat mengirimkan dan menerima pesan, baik itu pesan pribadi ataupun pesan *broadcast*. Berikut pada Gambar 12 merupakan tampilan pengiriman dan penerimaan pesan.

Gbr. 12 Tampilan Pengiriman dan Penerimaan Pesan

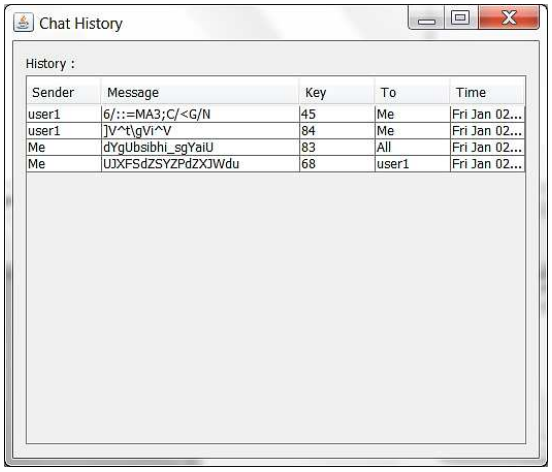
4.3. History

Berikut pada Gambar 13 merupakan tampilan awal halaman history pada saat belum terjadi percakapan.



Gbr. 13 Tampilan Awal Halaman History

Tampilan halaman *history* pada saat telah terjadi penerimaan dan pengiriman pesan dapat dilihat pada Gambar 14.



Gbr. 14 Tampilan Halaman History saat Mengirimkan dan Menerima Pesan

4.4. Hasil Uji Coba Enkripsi Pesan

Untuk dapat mengetahui hasil dari pesan yang telah di enkripsi. Berikut tabel 2 merupakan tabel hasil enkripsi beberapa pesan yang berisikan *plaintext*, *key*, dan *ciphertext*.

TABEL II
TABEL HASIL ENKRIPSI
CONTOH KALIMAT DENGAN ENKRIPSI CAESAR CIPH

Plaintext	Key	Ciphertext
hai	61	F?G
apa kabar kalian semua	76	N]NlXNON_lXNYVN[l`RZbN
saya baik baik saja	86	jXpXvYX`bvYX`bvjXaX
ini pesan rahasia	71	QVQgXM[IVgZIPI[QI
pesan telah dienkripsi	44	=2@.;LA29.5L162;8?6=@6

V. KESIMPULAN

Berdasarkan analisa dari permasalahan dan penyelesaian masalah pada bab-bab sebelumnya, maka dapat ditarik kesimpulan sebagai berikut:

a. Pengamanan pesan sangat diperlukan untuk menjaga kerasiaan pesan.
b. Enkripsi dengan *caesar cipher* dengan menggunakan *key* acak menjadikan pesan yang sama menghasilkan *ciphertext* yang berbeda sehinga menjadi lebih aman.

5.1. Saran

Aplikasi *chatting* dengan enkripsi *caesar cipher* masih memiliki banyak kekurangan, dan diperlukan pengembangan lebih lanjut guna mencapai hasil pengamanan yang maksimal. Berikut ini saran yang dijadikan acuan untuk pengembangan aplikasi selanjutnya:

a. Adanya penambahan fitur agar *client* dapat mengakses aplikasi berbeda jaringan.
b. Adanya penambahan fitur agar memungkinkan beberapa *client* dapat menggunakan history yang berbeda pada satu komputer.
c. Adanya penambahan fitur agar dapat menghapus history.
d. Adanya penambahann fitur-fitur lain agar aplikasi ini menjadi lebih menarik dan semakin sempurna.

REFERENSI

[1] Ukar, K., 2009. Pengenalan Komputer. Jakarta: Elex Media Komputindo.
[2] Sanggita, H., 2014. Pengertian Simulasi Digital.http://www.academia.edu/9080716/Pengertian_Simulasi_Digital[Diakses 3 Desember 2014].
[3] Heriyanto, T., 1999. Pengenalan Kriptografi. Volume 1.
[4] Kromodimoeljo, S., 2009. Teori dan Aplikasi Kriptografi. Jakarta: SPK IT Consulting.
[5] Husein, M., 2014. Implementasi Caesar Cipher untuk Penyembunyian Pesan Teks Rahasia pada Citra dengan Menggunakan Metode Least Significant Bit. Volume V