Penerapan Metode Vigenere Chiper pada Aplikasi Chat Messenger Sederhana

Nur Adi Susliawan Dwi Caksono / 13508081
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
If18081@students.if.itb.ac.id

***Abstract*— Saat ini, media komunikasi yang paling cepat perkembangannya bagi pengguna internet adalah Instant Messaging. Instant Messaging merupakan suatu aplikasi yang tujuannya ialah memudahkan pengguna dalam berkomunikasi. Sekarang ini, sudah banyak aplikasi yang menerapkan layanan instant messaging, diantaranya ialah Yahoo Messenger, Windows Live Messenger, dan Google Talk. Instant Messaging memiliki banyak fitur diantaranya ialah bertukar pesan singkat atau biasa disebut dengan istilah chat, berkirim pesan layaknya menggunakan email, teleconference dimana user-user dalam suatu kelompok dapat berkomunikasi satu sama lain dalam satu waktu, video call, bahkan sekarang voice call pun termasuk fitur dari instant messaging. Namun disamping kekayaan fitur yang ditawarkan oleh instant messaging, belum banyak yang mengetahui sejauh mana tingkat keamanan layanan-layanan tersebut. Hal ini akan menjadi sangat penting jika informasi yang dikirim merupakan informasi yang sifatnya sensitive dan rahasia. Pada makalah ini saya akan mencoba untuk membuat suatu aplikasi chat messenger sederhana dimana pada lalu lintas pertukaran data antara user yang satu dengan user yang lain akan melalui proses enkripsi menggunakan metode enkripsi kriptografi klasik yaitu vigenere chiper**

Kata kunci : Instant Messaging, Chat, Vigenere Chiper

I. PENDAHULUAN

Tidak diragukan lagi perkembangan internet yang sangat pesat telah menyebabkan cara masyarakat berkomunikasi berubah. Dimulai dengan pengiriman surat konvensional yang harus dikirim melalui pos, kemudian berkembang surat elektronik (e-mail). Jutaan e-mail dikirimkan setiap harinya dalam jaringan internet. Namun penggunaan email ini terkadang masih dianggap kurang cepat dan kurang praktis karena email masih menerapkan system komunikasi yang sifatnya offline, sehingga pengirim tidak bisa mengharapkan pesan emailnya dapat ditanggapi sesegera mungkin.

Instant messaging merupakan suatu perangkat lunak (*software*) yang memfasilitasi pengiriman pesan singkat antara dua user atau lebih. Sekarang ini fitur dari instant messaging tidak hanya berkisar pada pengiriman teks saja tetapi sudah mencapai pengriman teks oleh multiuser pada saat yang sama atau biasa disebut dengan konferensi, voice call dan video call. Penggunaan teknologi ini memiliki suatu kelebihan dibandingkan dengan surat elektronik (e-mail) karena komunikasi dapat terjalin secara langsung atau real time. Hal ini lah yang menyebabkan instant messaging merupakan media komunikasi yang paling cepat perkembangannya.

Instant messaging merupakan kumpulan teknologi berbasis teks yang digunakan oleh dua atau lebih partisipan yang dihubungkan dengan jaringan internet. Instant messaging menawarkan teknologi komunikasi yang efektif dan efisien.Instant messaging melalui jaringan internet pertama kali muncul di system operasi yang sifatnya multi-user seperti CTSS dan Multics pada pertengahan tahun 1960 Pada awalnya, beberapa dari system ini digunakan sebagai system nofitifikasi untuk layanan seperti printing, tetapi system ini malah digunakan untuk memfasilitasi komunikasi dengan pengguna lain yang log in di mesin yang sama.

Sistem instant messaging juga memungkinkan *user*(pengguna) untuk menyimpan daftar orang yang dapat diajak berkomunikasi dalam sebuah daftar kontak. Sistem akan memberikan informasi jika ada kontak yang sedang *online* sehingga pengguna dapat memuliai percakapan dengan kontak tersebut dan kemudian saling bertukar pesan. Komunikasi pun terjadi secara instan dan *real time*. Fasilitas instant messaging ini disebut *presence information*.

Seiring dengan semakin maraknya layanan *instant messaging*, maka aspek keamanan menjadi sangat penting untuk dipertimbangkan apalagi apabila data yang dikirimkan tersebut sifatnya sensitive dan rahasia. Apalagi jika pemakainya merupakan perusahaan atau enterprise, maka keamanan instant messaging tidak boleh dikesampingkan.

Tetapi kenyataannya system instant messaging saat ini didesain bukan berdasarkan pada aspek keamanan

melainkan aspek skalabilitas untuk menunjang jumlah pengguna yang begitu besar. Hampir semua layanan instant messaging public tidak memiliki fasilitas enkripsi seperti Yahoo Messenger dan Windows Live Messenger.

Oleh karena itu, pada makalah ini penulis akan mencoba untuk membuat aplikasi chat messenger sederhana yang lalu lintas pengiriman data nya akan dienkripsi dengan salah satu metode algoritma kriptografi klasik yaitu Vigenere Chiper.

II. DASAR TEORI

2.1 VIGENERE CIPHER

Vigenere Cipher adalah salah satu jenis kriptografi klasik yang pada dasarnya ialah melakukan substitusi chpher abjad majemuk (*polyalphabetic substitution*). Metode ini pertama kali dipublikasikan oleh seorang diplomat (sekaligus seorang kriptologis) Perancis, Blaise de Vigenere pada abad ke 16, tepatnya pada tahun 1586, tetapi sebenarnya Giovan Batista Belaso telah menggambarkannya pertama kali pada tahun 1553 seperti ditulis di dalam bukunya *La Cifra del Sig*. Metode Vigenere Cipher ini berhasil dipecahkan oleh matematikawan Inggris Charles Babbage dan Kasiski pada pertengahan abad 19. Vigenere cipher ini digunakan oleh tentara konfiderasi pada perang sipil Amerika. Perang sipil akhirnya berhasil dihentikan setelah vigenere cipher berhasil dipecahkan.

Di metode kriptografi klasi Caeasr cipher, setiap huruf alphabet akan disubstitusi sepanjang 3 huruf sesudah huruf tersebut. Contoh, huruf A akan diganti dengan huruf D, B akan diganti dengan huruf E, Y akan diganti dengan huruf B dengan metode Caesar cipher.Vigenere Cipher ini menerapkan prinsip Caesar Cipher dalam pengenkripsiannya.

Untuk memudahkan dalam proses pengenkripsiannya, maka dapat digunakan alat bantu berupa bujur sangkar Vigener yang dikenal juga dengan nama tabula recta.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Gambar 2.1 tabula recta

Baris pada gambar tabula recta menyatakan huruf plainteks yang akan dienkripsi, dan kolom menyatakan huruf kunci enkripsi. Dan perpotongan antara baris dan kolom menyatakan huruf yang sudah terenkripsi atau diistilahkan dengan cipherteks.

Jika panjang kunci lebih pendek daripada panjang plainteks, maka kunci akan diulang secara periodic.

Contoh :

Kunci	:	sony
Plainteks	:	This is plainteks
Kunci pada saat enkripsi	:	sony so nysonyson
Cipherteks	:	LVVQ HZNGFHRVL

Pada dasarnya, setiap enkripsi huruf adalah *Caesar Cipher* dengan kunci yang berbeda-beda.

c(‘T’) = (‘T’ + ‘s’) mod 26 = L

c(‘H’) = (‘H’ + ‘o’) mod 26 = V, dst

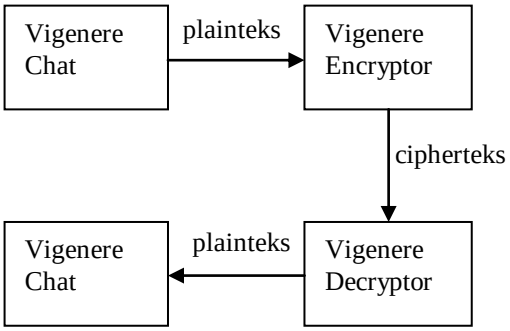
Jadi dengan vigenere cipher, huruf yang sama pada plainteks tidak selalu dienkripsi menjadi huruf cipherteks yang sama pula. Contoh huruf plainteks T dapat dienkripsi menjadi L atau J, huruf cipherteks V dapat merepresentasikan huruf plainteks H, I, dan X. Hal ini merupakan karakteristik dari cipher abjad majemuk dimana setiap huruf plainteks dapat memiliki kemungkinan banyak huruf palinteks. Hal ini berbeda dengan cipher substitusi sederhana dimana setiap huruf cipherteks selalu menggantikan huruf plainteks tertentu.

Vigenere Cipher yang akan dipakai pada aplikasi ini adalah vigenere Cipher extended dimana pengenkripsian tidak hanya untuk huruf alphabet tetapi termasuk juga karakter-karakter ASCII. Jadi batas pengenkripsian tidak terbatas untuk 26 karakter tetapi mencapai 256 karakter.

2.2 DESKRIPSI VIGENERE CHAT

Seperti yang telah dijelaskan pada bab pendahuluan, penulis akan mencoba membuat aplikasi chat sederhana yang pengiriman datanya akan melalui proses pengenkripsian terlebih dahulu dimana metode enkripsi yang dipakai adalah vigenere cipher. Program ini penulis namakan dengan nama vigenere chat. Program ini akan mengizinkan 2 pihak untuk saling chat satu sama lain melalui koneksi TCP/IP. Vigenere chat ini akan dibuat dalam bahasa java.

Ilustrasi pemakaian program ini dapat dimisalkan seperti ini : Misalkan Zakiy dan Ifan ingin menggunakan vigenere chat untuk mengamankan percakapan mereka sehingga tidak disadap oleh orang lain. Satu pihak diantara mereka berdua setuju bertindak sebagai client dan pihak yang lain menjadi server. Pertama-tama server harus menyediakan port yang harus dibuka sehingga mengizinkan client dapat membuat koneksi dengan server.



Gambar 2.2 Deskripsi Umum Vigenere Chat

Deskripsi umum aplikasi vigenere chat dapat Anda lihat pada gambar 2.2 diatas. Pesan yang dikirimkan dari suatu aplikasi akan terlebih dahulu melalui suatu blok yang dinamakan Vigenere Encryptor. Blok ini akan mengenkripsi pesan plainteks dengan menggunakan algoritma vigenere cipher menjadi sebuah cipherteks. Cipherteks inilah yang akan dikirim ke jaringan, sehingga apabila ada seseorang yang ingin menyadap pesan ini, kesulitan untuk melakukannya karena pesan sudah teracak. Kemudian pesan cipherteks yang dikirim ini akan diterima oleh blok yang bernama vigenere decryptor yang akan mendekripsi pesan cipherteks menjadi plainteks semula yang dikirim oleh sender. Pada blok vigenere encryptor, plainteks akan dienkripsi sesuai dengan kunci yang dimasukkan oleh pengguna aplikasi. Begitu pula untuk blok vigenere decryptor, cipherteks hanya dapat didekripsi menghasilkan plainteks semula yang dikirim oleh sender dengan menggunakan kunci yang sama dengan yang dimasukkan oleh pengguna di bagian pengirim.

III. IMPLEMENTASI

Implementasi aplikasi Vigenere Chat ini dapat kita bagi menjadi 2 bagian yaitu bagian kelas Socket yang akan membangun koneksi antara aplikasi yang satu dengan aplikasi yang lain dan kelas Vigenere Cipher yang akan melakukan proses enkripsi dan dekripsi pesan.

3.1 KELAS VIGENERE CIPHER

Seperti yan telah dijelaskan di bab 2 dasar teori, aplikasi ini akan menggunakan metode enkripsi vigenere cipher teks extended untuk penngenkripsian pesannya. Berikut adalah pseudocode untuk enkripsi pada kelas vigenere cipher extended ini yang ditulis dengan bahasa Java.

```
public String CipherTeksExtended(String Teks, String Key)
{
    Teks = Teks.toLowerCase();
    Key = Key.toLowerCase();
     String temp = "";
     int i = 0;
     int j = 0;
     while(j < Key.length() && i < Teks.length())
     {
          char cTeks = Teks.charAt(i);
          if(cTeks != ' ' && cTeks != '¥n')
          {
             char cKey = Key.charAt(j);
             int ATeks = (int) cTeks;
             int AKey = (int) cKey ;
             ATeks = ((ATeks + AKey) % 256);
             cTeks = (char) ATeks;
             String sTeks = Character.toString(cTeks);
             temp = temp.concat(sTeks);
             j++;
          }
          f(cTeks == '¥n' ){
               temp = temp.concat("¥n");
          }
          i++;
       }
       return temp;
}
```

Gambar 3.1 Pseudocode Enkripsi Vigenere Cipher Extended

Gambar 3.1 diatas merupakan suatu fungsi yang akan menghasilkan suatu string yang merupakan hasil enkripsi dari inputan yang berupa teks dan kunci. Jadi fungsi diatas akan menerima inputan berupa teks dan key yang memiliki panjang teks yang sama. Kemudian terdapat suatu iterator i yang akan menghitung sampai i mencapai end of teks. Jadi, selama i belum mencapai end of teks, akan dilakukan proses enkripsi dengan cara menambah ASCII teks pada posisi i dengan ASCII key pada posisi I kemudian di mod dengan 256 karena jumlah karakter ASCII standar itu berjumlah 256.

Fungsi enkripsi ini akan dipakai dalam pada blok Vigenere Encryptor pada gambar 2.2.

Apabila terdapat proses pengenkripsian pasti terdapat pula proses dekripsi. Berikut adalah pseudocode untuk dekripsi pada kelas vigenere cipher extended

```
public String DeCipherTeksExtended(String Teks, String Key)
{
    Teks = Teks.toLowerCase();
    Key = Key.toLowerCase();
    String temp = "";
    int i = 0;
    int j = 0;
    while(j < Key.length() && i < Teks.length())
    {
        char cTeks = Teks.charAt(i);
        if(cTeks != ' ' && cTeks != '¥n')
        {
```

```
            char cKey = Key.charAt(j);
            int ATeks = (int) cTeks;
            int AKey = (int) cKey;
            ATeks = ATeks - AKey;
            if(ATeks < 0){
                ATeks = ATeks + 256;
            }
            cTeks = (char) ATeks;
            String sTeks = Character.toString(cTeks);
            temp = temp.concat(sTeks);
            j++;
        }
        i++;
    }
    temp = temp.toUpperCase();
    return temp;
}
```

Gambar 3.2 Pseudocode Dekripsi Vigenere Cipher Extended

Gambar 3.2 merupakan proses dekripsi vigenere cipher extended. Sebenarnya proses dekripsi ini merupakan kebalikan dari proses enkripsi pada gambar 3.1. Jadi ASCII dari cipher teks akan dikurangi dengan key kemudian apabila hasil pengurangan ini lebih kecil dari 0, maka akan ditambah dengan 256.

3.2 KELAS SOCKET

Pada aplikasi vigenere chat sederhana ini digunakan pemrograman socket untuk pengiriman dan penerimaan pesan antara 2 node. Jadi kita mengunakan kelas socket yan sudah terdapat di library java. Java telah memfasilitasi kita untuk melakukan pemrograman socket melalui java.net.package. Socket merupakan cara komunikasi antara dua program melalui jaringan yang sifatnya endpoint. Atau dengan kata lain untuk mencapai jaringan dan dapat menyalurkan data, suatu aplikasi harus melalui socket terlebih dahulu.

Java.net.package sudah menyediakan semua kelas yang dibutuhkan untuk membangun suatu jaringan, dan socket terdapat juga didalamnya. Kelas socket menyediakan socket yang bersifat client-side atau socket yang sederhana.

Di aplikasi Vigenere Chat ini, diterapkan kelas ServerSocket dimana kelas ini menyediakan server socket atau socket di sisi server. Socket ini akan menunggu request dari jaringan. Apabila request tersebut datang, maka server socket akan membungkus option-option required untuk membentuk socket server-side.

Jadi ketika suatu aplikasi vigenere chat ini dijalankan, salah satu aplikasi harus terlebih dahulu melakukan listen dimana aplikasi ini akan menunggu request connect dari aplikasi lain.

Berikut adalah operasi untuk operasi Listen yang diterapkan dalam bahasa Java.

```
private void listen()
{
    try
    {
        // create a server socket
        ServerSocket listenSocket = new ServerSocket(
            new Integer(txtPort.getText()).intValue(), 1);
        try
        {
            // inform user we are listening for a connection
            log("Listening on port " +
                txtPort.getText().trim() + "...");

            // accept an incoming connection
            remoteSocket = listenSocket.accept();

            // set remote machine textbox
            txtRemoteMachine.setText(
                remoteSocket.getInetAddress().getHostName());

            // create the input/output streams
            initStreams(true);
        }
        finally
        {
            // close the listener socket
            listenSocket.close();
        }
      }
        catch(Exception e)
      {
        reportException("listen()", e);
       }
}
```

Gambar 3.3 Operasi Listen Vigenere Chat

Seperti yang terlihat pada gambar 3.3 diatas operasi listen akan membentuk serversocket sesuai dengan port yang telah ditentukan oleh masing-masing aplikasi. Di operasi listen ini, server socket akan memberitahu remote socket untuk menerima connection mealui method listenSocket.

Seperti yang telah dijelaskan pada awal bab ini, connection akan terhubung apabila terdapat sinkronisasi antara operasi listen dan operasi connect. Berikut adalah operasi connect yang diterapkan dalam bahasa java

```
private void connect()
    {
        try
        {
            // inform user connection being attempted
            log("Attempting to contact " + txtRemoteMachine.getText());

            // open a socket to the server
            remoteSocket = new Socket(txtRemoteMachine.getText(), new
Integer(txtPort.getText()).intValue());

            // create the input/output streams
            initStreams(false);
        }
        catch(Exception e)
        {
            reportException("connect()", e);
        }
}
```

Gambar 3.4 Operasi Connect Vigenere Chat

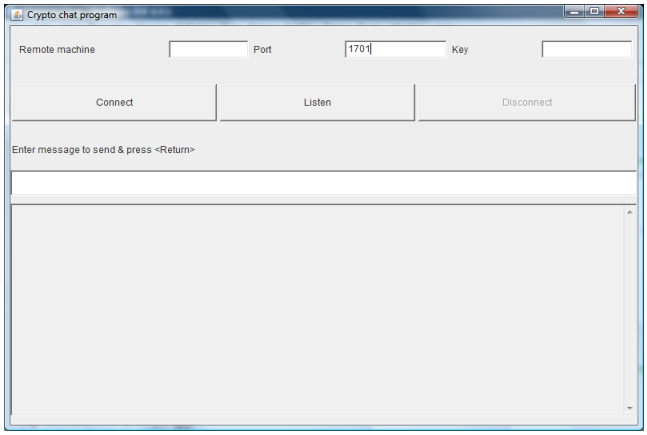
Jadi dapat disimpulkan secara garis besar, aplikasi yang melakukan listening akan menjadi server dan aplikasi yang melakukan connect akan menjadi client. Socket pada sisi client hanya perlu mengetahui host name(nama mesin dimana server ini berjalan) dan port dimana server melakukan listening.

IV.PENGUJIAN

Pada tahap pengujian ini, penulis akan menguji aplikasi vigenere chat yang telah dibuat. Pengujian dibagi menjadi 3 tahap yaitu tahap membangun koneksi antara 2 aplikasi, tahap saling bertukar pesan chat dengan kunci yang sama dan yang terakhir tahap saling bertukar pesan chat dengan kunci yang berbeda.

4.1 PENGUJIAN MEMBANGUN HUBUNGAN KONEKSI ANTAR 2 APLIKASI

Pengujian tahap ini dilakukan dengan menjalankan 2 aplikasi vigenere chat yang telah dibuat. Tampilan aplikasi vigenere chat sederhana dapat dilihat pada gambar 4.1 dibawah ini

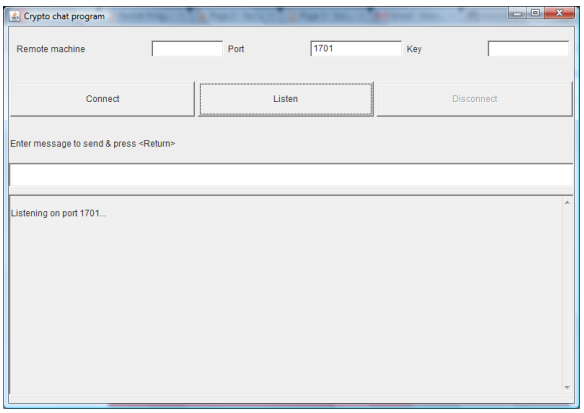


Gambar 4.1 Tampilan Vigenere Chat

Untuk membangun koneksi antara 2 aplikasi, salah satu aplikasi harus melakukan listen yang akan menjadikan aplikasinya sebagai server dan aplikasi lain yang bertindak sebagai client akan melakukan connect yang akan menghubungkannya dengan server. Berikut adalah contoh pembangunan koneksi antara server dan client.

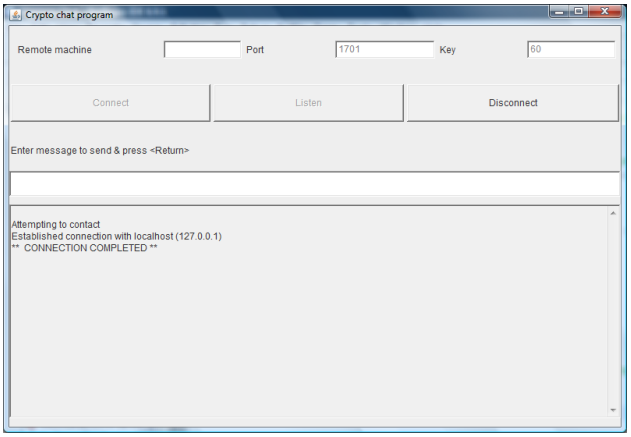
Ketika user mengklik button listen pada aplikasi maka secara otomatis aplikasi tersebut bertindak sebagai server dan menunggu aplikasi lain yang meminta request untuk melakukan connect.

Untuk lebih jelasnya dapat Anda lihat pada gambar 4.2



Gambar 4.2 Listen Aplikasi Vigenere Chat Server

Ketika aplikasi vigenere lain mengklik button connect maka secara otomatis aplikasi tersebut akan terhubung ke server dan bertindak sebagai client.



Gambar 4.2 Connect Aplikasi Vigenere Chat Client

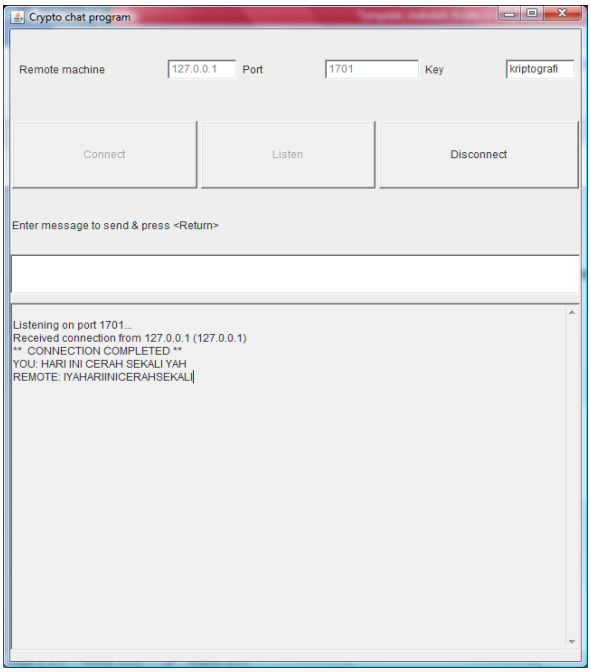
Gambar 4.2 merupakan tampilan ketika aplikasi vigenere chat yang lain yang bertindak sebagai client menekan tombol connect. Pada tahap ini antara aplikasi server dan aplikasi client dapat saling bertukar pesan.

4.2 PENGUJIAN CHAT DENGAN KUNCI YANG SAMA

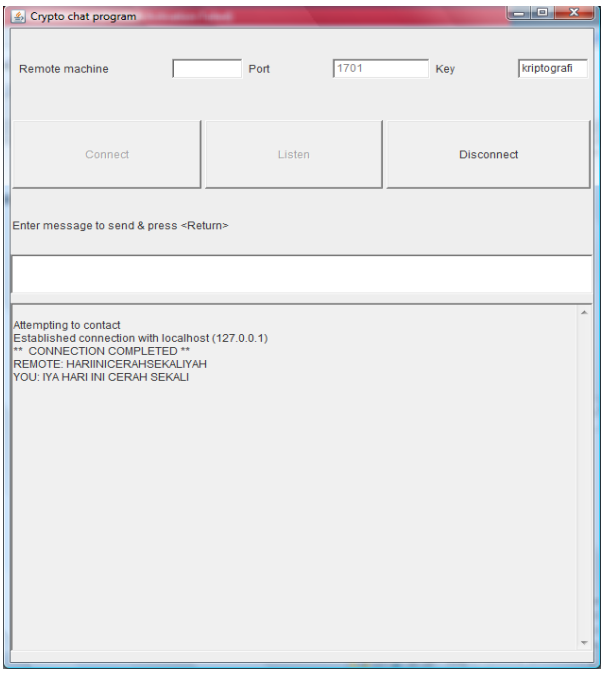
Setelah koneksi antara server dan client terhubung maka server dan client dapat langsung bertukar pesan dengan terlebih dahulu memasukkan kunci yang telah disepakati masing-masing pihak.

Berikut adalah pengujian chat yang dilakukan antara server dan client ini dengan menggunakan kunci yang sama. Tujuan dari pengujian ini ialah untuk membuktikan bahwa metode algoritma yang dipakai yaitu Vigenere Cipher Extended sudah berfungsi dengan benar. Pertukaran pesan dapat dilihat pada gambar dibawah ini:

SERVER : HARI INI CERAH SEKALI YAH
REMOTE : IYA HARI INI CERAH SEKALI
Key : kriptografi



Gambar 4.2 Pengujian Aplikasi Vigenere Chat Server



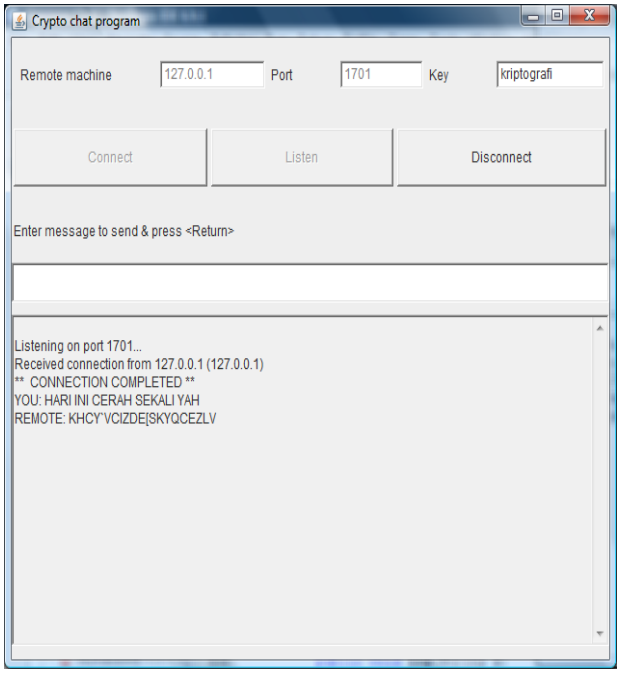
Gambar 4.3 Pengujian Aplikasi Vigenere Chat Client

Pada gambar 4.2 dan gambar 4.3 pertama-tama server mengirimkan pesan chat kepada client yang berisi HARI INI CERAH SEKALI YAH dengan kunci ‘kriptografi’, pesan tersebut sampai di client dengan pesan HARIIINICERAHSEKALIYAH. Kemudian client membalas chat server dengan pesan IYA HARI INI CERAH SEKALI, pesan tersebut sampai ke server dengan pesan IYAHARIIINICERAHSEKALI.

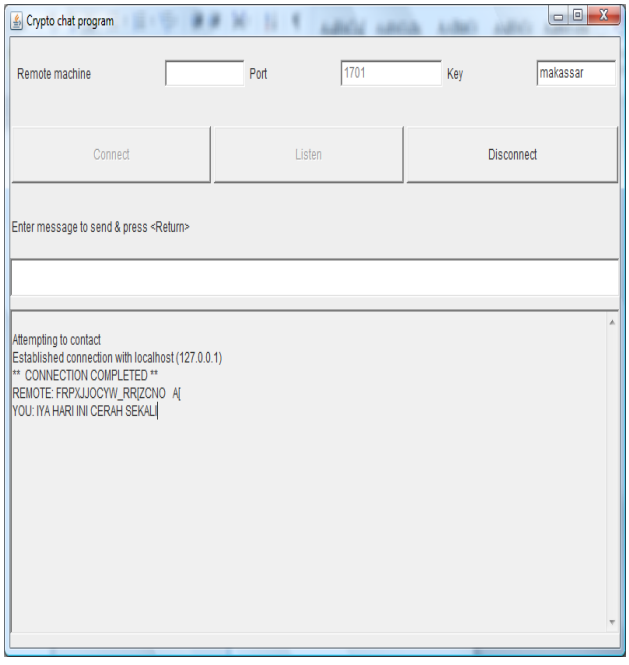
4.2 PENGUJIAN CHAT DENGAN KUNCI YANG BERBEDA

Setelah penujian chat dengan kunci yang sama, sekarang akan dilakukan pengujian dengan kunci yang berbeda.

Server Key : kriptografi
Client Key : makassar



Gambar 4.4 Pengujian Aplikasi Vigenere Chat Server



Gambar 4.5 Pengujian Aplikasi Vigenere Chat Client

Pada gambar 4.4 dan 4.5 pertama-tama server mengirimkan pesan chat kepada client yang berisi HARI INI CERAH SEKALI YAH dengan kunci ‘kriptografi’, di client pesan tersebut diterima dan didekripsi dengan

kunci yang berbeda yaitu ‘makassar’ dan menghasilkan suatu pesan yang sulit untuk dibaca yaitu FRPXJJOCYW_RR[ZCNOA[. Kemudian client membalas chat pesan dari server dengan pesan IYA HARI INI CERAH SEKALI yang akan dienkripsi dengan kunci ‘makassar’. Pesan tersebut sampai ke server dan didekripsi dengan kunci ‘kriptografi’. Pesan yang sudah didekripsi pun ternyata sulit dibaca karena kunci server dan kunci client berbeda. Pesan yang tertera itu berbunyi KHCY`VCIZDE[SKYQCEZLV.

V. KESIMPULAN

Salah satu fitur dari aplikasi instant messaging ialah chat. Dari banyak fitur yang disediakan oleh aplikasi instant messaging yang paling banyak digunakan oleh masyarakat adalah chat. Sayangnya developer aplikasi instant messaging saat ini melupakan satu hal yang penting yaitu masalah keamanan pengiriman data. Seperti pada Yahoo Messenger, pengiriman pesan chat antara satu user dengan user yang lain tidak dilakukan enkripsi sama sekali yang tentu saja rawan apabila pesan yang dikirim yang sifatnya pribadi dan rahasia disadap oleh orang yang tidak diinginkan.

Pada makalah ini, penulis mencoba untuk membuat suatu aplikasi chat messenger sederhana dimana lalu lintas datanya diamankan dengan metode enkripsi klasik yaitu vigenere cipher. Namun metode vigenere cipher ini sekarang sudah obsolete(kuno) karena sudah terdapat metode pemecahannya yaitu dengan metode kasiski. Oleh karena itu penulis menyarankan agar algoritma pengenkripsian yang digunakan kedepannya pada fitur chat instant messaging ialah metode pengenkripsian modern seperti DES atau RSA yang jauh lebih powerful.

REFERENSI

[1] http://en.wikipedia.org/wiki/Instant_messaging

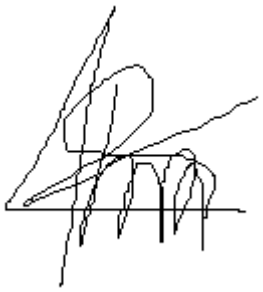
[2] http://www.docstoc.com/docs/20676376/Keamanan-pada-Layanan-Instant-Messaging-Studi-Kasus-Yahoo

[3] http://www.scribd.com/doc/41784523/Implementasi-Sistem-Enkripsi-Pengirim-Pesan-Instan-Dengan-Algoritma-Blowfis

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, 22 Maret 2010



Nur Adi Susliawan D C
13508081