BAB III

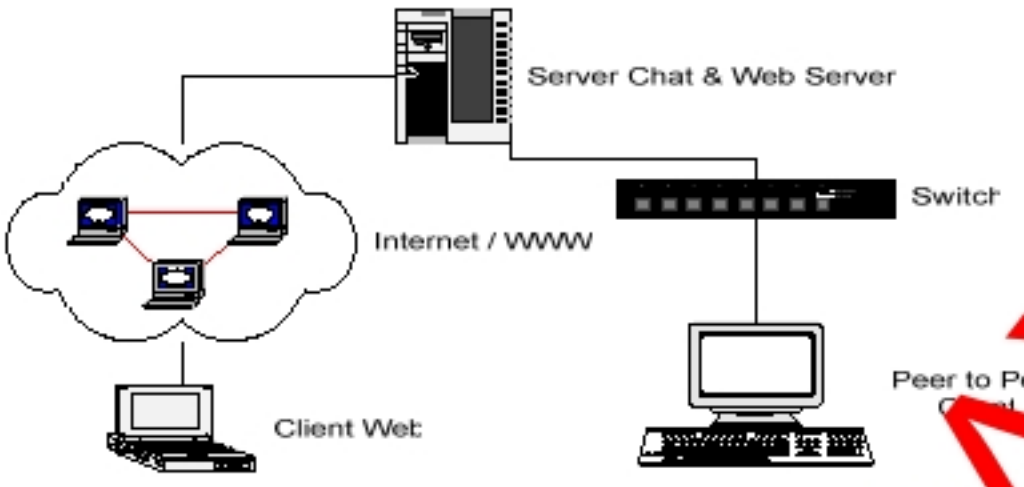
METODE PENELITIAN

Metode penelitian yang digunakan pada pembuatan tugas akhir ini dimulai dengan studi pustaka untuk mencari dan mengumpulkan referensi sebagai bahan pembuatan tugas akhir, baik dalam bentuk *text book* maupun referensi dari internet. Melalui metode ini penulis berusaha untuk mendapatkan informasi dan data sebagai petunjuk dalam menyelesaikan permasalahan yang dihadapi dalam pembuatan tugas akhir.

Dari data – data yang telah diperoleh kemudian dilakukan perancangan, pembuatan serta pengujian sistem untuk mengetahui kinerja dari sistem yang dibuat. Dan yang terakhir dilakuan evaluasi terhadap sistem untuk meningkatkan kinerja sistem jika masih dianggap kurang baik.

3.1 Perancangan Sistem

Sistem yang dibuat nantinya diharapkan dapat diimplementasikan secara nyata pada jaringan komputer yang terhubung langsung dengan internet maupun pada jaringan *Local Area Network* (LAN). Sistem dapat diimplementasikan secara fleksibel pada dua jaringan yang berbeda, yaitu pada jaringan LAN dan internet, program di implementasikan melalui aplikasi berbasis web dan aplikasi desktop.



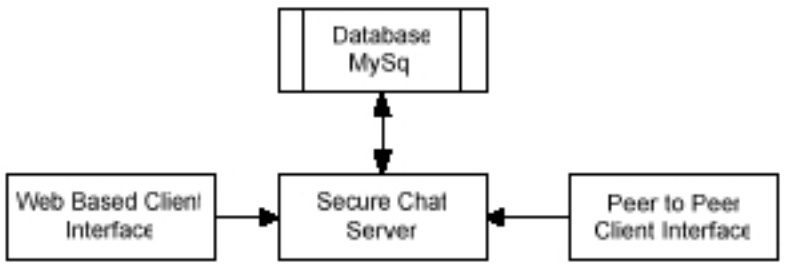
Gambar 3.1 Implementasi sistem

Program ini akan melakukan pengiriman packet data melewati jaringan, sebelum packet data dikirim terlebih dahulu dimonitoring pengirim dan tujuan dari packet data dan mengidentifikasikan jenis algoritma enkripsi yang digunakan, sehingga packet data yang dilewatkan melalui jaringan akan terlebih dahulu di dekripsi sesuai dengan algoritma enkripsi dari pengirim, adapun sebaliknya packet data akan di enkripsi sesuai dengan algoritma enkripsi dari penerima packet data.

3.1.1 MySql 5.0

MySQL adalah aplikasi database open source untuk skala enterprise maupun menengah. Fungsi *database* digunakan untuk menyimpan data user yang pernah menggunakan aplikasi secure chat.

Database digunakan untuk mengetahui user siapa saja yang pernah menggunakan berserta waktu login ke server dan juga dapat meningkatkan keamanan jaringan komputer itu sendiri. Database yang digunakan adalah MySQL versi 5.0. Gambar 3.2 menunjukkan bagaimana database berfungsi untuk menyimpan data user pengguna aplikasi.



Gambar 3.2 MySQL sebagai database user

Struktur database aplikasi yang dibuat memiliki dua buah table yaitu tbluser dan loglogin, tbluser digunakan untuk menyimpan data user yang mendaftar ke server melalui website, tblloglogin digunakan untuk menyimpan data log user yang berhasil login ke server chat. Berikut ini adalah desain struktur database aplikasi secure chat.

Tabel 2. Tabel tblloglogin

Field Name	Type	No Nulls	Key	Default Values
username	VARCHAR(25)	Yes	-	-
waktu	TIME	Yes	-	00:00:00
ipaddress	VARCHAR(18)	Yes	-	-
tanggal	DATE	Yes	-	0000-00-00

Tabel ini berisi daftar user yang pernah login ke server chat

Keterangan :

a. username : Mencatat username yang login ke server
b. waktu : Mencatat jam user yang login ke server
c. ipaddress : Mencatat alamat ipaddress user
d. tanggal : Mencatat tanggal user yang login ke server

Tabel 3. Tabel tbluser

Field Name	Type	Not Nulls	Key	Default Values
username	VARCHAR(25)	Yes	Primary	-
password	VARCHAR(25)	Yes	-	-
fullname	VARCHAR(45)	Yes	-	-
alamat	VARCHAR(50)	Yes	-	-
kota	VARCHAR(45)	Yes	-	-
provinsi	VARCHAR(45)	Yes	-	-
negara	VARCHAR(45)	Yes	-	-
kelamin	VARCHAR(9)	Yes	-	
lahir	DATE	Yes	-	0000-0
pekerjaan	VARCHAR(45)	Yes	-	
email	VARCHAR(60)	Yes		-

Tabel ini berisi daftar user yang te daftar di server chat

Keterangan :

a username : Menc tat na user

b password : Mencatat pass ord user

c fullname : Me catat nama lengkap user

d alama : M ncatat data alamat user

e k ta : M ncatat data kota user

f p vinsi : Mencatat data provinsi user

g negar : Mencatat data negara user

h kelamin : Mencatat data kelamin user

i lahir : Mencatat data tanggal lahir

j pekerjaan : Mencatat data keterangan pekerjaan user

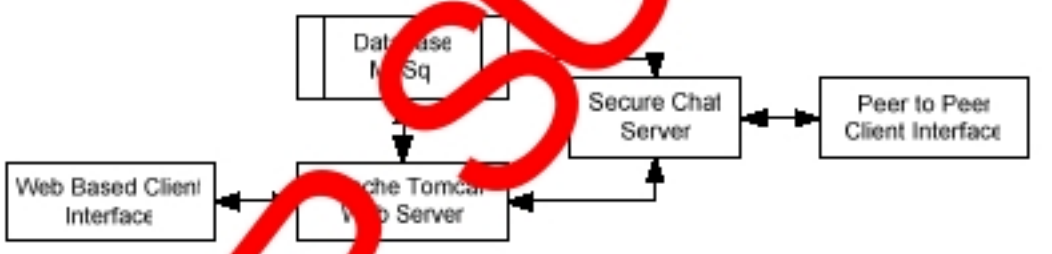
k email : Mencatat alamat email user

3.1.2 Ether Detect Packet Sniffer 1.2

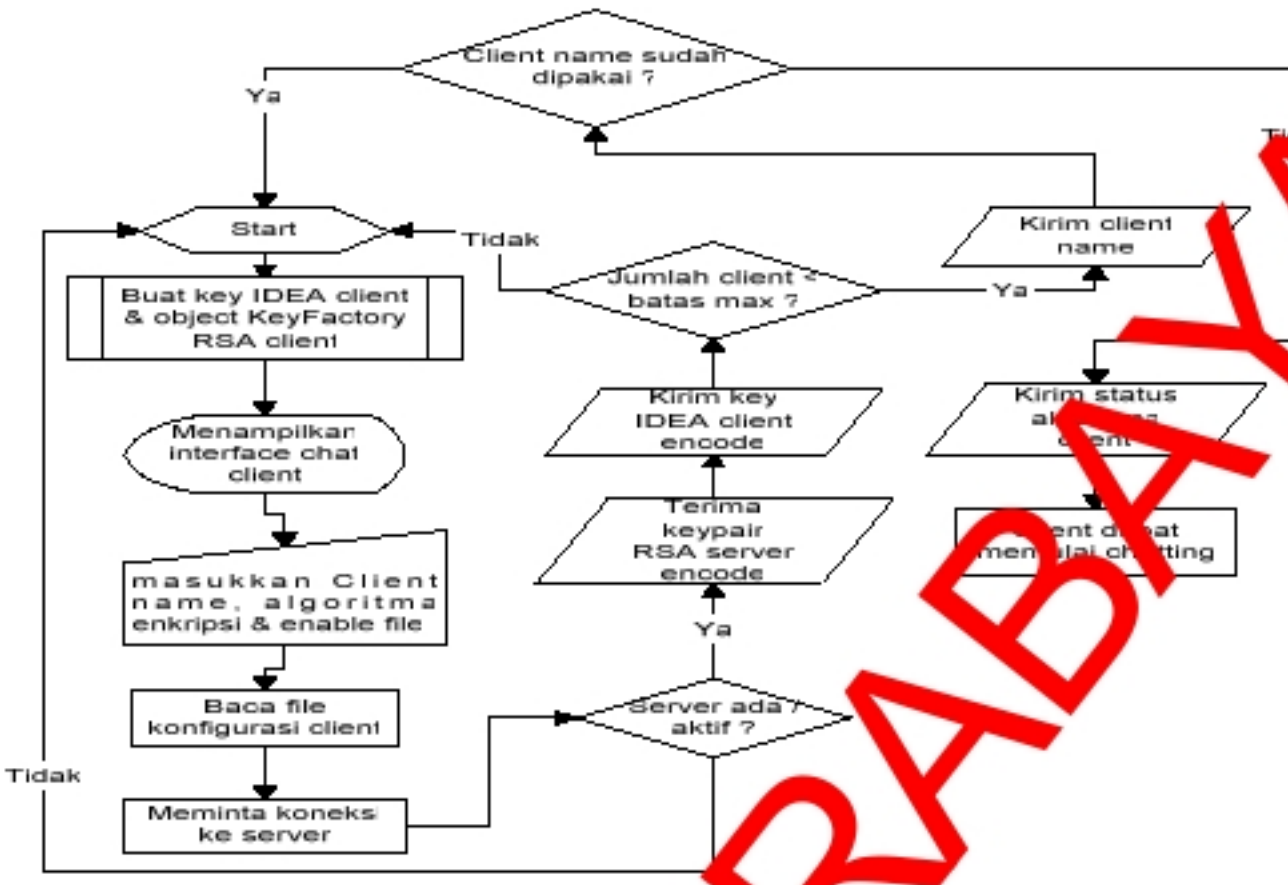
Ether Detect adalah *software* aplikasi sniffing (menyadap) dan monitoring, yang digunakan untuk menyadap packet data yang melewati suatu jaringan tertentu. Software ini akan digunakan untuk membandingkan packet data yang tidak dienkripsi dengan packet data yang telah dienkripsi

3.1.3 Perangkat Lunak

Perangkat lunak yang digunakan dalam tugas akhir ini akan diinstall pada PC Server . Seluruh aktivitas untuk melakukan pemantauan dan pengendalian aplikasi server akan dilakukan disini. Hubungan dari masing–masing perangkat lunak yang terdapat dalam PC Server ditunjukkan pada gambar dibawah.

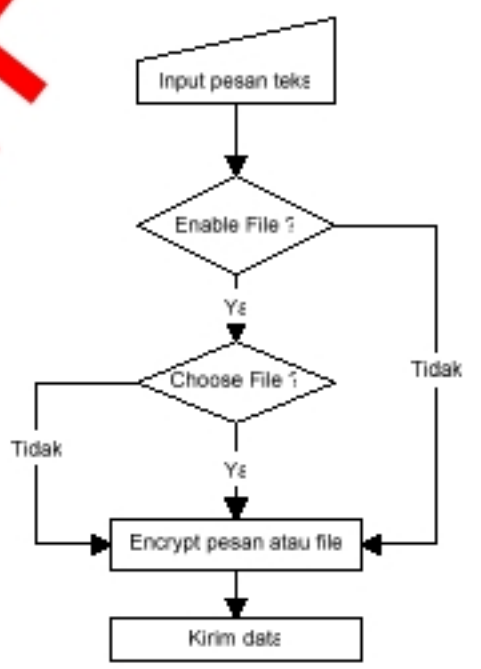


Gambar 3.3 Blok Diagram Sistem



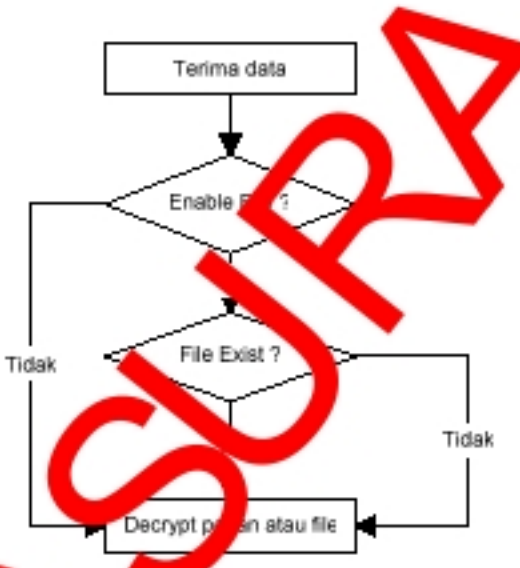
Gambar 3.4 Flowchart aplikasi secure chat client

Gambar 3.5 menjelaskan algoritma pengiriman data untuk aplikasi yang berjalan di desktop yang digunakan *client* :



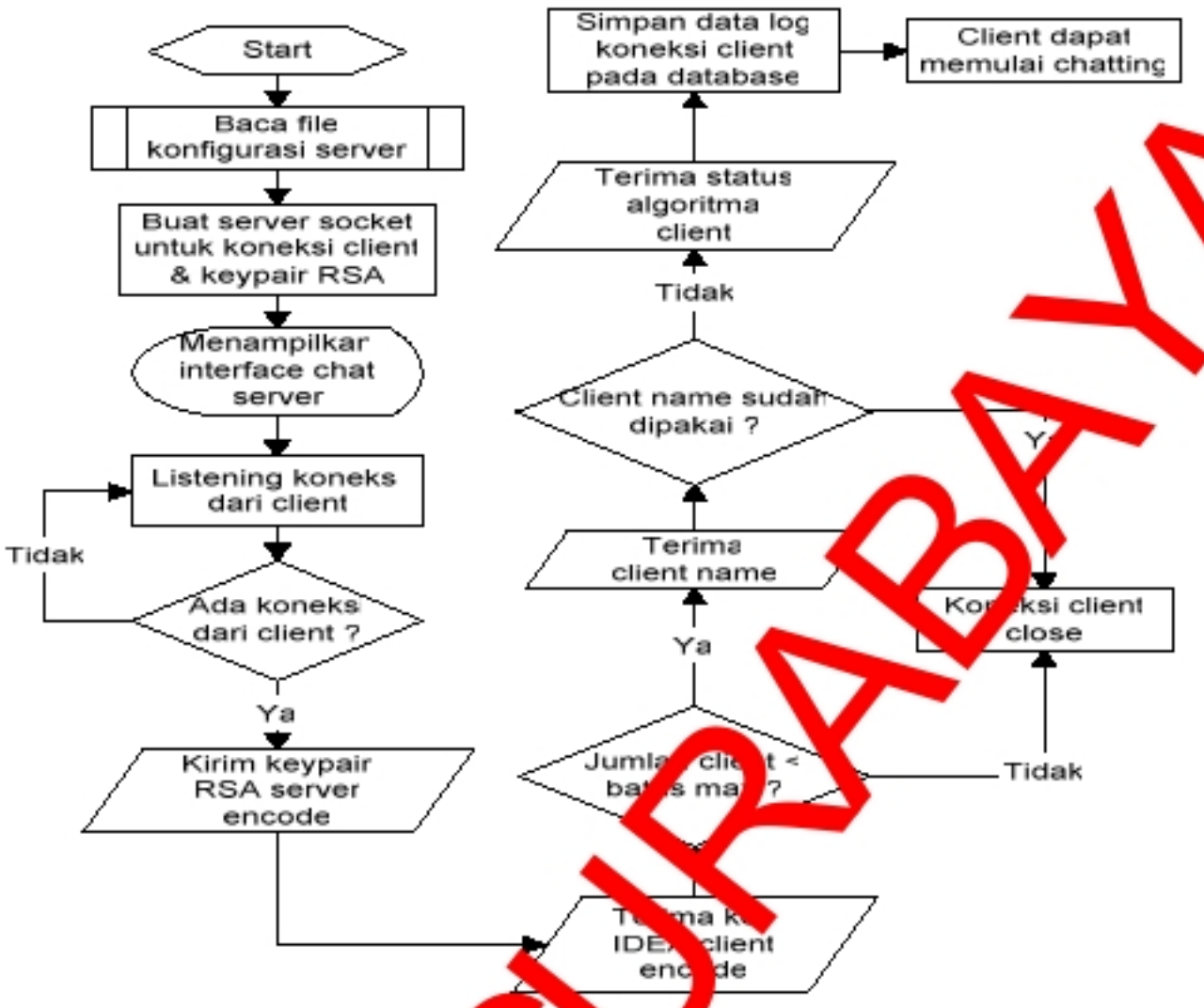
Gambar 3.5 Flowchart pengiriman data aplikasi secure chat client

Cara kerja pengiriman data adalah user memasukkan pesan teks yang akan dikirim kepada user lain, jika user ingin mengirimkan file kepada user lain, maka user sebelumnya harus telah mengaktifkan fitur pengiriman file, jika fitur sudah diaktifkan maka user dapat memilih file yang akan dikirimkan, file yang telah dipilih akan dienkripsi dengan menggunakan algoritma IDEA, sehingga file yang akan melalui jaringan dipastikan aman dari penyadapan, selanjutnya data akan dikirimkan melalui jaringan.



Gambar 3.6 Flowchart penerimaan data aplikasi secure chat client

Adapun sebaliknya data yang diterima dari jaringan akan diverifikasi apakah user penerima telah mengaktifkan fitur pengiriman file, jika fitur telah diaktifkan maka data yang diterima akan diperiksa apakah terdapat file didalamnya, jika ada maka file yang diterima akan didekripsi dengan menggunakan algoritma IDEA sedangkan pesan teks yang diterima akan didekripsi sesuai dengan algoritma enkripsi dari pihak penerima.



Gambar 3.7 Flowchart aplikasi secure chat server

Cara kerja dari sistem yang dibuat adalah : Pertama kali *client* harus menginputkan *client name*, jenis algoritma dan fitur pengiriman *file*, kemudian *client* akan meminta koneksi kepada server, jika server tidak aktif maka akan muncul pesan error, jika server aktif atau server ditemukan maka koneksi *client* akan diterima, kemudian server akan mengirimkan keypair RSA yang telah di encode agar kunci yang ditransfer melalui jaringan dipastikan aman dan tidak dapat dibaca.

Setelah *client* menerima keypair RSA dari server maka *client* akan mengirimkan ke server IDEA *key* dan *byte* iv *secure random* yang sudah di *encode* milik *client* , selanjutnya *client* akan mengirimkan *client name* ke server, server akan melakukan validasi apakah nama *client* sudah terpakai atau belum

dan apakah jumlah *client* masih dibawah batas maksimal *client* yang dapat dihandle server, jika kondisi diatas tidak terpenuhi maka akan muncul pesan konfirmasi error, jika kondisi diatas terpenuhi maka *client* dapat melakukan komunikasi dengan user lain baik pesan bersifat *broadcast* maupun pesan *private*.

3.2 Pembuatan Sistem

3.2.1 Konfigurasi Java

Langkah – langkah yang harus dilakukan agar java dapat mengkompilasi seluruh *file* java adalah sebagai berikut :

1. **Setting library Bouncy Castle :**

   Copy library dari Bouncy Castle dengan nama file **bcprov-jdk15- 131.jar** ke direktory "C:\Program Files\Java\jdk1.5.0_02\jre\lib\ext" dan ke direktory "C:\Program Files\Java\jre1.5.0_02\lib\ext".

2. **Setting Java Security :**

   Tambahkan baris dibawah ini :

   "security.provider.7=org.bouncycastle.jce.provider.BouncyCastleProvider"

   Pada file java.security yang berada pada "C:\Program Files\Java\jdk1.5.0_02\jre\lib\security" dan dalam direktory "C:\Program Files\Java\jre1.5.0_02\lib\security".

3. **Setting Java Servlet :**

   Copy library dari Apache Jakarta Tomcat 5.15 dengan nama file **servlet-api.jar** ke dalam direktory "C:\Program Files\Java\jdk1.5.0_02\jre\lib\ext" dan ke dalam direktory "C:\Program Files\Java\jre1.5.0_02\lib\ext".

4. Setting Database connector MySQL :

Copy library dari MySQL dengan nama file **mysql-connector-java-3.1.12-bin.jar** ke dalam directory "C:\Program Files\Java\jdk1.5.0_02\jre\lib\ext" dan ke directory "C:\Program Files\Java\jre1.5.0_02\lib\ext".

3.2.2 Konfigurasi Apache Jakarta Tomcat

Langkah – langkah yang harus dilakukan untuk menggunakan Jakarta Tomcat adalah sebagai berikut :

1. Menjalankan dan Menghentikan Server

Untuk menjalankan dan menghentikan server Tomcat kita perlu masuk kedalam folder **CATALINA_HOME\bin**. Disana terdapat file **startup.sh** dan **startup.bat** (untuk versi dibawah 5.5) untuk menjalankan server. Gunakan **startup.sh** jika anda berjalan diatas sistem operasi linux, sebaliknya jika anda menggunakan windows anda bisa jalankan **startup.bat** atau dengan menggunakan file **tomcat5w.exe** .

Untuk mematikan server masih di **CATALINA_HOME\bin** terdapat file **shutdown.sh** dan **shutdown.bat**. Gunakan file tersebut untuk mematikan server maka server akan menghentikan servis request-response, melepas semua resource yang dipakai, menuliskan log-log dan akhirnya menutup dirinya sendiri.

Apabila kita berusaha menutupnya dengan paksa tanpa melalui file shutdown tersebut, ada kemungkinan beberapa kegiatan tidak dilakukan. Untuk server Tomcat yang digunakan dalam implementasi system adalah versi 5.15 ke atas jika anda menggunakan system operasi windows. Untuk

menjalankan dan menghentikan server anda hanya perlu mematikan service tomcat.

2. Membuat Virtual Direktori (Context)

Untuk membuat aplikasi web anda memerlukan virtual direktori dan direktori ini akan menjadi home direktori bagi aplikasi, direktori ini biasa disebut dengan **docBase** adapun langkah-langkahnya sebagai berikut :

A. Buat context pada **CATALINA_HOME\conf\Catalina\localhost** dengan nama securewebchat.xml, berikut adalah isi dari file tersebut :

```
<!--
    Context configuration file for the Tomcat Manager Web App
    $Id: manager.xml 303123 2004-08-27 17:03:35Z remm $
-->
<Context    path="/securewebchat"    docBase="D:\SKRIPSI-NOVAN\SKRIPSI-
NOVAN\Program TA\SecureWebChat" reloadable="true">
</Context>
```

B. Membuat struktur direktori sebagai berikut :

docBase
|_____/WEB-INF
|_____web.xml
|_____/classes
|_____ServletName.class
|_____/lib
|_____/app.jar
|_____/src

C. Membuat Web Application Deployment Descriptor, agar aplikasi web dapat berjalan maka kita perlu membuat file web.xml yang menjelaskan kepada server tentang segala hal seperti security dan lain-lain, berikut isi dari file web.xml :

```
<web-app>
  <display-name>SecureWebChat</display-name>
  <description>
     This is Secure Web Chat website using java server page.
  </description>
  <welcome-file-list>
       <welcome-file>index.htm</welcome-file>
       <welcome-file>index.jsp</welcome-file>
 </welcome-file-list>
 </web-app>
```

D. Untuk menjalankan servlet / JSP ketikkan :

http://IpAddressServer/securewebchat

3.2.4 Aplikasi Desktop

Di bawah ini akan dijelaskan mengenai cara kerja dan fungsi dari setiap file JAVA maupun yang berisi listing program yang digunakan pada aplikasi secure chat.

A. ChatRequest.java

File ini merupakan definisi object class ChatRequest yang berisi informasi user yang akan melakukan chat dengan user lain, class ChatRequest memiliki *source code* sebagai berikut :

```
import java.io.*;
public class ChatRequest implements Serializable{
        public int senderId;
        public int recieverId; }
```

B. Client Info.java

File ini merupakan definisi object class Client Info yang berisi informasi tentang user, class Client Info memiliki *source code* sebagai berikut :

```
import java.io.*;
import java.security.*;
import java.security.spec.*;
import javax.crypto.*;
import javax.crypto.spec.*;
public class ClientInfo implements Serializable{
        public int client Id;
        public String client Name;
        public byte[] publicKeyPair; //RSA public key
        public byte[] privateKeyPair; // RSA private key
        public Key key; // IDEA key information
        public byte[] iv; // IDEA key information
        public boolean algorithm; /* true=RSA , false=IDEA */}
```

C. ConnectionNotice.java

File ini merupakan definisi object class ConnectionNotice yang berisi informasi status koneksi *client*, class ConnectionNotice memiliki *source code* sebagai berikut :

H. ServerShutDown.java

File ini merupakan definisi object class **ServerShutDown** yang digunakan server ketika server dimatikan, class ServerShutDown memiliki *source code* sebagai berikut :

```
import java.io.*;
public class ServerShutDown implements Serializable{}
```

I. UpdateList.java

File ini merupakan definisi object class UpdateList yang berisi informasi adanya klien baru bergabung, class UpdateList memiliki *source code* sebagai berikut :

```
import java.io.*;
public class UpdateList implements Serializable{
        public boolean requestType;
        public String newClient ; }
```

J. UserIDEAKey.java

File ini merupakan definisi object class UserIDEAKey yang berisi informasi key dari algoritma enkripsi IDEA yang digunakan klien, class UserIDEAKey memiliki *source code* sebagai berikut :

```
import java.security.*;
import javax.crypto.*;
import java.io.*;
import javax.crypto.spec.*;
public class UserIDEAKey implements Serializable{
        public Key ideaKey; // IDEA secret key
        public byte[] iv; /* IDEA iv spec */}
```

K. UserRSAKey.java

File ini merupakan definisi object class UserRSAKey yang berisi informasi key dari algoritma enkripsi RSA yang digunakan klien, class UserRSAKey memiliki *source code* sebagai berikut :

```
import java.security.*;
import java.io.*;
public class UserRSAKey implements Serializable{
        public byte[] publicKeyPair; //RSA public key pair
        public byte[] privateKeyPair; /* RSA private key pair */}
```

L. FileNotice.java

File ini merupakan definisi object class FileNotice yang berisi status apakah user diperkenankan melakukan pengiriman file yang telah terenkripsi, file ini hanya digunakan pada aplikasi klien, class FileNotice memiliki *source code* sebagai berikut :

```
import java.io.*;
public class FileNotice implements Serializable{
        public boolean status; /*false=disable file, true=enable file */}
```

M. EncryptFile.java

File ini merupakan definisi object class EncryptFile yang berfungsi untuk melakukan enkripsi dan dekripsi file dengan menggunakan algoritma enkripsi IDEA, *source code* class EncryptFile dapat dilihat pada lampiran buku.

N. ServerInterface.java

File ini merupakan definisi object class ServerInterface yang berisi langkah-langkah pembuatan interface untuk server, class ServerInterface merupakan turunan dari class JFrame, class ServerInterface digunakan sebagai interface untuk aplikasi utama chatting (chat server) yang berjalan di server, *source code* class ServerInterface dapat dilihat pada lampiran buku.

O. Client Interface.java

File ini merupakan definisi object class *Client Interface* yang berisi langkah-langkah pembuatan interface untuk client , class *Client Interface* merupakan turunan dari class *JFrame*, *class Client Interface* digunakan sebagai *interface* untuk aplikasi utama chatting untuk *user* (*chat client*) yang berjalan di komputer *client* , *source code* class *Client Interface* dapat dilihat pada lampiran buku.

P. ChatServer.java

File ini merupakan definisi object class *ChatServer* yang berisi langkah-langkah kerja server guna menangani permintaan *chatting* dari client , class *ChatServer* merupakan program yang bekerja *multithread* (mampu menangani lebih dari satu proses secara bersama-sama). Pada class ini bertugas mengendalikan seluruh proses *chatting* dari *client* dan bekerja pada komputer *server*, *source code class* ChatServer dapat dilihat pada lampiran buku.

Q. ChatClient .java

File ini merupakan definisi *object class ChatClient* yang berisi langkah-langkah kerja aplikasi *client* guna menangani permintaan *user* agar dapat melakukan *chatting* dengan *user* lain melalui *server*, *class ChatClient* merupakan program yang ini bertugas mengendalikan seluruh proses *chatting* pada *client* dan bekerja pada komputer *client* , *source code class* ChatClient dapat dilihat pada lampiran buku.

R. WebClient Interface.java

File ini merupakan definisi object class *WebClient Interface* yang berisi langkah-langkah pembuatan interface untuk client , class *WebClient Interface* merupakan turunan dari class *JPanel*, *class WebClient Interface* digunakan sebagai *interface* untuk aplikasi utama chatting untuk *user* (*chat client*) yang berjalan di komputer *client* , *source code class WebClient Interface* dapat dilihat pada lampiran buku.

S. Client Applet.java

File ini merupakan definisi *object class Client Applet* yang berisi langkah-langkah kerja aplikasi *client* guna menangani permintaan *user* agar dapat melakukan *chatting* dengan *user* lain melalui *server*, *class Client Applet*

merupakan program yang ini bertugas mengendalikan seluruh proses *chatting* pada *client* dan bekerja pada komputer *client* , class *Client Applet* merupakan turunan dari class *JApplet, source code class Client Applet* dapat dilihat pada lampiran buku.

3.2.4 Aplikasi Website

Di bawah ini akan dijelaskan mengenai cara kerja dan fungsi dari setiap file Java Server Page (JSP) yang berisi listing program yang digunakan pada aplikasi secure chat berbasis web.

A. index.jsp

File ini merupakan file utama/default tampilan dari aplikasi secure chat. Untuk menggunakan aplikasi secure chat, *user* harus melakukan pendaftaran terlebih dahulu, setelah *user* mendaftarkan *username* dan *password*, maka dapat melakukan login ke chat server.

B. about.jsp

File ini merupakan file JSP yang berisikan tampilan informasi tentang aplikasi secure web chat.

C. aboutus. jsp

File ini merupakan file JSP yang berisikan tampilan informasi tentang webmaster secure web chat.

D. application. jsp

File ini merupakan file JSP yang berisikan tampilan informasi link halaman website untuk dapat menggunkan aplikasi secure web chat, mendapatkan dokumentasi aplikasi dan mendownload aplikasi. Untuk dapat memasuki halaman ini, user harus terlebih dahulu login ke server.

E. connect. jsp

File ini merupakan file JSP yang berisikan langkah-langkah validasi username dan password user yang login ke server.

F. contact. jsp

File ini merupakan file JSP yang berisikan tampilan informasi tentang *contact person website*

G. development. jsp

File ini merupakan file JSP yang berisikan tampilan informasi tentang perkembangan aplikasi *secure chat*

H. how2login. jsp

File ini merupakan file JSP yang berisikan tampilan informasi tentang cara-cara seorang user dapat melakukan login ke server.

I. how2use. jsp

File ini merupakan file JSP yang berisikan tampilan informasi tentang penggunaan aplikasi secure web chat.

J. support. jsp

File ini merupakan file JSP yang berisikan tampilan informasi tentang *contact person webmaster*.

K. document. jsp

File ini merupakan file JSP yang berisikan tampilan informasi link halaman website untuk dapat mendapatkan dokumentasi aplikasi. Untuk dapat memiliki halaman ini, user harus terlebih dahulu login ke server

L. source. jsp

File ini merupakan file JSP yang berisikan tampilan informasi link halaman website untuk dapat mendownload aplikasi. Untuk dapat memasuki halaman ini, user harus terlebih dahulu login ke server

M. signup. jsp

File ini merupakan file JSP yang berisikan tampilan informasi form bagi user yang ingin mendaftar sebagai *member* website secure chat

N. otentifikasi. jsp

File ini merupakan file JSP yang memproses data user baru yang mendaftar sebagai member website secure chat.

O. profile. jsp

File ini merupakan file JSP yang berisikan tampilan default *user* yang telah berhasil *login* ke *website secure chat*. *source code* file profile.jsp dapat dilihat pada lampiran buku.

P. editprofile. jsp

File ini merupakan file JSP yang berisikan tampilan *form* bagi user yang ingin melakukan perubahan data pribadi *user website secure chat*.

Q. update. jsp

File ini merupakan file JSP yang memproses data user yang telah terdaftar sebagai member yang melakukan perubahan data pribadi user

R. errorlogin. jsp

File ini merupakan file JSP yang berisikan tampilan pesan error user yang gagal login ke server.

S. logout. jsp

File ini merupakan file JSP yang berisikan tampilan default user yang keluar dari website secure web chat, user yang telah keluar diharuskan login lagi agar dapat menggunakan aplikasi secure chat

T. forgot. jsp

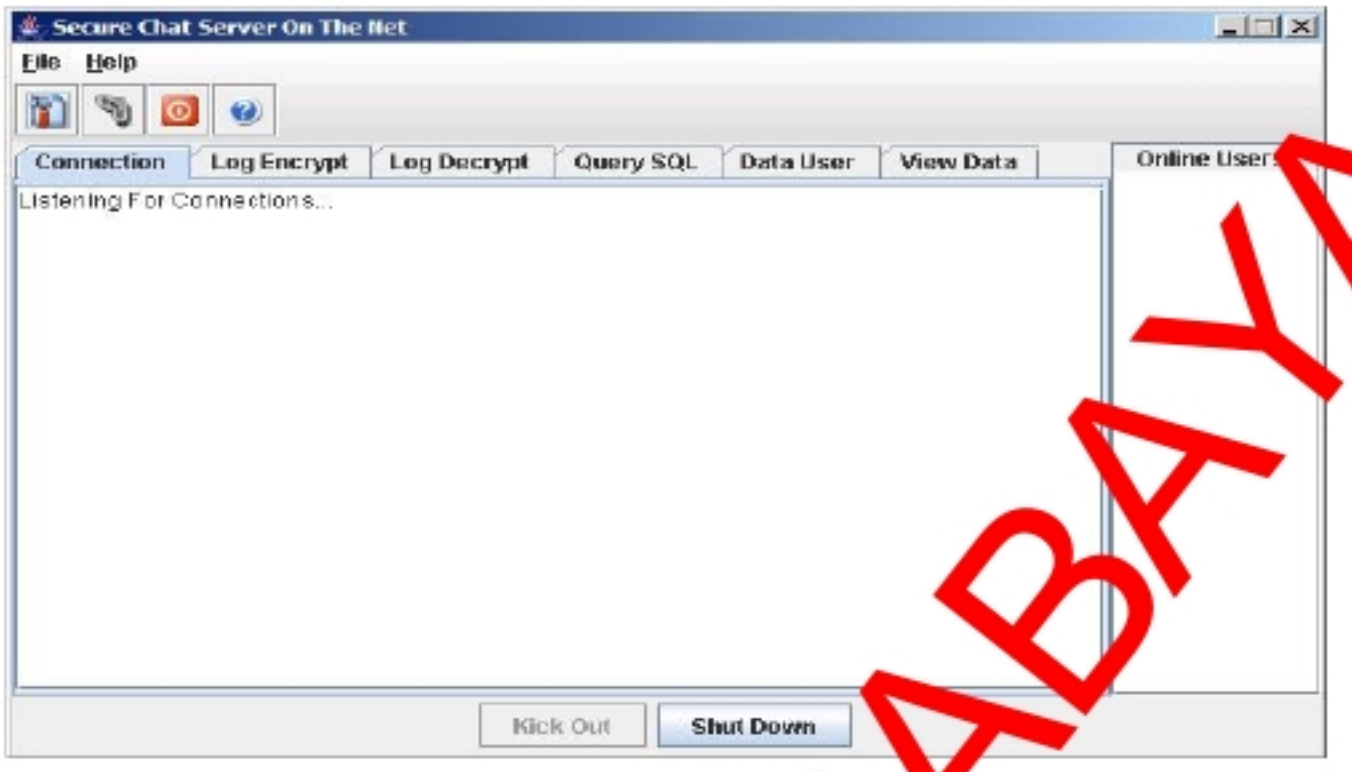
File ini merupakan file JSP yang berisikan tampilan form yang diisi user yang sudah terdaftar sebagai member. Apabila user lupa dengan informasi username serta password miliknya

3.3 Desain Interface Aplikasi

Desain interface aplikasi memiliki peranan penting dalam aplikasi yang dibuat, interface aplikasi harus dapat memberikan kemudahan untuk digunakan oleh user. Berikut akan digambarkan desain interface aplikasi baik yang bekerja pada sisi server maupun pada sisi *client* .

3.3.1 Desain Interface Chat Server

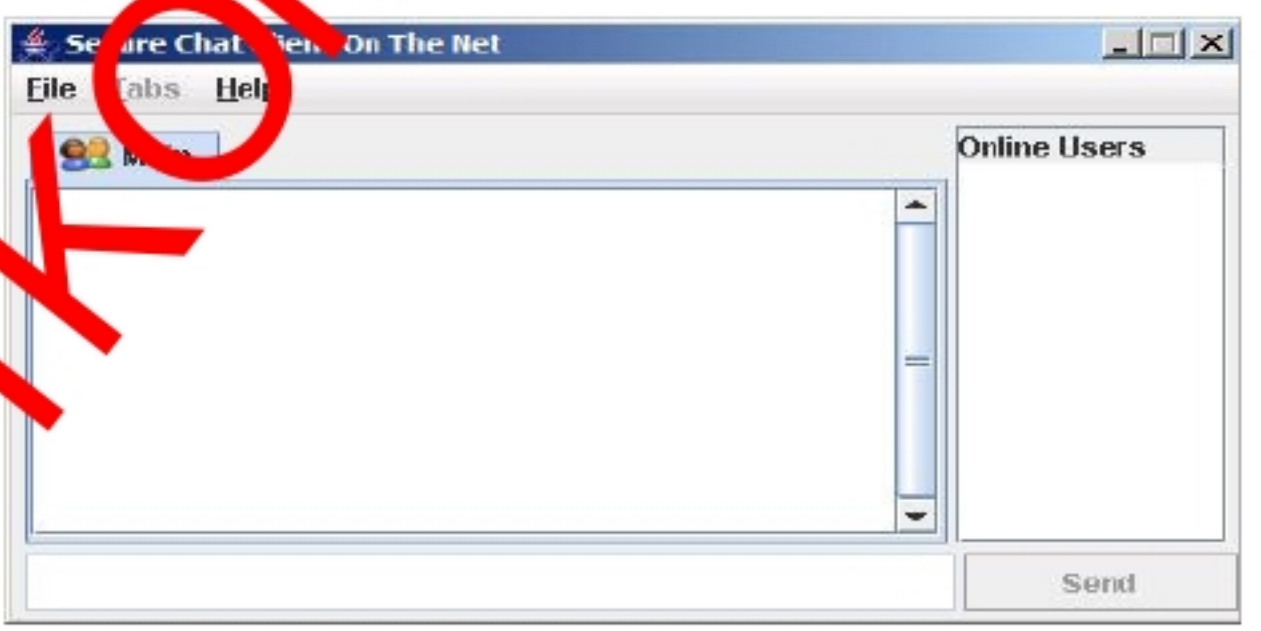
Gambar 3.8 adalah tampilan dari aplikasi secure chat yang berjalan di server, *client* dan web *browser*.



Gambar 3.8 Secure Chat S...

Aplikasi Secure Chat Server adalah aplikasi desktop yang berjalan pada sisi server, melalui aplikasi ini seorang administrator server dapat melakukan maintenance maupun konfigurasi terhadap server chat, seperti : mengubah konfigurasi server, menghapus user yang sedang login dan lain-lain.

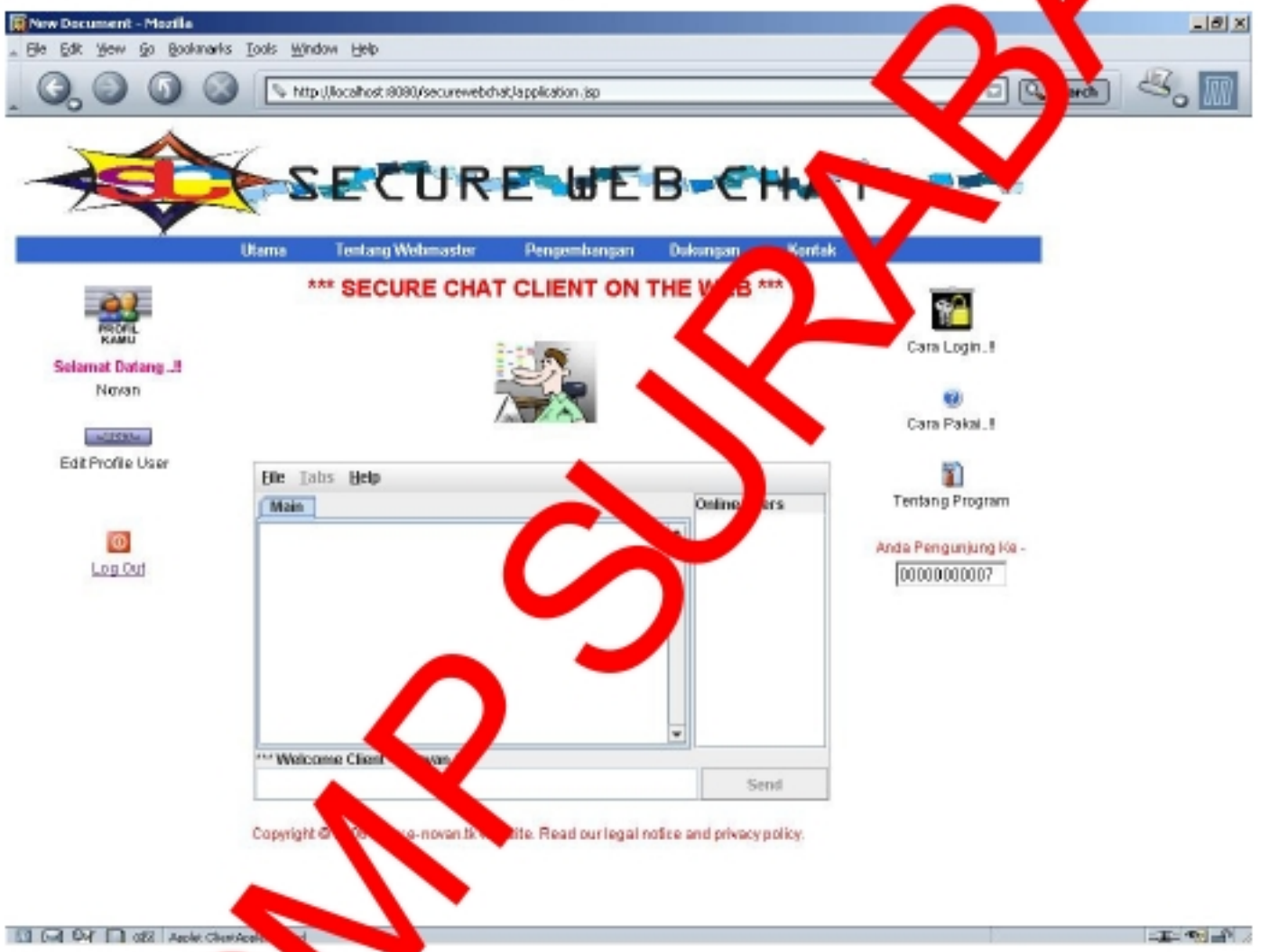
3.3.2 Desain Interface Chat Client



Gambar 3.9 Secure Chat Client

Aplikasi Secure Chat Client adalah aplikasi desktop yang berjalan pada sisi client , melalui aplikasi ini seorang user dapat melakukan chatting dengan terlebih dahulu login ke server.

3.3.3 Desain Interface Web Chat



Gambar 3.10 Web Chat Client

Aplikasi Secure Web Chat adalah aplikasi berbasis web yang hanya dapat diakses melalui web browser, melalui aplikasi ini seorang user dapat melakukan chatting dengan terlebih dahulu mendaftarkan diri sebagai member secure web chat website agar dapat login ke server chat.