# BAB 4
# IMPLEMENTASI DAN PENGUJIAN

## 4.1 Implementasi

Tahap implementasi merupakan tahap penerapan dari analisis yang sudah dilakukan. Penerapan metode yang digunakan pada penelitian, selanjutnya akan diimplementasi pada setiap perangkat yang terlibat untuk menghasilkan sistem yang sesuai dengan tahap perancangan.

### 4.1.1 Implementasi VoIP *Server*

VoIP *server* adalah komputer yang menyediakan layanan VoIP. VoIP *server* yang digunakan yaitu Asterisk yang berjalan pada sistem operasi Ubuntu 12.04 LTS. Spesfikasi perangkat keras komputer sebagai VoIP *server* adalah sebagai berikut:

**Tabel 4.1 Spesfikasi Perangkat Keras Komputer Sebagai VoIP *Server***

| No | Perangkat Keras | Spesifikasi |
|---|---|---|
| 1 | Prosesor | 2.30GHz |
| 2 | Memori | 1GB |
| 3 | Harddisk | 10GB |
| 4 | Sistem Operasi | Ubuntu 12.04 LTS |

#### 4.1.1.1 Instalasi VoIP *Server*

Instalasi Asterisk *Server* dimulai dengan dengan meng-*update* dan meng-*upgrade* Ubuntu terlebih dahulu. Ini dimaksudkan agar Ubuntu dapat mengakses berbagai *library* yang nantinya akan di gunakan untuk instalasi asterisk *server*. *Upgrade* dan *update* dapat dilakukan dengan menjalankan perintah "apt-get". Langkah untuk meng-install Asterisk *Server* adalah sebagai berikut :

1. Pada terminal, ketikan perintah sudo –s untuk mengakses hak *root* pada Ubuntu.
2. Setelah hak akses *root* didapatkan kemudian ketikkan perintah apt-get update && upgrade –y && reboot.

3. Setelah proses *update* dan *upgrade* selesai maka dilanjutkan dengan meng-instal *library-library* yang nantinya akan digunakan untuk proses instalasi Asterisk *Server*. Untuk meng-install *library-library* tersebut ketikkan perintah apt-get instal build-essential wget libssl-dev libncurses5-dev libnewt-dev libxml2-dev linux-headers-$(uname –r) libsqlite3-dev uuid-dev.
4. Setelah installasi *library* selesai kemudian ketikkan perintah cd /usr/src. Ini dimaksudkan agar Ubuntu berpindah *drive* ke usr/src.
5. Kemudian *download* Dahdi Linux, Libpri, serta Asterisk *Server* dengan mengetikkan perintah wget http://downloads.asterisk.org/pub/telephony/dahdi-linux-complete/dahdi-linux-complete-current.tar.gz, wget http://downloads.asterisk.org/pub/telephony/libpri/libpri-1.4-current.tar.gz , dan wget http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-1.8.32.3.tar.gz .
6. Setelah *download* selesai *extract* Dahdi Linux, Libpri, dan Asterisk *Server* dengan mengetikkan perintah tar –zxvf dahdi-linux-complete-current.tar.gz, tar –zxvf libpri-1.4-current.tar.gz, dan tar –zxvf asterisk-1.8.32.3.tar.gz.
7. Kemudian masuk ke dalam *directory* Dahdi Linux dengan mengetikkan perintah cd dahdi-linux-complete-2.10.2+2.10.2/ lalu install Dahdi Linux dengan mengetikkan perintah make && make install && make config.
8. Setelah installasi dahdi linux selesai kemudian dilanjutkan dengan melakukan proses installasi Libpri dengan mengetikkan perintah cd libpri-1.4.15/ dan dilanjutkkan dengan meng-install Libpri dengan mengetikkan perintah make && make install.
9. Setelah proses installasi Libpri selesai dilanjutkan dengan melakukan proses installasi *Asterisk Server* dengan cara mengetikkan perintah cd asterisk-1.8.32.3/ dan dilajutkan dengan meng-install *Asterisk Server* dengan mengetikkan perintah ./configure && make menuselect && make && make install && make config && make samples.

10. Setelah installasi selesai kemudian jalankan Dahdi Linux dan *Asterisk Server* dan dengan mengetikkan /etc/init.d/dahdi start, dan /etc/init.d/asterisk start.
11. Jika Dahdi Linux dan *Asterisk Server* sudah berjalan kemudian jalankan *asterisk* CLI dengan mengetikkan perintah asterisk –rvvv.

#### 4.1.1.2 Pembuatan SIP ID

SIP ID digunakan sebagai pengalamatan pada VoIP *server*. Pengalamatan SIP ID yang digunakan yaitu sebagai berikut;

**Tabel 4.2 Pengalamatan SIP ID**

| No. | IP Address | SIP ID | Keterangan |
|---|---|---|---|
| 1 | 192.168.20.11 | 1001 | *Client* 1 |
| 2 | 192.168.20.12 | 1002 | *Client* 2 |

Pembuatan SIP ID pada Asterisk dilakukan dengan cara mengubah file sip.conf dengan perintah "gedit /etc/asterisk/sip.conf" dan untuk pengaturan akunnya dengan cara mengubah file extensions.conf dengan perintah "gedit /etc/asterisk/extensions.conf". Berikut ini adalah langkah dalam pembuatan SIP ID:

1. Pada terminal, buka file "sip.conf" dengan menggunakan perintah "gedit /etc/asisk/sip.conf untuk membuat akun SIP.
2. Setelah file terbuka, tambahkan konfigurasi pada [general]:

```
[general]
context=internal
allowguest=no
allowoverlap=no
bindport=5060
bindaddr=0.0.0.0
srvlookup=no
dissalow=all
allow=ulaw
alwaysauthreject=yes
canreinvite=no
nat=yes
session-timers=refuse
localnet=192.168.20.0/255.255.255.0
```

3. Untuk menambahkan SIP ID 1001 sebagai *Client* 1, berikut konfigurasinya:

```
[1001]
type=friend
host=dynamic
secret=Skatel123
```

```
context=internal
encrypted=yes
```

4. Untuk menambahkan SIP ID 1002 sebagai *Client* 2, berikut konfigurasinya:

```
[1002]
type=friend
host=dynamic
secret=Skatel123
context=internal
encrypted=yes
```

5. Simpan file dengan cara menekan ctrl+s

sip.conf

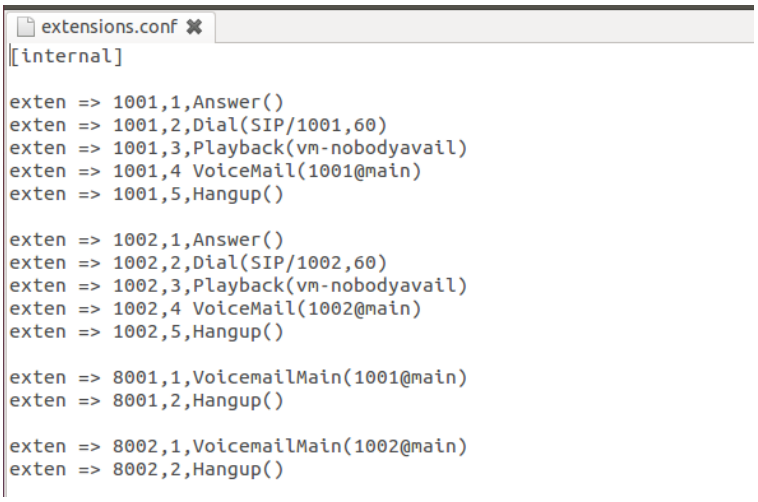
```
[general]
context=internal
allowguest=no
allowoverlap=no
bindport=5060
bindaddr=0.0.0.0
srvlookup=no
dissalow=all
allow=ulaw
alwaysauthreject=yes
canreinvite=no
nat=yes
session-timers=refuse
localnet=192.168.20.0/255.255.255.0

[1001]
type=friend
host=dynamic
secret=Skatel123
context=internal
encrypted=yes


[1002]
type=friend
host=dynamic
secret=Skatel123
context=internal
encrypted=yes
```

**Gambar 4.1 Konfigurasi File sip.conf**

6. Setelah konfigurasi sip.conf selesai, kemudian dilanjutkan dengan mengubah file “extensions.conf” dengan menggunakan perintah “gedit /etc/asisk/ extensions.conf untuk konfigurasi akun SIP.
7. Setelah file terbuka, tambahkan konfigurasi pada [internal]. Ini ditujukan agar setiap akun dapat melakukan perintah-perintah seperti: menerima telepon, memutuskan panggilan telepon dan lain-lain. Dapat dilihat pada Gambar 4.2




**Gambar 4.2 Konfigurasi File extensions.conf**

### 4.1.2 Implementasi *Class*

Implementasi *class* pada aplikasi *SSipCall* merupakan implementasi dari analisis kelas yang ada pada *class diagram*. File-file kelas yang ada pada aplikasi ini menggunakan ekstensi .java. berikut ini adalah kelas-kelas yang telah dibuat beserta nama fisiknya:

**Tabel 4.3 Implementasi *Class* Pada Aplikasi *SSipCall***

| No. | Nama *Class* | Nama File |
|---|---|---|
| 1 | WalkieTalkieActivity | WalkieTalkieActivity.java |
| 2 | SipSettings | SipSettings.java |
| 3 | IncomingCallReceiver | IncomingCallReceiver.java |
| 4 | DBAdapter | DBAdapter.java |
| 5 | PjSipAccount | PjSipAccount.java |
| 6 | PjSipService | PjSipService.java |
| 7 | DatabaseHelper | DatabaseHelper.java |
| 8 | csipsimple_acc_config | csipsimple_acc_config.java |
| 9 | pjmedia_srtp_use | pjmedia_srtp_use.java |
| 10 | pjsua_acc_config | pjsua_acc_config.java |
| 11 | Pjsua | pjsua.java |
| 12 | SwigNext | SwigNext.java |
| 13 | pjsuaConstants | pjsuaConstants.java |
| 14 | csipsimple_config | csipsimple_config.java |
| 15 | pjsua_config | pjsua_config.java |
| 16 | pjsua_logging_config | pjsua_logging_config.java |
| 17 | pjsua_media_config | pjsua_media_config.java |

### 4.1.3 Implementasi VoIP *Client*

Perangkat keras yang digunakan untuk mengimplementasikan sistem pada aplikasi *SSipCall* dan *Linphone* dapat dilihat pada tabel dibawah ini:

**Tabel 4.4 Spesfikasi Perangkat Keras Minimum *Smartphone* Sebagai VoIP *Client***

| No | Perangkat Keras | Spesifikasi |
|---|---|---|
| 1 | Dimensi layar | 3,5” |
| 2 | RAM | 512MB |
| 3 | Memory Internal | 1GB |
| 4 | Prosesor | 800Mhz |
| 5 | Resolusi | 480 x 320 pixel |
| 6 | Sistem Operasi | GingerBread 2.3 |

**Tabel 4.5 Spesfikasi Perangkat Keras Komputer Sebagai VoIP *Client***

| No | Perangkat Keras | Spesifikasi |
|---|---|---|
| 1 | Prosesor | i3 2.30Ghz |
| 2 | Memori | 2GB |
| 3 | Harddisk | 20GB |
| 4 | Sistem Operasi | Windows 7 |

#### 4.1.3.1 Pendaftaran SIP ID Pada *SSipCall*

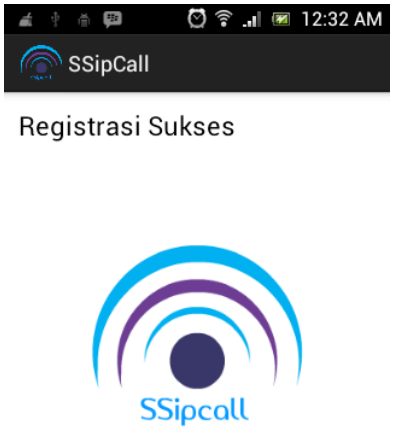
SIP ID yang telah dibuat pada VoIP *server* lalu diimplementasikan pada VoIP *client* untuk android. Berikut ini adalah langkah untuk mengimplementasikan SIP ID *client* 1 pada VoIP *client:*

1. Pada *SSipCall* pilih menu pengaturan akun, lalu akan muncul halaman Login.
2. Pada halaman Login terdapat *field* ID Pengguna, *Server*, Kata Sandi dan Keamanan. Lihat gambar 4.3




**Gambar 4.3 Halaman Login Pada *SsipCall***

3. Isi field ID Pengguna = 1001, *Server* = 192.168.20.1, dan Kata Sandi = Skatel123
4. Setelah SIP ID telah berhasil diimplementasikan pada halaman utama akan muncul pemberitahuan bahwa registrasi sukses seperti pada gambar 4.4

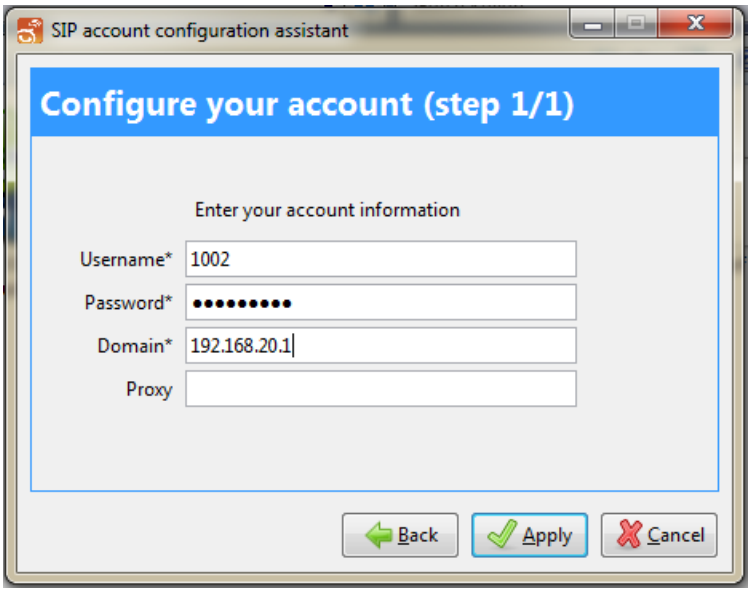



**Gambar 4.4 Status Registrasi Sukses**

**4.1.3.2** Pendaftaran SIP ID Pada Aplikasi *Lindphone*

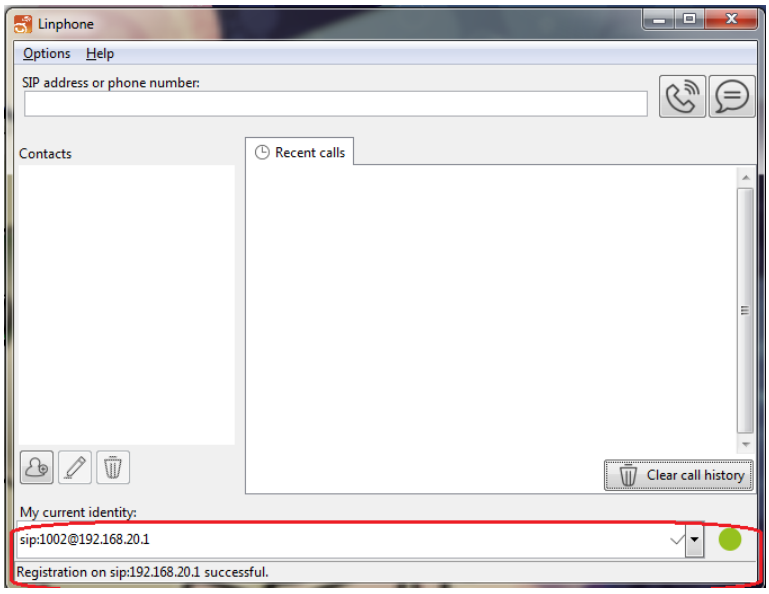
SIP ID yang telah dibuat pada VoIP *server* lalu diimplementasikan pada VoIP *client* untuk windows. Berikut ini adalah langkah untuk mengimplementasikan SIP ID *client* 2 pada VoIP *client*:

1. Pada apliksi Linphone, buka menu *Options* -> *Preferences* -> *Wizard* -> *Forward* -> *I have already sip account and I just want to use it* -> *Forward*.
2. Pada jendela SIP *Account Configuration Assistant* isikan *username* = 1002, *Password* = Skatel123, *Domain* = 192.168.20.1. Seperti pada Gambar berikut




**Gambar 4.5 SIP *Account Configuration Assistant***

3. Lalu pilih Apply -> Close -> Done
4. Setelah SIP ID telah berhasil diimplementasikan akan muncul pemberitahuan Registration on SIP:192.168.20.1 pada jendela utama Linphone seperti pada gambar 4.6




**Gambar 4.6 *Linphone***

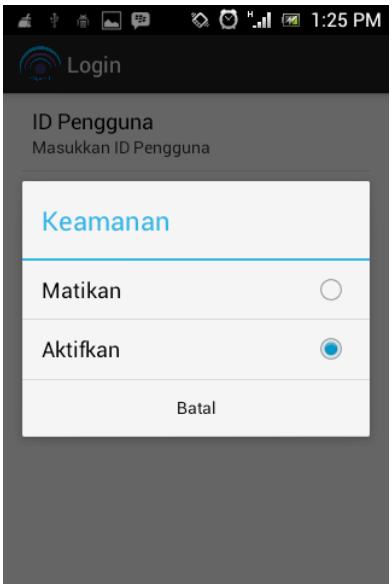
**4.1.3.3** Konfigurasi *Secure Call*

Komunikasi VoIP dengan *secure call* dapat dilakukan ketika VoIP *server* VoIP *client* telah mendukung. Protokol SRTP sudah didukung oleh *Asterisk* tanpa memerlukan konfigurasi tambahan. Pada sisi VoIP *client* yang telah dibangun dengan versi android yaitu *SSipCall* dan untuk versi windows menggunakan *Linphone*, diperlukan konfigurasi untuk mendukung protokol SRTP pada komunikasi VoIP.

**4.1.3.3.1 Konfigurasi *SSipCall***

Konfigurasi pada *SSipCall* untuk melakukan komunikasi dengan *secure call*, langkahnya adalah sebagai berikut:

1. Pilih menu Pengaturan Akun
2. Pilih keamanan, pada keamanan pilih aktifkan seperti pada Gambar berikut.

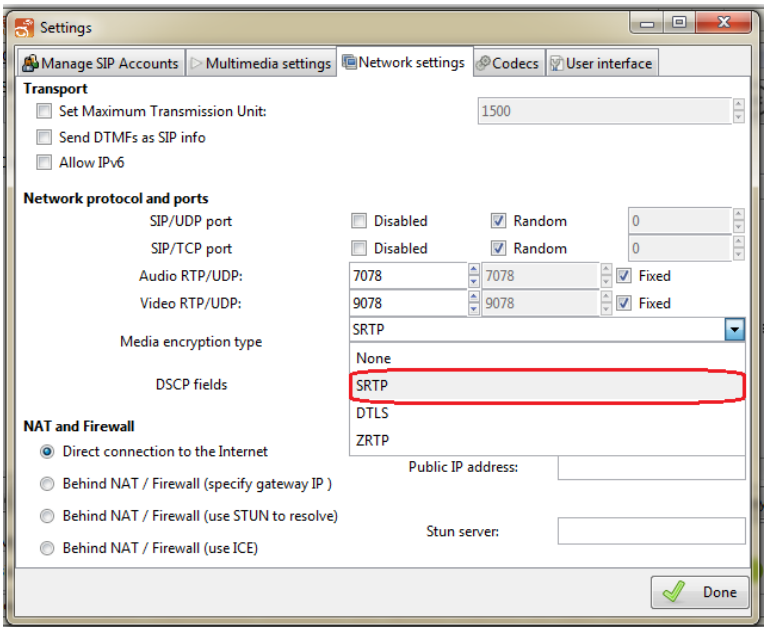



**Gambar 4.7 Konfigurasi *SsipCall***

#### 4.1.3.3.2 Konfigurasi *Linphone*

Konfigurasi pada *Linphone* untuk melakukan komunikasi dengan *secure call*, langkahnya adalah sebagai berikut:

1. Pilih menu *Options* -> *Prefences* -> *Network Setting*
2. Pada *media encryption type* pilih SRTP -> *Done*, seperti pada Gambar Berikut




**Gambar 4.8 Konfigurasi *Lindphone***

### 4.1.4 Implementasi Antarmuka

Implementasi antarmuka bertujuan untuk mengetahui setiap tampilan yang telah dibangun dan pengkodeannya ke dalam bentuk file program, sehingga akan diketahui apakah sistem yang telah dibuat benar-benar sesuai dengan yang direncanakan. Implementasi antarmuka dari aplikasi *SSipCall* ini dapat dilihat pada Tabel 4.6

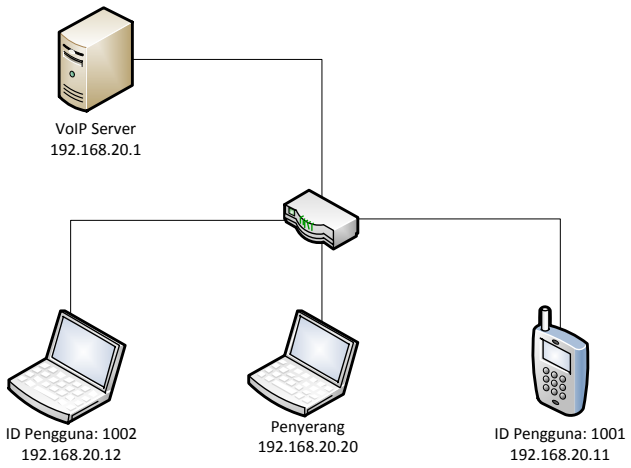
**Tabel 4.6 Implementasi Antarmuka**

| No. | Halaman | Deskripsi | Nama File |
|---|---|---|---|
| 1 | Utama | Halaman antarmuka awal pada saat pertama kali masuk aplikasi. | Walkie-talkie.xml |
| 2 | Panggil | Halaman yang digunakan untuk melakukan panggilan. | call_address_dialog.xml |
| 3 | Menerima Panggilan | Halaman yang digunakan untuk menerima panggilan. | receiver_call.xml |
| 3 | Pengaturan Akun | Halaman yang digunakan untuk memasukan Nama Pengguna, *Server*, Kata Sandi, dan Keamanan. | Preferences.xml |

## 4.2 Pengujian

Pada tahap ini dilakukan pengujian terhadap sistem komunikasi VoIP yang tidak menggunakan keamanan dan yang menggunakan keamanan. Uji coba panggilan dilakukan dari satu *Laptop* ke satu *Smartphone*. Tujuan dari pengujian

ini adalah untuk mengetahui tingkat keamanan dari masing-masing sistem VoIP. Arsitektur pengujian yang digunakan dapat dilihat pada gambar 4.9




**Gambar 4.9 Asitektur pengujian VoIP**

### 4.2.1 Skenario Pengujian

Pada tahap pengujian dilakukan beberapa skenario untuk mengetahui tingkat keamanan sistem VoIP pada dua kondisi yang berbeda. Skenario pengujian yang akan dilakukan dijelaskan pada Tabel 4.7

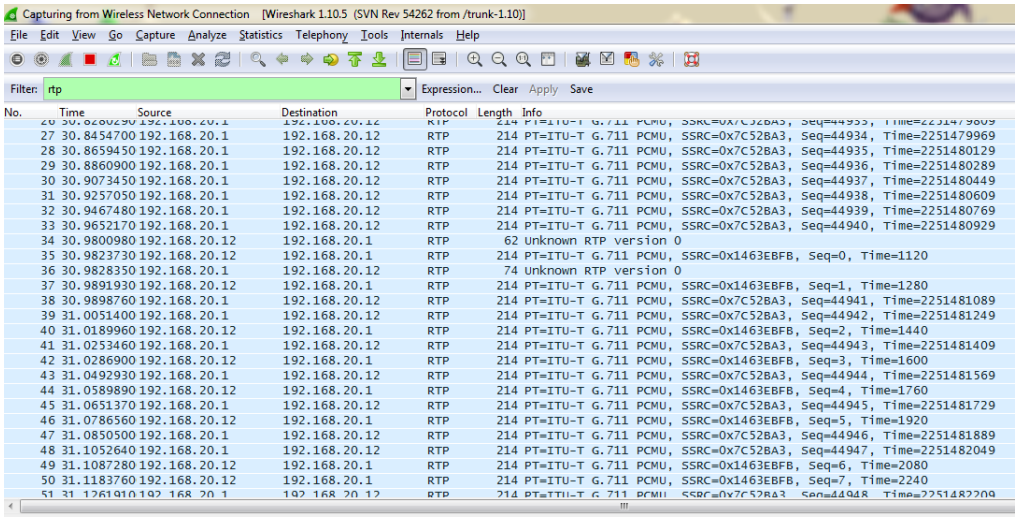
**Tabel 4.7 Skenario Pengujian VoIP**

| No | Sistem VoIP | Waktu Penyerangan |
|---|---|---|
| 1 | Tidak menggunakan *Secure Call* | Pada saat komunikasi berlangsung. |
| 2 | Menggunakan *Secure Call* | Pada saat komunikasi berlangsung. |

#### 4.2.1.1 Pengujian Sistem VoIP Tanpa Menggunakan *Secure Call*

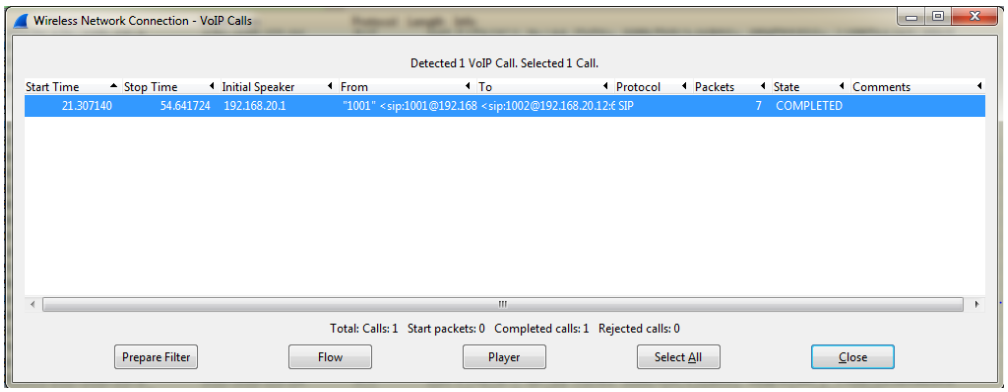
Pada tahap ini pengujian dilakukan dengan menggunakan metode *Sniffing*. Penyerangan dengan metode *sniffing* dilakukan untuk mendapatkan paket RTP pada saat kedua *enduser* melakukan komunikasi suara. Setelah paket RTP didapatkan langkah selanjutnya adalah men*decode* paket RTP tersebut agar dapat didengarkan audionya. Pengujian dilakukan pada saat komunikasi antara pengguna 1001 dan pengguna 1002 terjadi dengan menggunakan *Wireshark*. Berikut adalah langkah pengujiannya:

1. Buka *Wireshark*, lalu pilih *Capture -> Interfaces -> Wirelass Network Connection -> start* untuk memulai proses *packet sniffing.* Proses ini dilakukan ketika proses komunikasi VoIP berlangsung sampai dengan selesai.
2. Saat komunikasi VoIP berlangsung akan terlihat paket RTP pada jendela *wireshark* seperti pada Gambar 4.10. Agar paket yang ditampilkan hanya paket RTP pada *field Filter* tuliskan RTP.



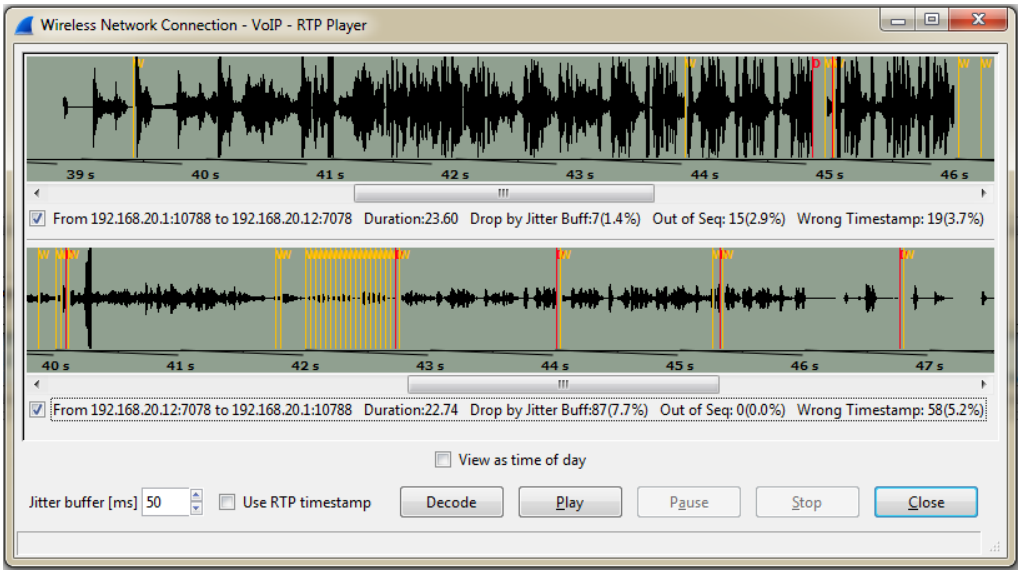

**Gambar 4.10 Proses *Packet Sniffing***

3. Setelah *wireshark* menampiilkan paket RTP, selanjutnya adalah mendecode paket RTP dengan memilih menu *Telephony -> Voip Calls.* Kemudian *wireshark* menampilkan komunikasi VoIP yang terdeteksi seperti pada Gambar 4.11




**Gambar 4.11 Daftar Komunikasi VoIP**

4. Pilih komunikasi VoIP, lalu klik tombol *Player* -> *Decode* maka akan muncul jendela baru seperti pada Gambar 4.12



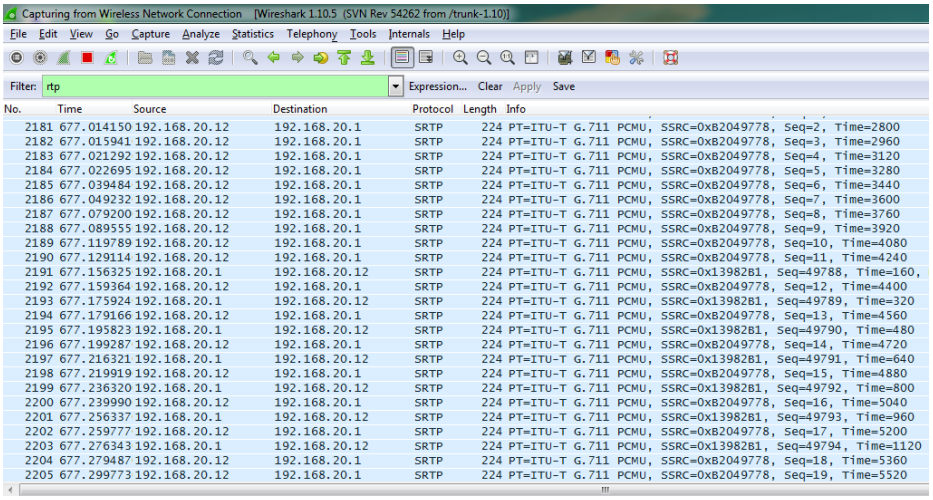

**Gambar 4.12 RTP *Player***

5. Klik *Play* untuk mendengarkan percakapan antara pengguna 1001 dan pengguna 1002.

### 4.2.1.2 Pengujian Sistem VoIP Dengan Menggunakan *Secure Call*

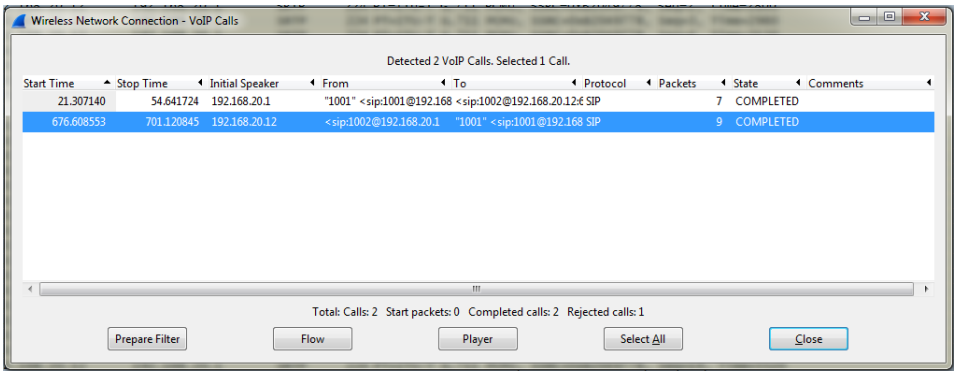
Pengujian sistem VoIP dengan menggunakan *secure call* menggunakan metode dan *Tools* yang sama dengan pengujian sistem VoIP tidak menggunakan *secure call*. Langkah yang dilakukan juga sama, yang membedakan bagian ini adalah suara yang dihasilkan dari proses *packet sniffing.* Berikut adalah langkanya:

1. Buka *Wireshark*, lalu pilih *Capture* -> *Interfaces* -> *Wirelass Network Connection* -> *start* untuk memulai proses *packet sniffing*. Proses ini dilakukan ketika proses komunikasi VoIP berlangsung sampai dengan selesai.
2. Saat komunikasi VoIP berlangsung dengan menggunakan *secure call*, yang terlihat adalah paket SRTP pada jendela *wireshark* yang artinya komunikasi VoIP berlangsug dengan aman seperti pada Gambar 4.13. Agar paket yang ditampilkan hanya paket RTP pada *field Filter* tuliskan RTP.



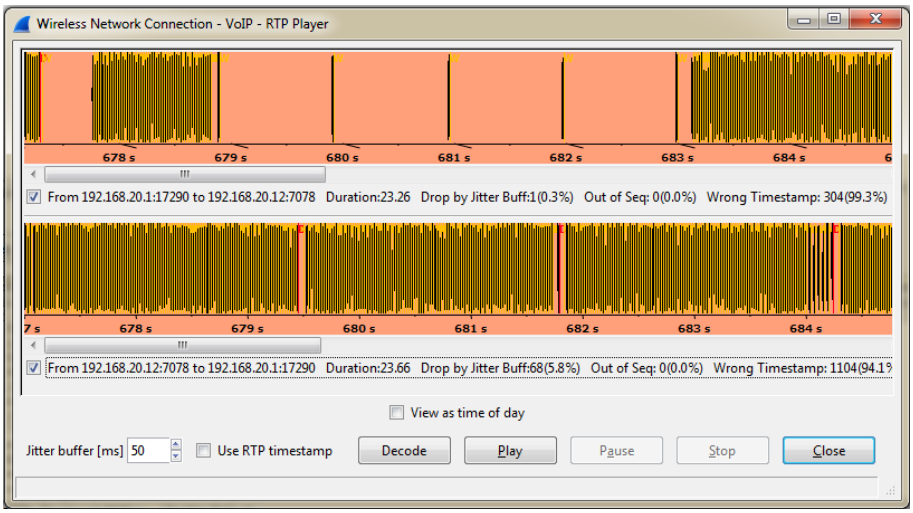

**Gambar 4.13 Proses *Packet Sniffing***

3. Setelah *wireshark* menampiilkan paket SRTP, selanjutnya adalah mendecode paket SRTP dengan memilih menu *Telephony -> Voip Calls.* Kemudian *wireshark* menampilkan komunikasi VoIP yang terdeteksi seperti pada Gambar 4.14




**Gambar 4.14 Daftar Komunikasi VoIP**

4. Pilih komunikasi VoIP, lalu klik tombol *Player -> Decode* maka akan muncul jendela baru seperti pada Gambar 4.15




**Gambar 4.15 RTP *Player***

5. Klik *Play* untuk mendengarkan percakapan antara pengguna 1001 dan pengguna 1002

### 4.2.1.3 Hasil Pengujian

Hasil pengujian pada penelitian ini diperoleh berdasarkan pengujian yang telah dilakukan berdasarkan skenario pada Tabel 4.7 Skenario Pengujian VoIP. Hasil pengujian dapat dilihat pada Tabel 4.8

**Tabel 4.8 Hasil Pengujian**

| No. | Skenario Pengujian | Komunikasi VoIP | Percobaan Penyadapan | Hasil Suara |
|---|---|---|---|---|
| 1 | Tidak menggunakan *Secure Call* | Berhasil melakukan komunikasi suara | Penyadapan berhasil | Dapat didengarkan |
| 2 | Menggunakan *Secure Call* | Berhasil melakukan komunikasi suara | Penyadapan berhasil | Tidak dapat didengarkan |

Berdasarkan Tabel 4.8 bahwa hasil pengujian VoIP tidak menggunakan *secure call* berhasil dilakukan oleh *client* 1 dengan *client* 2, dan penyerang berhasil melakukan penyadapan sehingga percakapan antara *client* 1 dengan *client* 2 dapat didengarkan. Sedangkan pengujian VoIP menggunakan *secure call* berhasil dilakukan oleh *client* 1 dengan *client* 2, dan penyerang berhasil melakukan penyadapan tetapi suara yang dihasilkan dari percakapan antara *client* 1 dengan *client* 2 tidak dapat didengarkan karena rusak.