BAB III
DESAIN SISTEM DAN IMPLEMENTASI

Pada bab ini dijelaskan tentang identifikasi, desain dan implementasi dari aplikasi steganografi pada VoIP.

Langkah pertama yang dilakukan adalah mengidentifikasi hal-hal yang terkait dengan sistem yang akan dibangun. Dengan melakukan identifikasi tersebut diharapkan sistem akan berjalan sesuai dengan kriteria yang telah ditentukan.

Langkah kedua adalah membuat desain sesuai dengan hasil proses identifikasi. Desain yang dibuat terbagi menjadi dua bagian, yaitu desain sistem VoIP dan yang kedua adalah desain aplikasi steganografi. Desain sistem VoIP dilakukan berdasarkan *object oriented* [15] dengan mengadopsi *source code* yang bersifat *open source* yang mengevaluasi model *Speex* [6] sebagai *voice codec*-nya.

Langkah ketiga adalah implementasi desain sistem yang telah dirancang. Implementasi akan dilakukan pada lingkungan perangkat *personal computer* (PC) berbasis sistem operasi Microsoft Windows XP.

3.1. Identifikasi

Saat ini telah banyak perangkat lunak yang mengimplementasikan teknologi VoIP, namun tidak semua perangkat lunak telah memanfaatkan proses steganografi pada VoIP pada aplikasinya untuk digunakan sebagai media pengirim berita rahasia.

Tabel 3.1. Rancangan Fitur VoIP Steganografi

No.	Fitur	VoIP
1.	Mode komunikasi	*Peer to peer*
2.	Mode VoIP	*Full duplex*
3.	Kompresi suara	Speex
4.	Konferensi	Tidak bisa
5.	Kontrol akses aplikasi	Tidak ada
6.	Protokol yang digunakan	Protokol *peer to peer* khusus
7.	Metode steganografi	*Covert Channel : Least Significant Bit* (LSB)

Oleh karena itu penulis akan membuat perangkat lunak yang mengimplementasikan proses steganografi pada VoIP. Pada Tabel 3.1 menunjukkan rancangan perangkat lunak VoIP Steganografi yang akan dibuat. Dengan memperhatikan kebutuhan dimaksud, maka dalam perancangan sistem VoIP akan meliputi hal-hal sebagai berikut :

1. Sistem VoIP terdiri dari *user* pemanggil dan *user* penerima panggilan. Hubungan antara dua entitas tersebut bertipe *client-server*.
2. Komunikasi antar *user* dilakukan secara *point-to-point* sehingga komunikasi hanya dapat dilakukan oleh dua *user* untuk setiap panggilan komunikasi (tidak dapat melakukan konferensi).
3. Sistem VoIP menggunakan protokol khusus yang mengadopsi standar protokol TCP/IP.
4. Untuk menghasilkan kualitas suara yang baik, maka teknik kompresi suara yang digunakan adalah Speex. Teknik ini digunakan karena bisa diterapkan untuk mode komunikasi *full duplex* dengan *bandwidth* yang kecil.
5. Data yang digunakan sebagai masukan proses steganografi adalah berupa teks yang langsung diketik dari *keyboard*.

3.2. Desain Sistem

Desain sistem yang dirancang dibagi dalam dua bagian pokok, yaitu sistem komunikasi VoIP dan sistem aplikasi steganografi dengan metode *Least Significant Bit* (LSB) [5].

Sistem komunikasi VoIP yang dibangun terbagi menjadi dua fase sebagai berikut :

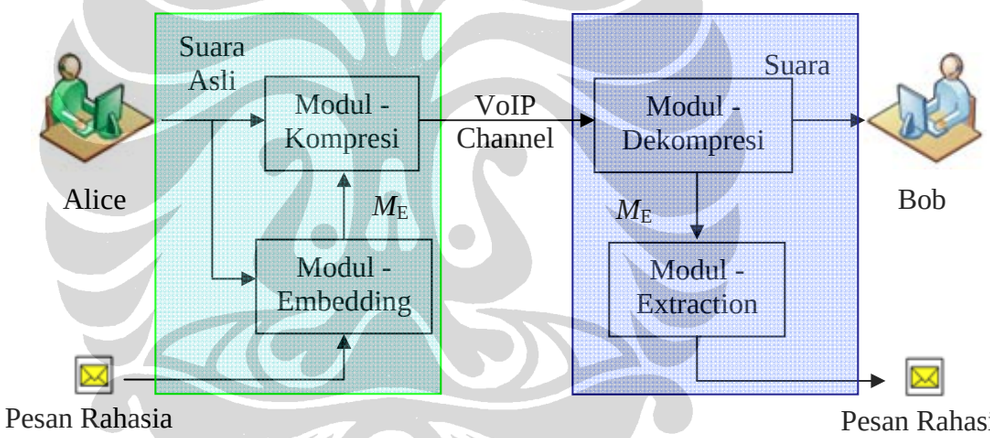
1. Identifikasi pengguna

   Pengguna VoIP melakukan identifikasi masing-masing sebagai pihak yang memanggil atau yang dipanggil. Ketentuan yang diberlakukan dalam identifikasi ini adalah bahwa alamat IP pihak-pihak yang akan berkomunikasi telah saling diketahui. Pihak pemanggil akan mengisikan alamat IP pihak yang dipanggil, sedangkan pihak yang dipanggil hanya menunggu respon dari pihak pemanggil. Jika alamat IP yang diisikan sesuai maka koneksi bisa dijalankan dan *user* dapat menggunakan aplikasi VoIP Steganografi.

2. Komunikasi suara

   Setelah koneksi antara dua pengguna terhubung, setiap pengguna sudah siap untuk saling berkomunikasi. Dengan menggunakan Speex sebagai *voice codec*-nya, sumber suara yang masih berupa data *analog* dikompresi menjadi data *digital*. Setelah menjadi data *digital* inilah, data bisa diproses untuk penerapan steganografi dengan metode yang telah ditentukan, dalam hal ini adalah metode LSB. Komunikasi suara antara dua pengguna bersifat *full duplex*, artinya masing-masing pihak bisa saling bersuara dalam waktu yang bersamaan dan mereka masih saling mendengar tanpa harus menunggu salah satu pihak selesai berbicara. Pada saat komunikasi suara dilakukan, setiap pengguna bisa saling bertukar berita yang di-*input*/diketik melalui *keyboard*. Dalam prosesnya, data berita yang di-*input* inilah yang akan ditempelkan pada data suara sebelum paket suara dikirimkan melalui sistem VoIP.

Sedangkan sistem aplikasi steganografi dibagi menjadi dua proses [5], yaitu proses dalam modul penempelan pesan (*Modul-Embedding*) dan proses dalam modul ekstraksi pesan (*Modul-Exctraction*).



Gambar 3.1. Model Komunikasi Rahasia Secara Umum dalam VoIP [5]

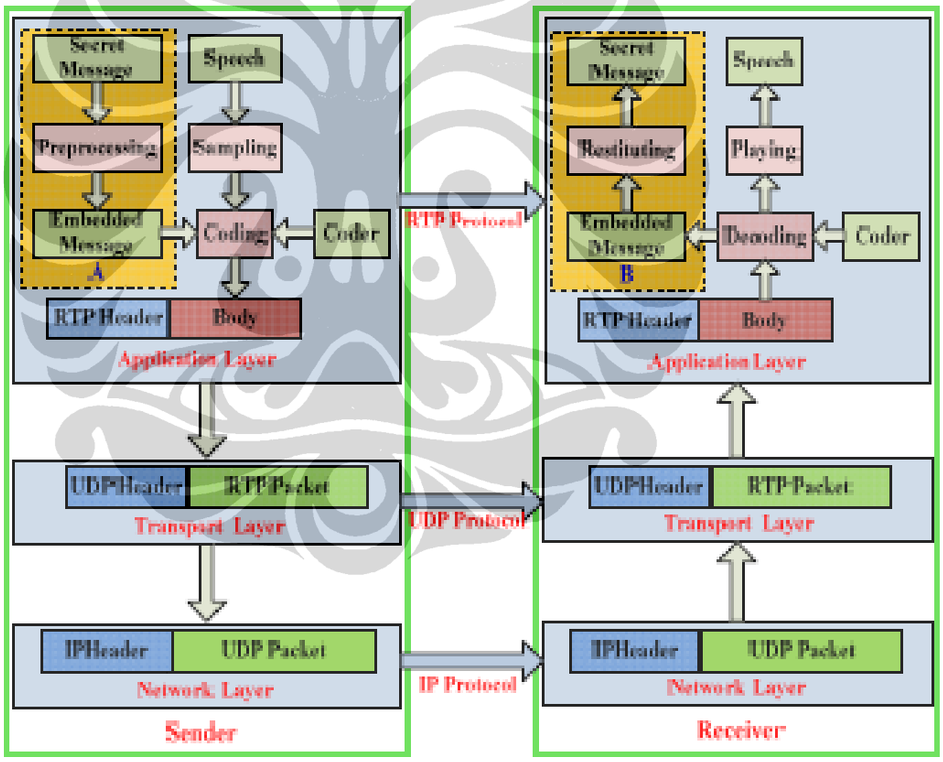
Pada Gambar 3.1 mengilustrasikan bahwa Alice (sebagai pengirim) diasumsikan akan mengirim pesan rahasia kepada Bob (sebagai penerima), sementara itu mereka berbicara tentang topik yang tidak mencurigakan melalui sistem VoIP. Proses transmisi akan digambarkan sebagai berikut : Alice menyediakan Suara Asli (*S*) dan

Pesan Rahasia (*P*) sebagai masukan untuk Modul-Kompresi dan *Modul-Embedding*. Setelah proses di Modul-*Embedding*, *embedding Message* (M_E) dapat diperoleh. M_E adalah obyek final yang ditempelkan ke dalam *S* oleh *Modul-Embedding* (ME_A). Setelah suara dengan M_E terkirim melalui saluran VoIP, Bob mendekompresi M_E melalui Modul-Dekompresi (MD) dan mengambil *P* melalui *Modul-Extraction* (ME_B).

Modul-Kompresi dan Modul-Dekompresi dikerjakan oleh algoritma *voice codec* yang digunakan yaitu Speex [6]. Pada Modul-Kompresi, proses yang terjadi adalah mengkonversi suara *analog* menjadi *digital* dalam bentuk karakter. Sedangkan proses yang terjadi pada Modul-Dekompresi adalah sebaliknya, yaitu data suara yang diterima dalam bentuk *digital* diolah kembali ke dalam bentuk *analog* sebagai *output* yang akan didengar oleh pengguna.

Proses yang akan dijelaskan dalam makalah ini adalah proses pada *Modul-Embedding* dan *Modul-Extraction*.

Gambaran lebih jelas dapat dilihat secara skematis pada Gambar 3.2 berikut ini.



Gambar 3.2. Skema Arsitektur VoIP Steganografi [5]

Proses yang terjadi pada *Modul-Embedding* dan *Modul-Extraction* dapat dijelaskan sebagai berikut :

Modul-Embedding

Data suara *digital* yang diterima dari hasil *codec* adalah berupa aliran paket dalam bentuk karakter. Besarnya tiap-tiap paket yang mengalir dari hasil *codec* bervariasi. Karakter terakhir dari setiap paket inilah yang nantinya akan diganti dengan teks stego yang akan dikirimkan. Teks yang diketik sebagai masukan data stego akan dikumpulkan dalam susunan *array* dan ditambahkan dengan tanda mulai stego dan tanda akhir stego. Data dalam susunan *array* tersebut kemudian ditempelkan satu persatu pada aliran paket data suara *digital* yang diterima. Untuk setiap paket hasil *codec* yang mengalir pada *modul-embedding*, jika tidak ditempeli oleh data stego maka karakter terakhir akan diganti dengan tanda tanpa stego. Hal ini dimaksudkan untuk memastikan bahwa dalam setiap aliran paket hanya akan teridentifikasi antara data stego atau bukan stego, yang nantinya akan digunakan oleh penerima pada *modul-extraction* sebagai tanda pengenalnya. Selanjutnya aliran paket data ini akan dikirimkan melalui mekanisme protokol jaringan IP dengan menambahkan *header* sebagaimana mestinya.

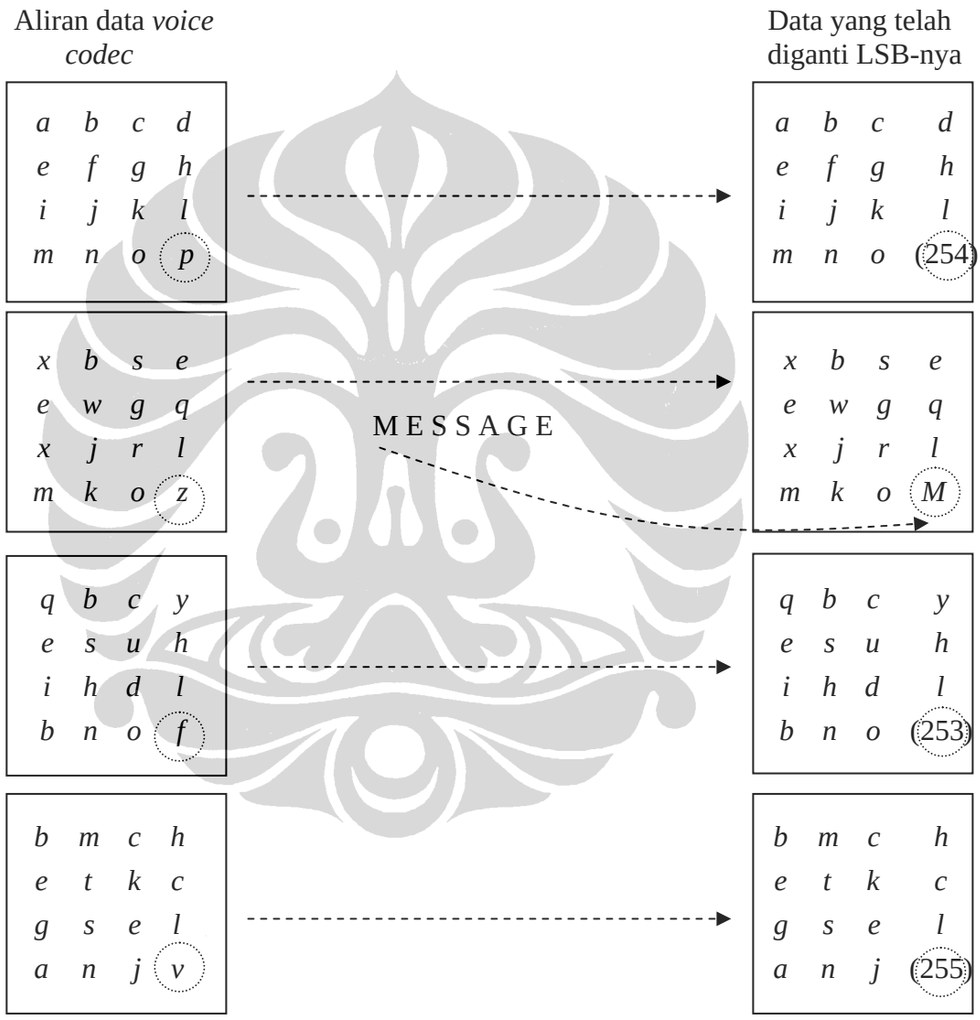
Modul-Extraction

Proses yang terjadi pada *modul-extraction* adalah kebalikan dari proses yang terjadi pada *modul-embbeding*. Setiap aliran paket data yang diterima akan disaring untuk dikenali apakah mengandung data stego atau tidak. Jika pada suatu paket terdapat tanda mulai stego pada karakter akhirnya, maka aliran paket berikutnya akan diambil untuk dikumpulkan sebagai teks/pesan yang diterima sampai dikenali/ditemukan paket yang mengandung tanda akhir stego. Aliran paket-paket yang telah diterima ini, baik yang mengandung stego ataupun tidak kemudian akan diproses oleh *codec* untuk dikeluarkan menjadi suara *analog* kembali yang bisa didengarkan oleh penerima.

Karakter yang digunakan sebagai tanda untuk awal stego, akhir stego, dan tanpa stego bisa ditentukan sesuai kebutuhan. Karakter yang bisa digunakan adalah karakter yang tidak digunakan dalam penulisan teks pesan yang akan dikirim. Sebagai contoh dalam format ASCII [16] adalah karakter dengan nilai desimal 0 – 31 atau 128 – 255. Sedangkan karakter ASCII dengan nilai desimal 32 – 127 digunakan

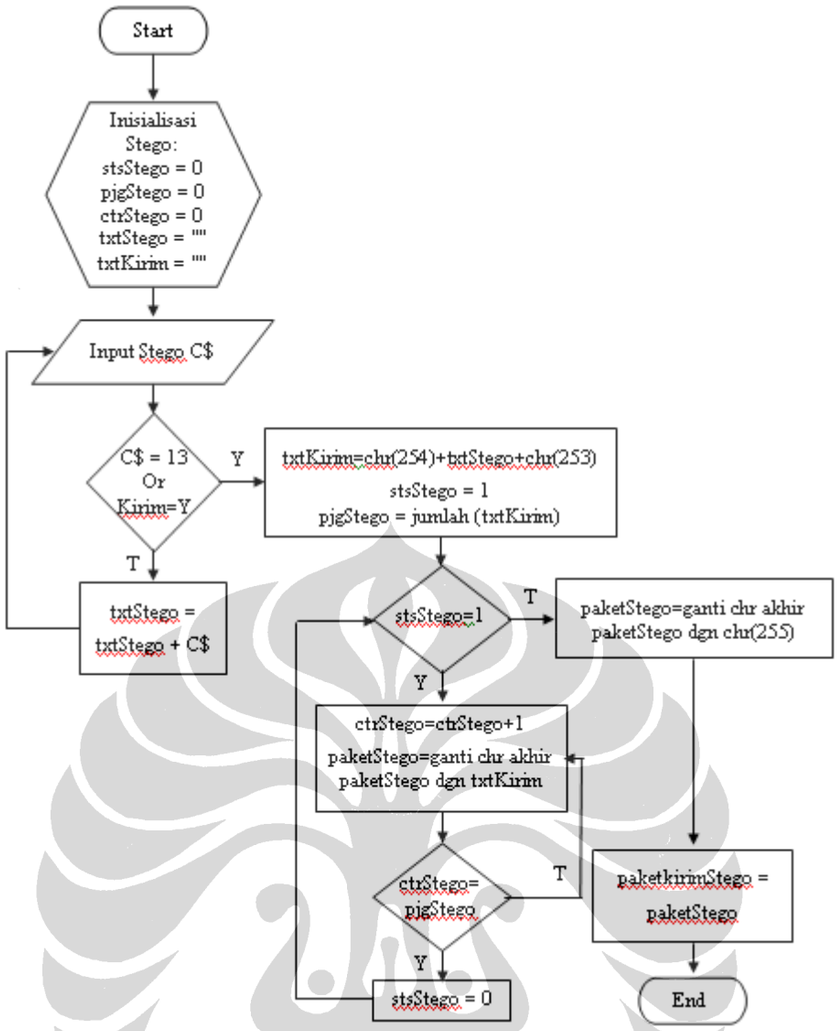
untuk penulisan teks yang akan dikirim, yaitu berupa huruf-huruf beserta tanda bacanya.

Pada Gambar 3.3 terlihat bahwa karakter terakhir dalam paket data yang pertama adalah huruf ’*p*’ kemudian dengan metode LSB akan diganti menjadi karakter dengan nilai desimal 254 sebagai tanda awal stego. Karakter terakhir dalam paket data yang kedua adalah huruf ’*z*’ yang akan diganti menjadi huruf M sebagai huruf masukan dari pesan yang akan dikirim. Untuk menandai akhir stego, karakter terakhir dalam paket data ke-n (n = jumlah karakter dalam teks yang akan dikirim) akan diganti dengan karakter dengan nilai desimal 253. Pada paket data selanjutnya, dengan karakter terakhirnya adalah huruf ’*v*’ akan diganti dengan karakter yang nilai desimalnya 255 sebagai tanda tidak adanya proses stego (tanda tanpa stego).



Gambar 3.3. Skema Proses Penggantian Karakter dengan Metode LSB

Diagram alir proses *embedding* (penempelan) data teks pada hasil *voice codec* seperti terlihat pada Gambar 3.4 berikut :



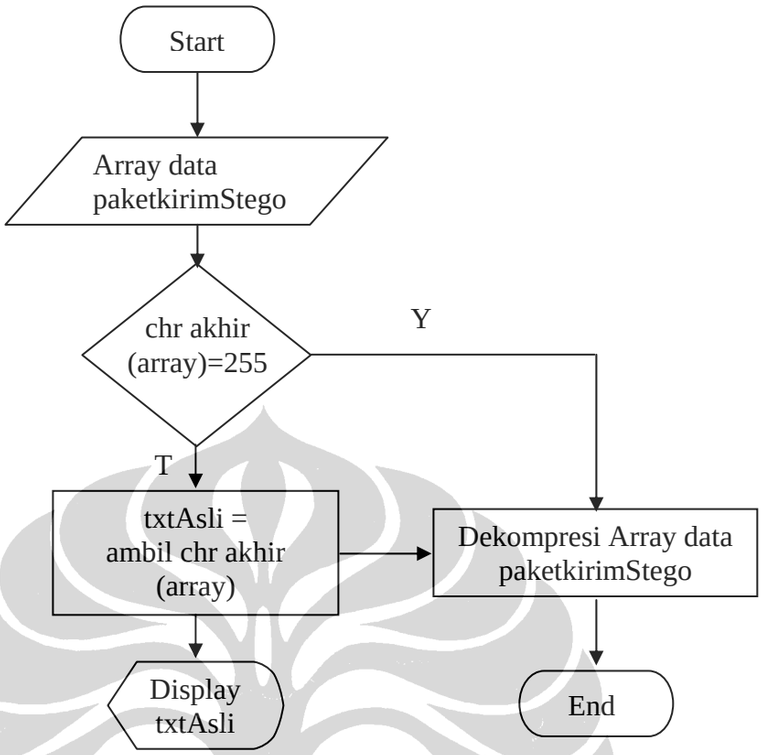
Gambar 3.4. Diagram Alir Proses *Embedding* Data Teks

Sedangkan diagram alir proses monitoring, deteksi data steganografi, dan pengambilan data teks yang dikirimkan dapat dilihat pada Gambar 3.5.

3.3. Implementasi

Implementasi perangkat lunak dilakukan pada *personal computer* (PC) yang terkoneksi dengan sistem jaringan yang menggunakan protokol internet. Pada proses penelitian, aplikasi dijalankan pada PC dengan sistem operasi Microsoft Windows XP.

Pada pembuatan programnya dilakukan dengan menggunakan bahasa pemrograman Microsoft Visual Basic 6.0. Pembuatan program untuk proses *codec* dan aplikasi VoIP menggunakan *class library* [17] yang diperoleh dari sumber terbuka, yaitu internet.



Gambar 3.5. Diagram Alir Proses *Monitoring*, Deteksi Data Steganografi, dan Pengambilan Data Teks

3.3.1. Pembuatan Aplikasi

1) Spesifikasi perangkat keras yang digunakan pada proses pembuatan *prototype* aplikasi dan ujicoba VoIP Steganografi adalah sebagai berikut :
   - Laptop ACER Aspire 4710
   - Prosessor 2.0 GHz
   - RAM 512 MB DDR2
   - *Harddisk* 120 GB
   - *Ethernet card*
   - *Wireless* LAN 802.11
   - Sistem Operasi Microsoft Windows XP Professional

2) Spesifikasi perangkat lunak yang digunakan dalam proses pembuatan *prototype* aplikasi dan ujicoba VoIP Steganografi adalah Microsoft Visual Basic 6.0.

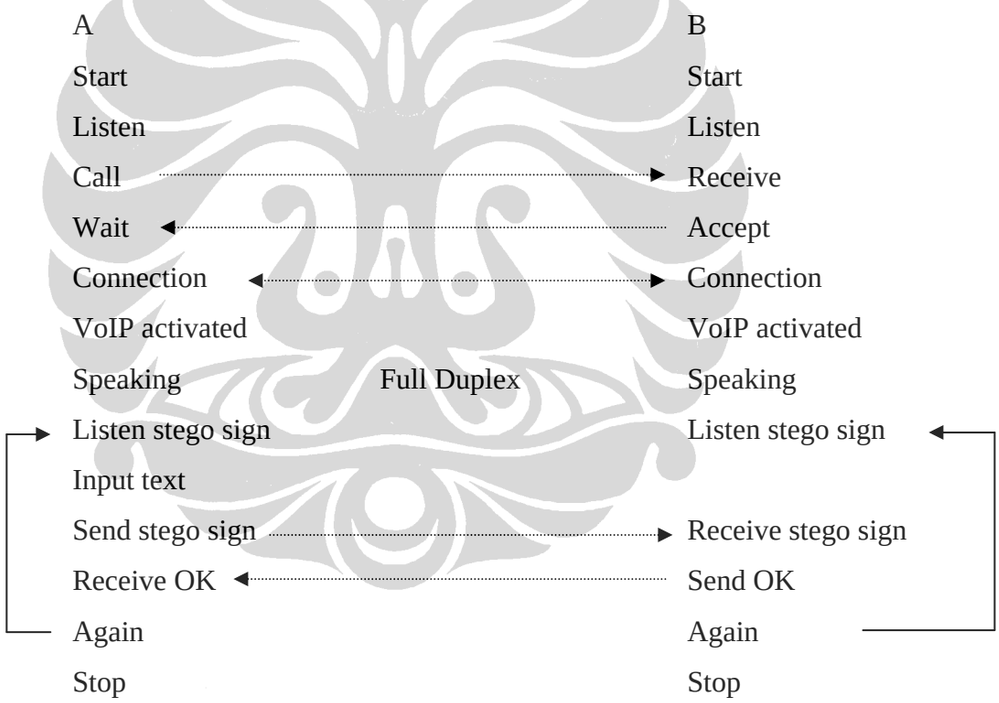
3.3.2. Lingkungan Implementasi

Aplikasi VoIP Steganografi yang dirancang dapat berjalan pada PC yang mendukung antara lain sebagai berikut :

1) Sistem operasi Microsoft Windows;
2) Koneksi sistem jaringan komputer berbasis protokol internet;
3) Tersedia *sound card;*
4) Tersedia *microphone* yang digunakan sebagai *input* suara;
5) Tersedia *speaker* yang digunakan sebagai *output* suara.

3.3.3. Skenario Implementasi

Proses terjadinya komunikasi VoIP Steganografi secara bertahap terlihat seperti berikut ini :



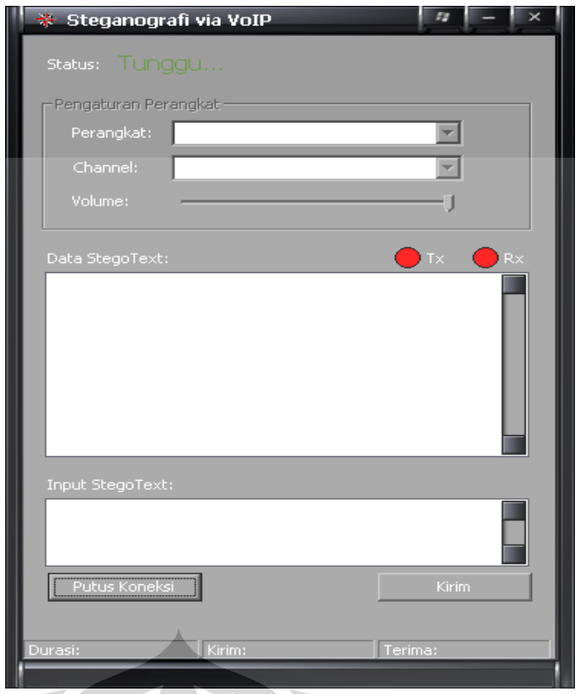
Komunikasi suara dengan VoIP Steganografi dapat digambarkan dengan suatu skenario implementasi. Pada makalah ini, permasalahan dibatasi dengan asumsi bahwa setiap pengguna telah mengetahui alamat IP yang digunakan masing-masing. Pada skenario tersebut akan dilakukan proses komunikasi suara dengan menggunakan VoIP Steganografi antara dua entitas *user*.

Pada saat *user* akan menggunakan aplikasi VoIP Steganografi, maka *user* tersebut harus menentukan pilihan, apakah sebagai pihak pemanggil atau pihak yang menunggu panggilan. Jika *user* sebagai pihak pemanggil, maka harus mengisikan alamat IP *user* tujuan yang telah diketahui terhadap aplikasi kemudian klik tombol koneksi ke tujuan. Sedangkan jika *user* sebagai pihak yang menunggu panggilan hanya diperlukan menekan tombol tunggu panggilan. Tampilan awal untuk pembuatan koneksi VoIP Steganografi dapat dilihat pada Gambar 3.6.

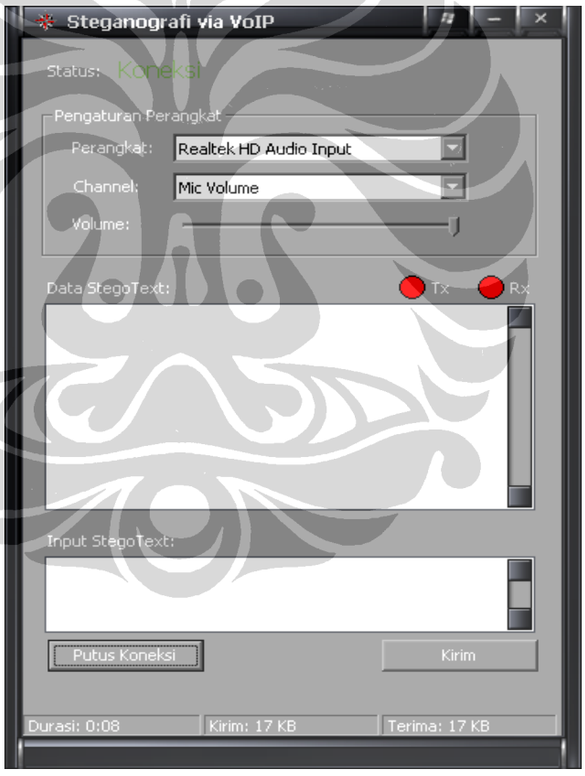
Gambar 3.6. Tampilan Pembuatan Koneksi VoIP Steganografi

Jika *user* berlaku sebagai pihak yang menunggu panggilan, maka setelah tombol tunggu panggilan di-klik akan muncul tampilan dengan status tunggu seperti terlihat pada Gambar 3.7.

Sedangkan pada *user* yang memilih sebagai pihak pemanggil, setelah pilihan ditentukan dengan menekan tombol koneksi ke tujuan, koneksi antara dua PC akan terbentuk. Tampilan pada setiap PC *user* setelah terkoneksi dapat dilihat pada Gambar 3.8. Pada tahapan ini aplikasi VoIP Steganografi sudah siap untuk digunakan. Pada tampilan yang ada terdapat beberapa fitur pengaturan yang bisa diatur untuk perangkat yang digunakan antara lain perangkat *soundcard*, *input* suara, dan volume suara.



Gambar 3.7. Tampilan Status Sedang Menunggu Untuk Koneksi



Gambar 3.8. Tampilan Setelah Koneksi VoIP Steganografi Terbentuk

Tanda Tx dan Rx dengan lingkaran merah adalah untuk mengetahui proses yang sedang dijalankan. Tanda merah akan berubah menjadi hijau sesuai dengan proses yang sedang berjalan. Jendela Data StegoText berfungsi untuk menampilkan data-data teks yang telah dikirimkan atau diterima, sedangkan jendela Input StegoText digunakan untuk mengetik teks pesan yang akan dikirimkan. Jika telah selesai menggunakan aplikasi, tombol yang digunakan untuk memutus koneksi adalah tombol putus koneksi. Komunikasi suara *peer-to-peer* antara dua *user* dilakukan secara *full-duplex*, sehingga proses komunikasi suara tidak perlu dilaksanakan secara bergantian demikian pula untuk proses kirim terima teks pesan yang akan dilakukan.