



SIMULASI JARINGAN VOIP DENGAN SERVER BRIKER MENGGUNAKAN CARD OPENVOX

Nur Rachmad, Irda Chynthya Dewi
nbalistik@yahoo.com

Akademi Teknik Telekomunikasi Sandhy Putra Jakarta

ABSTRAK

Teknologi jaringan komputer dan internet telah mengalami perkembangan yang sangat pesat, sehingga mampu menyambungkan semua komputer yang ada di dunia untuk dapat berkomunikasi dan bertukar informasi. Satu kelebihan yang dimiliki oleh sistem komunikasi lewat jaringan internet yaitu biaya perawatan yang relatif lebih murah dibandingkan biaya perawatan yang dibutuhkan oleh jaringan komunikasi konvensional, contohnya jaringan telepon.

VoIP atau *Voice Over Internet Protocol* pada hakikatnya adalah penghantaran paket data suara melalui protokol IP. Dengan menggunakan briker yang disambungkan dengan VoIP gateway dan GSM gateway mampu mengabungkan jaringan *packet data* dengan jaringan *circuit switch*. Tujuannya, dengan menggunakan server briker yang berbasis open source dapat mengkonfigurasi jaringan VoIP ke jaringan GSM seluler serta menganalisa *bandwidth*, *packet lost*, BER, delay dan *jitter* yang terjadi pada pengkonfigurasi jaringan VoIP ke jaringan GSM seluler.

GSM gateway dihubungkan dengan card openvox di dalam CPU menggunakan kabel UTP (*Unshielded Twisted Pair*) connector RJ 11 yang terdapat server briker dan CPU tersebut dihubungkan menggunakan kabel UTP (*Unshielded Twisted Pair*) connector RJ 45 dengan switch yang terhubung pada beberapa client. Dengan cara tersebut diharapkan dapat membuat jaringan komunikasi dengan biaya yang lebih murah.

Kata kunci : VoIP, Briker, Openvox, GSM Gateway

ABSTRACT

Network technology computer and the internet has undergo development which is very fast, so being able to connect all computers in the world to be able to communicate and exchange information. One advantage is owned by system communication through the internet network that is maintenance cost are comparatively cheaper than the cost of care required by conventional network communication, for example telephone network.

VoIP or *Voice Over Internet Protocol* virtually is voice transfer through the IP protocol. By using briker connected with VoIP gateway and GSM gateway is able to combine packet data network and network switch circuit. The goal using server briker based open-source can configuring VoIP network to GSM mobile network and analyzing bandwidth, packets data, BER, delay and jitter which happened to configuring VoIP network to GSM mobile seluler network.

GSM gateway connected with card openvox in the CPU using cable UTP (*Unshielded Twisted Pair*) connector RJ 11 embedded server briker dan the CPU are connected using cable UTP (*Unshielded Twisted Pair*) connector RJ 45 with a switch connected at some client. In this manner it is expected can create network communication with cheaper cost.

Keyword : VoIP, Briker, Openvox, GSM Gateway

1. PENDAHULUAN

1.1 Latar Belakang

Teknologi jaringan komputer dan internet telah mengalami perkembangan yang sangat pesat, sehingga mampu menyambungkan semua komputer yang ada di dunia untuk dapat berkomunikasi dan bertukar informasi. Satu kelebihan yang dimiliki oleh sistem komunikasi lewat jaringan internet yaitu biaya perawatan yang relatif lebih murah dibandingkan biaya perawatan yang dibutuhkan oleh jaringan komunikasi konvensional, contohnya jaringan telepon.

Munculnya teknologi generasi masa depan atau biasa dikenal sebagai *Next Generation Network* (NGN) merupakan awal pemikiran untuk bermigrasi dari teknologi jaringan konvensional yang berbasis jaringan *circuit switch*, seperti PSTN menuju jaringan yang berbasis *packet switch* dengan menggunakan jaringan *internet protocol* (IP).

VoIP atau *Voice Over Internet Protocol* pada hakikatnya adalah penghantaran paket data suara melalui protokol IP. Dengan menggunakan briker yang disambungkan dengan VoIP gateway dan GSM gateway mampu mengubungkan jaringan *packet data* dengan jaringan *circuit switch*. Dengan mengintegrasikan jaringan VoIP ke GSM Gateway diharapkan dapat membuat jaringan komunikasi dengan biaya yang lebih murah dibandingkan dengan jaringan komunikasi konvensional lainnya.

1.2 Tujuan Penelitian

Maksud dan tujuan dalam penyusunan proyek akhir ini adalah :

1. Membuat jaringan voip dengan open source
2. Mengkonfigurasi jaringan voip
3. Menganalisa kecepatan *bandwidth*, *delay*, *jitter*, *throughput* dan *packet loss* yang terjadi pada jaringan voip

1.3 Rumusan Masalah

Permasalahan yang ada dan akan dipecahkan dalam proyek akhir ini antara lain yaitu :

1. Bagaimana cara mengkonfigurasi server briker berbasis open source
2. Bagaimana cara mengkonfigurasi SIP server dengan briker dan SIP client dengan X-Lite
3. Bagaimana cara mengkonfigurasi jaringan voip dari *point to point* dengan menggunakan server briker
4. Bagaimana cara mengkonfigurasi jaringan voip

1.4 Batasan Masalah

Dalam penyusunan proyek akhir ini mempunyai batasan masalah, diantaranya adalah :

1. Membahas server briker
2. Membahas jaringan voip
3. Hanya membahas SIP protokol pada server briker
4. Tidak membahas detail tentang TCP/IP
5. Membahas kelayakan penggunaan dari server briker untuk integrasi ke jaringan GSM
6. Hanya menganalisa *bandwidth*, *delay*, *jitter*, *packetloss* dan *throughput* pada jaringan voip *point to point*

1.5 Metodologi Penelitian

Dalam pelaksanaan Proyek akhir ini, penulis melakukan beberapa metode penelitian untuk merealisasikan Proyek akhir ini, yaitu:

1. Studi Literatur, Metode ini dilakukan dengan melakukan studi literatur di Perpustakaan kampus atau di Perpustakaan lain yang berhubungan dengan permasalahan yang akan dibahas, dan membaca buku referensi serta mencari data di situs internet yang dapat mendukung perealisasi proyek akhir ini.
2. Riset dan Aplikasi, Melakukan penelitian tentang proses yang dilakukan dengan dibimbing oleh staf yang sudah ahli di bidangnya.

1.6 Sistematika Penulisan

Secara umum sistematika penulisan proyek akhir ini terdiri dari bab-bab dengan metode penyampaian sebagai berikut:

BAB I PENDAHULUAN

Pada bab ini mengemukakan latar belakang masalah, maksud dan tujuan, rumusan masalah, pembatasan masalah, metodologi penelitian, sistematika penulisan.

BAB II SERVER BRIKER MENGGUNAKAN CARD OPENVOX YANG TERINTEGRASI DENGAN GSM GATEWAY

Pada bab ini membahas tentang teori-teori penunjang yang akan membahas mengenai server briker, VoIP, VoIP gateway, GSM Gateway dan *codec*.

BAB III KONFIGURASI SERVER BRIKER DENGAN OPENVOX DAN GSM GATEWAY

Pada bab ini membahas tentang cara-cara pengkonfigurasi server briker sebagai server voip, konfigurasi softphone x-lite point to point, konfigurasi *softphone x-lite* ke GSM Gateway.

BAB IV ANALISA PENGUKURAN PARAMETER BANDWIDTH, DELAY, JITTER, PACKET LOSS DAN THROUGHPUT PADA JARINGAN VOIP

Pada bab ini membahas tentang analisis pengujian parameter *bandwidth*, *jitter*, *delay*, *packet loss* dan *throughput* pada jaringan voip point to point menggunakan server briker dengan wireshark dan menganalisa terjadinya kesalahan atau kegagalan dalam mengintegrasikan jaringan voip menggunakan server briker ke jaringan GSM.

BAB V PENUTUP

Pada bab ini mengemukakan kesimpulan dan saran-saran yang konstruktif untuk kesempurnaan proyek akhir ini.

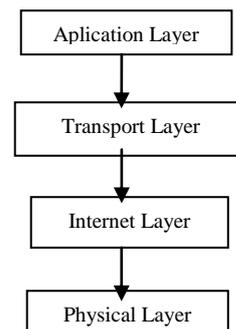
2. SERVER BRIKER MENGGUNAKAN CARD OPENVOX YANG TERINTEGRASI DENGAN GSM GATEWAY

2.1 VOIP (*Voice over Internet Protocol*)

VoIP adalah singkatan dari *Voice Over Internet Protocol* atau sering dikenal sebagai *Internet Telephony*, *IP Telephony*, atau *Digital Phone*. Proses pengiriman suara pada VoIP adalah dengan cara mengirim suara melalui jaringan *internet protocol* (IP). Sehingga dengan teknologi VoIP ini memungkinkan percakapan suara jarak jauh melalui media internet. Jaringan yang biasa digunakan bias berupa internet atau intranet. Teknologi ini bekerja dengan cara merubah data suara yang menjadi kode digital yang dikirimkan melalui jaringan IP. Teknologi komunikasi data merupakan teknologi yang berkembang paling pesat saat ini, dan pemicunya adalah internet, yaitu jaringan komunikasi data yang berbasis protokol komunikasi TCP/IP. Pada komunikasi VoIP ada dua macam protokol yang digunakan untuk signaling diantaranya yaitu H.323 dan SIP dan protokol yang digunakan untuk mengirim paket data audio/video yaitu RTP (*Real Time Transport*).

2.2 TCP/IP

Model TCP/IP (*Transmission Control Protocol / Internet Protocol*) diciptakan oleh US *Departement of Defence* (DoD), karena DoD ingin mendesain *network* yang dapat tetap berfungsi dalam kondisi apapun.



Gambar 2.1 TCI/IP Layer

TCP/IP terdiri dari 4 layer, diantaranya adalah :

1. *Network Access Layer (Physical / Link Layer)*, Layer ini bertugas untuk mengatur semua hal-hal yang diperlukan sebuah paket IP agar dapat dikirimkan melalui sebuah medium fisik jaringan, termasuk di dalamnya *detail* teknologi LAN dan WAN
2. *Internet Layer*, Layer ini memiliki tugas utama untuk memilih jalur terbaik yang akan dilewati oleh sebuah paket data dalam sebuah jaringan. Selain itu *layer* ini bertugas untuk melakukan *packet switching* untuk mendukung tugas utama tersebut.
3. *Transport Layer*, Layer ini menyediakan layanan pengiriman dari sumber data menuju ke tujuan data dengan cara membuat *logical connection* antara keduanya. Layer ini bertugas untuk memecah data dan membangun kembali data yang diterima dari *application layer* ke dalam aliran data yang sama antara sumber dan pengiriman data.
4. *Application Layer*, Layer ini berfungsi untuk menangani *high-level protocol*, masalah representasi data, proses *encoding* dan dialog *control* yang memungkinkan terjadinya komunikasi antar aplikasi jaringan.

2.3 Server Briker

Salah satu VoIP yang berbasis open source adalah briker. Briker adalah salah satu inovasi baru dalam komunikasi. Sama seperti VoIP rakyat, briker dikembangkan berbasis open source dengan mengintegrasikan beragam software open source. Briker adalah salah satu distribusi linux yang didalamnya terdapat server yang memungkinkan bagi pengguna untuk mengimplementasikan layanan VoIP, dan membangun sentral telepon sendiri. Fitur-fitur dalam briker setara dengan fitur-fitur yang terdapat pada PABX. Server briker adalah PBX berbasis IP (IPPBX) yang berbentuk software. Dengan menginstalnya pada komputer, maka komputer tersebut dapat berubah menjadi mesin PBX (*Private Branch Exchange*) dengan kemampuan telekomunikasi via jaringan IP.

Beberapa jenis protokol yang digunakan dalam server briker, diantaranya adalah SIP (*Session Initiation Protocol*), IAX (*Inter Asterisk Exchange*) dan H.323. Namun yang selalu digunakan pada server briker adalah SIP (*Session Initiation Protocol*).

2.3.1 H.323

H.323 adalah salah satu rekomendasi dari ITU-T (*International Telecommunications Union Telecommunications*). H.323 dirancang untuk digunakan khusus pada aplikasi *video conference*, tetapi pada prakteknya H.323 hanya berkembang untuk *voice*-nya saja. merupakan standar yang menentukan komponen, protokol, dan prosedur yang menyediakan layanan komunikasi multimedia. Layanan tersebut adalah komunikasi audio, video, dan data *real-time*, melalui jaringan berbasis paket (*packet-based network*). H.323 berjalan pada jaringan intranet dan jaringan *packet-switched* tanpa mengatur media jaringan yang di gunakan sebagai sarana transportasi maupun protokol *network layer*.

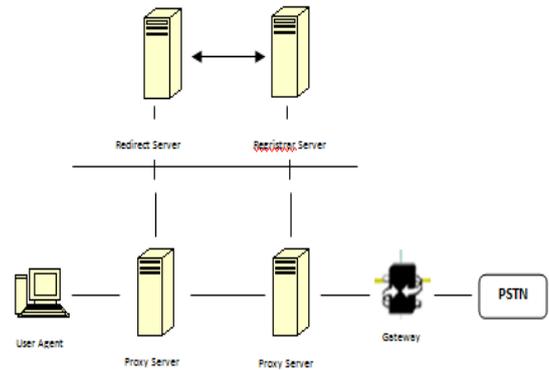
Protokol H.323 menggunakan protokol standar ITU-T lainnya seperti H.255-0 untuk *call control* protocol dan H.245 untuk *logical channel protocol*. H.255-0 digunakan untuk call setup, sedangkan H.245 digunakan untuk *capability* dan *bandwidth control*. Selain itu juga digunakan untuk mengatur jumlah channel yang akan digunakan untuk percakapan.

Elemen dasar jaringan dari protokol H.323 adalah terminal, *gatekeeper* dan *gateway*. Terminal adalah perangkat yang berfungsi sebagai *endpoint*, yang melayani satu atau lebih *user* untuk melakukan pembicaraan dengan *user* lain. *Gatekeeper* berfungsi sebagai pengatur antar *gateway*. *Gateway* adalah pihak

yang berfungsi menghubungkan antara protokol H323 dengan sistem lainnya seperti PSTN dan SIP.[1]

2.3.2 SIP (*Session Initiation Protocol*)

SIP atau *Session Initiation Protocol*, dikembangkan oleh *Internet Engineering Task Force* (IETF) yang merupakan lembaga tertinggi di *Internet*. SIP di desain sebagai protokol multimedia yang dapat memanfaatkan kegunaan dari arsitektur aplikasi internet yang sudah ada. SIP hanya mengatur bagaimana cara membangun dan menutup sebuah sesi komunikasi. SIP menggunakan RTP (*Real Time Transport Protocol*) sebagai media *transfer*. SIP adalah sebuah *signaling protocol* (*application-layer control*) untuk menciptakan, mengatur dan menghentikan sesi komunikasi multimedia antara dua peserta atau lebih. Komponen SIP diantaranya adalah *proxy server*, *registrar server*, *redirect server* dan *user agent*. [2]



Gambar 2.2 Arsitektur SIP

Keterangan.

1. *User agent*, merupakan komponen SIP yang memulai, menerima dan menutup sesi komunikasi. *User agent* terbagi menjadi dua yaitu *User Agent Client* (UAC) dan *User Agent Server* (UAS). *User Agent Client* (UAC) adalah *user agent* yang bertugas untuk memulai sesi komunikasi dan *User Agent Server* (UAS) adalah *user agent* yang bertugas untuk menerima atau menanggapi sesi komunikasi. Diantara UAC dan UAS sama-sama dapat menutup atau mengakhiri sesi komunikasi.
2. *Proxy server*, merupakan komponen penengah antar *user agent*, bertindak sebagai *server* dan *client* yang menerima *request message* dari *user agent* dan menyampaikan pada *user agent* lainnya.
3. *Redirect server*, merupakan komponen SIP yang menerima *request message* dari *user agent*, memetakan alamat SIP *user agent* atau *proxy* tujuan kemudian menyampaikan hasil pemetaan kembali pada *user agent* pengirim (UAC).
4. *Registrar server*, merupakan komponen SIP yang menerima *request message* dari REGISTER. *Registrar server* dapat menambahkan fungsi otentikasi *user* untuk validasi. Dengan kata lain *registrar server* hanya menyimpan *database user* dan otentikasi dan lokasi sebenarnya yang berupa IP dan *port* agar pengguna yang terdaftar dapat dihubungi oleh komponen SIP lainnya.

2.3.3 IAX (*Inter Asterisk eXchange*)

IAX adalah protokol kontrol dan transmisi untuk media *streaming* melalui jaringan *Internet Protocol* (IP). IAX dapat digunakan untuk semua tipe media *streaming* termasuk *video* tetapi tujuan utama dari protokol ini adalah untuk mengontrol IP *voice call*. IAX biasanya digunakan untuk komunikasi VoIP antar asterisk. Pada protokol ini sebenarnya sudah mencakup 2 protokol yaitu untuk kontrol dan untuk transmisi, tidak seperti *Session Initiation Protocol* (SIP) dan *Media Gateway Control Protocol* (MGCP) yang merupakan protokol untuk kontrol sedangkan untuk transmisinya masih dibutuhkan *Real-Time Transfer Protocol* (RTP). Tujuan utama

didesainnya protokol ini adalah untuk meminimalisasi penggunaan *bandwidth*.

2.4 VoIP Gateway

VoIP *Gateway* adalah suatu alat yang digunakan untuk menghubungkan ke jaringan suara konvensional. Secara sederhana dapat dipahami bahwa didalam perangkat VoIP *gateway* terdapat perangkat FXO akan menghasilkan sinyal yang dibutuhkan oleh perangkat FXS dan demikian juga sebaliknya perangkat FXS akan menghasilkan sinyal yang dibutuhkan oleh FXO. FXS (*Foreign eXchange Subscriber*) adalah port yang dihubungkan langsung dengan pesawat telepon. Sedangkan FXO (*Foreign eXchange Office*) adalah port yang dihubungkan langsung dengan jaringan PSTN. Karena FXO adalah port yang dihubungkan langsung dengan jaringan PSTN, jadi pada *port* ini tidak ada sumber tegangan atau tidak terdapat aliran arus [4]. Sedangkan FXS merupakan *port* yang memiliki sumber tegangan atau terdapat aliran arus di dalamnya. Karena *port* ini biasanya langsung terhubung ke pesawat telepon analog. Dari pengertian FXS dan FXO diatas jelas terlihat perbedaannya, FXO yang disambungkan dengan jaringan PSTN dan FXS yang memberikan signal dan tegangan.

FXS (*Foreign eXchange Subscriber*) dan FXO (*Foreign eXchange Office*) dikenal sebagai *Plain Old Telephone Service* (POTS). POTS (*Plain Old Telephone Service*) adalah layanan lokal perusahaan telepon yang pada umumnya mentransmisikan data ke rumah-rumah dengan menggunakan kabel tembaga.

2.5 GSM Gateway

Media gateway GSM to SIP atau SIP to GSM merupakan sebuah *device* yang berfungsi sebagai penghubung antara GSM *network* dengan SIP *server*. *Media gateway* disebut dengan GSM *Gateway*. GSM *Gateway* mampu *me-routing* antara IP, digital, analog dan GSM *networks*. Secara umum GSM *Gateway* terdiri dari *port* WAN, *port* LAN dan slot SIM *card*. Beberapa produk GSM *Gateway* memiliki *port* FXS dan multi slot SIM *Card*.

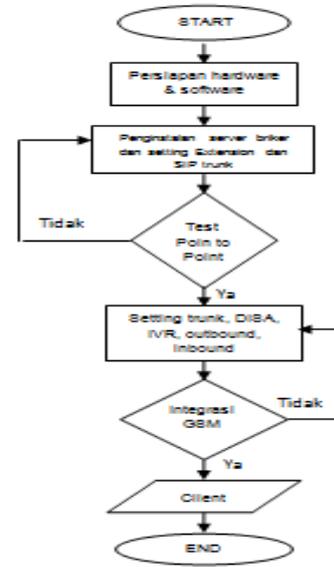
Slot SIM *card* akan terintegrasi ke nomor *extension* yang terdaftar pada SIP *server*. Sehingga seseorang dapat melakukan panggilan dan menerima panggilan melalui *softphone*, SIP *phone*, ataupun telepon analog. Dengan menggunakan *device* ini, seseorang mampu melakukan panggilan atau menerima panggilan melalui VoIP.



Gambar 2.3 GSM Gateway

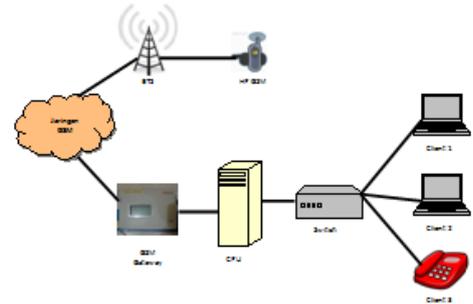
3. KONFIGURASI SERVER BRIKER DENGAN OPENVOX DAN GSM GATEWAY

3.1 Flowchart Server Briker Dengan Card Openvox



Gambar 3.1 Flowchart Server Briker Dengan Card Openvox

3.2 Topologi Server briker Dengan Card Openvox



Gambar 3.2 Topologi Server Briker Dengan Card Openvox

3.3 Konfigurasi Server Briker Sebagai Server VoIP

Server Briker adalah distro dari server voip yang berbasis *open source*. Server briker merupakan salah satu turunan Linux. Berikut adalah beberapa cara untuk menjadikan server briker sebagai server voip.

3.3.1 Proses Instalasi

Untuk menggunakan server briker sebagai server voip dibutuhkan penginstalan server briker tersebut terlebih dahulu. Berikut adalah cara menginstal server briker. Untuk menggunakan server briker sebagai server voip dibutuhkan penginstalan server briker tersebut terlebih dahulu. Berikut adalah cara menginstal server briker.

1. Konfigurasi bios agar melakukan *booting* pertama kali dari CDROM, lalu masukan CD briker IPPBX ke CDROM. Kemudian instal server briker pada *harddisk* dan ketikkan *wizard* kemudian enter.
2. Kemudian briker akan otomatis memeriksa *hardware* yang terpasang dengan cara memeriksa CD-ROM.
3. Lalu briker akan memberikan pilihan untuk mempartisi *harddisk*. Lalu pilih *manual* untuk mempartisi *harddisk*
4. Kemudian briker akan memformat seluruh isi *harddisk* dan mempartisi *harddisk* tersebut
5. Kemudian briker akan otomatis menginstal *base-system* pada *harddisk*
6. Briker juga akan menginstal *software* yang lainnya.
7. Setelah menginstal *software* lainnya, briker juga menginstal GRUB *Boot Loader* pada *harddisk*.

8. Setelah itu akan muncul tampilan *loading finishing* dari briker tersebut.
9. Setelah itu akan muncul tampilan seperti dibawah ini dan pilih *yes* untuk menyelesaikan proses instalasi tersebut.

3.3.2 Web Login

1. Untuk log in briker buka *web browser* terlebih dahulu dan ketikkan alamat IP 192.168.2.2 pada kotak URL
2. Kemudian akan muncul kota dialog seperti gambar dibawah ini, lalu isikan *username* dengan *administrator* dan isikan *password* dengan Briker. Lalu klik login.
3. Kemudian akan masuk ke dalam *home* briker.

3.3.3 Membuat Extension

1. Setelah masuk ke dalam IPPBX Administrator lalu pilih dan klik *extension* setelah itu klik *Add Extension* dan isi kotak *User Extension*, *Display Name*, *Account Code* dan *Secret* kemudian setelah selesai klik *submit*.
2. Lalu ikuti erintah selanjutnya yaitu klik *Apply Configuration Changes*, setelah itu klik *Continue With Reload* dan tunggu hingga proses *loading* selesai. Kemudian lakukan cara seperti diatas untuk membuat lebih banyak *extension*.

3.3.4 Membuat SIP Trunk

1. Setelah masuk ke dalam IPPBX Administrator lalu pilih dan klik menu *Trunk*, setelah itu pilih *Add SIP Trunk*.
2. Setelah itu isi kotak *Outbound Caller ID*. Kemudian pada menu *outgoing* isi *host*, *user name*, *number extension* dan *secret*.
3. Lalu pada menu *registration* isi kotak *Register String* dengan *Username@Secret@Trunk_Name*. setelah itu tekan *submit*.
4. Kemudian klik *Apply Configuration Changes*, lalu klik *Continue With Reload* dan tunggu hingga proses *loading* selesai.

3.3.5 Membuat ZAP Trunk

1. Setelah masuk ke dalam IPPBX Administrator lalu pilih dan klik menu *Trunk*, kemudian pilih *Add Zap Trunk* dan kemudian pilih ZAP/g0.
2. Kemudian pada kotak *Dial Rules Wizard* pilih menu *Always Dial With Prefix*.
3. Setelah itu akan muncul kotak dialog seperti gambar dibawah ini, dan masukan 8 digit belakang nomor yang dituju setelah itu tekan ok.
4. Lalu akan muncul tampilan berikutnya seperti yang ada pada gambar di bawah ini, dan isikan *prefix* nomor yang dituju setelah itu tekan ok.
5. Jika sudah, akan muncul nomor yang sudah dimasukan pada kotak *Dial Pattern*. Setelah itu tekan *Submit*.
6. Kemudian klik *Apply Configuration Changes*, lalu klik *Continue with Reload* dan tunggu hingga proses *loading* selesai.

3.3.6 Mengatur Inbound Routes

1. Setelah masuk ke dalam IPPBX Administrator kemudian pilih dan klik *Outbound Routes*, lalu pilih *outbound routes* yang sudah ada yaitu PSTN_PLMN.
2. Kemudian isi *route password* dengan 1234 lalu tekan *submit*.
3. Lalu pilih dan klik menu selanjutnya yaitu *Apply Configuration Changes*, setelah itu klik *Continue With Reload* dan tunggu hingga proses *loading* selesai.

3.3.7 Mengatur IVR (Interactive Voice Response)

1. Setelah masuk ke dalam IPPBX Administrator kemudian pilih dan klik menu IVR.
2. Lalu isi kotak *Change Name* dan *Timeout*. Kemudian klik *save*.
3. Setelah itu klik perintah selanjutnya yaitu *Apply Configuration Changes*, lalu klik *Continue With Reload* dan tunggu hingga proses *loading* selesai.

3.3.8 Mengatur Ring Group

1. Setelah masuk ke dalam IPPBX Administrator kemudian pilih menu *Ring Group*.
2. Kemudian klik *Add Ring Group*, setelah itu isi pada kotak *Group Description* dan *Extension List* dengan daftar *extension* yang akan digunakan setelah dibuat pada menu *extension*.
3. Setelah itu tekan *submit*.
4. Lalu pilih dan klik perintah selanjutnya yaitu *Apply Configuration Changes*, kemudian klik *Continue With Reload* dan tunggu hingga proses *loading* selesai.

3.3.9 Mengatur DISA

1. Setelah masuk ke dalam IPPBX Administrator kemudian pilih menu DISA.
2. Setelah itu klik *Add DISA*, kemudian isi pada kotak *Name* dan PIN yang terdapat didalam menu DISA tersebut lalu tekan *submit*.
3. Setelah itu klik perintah selanjutnya *Apply Configuration changes*, lalu klik *Continue With Reload* dan tunggu hingga proses *loading* selesai.

3.3.10 Mengatur Inbound

1. Setelah masuk ke dalam IPPBX Administrator lalu pilih dan klik menu *Inbound*, kemudian klik *Add Routes* dan isikan kotak *Description*, *DID number* dan *Caller ID*.
2. Lalu pada *Set Destination* pilih dan klik *Ring Group*, kemudian tekan *submit*.
3. Lalu pilih dan klik perintah selanjutnya yaitu *Apply Configuration Changes*, dan klik *Continue With Reload* lalu tunggu hingga proses *loading* selesai.

3.4 Konfigurasi Softphone X-Lite Point to Point

Setelah menginstal server briker, lalu server tersebut dites apakah server tersebut bias menelon dari satu *extension* ke *extension* lainnya. Berikut adalah konfigurasi *softphone X-Lite point to point*.

1. Buka *softphone* terlebih dahulu, kemudian klik *show menu* pada *softphone x-lite* setelah itu pilih *SIP account setting*.
2. Kemudian akan keluar menu *SIP account*, lalu klik *add*.
3. Kemudian akan keluar kotak dialog seperti gambar di bawah ini, kemudian isikan *Display name*, *User name*, *Password*, *Authorization user name*, dengan nama, nomor dan *password extension* yang diinginkan, lalu pada kotak *Domain* isi dengan alamat IP server. Setelah itu pilih *domain* pada bagian *Send Outbound*.
4. Lalu pilih menu *topology* pada menu *properties of account*, kemudian gunakan *use local ip address*. Lalu klik *apply* untuk menyimpan pengaturan tersebut kemudian ok.
5. Lalu akan muncul kotak dialog seperti gambar berikut yang menandakan bahwa *extension* yang telah dibuat berhasil disimpan, lalu klik *close*.
6. Setelah di *close* maka pada *softphone x-lite* akan muncul keterangan seperti gambar berikut yang menandakan bahwa *extension* untuk client a sudah berhasil dibuat. Untuk mengatur *softphone* pada *client b* ulangi langkah tersebut pada *softphone* yang berbeda.
7. Setelah kedua *client* siap digunakan, maka masukan nomor *extension client b* pada *softphone client a* lalu tekan tombol *dial*.
8. Setelah *client b* mendapatkan peringatan bahwa ada telepon masuk, untuk menjawab telepon tersebut *client b* menekan tombol *dial*. Untuk menakhiri telepon tekan tombol *hang-up*.

3.5 Konfigurasi Softphone X-Lite Ke Nomor GSM

Konfigurasi *softphone x-lite* ke nomor gsm adalah bentuk simulasi integrasi jaringan voip ke jaringan gsm. Berikut adalah langkah-langkahnya.

1. Buka terlebih dahulu *softphone x-lite*, kemudian buat dan atur *extension* yang akan digunakan untuk menghubungi ke nomor gsm.
2. Setelah *extension* pada *softphone* siap digunakan, lalu masukan nomor gsm yang dituju kemudian tekan *dial*.

3. Setelah menekan tombol *dial*, lalu operator pada *softphone* akan meminta anda untuk memasukkan *password* yang telah diatur sebelumnya pada menu *Outbound Route*. Setelah memasukkan *password* kemudian nomor *extension* tersebut akan terhubung ke nomor gsm yang dituju. Seperti yang ada pada gambar di bawah ini, yang menandakan bahwa *softphone* tersebut sudah tersambung dengan nomor gsm yang dituju

4. ANALISA BANDWIDTH, DELAY, JITTER, PACKET LOSS DAN THROUGHPUT PADA JARINGAN VOIP

4.1 Cara Mengukur Parameter Bandwidth, Delay, Jitter, Packet Loss dan Throughput Pada Jaringan Voip Dengan Menggunakan Wireshark

1. Buka terlebih dahulu *software* wireshark, kemudian *client a* dan *client b* saling berkomunikasi dengan menggunakan *softphone x-lite*.
2. Setelah itu klik menu *capture*, kemudian pilih *interfaces* dan lihat bagian *client* mana yang data pakatnya berjalan atau aktif. Lalu klik *start* untuk melihat besarnya parameter QoS (*Quality of Service*) seperti *bandwidth*, *jitter*, *delay*, *throughput* dan *packet loss*.
3. Untuk melihat hasil perhitungan *bandwidth*, *packet loss* dan *throughput* pilih menu *statistic*, lalu pilih IO *Graph*. Untuk melihat perhitungan *bandwidth*, *packet loss* dan *throughput* dapat mengganti pada pilihan yang terdapat pada bagian Y Axis pada kotak unit. Untuk menyimpan gambar perhitungan *bandwidth*, *packet loss* dan *throughput* dapat mengklik pilihan *save*.
4. Untuk melihat perhitungan *delay* dapat dilihat dengan cara mengklik *statistic*, kemudian memilih *summary*. Jika sudah selesai, kemudian memilih *close*.

4.2 Pengukuran Parameter Bandwidth, Delay, Jitter, Packet Loss Dan Throughput Pada Jaringan Voip Point to Point Dengan Menggunakan Wireshark.

4.2.1 Bandwidth

Bandwidth adalah suatu ukuran dari banyaknya informasi yang dapat mengalir dari suatu tempat ke tempat lain dalam suatu waktu tertentu. Satuan yang digunakan untuk bandwidth adalah bits per second atau sering disingkat dengan bps. Bagus atau tidaknya *bandwidth* sangat dipengaruhi oleh system jaringannya. *Bandwidth* berbanding lurus dengan system jaringannya. Apabila system jaringan bagus, maka kualitas *bandwidth* akan baik pula. Namun apabila kualitas system jaringan jelek, maka *bandwidth* juga akan menjadi jelek. *Bandwidth* juga sangat dipengaruhi oleh protokol dan *codec* yang digunakan.

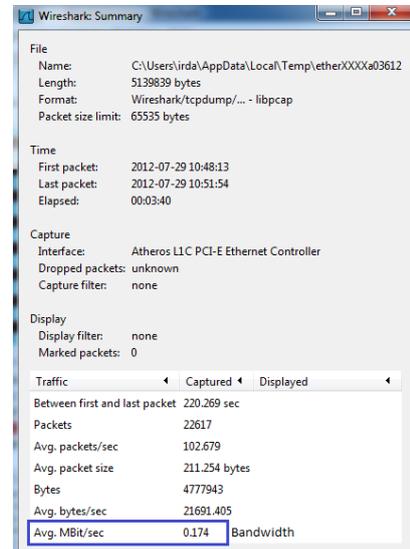
Analisa hasil pengukuran *bandwidth* dengan menggunakan wireshark diantaranya ada pengukuran berdasarkan pengambilan gambar *summary*. Berdasarkan hasil dari pengukuran yang terdapat pada *summary bandwidth* yang didapat rata-ratanya adalah 0,174 Mbps atau sama dengan 174 Kbps. Jika dilihat dari standar *codec* yang digunakan pada server briker dalam jaringan voip yaitu *codec G.711* dengan standar kecepatan *bandwidth* adalah 64 Kbps maka hasil yang didpata jika dibandingkan dengan hasil *capture summary* adalah bagus.

Tabel 4.1 Standart Codec

Jenis Codec	Bandwidth
G.711	64 Kbps
G.729	8 Kbps
G.723.1	6,4 Kbps
G.723.1	5,3 Kbps
G.726	32 Kbps
G.728	16 Kbps

(Sumber : TIPHON) [6]

Karena pada *codec G.711* standar kecepatannya ada 64 Kbps sedangkan pada *summary* adalah 174 Kbps. *Bandwidth* berkualitas bagus apabila delay yang terdapat pada jaringan tersebut sedikit sehingga dapat menghasilkan *bandwidth* yang berkualitas.



Gambar 4.1 Summary Bandwidth

4.2.2 Packet Loss

Packet loss adalah perbandingan seluruh paket IP yang hilang dengan seluruh paket IP yang dikirimkan antara *source* dan *destination*. Penyebab terjadinya *packet loss* adalah terjadinya *overload* trafik di dalam jaringan, tabrakan (*congestion*) dalam jaringan, *error* yang terjadi pada media fisik dan kegagalan yang terjadi pada sisi penerima yang disebabkan karena *overflow* yang terjadi pada buffer.

Tabel 4.2 Performansi jaringan IP berdasarkan *packet loss*

KATEGORI DEGRADASI	PACKET LOSS
Sangat bagus	0
Bagus	3%
Sedang	15%
Jelek	25%

Sumber : TIPHON)

Rumus Loss :

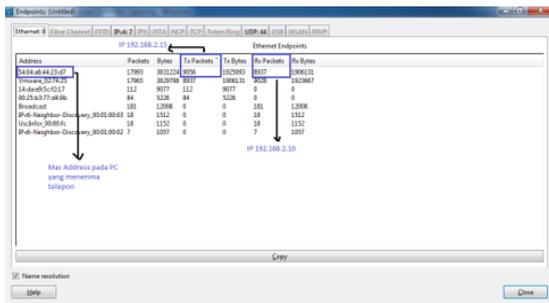
$$Packet Loss = \frac{Packet Tx - Packet Rx}{Packet Tx} \times 100 \%$$

Data untuk menghitung *packet loss* dengan menggunakan perhitungan manual.

Maka, *packet loss* adalah

$$= \frac{9056 - 8937}{9056} \times 100 \% = \frac{119}{9056} \times 100 \%$$

$$= 0,013 \times 100 \% = 1,3 \%$$



Gambar 4.2 Endpoints

Analisa hasil perhitungan diatas didapat *packet loss* sebesar 1,3 %. Ketika hasil perhitungan yang menggunakan rumus diatas tersebut dicocokkan dengan tabel 4.2 maka masuk kedalam kategori degradasi sangat bagus. Karena sedikitnya paket yang hilang dari paket yang telah dikirimkan. *Packet loss* dapat terjadi disebabkan oleh penurunan sinyal dalam media jaringan dan kesalahan *hardware* jaringan. Apabila hasil paket yang hilang lebih banyak dibandingkan dengan jumlah total paket yang dikirimkan maka jaringan tersebut akan masuk ke dalam kategori degradasi tidak bagus atau jelek. Karena akan banyak data yang hilang dan data yang sampai pada penerima tidak sempurna.

4.2.3 Delay

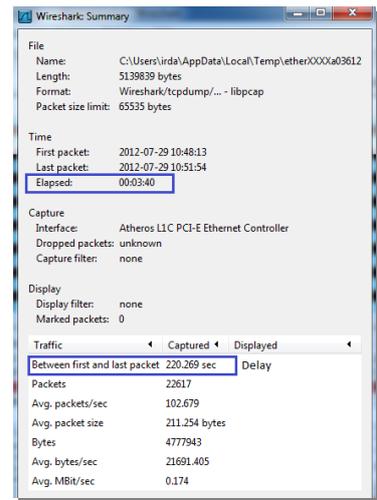
Delay adalah waktu tunda suatu paket yang diakibatkan oleh proses transmisi dari suatu sumber (pengirim) ke tujuan (penerima). Hal tersebut juga dapat disebut dengan *one-way delay*. Delay didalam jaringan dapat digolongkan sebagai berikut *delay processing*, *delay packetization*, *delay serialization*, *delay jitter buffer* dan *delay network*. Delay dapat menyebabkan penurunan kualitas suara. Rekomendasi ITU-T G.114 menetapkan batasan delay seperti pada table 4.3 berikut. [7]

Tabel 4.3 Batasan Delay Menurut Rekomendasi ITU-T G.114 (ITU-T Recommendation, 2003 : No page)

Delay Range	ITU-T Recommendation
0 - 150 ms	Kualitas Sangat Bagus
150 - 400 ms	Kualitas Bagus
> 400 ms	Kualitas Jelek

Hasil pengukuran *delay* dengan menggunakan wireshark sebesar 220,269sec = 220269 ms. Berdasarkan data yang terdapat pada hasil pengambilan gambar *summary* terdapat waktu yang digunakan dalam percakapan tersebut. Jadi, *delay* dibagi dengan waktu yang digunakan untuk percakapan maka mendapatkan hasil sebagai berikut :

$$= \frac{220269 \text{ ms}}{220000 \text{ ms}} = 1.001 \text{ ms} = 1 \text{ ms}$$



Gambar 4.3 Summary Delay

Jadi, dari hasil perhitungan diatas, didapatkan kesimpulan bahwa setiap 1 ms waktu yang digunakan untuk melakukan percakapan pada jaringan voip mempunyai *delay* sebesar 1 ms. Pada hasil perhitungan diatas, apabila dilihat serta dicocokkan dengan tabel 4.3 hasil *delay* mempunyai kualitas yang sangat bagus berdasarkan standar rekomendasi dari ITU-T G.114.

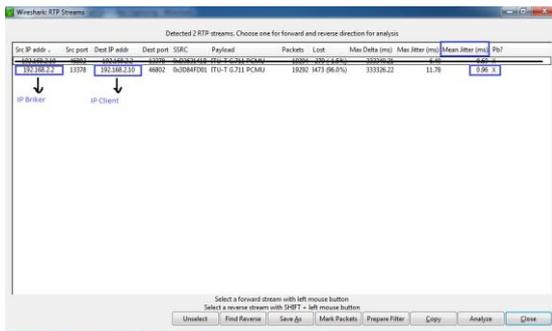
4.2.4 Jitter

Jitter merupakan variasi *delay* antar paket yang terjadi pada jaringan IP. Besarnya nilai *jitter* akan sangat dipengaruhi oleh variasi beban trafik dan besarnya tumbukan antar paket (*congestion*) yang ada di dalam jaringan IP. Terdapat empat kategori penurunan performansi jaringan berdasarkan nilai *peak jitter* sesuai dengan versi *TIPHON* (Joesman 2008) [6], yaitu :

Tabel 4.4 Performansi Jaringan IP Berdasarkan Parameter *jitter*

Kategori Degradasi	Peak Jitter
Sangat Bagus	0 ms
Bagus	0 - 75 ms
Sedang	76 - 125 ms
Jelek	125 - 225 ms

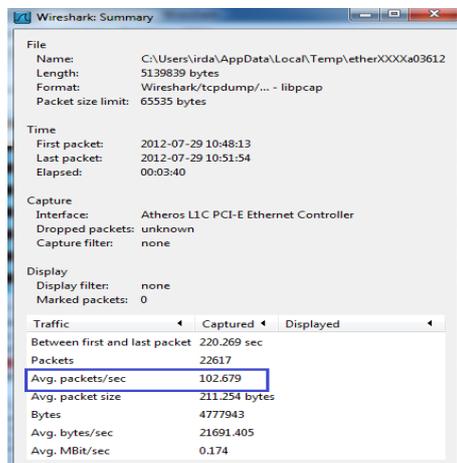
Dari hasil pengukuran *jitter* pada gambar 4.4 dengan menggunakan wireshark dapat disimpulkan bahwa hasil *jitter* yang didapat adalah sebesar 0,96. Jika dilihat pada tabel 4.4 kategori degradasi sangat bagus 0 ms, bagus jika 0 -75 ms, sedang 76 – 125 ms dan jelek jika 125-225 ms. Melihat dari tabel tersebut berdasarkan kategori degradasi diatas dapat disimpulkan bahwa hasil *jitter* yang diukur dengan menggunakan wireshark hasilnya bagus jika dibandingkan dengan tabel 4.4 diatas. Factor-faktor yang dapat mempengaruhi nilai *jitter* diakibatkan oleh variasi nilai dalam panjang antrian dan dalam waktu pengolahan data.



Gambar 4.4 Jitter

4.2.5 Troughput

Troughput adalah *bandwidth* actual yang terukur pada suatu ukuran tertentu dalam suatu hari menggunakan rute internet yang spesifik ketika sedang mendownload suatu file.



Gambar 4.5 Summary Troughput

Hasil pengukuran *troughput* dengan menggunakan wireshark antara *troughput* yang ada pada hasil pengukuran *summary* dan hasil pengukuran pada grafik *troughput* hal ini disebabkan karena dalam pengambilan gambar tidak bias dalam satu waktu yang bersamaa jadi hasil pengukuran tersebut berbeda. Kemudian, hal lainnya disebabkan oleh *noise*. *Noise* yang dapat menyebabkan hasil *troughput* berbeda adalah *noise thermal*. Karena *noise thermal* tidak terlalu pada transmisi *voice* namun sangat berpengaruh pada transmisi data. Sedangkan *troughput* adalah kemampuan suatu jaringan dalam melakukan pengiriman data. Jadi *noise thermal* sangat mempengaruhi besar kecilnya *troughput* pada jaringan komunikasi data.

4.3 Analisa Terjadinya Kesalahan Akibat Software, Hardware, Konfigurasi Integrasi Jaringan Voip Ke Jaringan GSM

Server briker adalah salah satu dari sekian distro voip yang berbasis *open source*. Server briker adal server untuk voip (*Voice Over Internet Protocol*). Untuk mengintegrasikan jaringan voip ke jaringan gsm, maka dibutuhkan sebuah *device* atau sebuah perangkat yaitu voip *gateway* dan GSM *gateway*. Voip *gateway* berfungsi untuk menghubungkan ke jaringan konvensional biasa. Sedangkan GSM *gateway* berfungsi untuk menghubungkan jaringan GSM dengan SIP trunk yang ada pada pengaturan server briker dalam voip.

Dalam hal ini, jaringan voip tidak dapat terhubung disebabkan oleh kerusakan pada perangkat atau *hardware* yaitu *card openvox*. Hal ini yang menyebabkan tidak dapat terhubungnya jaringan voip ke jaringan gsm. Karena inputan yang berupa sinyal analog dari *softphone* yang menghubungi

nomor gsm yang dituju kemudian dirubah menjadi paket data atau sinyal digital karena sinyal tersebut masuk ke dalam jaringan LAN yang berbasis IP kemudian ketika di transmisikan untuk ke jaringan gsm sinyal digital tersebut tidak dapat berubah. Hal ini disebabkan oleh *card openvox* yang rusak. Dimana *card openvox* berfungsi sebagai perubah sinyal digital ke sinyal analog. Sebab, dalam jaringan gsm sinyal yang digunakan adalah sinyal analog. Jadi, karena *hardware card openvox* yang rusak tidak dapat merubah sinyal digital menjadi sinyal analog untuk dapat di transmisikan ke dalam jaringan gsm.

Hal tersebut dapat diketahui ketika dalam penelitan mengganti server yang digunakan. Dari server briker menjadi server asterisk. Ketika menggunakan server asterisk, terlebih dahulu harus mendownload *drivers* openvox pada saat sedang dalam proses install software asterisk tersebut. Kemudian dalam kondisi mendownload *drivers* openvox tidak ada kendala apapun dan download berhasil dengan sukses. Namun ketika *drivers* tersebut diinstal dan menunggu proses *loading* dalam waktu ± 3 atau 4 jam hasil yang didapat adalah *FXO NOT DETECTED*. Sedangkan *card openvox* sudah terpasang di dalam cpu yang akan dijadikan sebagai server voip.

5. PENUTUP

5.1 Kesimpulan

Berdasarkan dari hasil penelitian simulasi yang telah dilakukan maka dapat disimpulkan sebagai berikut :

1. Dari simulasi jaringan voip yang dilakukan dapat disimpulkan bahwa hasil kecepatan *bandwidth* yang didapat dar hasil *capture summary* dengan menggunakan wireshark adalah termasuk ke dalam kategori degradedasi bagus. Karena standar kapasitas kecepatan *bandwidth* pada server yang digunakan yaitu dengan menggunakan *codec* G.711 yaitu 64 Kbps. Sedangkan yang dihasilkan dengan menggunakan wireshark adalah 174 Kbps. Sedangkan pada parameter *troughput* dapat dilihat bahwa yang dihasilkan dengan menggunakan wireshark adalah sebesar 109,679 *bytes/sec* atau 0,199 Kbps. Kualitas *troughput* dapat dilihat dari sistem jaringan yang digunakan. Karena *troughput* berbanding lurus dengan sistem jaringannya. Apabila jaringan yang digunakan bagus, maka *troughput* yang dihasilkan juga bagus dan kebalikannya.
2. Pada parameter *packet loss* dapat disimpulkan bahwa dalam jaringan voip *point to point* dengan menggunakan server briker sebesar 1,3%. Apabila dilihat dalam table 4.2 maka *packet loss* sebesar 1,3% termasuk kedalam kategori degradedasi agus. Karena semakin sedikit paket yang hilang dari jumlah keseluruhan paket yang dikirim maka semakin bagus pula hasil yang diterima oleh penerima (*Receiver*). Salah sau penyebab terjadinya *packet loss* adalah terjadinya *overload* trafik di dalam jaringan, terjadinya tabrakan dalam jaringan dan *eror* yang terjadi pada media fisik (*hardware*).
3. Pada parameter *delay* dapat dilihat hasil *delay* yang didapat dari jaringan voip *point to point* dengan menggunakan wireshar adalah sebesar 1ms. Apabila dibandingkan ke dalam tabel standarisasi *delay* menurut ITU-T *Recommendation* G.114 adalah termasuk kedalam kategori degradasi bagus. Jadi komunikasi dapat berlangsung dengan lancar karena waktu tunda yang di dapat hasilnya sedikit. Sedangkan pada parameter *jitter* saat diukur dengan menggunakan wireshark hasilnya sebesar 0,96. Setelah dibandingka ke dalam tabel standarisasi *jitter* 0,96 yang didapat dengan menggunakan wireshark termasuk ke dalam kategori bagus. Hal ini diarenakan *delay* yang didapat juga dalam kualitas bagus karena *jitter* adalah variasi *delay* antar paket yang terjadi di dalam jaringan IP.

5.2 Saran

Untuk mendapatkan hasil yang lebih baik, penulis menyarankan untuk melakukan hal-hal sebagai berikut :

1. Gunakan *software* yang lain untuk membuktikan apakah hasil yang didapatkan dengan menggunakan *software* sebelumnya akurat atau tidak.
2. Dibutuhkan pengamatan lebih lanjut untuk dapat melihat hasil analisis yang lebih baik dan lebih akurat dari hasil sebelumnya.

Daftar Pustaka

- [1] H.323, diakses dari :
<http://donupermana.wordpress.com/makalah/voip>
- [2] SIP (Session Initiation Protocol) diakses dari :
<http://www.karyatulisilmiah.com/pengertian-sip.html>
- [3] Sugeng Winarno, 2008, Membangun Telepon Berbasis VoIP, Informatika – Bandung
- [4] Foreign eXchange Office, diakses dari :
<http://www.kerjakeras.blogger-indonesia.com/2009/10/istilah-telekomunikasi-fxo-foreign.html>
- [5] www.briker.org
- [6] Joesman 2008, *simulasi Jaringan Berbasis Paket dengan mempergunakan simulator OPNET*, 3 April 2008, diakses dari : <http://joesman.wordpress.com/> viewed 20 Juli 2012
- [7] ITU-T Recommendation G.114 05/2003. *One way transmission time*, diakses dari :
https://docs.google.com/viewer?a=v&q=cache:rDjqnZGI08J:www.itu.int/rec/dologin_pub.asp?lang%3De%26id%3DT-REC-G.114-200305-1!!PDF-E%26type%3Ditem%3Ditem%3Ditu%3Dg%3D114%3Drecommendation%3Done%3Dway%3Dtransmission%3Dtime&hl=id&gl=id&pid=bl&srcid=ADGEEsj1nzV_dN6VH5nymwoXTHKkmQSEN8Q_wFmnb9_qqHxELZRacOJ9bvFVEUwLyydpEY7NdpiY2gQXKdbVNc7hKk4MPIkqd2owyFU1v-ZrtDFcdLaEm6te02_ABFGXQmTjo1wwwzcH&sig=AHIEtbQVx1DyUea1sa9gM-Va6jdsFEJQag viewed 20 Juli 2012.