Bab IV

IMPLEMENTASI DAN PENGUJIAN

Pada bab ini akan dilakukan implementasi dan pengujian terhadap sistem *VoIP* yang dibangun . Implementasi dilakukan berdasarkan analisis dan perancangan yang telah dibuat. Setelah implementasi selesai, maka selanjutnya akan dilakukan pengujian pada kinerja dari *QoS priority queuing* yang terdapat pada *router* mikrotik pada sistem *VoIP* yang telah dibangun.

4.1 Instalasi *VoIP Server* ASTERISK

Untuk memulai instalasi *server asterisk*, langkah pertama yang dilakukan adalah mempersiapkan file installer *asterisk for windows*. Sumber file ini dapat diperoleh dari koneksi langsung dari *internet* dengan merujuk kepada situs resmi *asterisk* di alamat http://www.*asterisk*win32.org atau dapat di unduh disitus-situs yang menyediakan layanan unduh seperti www.download.com atau yang lainnya.

4.1.1 Instalasi *ASTERISK*

Versi *asterisk* yang dipakai adalah *asterisk for windows*. lakukan proses instalasi dengan melakukan *double click* pada *file* yang telah diunduh lalu akan muncul jendela seperti berikut :



Gambar IV.1 Proses awal instalasi *asterisk*

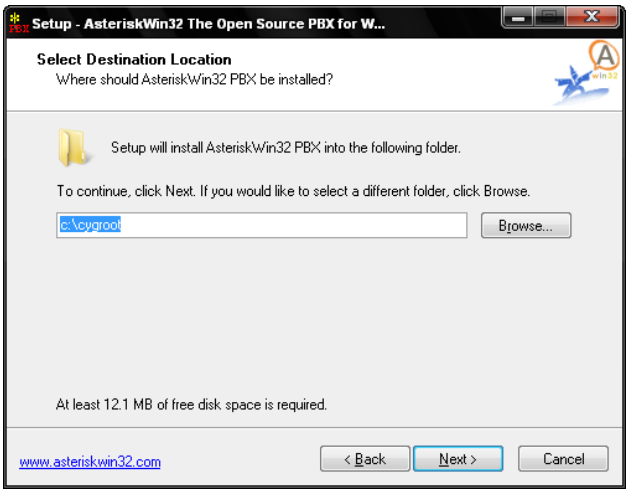
Tekan tombol "*Next*" untuk melanjutkan. Lalu akan muncul jendela lisensi persetujuan pengguna, pilih " *i accept the agreement*" lalu tekan tombol "*Next*".



Gambar IV.2 Perjanjian persetujuan *asterisk*

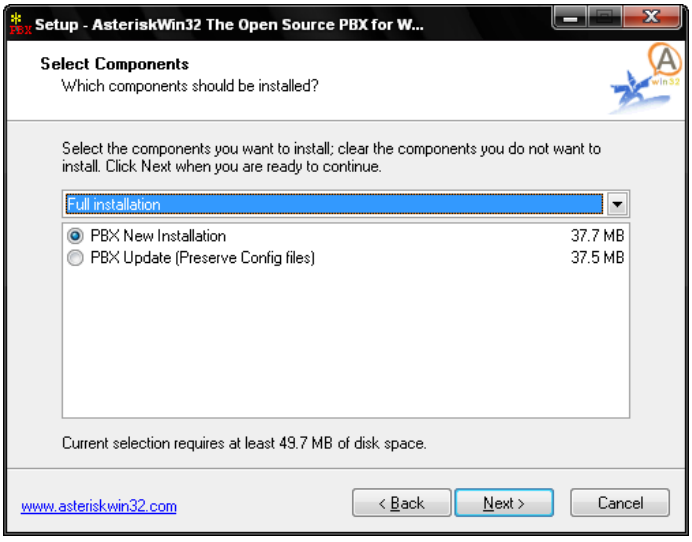
Kemudian tentukan folder tempat instalasi *server* yang akan digunakan yang bernama cygroot. Secara *default* proses instalasi akan memilih drive c sebagai

tempat instalasinya.Tekan tombol next untuk melanjutkan.



Gambar IV.3 Pemilihan lokasi folder instalasi

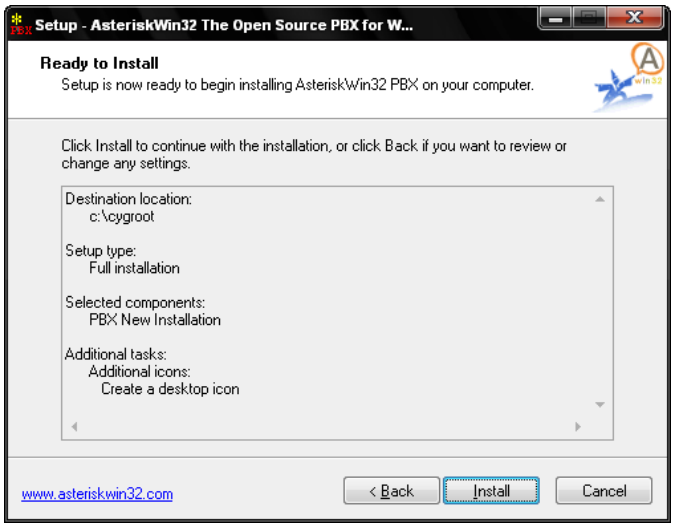
Pilih mode setup yang diinginkan, apakah instalasi baru atau *upgrade* dari versi sebelumnya. Pada contoh ini kita pilih instalasi baru karena kita belum pernah menginstall *asterisk* sebelummnya.



Gambar IV.4 Penentuan proses instalasi

Proses instalasi akan memberitahu opsi-opsi yang tadi telah kita pilih, jika anda sudah yakin dan tidak akan ada perubahan tekan tombol *install* untuk melanjutkan

proses instalasi.



Gambar IV.5 Konfirmasi instalasi

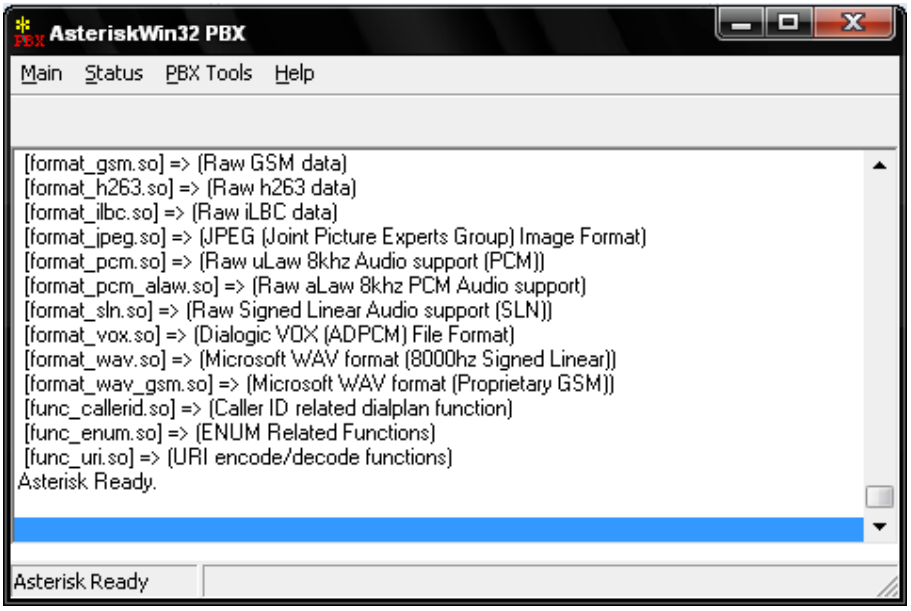
jika proses instalasi telah berhasil akan ditampilkan jendela seperti berikut .



Gambar IV.6 Instalasi berhasil

Jalankan *asterisk* dengan cara klik dua kali pada icon *asterisk* win32 yang ada pada *desktop* lalu tunggu beberapa saat sampai muncul jendela yang menyatakan bahwa sistem *asterisk* telah siap seperti yang terlihat pada gambar

berikut :



Gambar IV.6 *Console* utama *asterisk*

Secara default *asterisk* telah mempunyai tiga akun sip yang dapat diregistrasi yaitu *user* 3000, 3001 dan 3002.Secara default *asterisk* akan membuat satu folder di partisi utama komputer anda, sebagai contoh dalan hal in ada folder cygroot di drive c milik komputer penulis setelah instalasi *asterisk*. Folder ini berisi file-file penting yang akan digunakan oleh *asterisk server* untuk menjalankan fungsinya sebagai *VoIP server*.

4.1.2 Konfigurasi *ASTERISK*

Konfigurasi utama *asterisk* adalah konfigurasi SIP dan *dial plan*. Konfigurasi SIP dilakukan dengan mengkonfigurasi file *sip.conf* sedangkan untuk *dial plan* dilakukan dengan mengkonfigurasi file *extensions.conf*. kedua file ini terdapat pada folder C:\cygroot*asterisk*\etc.

4.1.2.1. Konfigurasi sip.conf

File sip.conf terbagi dalam beberapa blok. Blok yang pertama adalah blok [general]. Jka bagian ini dikonfigurasi akan memiliki dampak bagi semua data *account* yang terdaftar. Setiap baris konfigurasi memiliki arti sendiri-sendiri. Berikut akan dijelaskan setiap baris konfigurasi yang penting.

a. allowguest = yes

Memungkinkan panggilan dari *user* yang belum terdaftar atau belum *login. User* yang belum *login* dapat memanggil *user* Iain yang sedang *online.* Di sini. *user* akan teridentifikasi dengan nama *Asterisk*.

b. allowoverlap=no

Fungsi ini untuk menonaktifkan overlap dialing

d. bindport=5060

ini adalah fungsi untuk bind port UDP 5060. Port 5060 merupakan port standar protokol SIP.

f. bindaddr=0.0.0.0

Fungsi di atas berfungsi untuk menerima semua *address* untuk di *bind*.

g. Konfigurasi *codec* yang bisa digunakan melalui *Asterisk* adalah sebagai berikut :

disallow=all

allow=gsm

allow=ilbc

allow=ulaw

allow=speex

Fungsi di atas dimulai dengan *men-disable* semua *codec* terlebih dahulu, kemudian mengaktifkan *codec* yang akan didukung oleh *Asterisk*. Urutan di atas disusun berdasarkan prioritas *codec* yang ingin digunakan.

h. nat=yes

Mengasumsikan bahwa semua *user* berada dibelakang NAT

i. canreinvite=no

Perintah diatas bertujuan agar *voice* melalui *asterisk*, tidak *peer to peer*.

j. qualify=yes

Mengirim paket kecil seara periodik untuk mengetahui *link latency* dan membuat NAT tetap aktif.

Konfigurasi data account dilakukan pada file *sip.conf* juga. Tiap *user* dikonfigurasi dalam blok masing-masing. Attribut-attribut yang ada pada blok *user account* adalah attribut khusus yang unik untuk tiap *data account*. Jika setiap attribut pada *data account* sama. sebaiknya atribut itu diletakan pada blok [general]. Tapi jika ada atribut-atribut khusus pada masing-masing *user*, sebaiknya diletakan pada blok *data account user.*

Dalam konfigurasi data *account* di sini, setiap *user* memiliki attribut yang sama, sehingga semua telah dikonfigurasi pada blok [general]. Dengan

demikian pada blok data account, yang dikonfigurasi disini hanyalah informasi *user* account. Berikut contoh blok data account.

[3001]

Type=friend

Context=default

host=dynamic

username=3001

secret=3001

[3001] merupakan *context user*. Ini akan dipakai pada *extension.conf* untuk pengaturan nomor *VoIP*.

type=f riend merupakan salah satu tipe *client.*

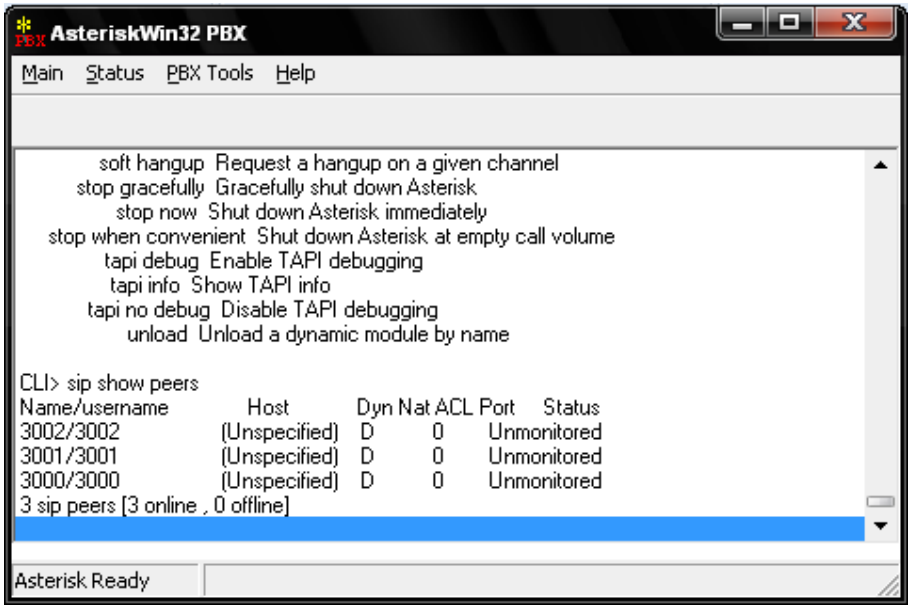
context=default merupakan *context* jaringan. Ini juga akan dipakai pada *extension.conf*, untuk merencanakan *dial plan*, dari masing-masing *context*, *default* menunjukan bahwa *context* yang digunakan adalah *context* standar. Konfigurasi tiap *context* bisa dibuat berbeda-beda. Karena pada perencanaan *dial plan* hanya akan ada satu tipe *dial plan*, maka cukup dibuat 1 *context* saja, yaitu *default*.

host=dynamic, menyatakan bahwa *user* dalam blok ini IP *Address-nya* boleh diubah-ubah.

*user*name=3001, *username* dari *context user* [3001].

secret=3001, *password* dari *context user* [3001].

Blok-blok *account* lainnya dibuat dengan aturan yang sama seperti di atas. Untuk memastikan *user* sip yang telah kita buat, dapat kita lihat dengan menggunakan perintah "*sip show peers*" pada CLI console .



Gambar IV.7 Daftar *user* SIP

4.1.2.2. Konfigurasi *Dial Plan*

Seperti yang telah dibicarakan sebelumnya, konfigurasi *dial plan* dilakukan pada file `extensions.conf` konfigurasi *dial plan*, memiliki format penulisan sebagai berikut :

Exten=> *extension*, *priority*, application

Penjelasannya sebagai berikut :

a. *extension* adalah nomor yang mewakili 1 *user* yang telah terdaftar
b. *priority* adalah nomor urut prioritas eksekusi aplikasi, yaitu menunjukan aplikasi apa yang harus dikerjakan terlebih dahulu
c. *application* aplikasi *dial plan* yang akan dieksekusi. Bentuk aplikasinya dapat dilihat pada bagian desain konfigurasi *server*.

Konfigurasi dial plan yang dilakukan adalah sebagai berikut :

```
exten => 3000,1,Dial(${ip3000},30,Ttm)
exten => 3000,2,Voicemail(u3000)
```

```
exten => 3000,3,Hangup
exten => 3000,102,Voicemail(b3000)
exten => 3000,103,Hangup
exten => 3001,1,Dial(${ip3001},30,Ttm)
exten => 3001,2,Voicemail(u3001)
exten => 3001,3,Hangup
exten => 3001,102,Voicemail(b3001)
exten => 3001,103,Hangup
exten => 3002,1,Dial(${ip3002},30,Ttm)
exten => 3002,2,Voicemail(u3002)
exten => 3002,3,Hangup
exten => 3002,102,Voicemail(b3002)
exten => 3002,103,Hangup
exten => 3003,1,Dial(${ip3003},30,Ttm)
exten => 3003,2,Voicemail(u3003)
exten => 3003,3,Hangup
exten => 3003,102,Voicemail(b3003)
exten => 3003,103,Hangup
```

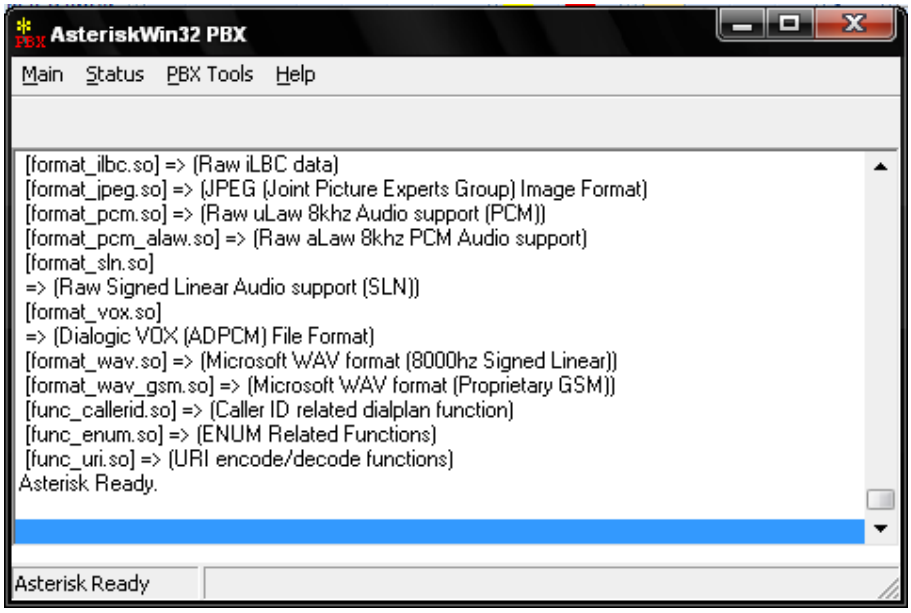
penjelasan dari konfigurasi diatas :

a. 3000, 3001, 3002, 3003 : merupakan nomor *extension*.
b. Dial(${ip3000}, 30,Ttm : artinya account yang bersangkutan menggunakan protocol SIP
c. Exten=> 3000,3,hangup : menyebabkan saat komunikasi berakhir.

Pada konfigurasi yang dilakukan, *account* yang dibuat sebanyak 4 buah account, yang bisa ditambah sewaktu-waktu dengan mudah.

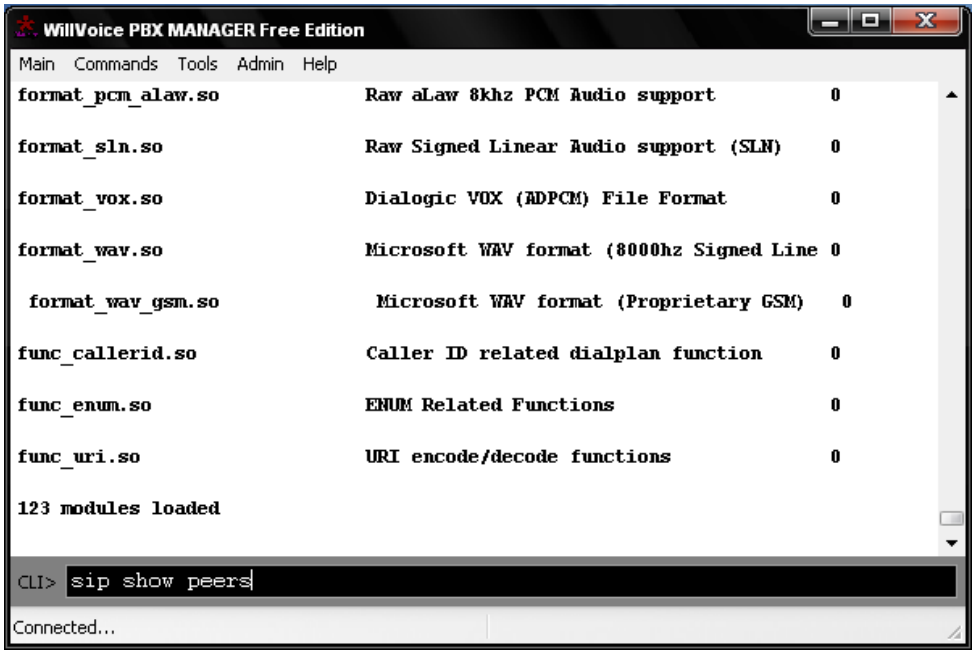
4.2 Pengujian *Asterisk*

Setelah diinstalasi *asterisk* akan membuat shortcut sendiri yang akan ditaruh di desktop komputer kita. Untuk menjalankan *asterisk* klik dua kali pada icon shortcut yang telah ada tersebut, tunggu beberapa saat sampai muncul tampilan seperti berikut :



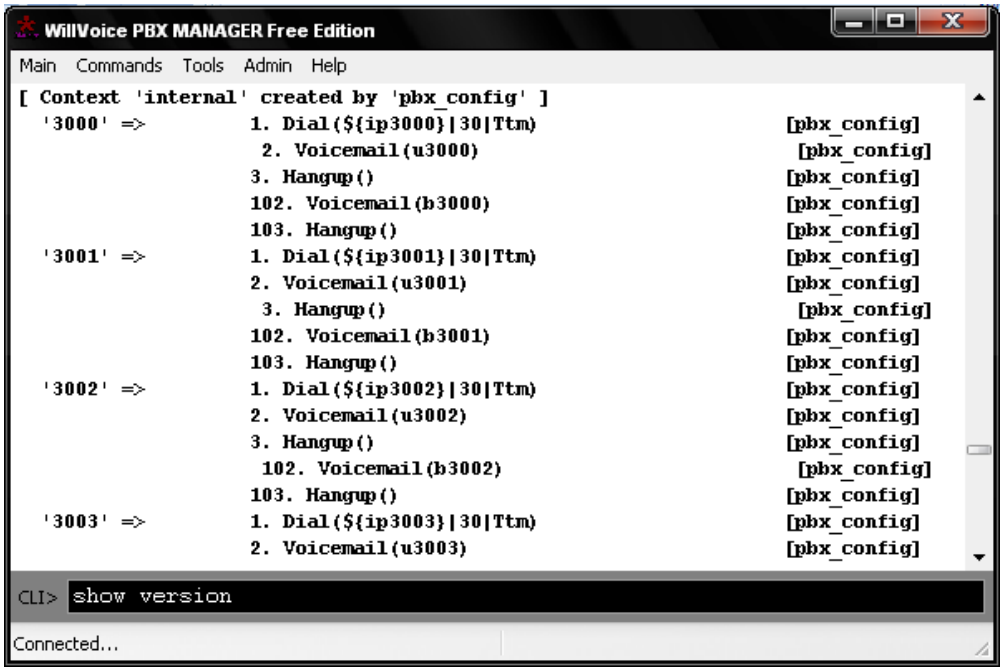
Gambar IV.8 Konsol utama *asterisk*

Konfigurasi *asterisk* dapat dilakukan menggunakan *PBX Manager*. Biasanya saat pertama kali dijalankan, *PBX Manager* tidak akan berhasil tersambung ke *Asterisk*. Melalui menu PBX Tools → start kita dapat menjalankan *asterisk*. Setelah *asterisk* dijalankan, barulah *PBX Manager* dapat tersambung ke *Asterisk* dan kita dapat mengkonfigurasi *Asterisk*.



Gambar IV.9 Konsol PBX Manager

Melalui menu *Commands* → *dial plan* kita dapat melihat konfigurasi detail dari *dial plan* yang terdapat pada *asterisk*win32.Bentuk keluarannya adalah sama dengan keluaran perintah " *show dialplan*" pada *command line interface* (CLI) *Asterisk.*



Gambar IV.10 Konfigurasi *dial plan*

4.3 Intalasi mikrotik

Untuk melakukan instalasi *Mikrotik Router*OS ke dalam PC ada hal yang perlu di persiapkan antara lain :

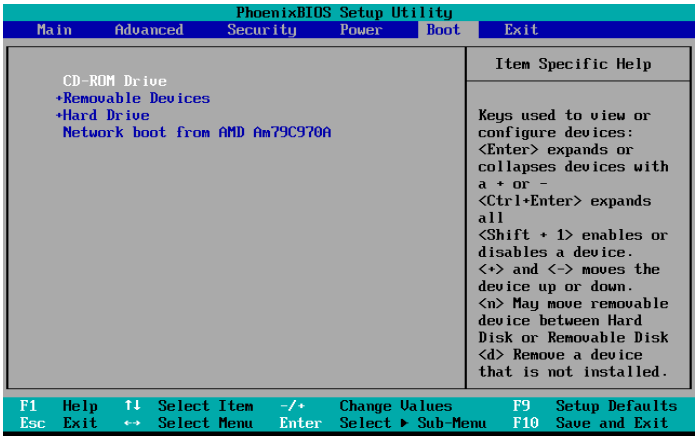
a. Menyiapkan 2 buah PC yang telah disediakan dengan 2 NIC, dengan spesifikasi seperti yang telah disebutkan di bab sebelummnya

b. Menyiapkan CD instaler Mikrotik yang bootable.

   Menyiapkan CD instaler Mikrotik *Router*OS yang bootable yang file isonya di download dari Internet dan membakarnya kedalam CD dengan aplikasi Nero.

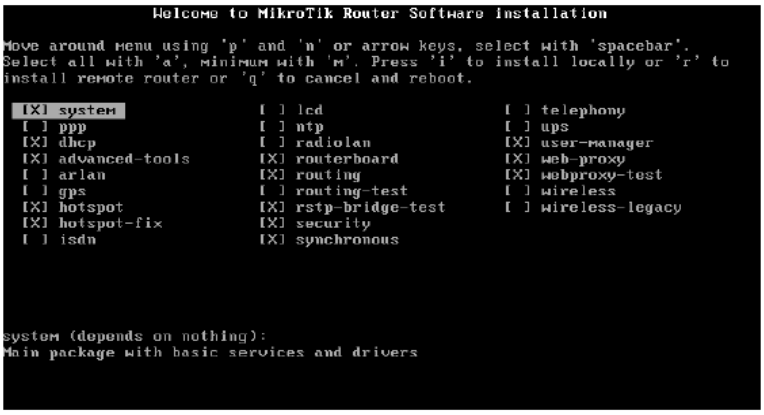
c. Konfigurasi BIOS pada PC.

   Mengatur konfigurasi BIOS agar boot sequence mengarah ke CD,



Gambar IV.11 Konfigurasi bios

setelah itu menyimpannya dengan menekan F10 lau tekan OK. System akan restart unutk melanjutkan proses instalasi sampai muncul layar seperti ni.



Gambar IV.12 Pemilihan paket-paket instalasi

d. Memilih paket-paket aplikasi dalam instalasi

Proses instalasi akan menampilkan pilihan modules apa saja yang akan kita install, adapun penjelasannya adalah sebagai berikut :

a) System : Merupakan paket utama dengan servis dasar dan juga driver –driver untuk periheral yang terpasang pada PC

router.

b) DHCP : Paket yang menjalankan DHCP client maupun DHCP *server*.

c) Advanced Tool : email client, pinger, netwach and utiliti lain.

d) Hotspot : Paket untuk hotspot.

e) Hotspot-fix : Paket perbaikan untuk hotspot versi 2.9.27.

f) *Router*board : Perlengkapan untuk *Router*Board

g) Routing : Paket yang mendukung RIP, OSPF dan BGP4.

h) Rtsp-bridge-test : Paket yang digunakan untuk mengetes RSTP bridge.

i) Security : Paket yang mendukung keamanan untuk IPSEC, SSH dan keamanan koneksi dengan WinBox.

j) Synchronous: Untuk synchronous dengan device lain.

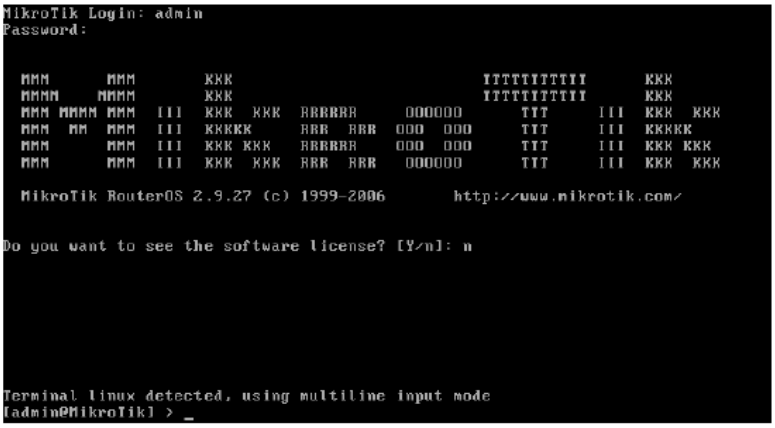
k) *User*-manager : Paket yang menjalankan majemen *user* pada *router*.

l) Web-Proxy : Paket untuk HTTP Web proxy.

m) Webproxy-test : Paket untuk mengetes HTTP Web proxy.

e. Menginstal paket-paket tersebut dengan menekan tombol I pada keyboard setelah itu tekan tombol “y”dan tekan tombol “y”lagi untuk melanjutkan instalasi.

f. Merestart komputer dengan menekan Enter.



Gambar IV.14 Halaman Login

Hasil instalasi Mikrotik *Router*OS dengan default *User* "admin" dan tanpa password .

4.3.1 Akses Mikrotik *Router*OS

Ada 4 cara pengaksesan Mikrotik *Router*OS, antara lain :

a. Via Console/Command Mikrotik

Jenis *router board* maupun PC bisa di akses langsung *via console/shell* maupun remote akses menggunakan PUTTY (www.putty.nl)

b. Via *Web Browser*

Mikrotik bisa diakses via *web/port* 80 pada *browser*. Contoh : ketik di *browser* dengan IP Address dari Mikrotik *Router*OS : 192.168.254.253

c. Via WinBox

Mikrotik bisa diakses/*remote* menggunakan tool winbox ,Winbox adalah sebuah *utility* untuk melakukan *remote* ke *server*

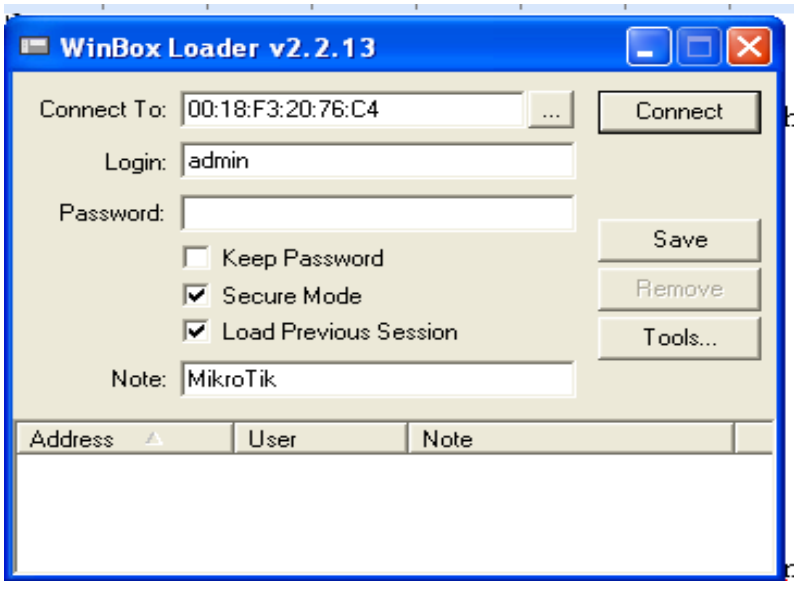
mikrotik dalam mode GUI.. Winbox bisa mendeteksi mikrotik yang sudah di install jika masih dalam satu *network*, yaitu dengan mendeteksi MAC address dari *ethernet* yang terpasang di Mikrotik *Router*OS.

d. Via Telnet

Mikrotik dapat di*remote* menggunakan *telnet* melalui program aplikasi "*command prompt*" (cmd) yang ada pada *windows*. Namun, penggunaan *telnet* tidak dianjurkan dalam jaringan karena masalah keamanannya. Contoh : c:\>telnet 192.168.254.253

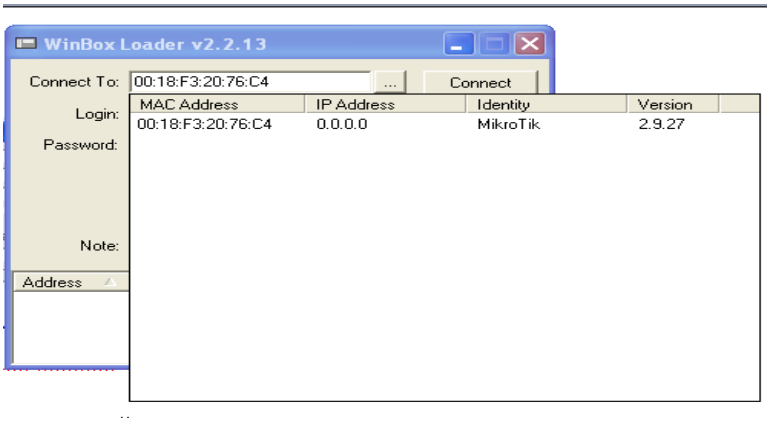
Dalam hal ini pengaksesan Mikrotik *Router*OS akan menggunakan WinBox karena praktis dan mudah digunakan, adapaun cara pengaksesan Mikrotik *Router*OS melalui Winbox adalah sebagai berikut :

a) Buka aplikasi WinBox



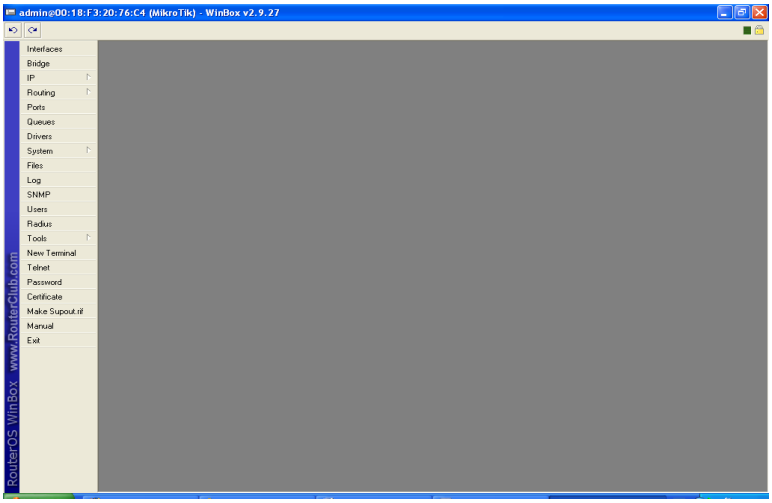
Gambar IV.15 Menu utama winbox

b) Klik tombol … untuk mencari Mikrotik *Router*OS



Gambar IV.16 Hasil pencarian mikrotik

c) Klik *MAC Address* yang tampil dan klik *connect* untuk koneksi ke Mikrotik *Router*OS.Winbox akan melakukan koneksi ke Mikrotik



Gambar IV.17 Tampilan awal mikrotik *router*OS pada winbox

Selanjutnya Konfigurasi akan dilakukan melalui WinBox.

4.4 Pengujian sistem *VoIP*

Pada penelitian ini akan dilakukan beberapa skenario ujicoba dengan tujuan untuk mendapatkan beberapa perbandingan data hasil pengukuran, adapun

skenario ujicoba yang akan dilakukan adalah sebagai berikut :

a. Pengujian sistem dengan *bandwidth* internet 512 Kbps.

   Sistem akan diujicoba pada koneksi *backbone* 512 kbps , lalu akan dibandingkan bagaimana performansi jaringan sebelum dan sesudah diterapkan *priority queuing*.

b. Pengujian sistem dengan *bandwidth* internet 256 Kbps.

   Sistem akan diujicoba pada koneksi *backbone* 256 kbps , lalu akan dibandingkan bagaimana performansi jaringan sebelum dan sesudah diterapkan *priority queuing*.

c. Pengujian sistem dengan *bandwidth* internet 128 Kbps.

   Sistem akan diujicoba pada koneksi *backbone* 256 kbps , lalu akan dibandingkan bagaimana performansi jaringan sebelum dan sesudah diterapkan *priority queuing*.

Analisa yang dilakukan meliputi pengukuran parameter-parameter berikut :

1. Paket *loss*

   Sistem akan diuji dan diamati dengan tujuan untuk mengetahui paket *loss* yang ada dengan mengacu pada refernsi sebagai berikut :

   a. Baik (0%-0,5%)
   b. Cukup (0,5%-1,5%)
   c. Buruk (>1 – 5%)

2. *Jitter*

   Sistem akan diuji dan diamati dengan tujuan untuk mengetahui *jitter*

yang ada dengan mengacu pada referensi sebagai berikut :

a. Baik (0 ms – 20 ms)

b. Cukup (20 ms – 50 ms)

c. Buruk (>50 ms)

3. Kebutuhan *bandwidth*

4. Kualitas dari suara yang diukur dalam satuan MOS

   Untuk pengukuran mos ini akan diukur nilai mos dengan referensi sebagai berikut [1]:

Tabel IV.1 Standarisasi MOS

MOS	Kualitas
5	Sangat bagus
4	Bagus
3	Cukup
2	Buruk
1	Sangat buruk

Pengukuran performansi dilakukan dengan mengubah parameter *bandwith*. Untuk mewakili kondisi jaringan secara *real* maka sistem akan dibebani oleh trafik data dari sebuah IP camera yang akan mengirimkan gambar jpeg secara terus menerus dan pengamatan data dilakukan saat mulai melakukan koneksi sampai dengan koneksi berakhir dengan mengambil sample paket sebanyak 20000 paket RTP yang nantinya akan digunakan untuk keperluan analisa

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

4.5 Pengukuran Pada Ujicoba Tanpa Penerapan *Priority Queueing*

Setelah dilakukan pengukuran pada koneksi *VoIP* yang terjadi antara SIP client 1 yang memiliki IP. 192.168.1.100 dengan SIP client 2 yang memiliki IP. 192.168.2.100 didapatkan hasil analisa dan pembahasannya sebagai berikut :

4.5.1. Pengamatan Paket *Loss* Pada Ujicoba Koneksi 512 Kbps

Dari pengukuran paket *loss* pada koneksi dengan konfigurasi limitasi *bandwith* 512 kbps sebelum diterapkan *priority queuing* berdasarkan analisa data dari *commview* didapatkan statistik :

Tabel IV.2 Jumlah Paket *Loss* pada ujicoba koneksi 512 kbps tanpa *QoS*

Ip sumber	Port asal	Ip tujuan	Port tujuan	Paket dikirim	Paket *loss*	Prosentase %
192.168.1.100	8000	192.168.1.2	12080	12279	1	0 %
192.168.2.100	8000	192.168.1.2	18468	11854	422	3,4 %
192.168.1.2	18468	192.168.2.100	8000	6140	1	0 %
192.168.1.2	12080	192.168.1.100	8000	5930	3	0,1 %

Berdasarkan tabel IV.2 diatas dapat diketahui bahwa pada saat ujicoba koneksi dengan limitasi *bandwith* 512 kbps antara client 1 dan *VoIP server* yang berfungsi sebagai *VoIP* gateway hampir tidak ada paket yang hilang. Hal ini disebabkan karena kedua *node* ini masih berada dalam satu jaringan lokal sehingga komunikasi suara dapat berlangsung dengan sangat baik tanpa ada suara yang hilang. Saat komunikasi berlangsung standar kompresi suara yang dipakai adalah G.711. Pada client 1 saat mengirim informasi suara menggunkan port 8000 yang ditujukan ke *VoIP server* pada port 12080. Total paket yang dikirim sebanyak 12279 paket sedangakan paket yang hilang hanya 1 paket saja.

Pada client 2 yang sudah terhubung ke luar jaringan dengan koneksi *bandwith* yang digunakan antar kedua jaringan sebesar 512 kbps, client 2

mengirim data pada port 8000 yang ditujukan ke *VoIP server* pada port 18468.Pada proses komunikasi ini terapat paket *loss* yang cukup tinggi yaitu sebesar 3,4 %. Hal ini terjadi karena paket data *VoIP* berebut jatah *bandwith* yang ada dengan paket data lainnya. Dalam kasus ini paket data yang berupa *VoIP* tersebut berebut data dengan paket http karena sistem dibebani oleh perangkat ip camera dengan ip 192.168.2.253 yang dibrowsing oleh salah satu pc di *network* 192.168.1 . Ip camera tersebut membebani jaringan dengan trafik data yang cukup tinggi sehingga paket data yang berupa *VoIP* terganggu proses pengirimannya.

4.5.2. Pengamatan *Jitter* Pada Ujicoba Koneksi 512 Kbps

Dari adanya koneksi komunikasi pada SIP client 1 dan SIP client 2 didapatkan statistik *jitter* pada saat mentransmisikan paket data selama terjadinya komunikasi, sebagai berikut :

Tabel IV.3 *jitter* pada ujicoba koeneksi 512 kbps tanpa priority queuing

Ip sumber	Port asal	Ip tujuan	Port tujuan	Paket dikirim	*Jitter* (ms)
192.168.1.100	8000	192.168.1.2	12080	12279	2,57
192.168.2.100	8000	192.168.1.2	18468	11854	17,5
192.168.1.2	18468	192.168.2.100	8000	6140	1,91
192.168.1.2	12080	192.168.1.100	8000	5930	20,29
Rata – rata *jitter*					10.7

Pada tabel IV.3 bisa diketahui dari seluruh total pengiriman paket pada ujicoba komunikasi *VoIP* dengan koneksi *backbone* 512 kbps, terdapat *jitter* yang berasal dari 4 port transmisi yang bervariasi dengan rata-rata *jitter* sebesar 10,7 ms. Dimana pada client 1 mengalami *jitter* 2,57 untuk transmisi ke *server*. Dari *server* ke client 1 terdapat *jitter* sebesar 20,29 ms. sedangkan pada client 2 mengalami *jitter* 17,5 ms untuk transmisi ke *server*. Dari *server* ke client 2 terdapat *jitter* sebesar 1,91.

4.5.3. Pengukuran *Bandwith* Pada Ujicoba Koneksi 512 Kbps

Pengukuran *bandwidth* pada ujicoba dengan *bandwith* interkoneksi sebesar 512 kbps, dilakukan dengan cara melakukan proses komunikasi suara antara kedua SIP client, kemudian didapatkan paket-paket informasi yang dikirim dan dari total paket yang dikirim dengan menggunakan commview dapat diketahui berapa besar kapasitas dari masing-masing paket, setelah itu nilai rata-rata tiap pengiriman bisa diketahui besarnya *bandwidth* yang dibutuhkan. Pengukuran *bandwidth* ini dilakukan sedemikian rupa agar tidak terjadi perpindahan data yang cukup besar antara kedua *client* tersebut sehingga aplikasi-aplikasi yang tidak diamati dalam pengukuran tersebut tidak diaktifkan misalnya : ftp (*file transfer protocol*) untuk pengiriman data, http (*hypertext transfer protocol*) untuk browsing dan lain sebagainya, karena pada penelitian in penulis hanya melakukan pengukuran untuk paket RTP saja. Berikut ini adalah besarnya *bandwidth* yang dibutuhkan saat terjadi ujicoba kominikasi dengan *bandwith* 512 kbps tanpa menggunakan *QoS priority* queuing.

Tabel IV.4 Pengukuran b*andwidth* pada ujicoba dengan koneksi 512 kbps

Ip asal	port asal	Ip tujuan	Port tujuan	Paket dikirim	*bandwith*	*Packet loss*
192.168.1.100	8000	192.168.1.2	12080	12279	104.73	1
192.168.2.100	8000	192.168.1.2	18468	11854	101.12	422
192.168.1.2	18468	192.168.2.100	8000	6140	83.65	1
192.168.1.2	12080	192.168.1.100	8000	5930	80.79	3
TOTAL				36203	370,79	427
RATA-RATA				9050,75	92,57	106,75

Dari tabel IV.3 bisa diketahui untuk melakukan proses komunikasi, masing – masing client akan mengirim paket ke *VoIP server* untuk kemudian oleh *VoIP server* paket tersebut diteruskan ke client yang dituju. *bandwidth* yang dibutuhkan saat melakukan komunikasi antara SIP client 1 ke *server* sebesar 104,73 kbps. Sedangkan dari SIP client 2 ke *server bandwith* yang dibutuhkan sebesar 101,12 kbps. *VOIP server* akan meneruskan paket yagn diterima dari sip client 2 ke sip client 1 sebesar 80,79 kbps, sedangkan untuk dari sip client 2 ke sip client 1 sbesar 83,65 kbps. Dari data diatas dapat kita lihat *bandwith* rata-rata yang dibutuhkan untuk satu kanal komunikasi *VoIP* adalah sebesar 92,57 kbps yang didapat dari rata-rata paket yang dikirim dan diterima oleh sip client.

4.5.4. Pengukuran MOS Pada Ujicoba Koneksi 512 Kbps

Kualitas suara pada saat ujicoba dengan *bandwith backbone* 512 kbps pada client 1 dan client 2, berdasarkan data dari commview didapatkan hasil penilaian kualitas suara atau MOS sebagai berikut :

Tabel IV.5 Pengukuran *MOS* pada ujicoba dengan koneksi 512 kbps

Ip asal	port asal	Ip tujuan	Port tujuan	Paket dikirim	*Packet loss*	MOS score
192.168.1.100	8000	192.168.1.2	12080	12279	1	4,4
192.168.2.100	8000	192.168.1.2	18468	11854	422	3,5
192.168.1.2	18468	192.168.2.100	8000	6140	1	4,4
192.168.1.2	12080	192.168.1.100	8000	5930	3	4,4
RATA-RATA				10654,6	106,75	4,2

Dari tabel IV.5 diatas dapat diketahui bahwa kualitas suara dan gambar yang dihasilkan saat terjadi koneksi komunikasi antara client 1 dan client 2 rata-rata nilai MOS score yang didapatkan adalah sebesar 4,2. Sehingga dapat diambil kesiimpulan MOS dari nilai tersebut dapat dikatakan kualitas suara yang dihasilkan adalah “Sangat Baik” untuk diaplikasikan.

4.5.5. Pengamatan Paket *Loss* Pada Ujicoba Koneksi 256 Kbps

Dari pengukuran paket *loss* pada koneksi dengan konfigurasi limitasi *bandwith* 256 kbps sebelum diterapkan *priority* queuing berdasarkan analisa data dari commview didapatkan statistik :

Tabel IV.6 Jumlah Paket *Loss* pada ujicoba koneksi 256 kbps tanpa *QoS*

Ip sumber	Port asal	Ip tujuan	Port tujuan	Paket dikirim	Paket *loss*	Prosentase %
192.168.1.100	8000	192.168.1.2	12080	12822	1	0 %
192.168.2.100	8000	192.168.1.2	18468	12828	7354	57,3 %
192.168.1.1	18468	192.168.2.100	8000	6412	0	0 %
192.168.1.2	12080	192.168.1.100	8000	2746	1	0,1 %

Berdasarkan tabel IV.6 diatas dapat diketahui bahwa pada saat ujicoba koneksi dengan limitasi *bandwith* 256 kbps antara client 1 dan *VoIP server* yang berfungsi sebagai *VoIP* gateway hampir tidak ada paket yang hilang. Hal ini disebabkan karena kedua *node* ini masih berada dalam satu jaringan lokal sehingga komunikasi suara dapat berlangsung dengan sangat baik tanpa ada suara yang hilang. Pada client 1 saat mengirim informasi suara menggunkan port 8000 yang ditujukan ke *VoIP server* pada port 12080. Total paket yang dikirim sebanyak 12822 paket sedangkan paket yang hilang hanya 1 paket saja.

Pada client 2 yang sudah terhubung ke luar jaringan dengan koneksi *bandwith* yang digunakan antar kedua jaringan sebesar 256 kbps, client 2 mengirim data pada port 8000 yang ditujukan ke *VoIP server* pada port 18468.Pada proses komunikasi ini terdapat paket *loss* yang sangat banyak yaitu sebanyak 7354 paket atau sebesar 57,3 %. Hal ini terjadi karena paket data *VoIP* berebut jatah *bandwith* yang ada dengan paket data lainnya yagnberupa trafik http dari ipcamera dalam sambungan *backbone bandwith* sebesar 256 kbps. Sehingga sambungan *backbone* tersebut tidak kuat menahan aliran arus trafik paket data yang ada. Yang berimbas langsung terhadap paket-paket yang dikirim dan

diterima oleh SIP client 2.

4.5.6. Pengukuran *Jitter* Pada Ujicoba Koneksi 256 Kbps

Dari adanya koneksi komunikasi pada SIP client 1 dan SIP client 2 didapatkan statistik *jitter* pada saat mentransmisikan paket data selama terjadinya komunikasi, sebagai berikut :

Tabel IV.7 *Jitter* pada ujicoba koneksi 256 kbps tanpa *priority queuing*

Ip sumber	Port asal	Ip tujuan	Port tujuan	Paket dikirim	*Jitter* (ms)
192.168.1.100	8000	192.168.1.2	12080	12822	1,92
192.168.2.100	8000	192.168.1.2	18468	12828	53,5
192.168.1.2	18468	192.168.2.100	8000	6412	1,84
192.168.1.2	12080	192.168.1.100	8000	2746	121,1
Rata – rata *jitter*					44,6

Pada tabel IV.7 bisa diketahui dari seluruh total pengiriman paket pada ujicoba komunikasi *VoIP* dengan koneksi *backbone* 256 kbps, terdapat *jitter* yang berasal dari 4 port transmisi yang bervariasi dengan rata-rata *jitter* sebesar 44,6 ms. Dimana pada client 1 mengalami *jitter* 1,92 untuk transmisi ke *server*. Dari *server* ke client 1 terdapat *jitter* sebesar 121,1 ms. sedangkan pada client 2 mengalami *jitter* 53,5 ms untuk transmisi ke *server*. Dari *server* ke client 2 terdapat *jitter* sebesar 1,84. Berdasarkan tabel referensi yang ada di atas maka dapat diambil kesimpulan *jitter* yang dihasilkan termasuk dalam kategori buruk.

4.5.7. Pengukuran *Bandwith* Pada Ujicoba Koneksi 256 Kbps

Pengukuran *bandwidth* pada ujicoba dengan *bandwith* interkoneksi sebesar 256 kbps, dilakukan dengan cara melakukan proses komunikasi suara antara kedua SIP client, kemudian didapatkan paket-paket informasi yang dikirim dan dari total paket yang dikirim dengan menggunakan commview dapat

diketahui berapa besar kapasitas dari masing-masing paket. Berikut ini adalah besarnya *bandwidth* yang dibutuhkan saat terjadi ujicoba kominikasi dengan *bandwith* 256 kbps tanpa menggunakan *QoS priority queuing*.

Tabel IV.8 Pengukuran b*andwidth* pada ujicoba dengan koneksi 256 kbps

Ip asal	port asal	Ip tujuan	Port tujuan	Paket dikirim	*bandwith*	*Packet loss*
192.168.1.100	8000	192.168.1.2	12080	12279	104.73	1
192.168.2.100	8000	192.168.1.2	18468	11854	101.12	7354
192.168.1.2	18468	192.168.2.100	8000	6140	83.65	0
192.168.1.2	12080	192.168.1.100	8000	5930	80.79	1
TOTAL				36203	370,79	7356
RATA-RATA				9050,75	92,57	1839

Dari tabel 4.8 bisa diketahui untuk melakukan proses komunikasi, masing –masing client akan mengirim paket ke *VoIP server* untuk kemudian oleh *VoIP server* paket tersebut diteruskan ke client yang dituju. *bandwidth* yang dibutuhkan saat melakukan komunikasi antara SIP client 1 ke *server* sebesar 104,73 kbps. Sedangkan dari SIP client 2 ke *server bandwith* yang dibutuhkan sebesar 101,12 kbps. *VOIP server* akan meneruskan paket yagn diterima dari sip client 2 ke sip client 1 sebesar 80,79 kbps, sedangkan untuk dari sip client 2 ke sip client 1 sbesar 83,65 kbps. Dari data diatas dapat kita lihat *bandwith* rata-rata yang dibutuhkan untuk satu kanal komunikasi *VoIP* adalah sebesar 92,57 kbps yang didapat dari rata-rata paket yang dikirim dan diterima oleh sip client.

4.5.8. Pengukuran MOS Score Pada Ujicoba Koneksi 256 Kbps

Kualitas suara pada saat ujicoba dengan *bandwith backbone* 256 kbps pada client 1 dan client 2, berdasarkan data dari commview didapatkan hasil penilaian kualitas suara atau MOS sebagai berikut :

Tabel IV.9 Pengukuran *MOS* pada ujicoba dengan koneksi 256 kbps

Ip asal	port asal	Ip tujuan	Port tujuan	Paket dikirim	*Packet loss*	MOS score
192.168.1.100	8000	192.168.1.2	12080	12279	1	4,4
192.168.2.100	8000	192.168.1.2	18468	11854	7354	1,1
192.168.1.2	18468	192.168.2.100	8000	6140	0	4,4
192.168.1.2	12080	192.168.1.100	8000	5930	1	3,2
RATA-RATA				10654,6	106,75	3,3

Dari tabel 4.9 diatas dapat diketahui bahwa kualitas suara dan gambar yang dihasilkan saat terjadi koneksi komunikasi antara client 1 dan client 2 rata-rata nilai MOS score yang didapatkan adalah sebesar 3,3. Sehingga dapat diambil kesiimpulan MOS dari nilai tersebut dapat dikatakan kualitas suara yang dihasilkan adalah “Cukup” untuk diaplikasikan.

4.5.9. Pengamatan Paket *Loss* Pada Ujicoba Koneksi 128 Kbps

Dari pengukuran paket *loss* pada koneksi dengan konfigurasi limitasi *bandwith* 128 kbps sebelum diterapkan *priority* queuing berdasarkan analisa data dari commview didapatkan statistik :

Tabel IV.10 Jumlah Paket *Loss* pada ujicoba koneksi 128 kbps tanpa *QoS*

Ip sumber	Port asal	Ip tujuan	Port tujuan	Paket dikirim	Paket *loss*	Prosentase %
192.168.1.100	8000	192.168.1.2	12080	12370	6222	50 %
192.168.2.100	8000	192.168.1.2	18468	12343	0	0 %
192.168.1.2	18468	192.168.2.100	8000	6172	1	0 %
192.168.1.2	12080	192.168.1.100	8000	3077	2	0,1 %

Berdasarkan tabel IV.10 diatas dapat diketahui bahwa pada saat ujicoba koneksi dengan limitasi *bandwith* 128 kbps antara client 1 dan *VoIP server* yang

berfungsi sebagai *VoIP* gateway terdapat banyak paket yang hilang.. Pada client 1 saat mengirim informasi suara menggunakan port 8000 yang ditujukan ke *VoIP server* pada port 12080. Total paket yang dikirim sebanyak 12080 paket sedangkan paket yang hilang sebanyak 6222 paket atau sebesar 50% dari total paket yang dikirim.

Pada client 2 yang sudah terhubung ke luar jaringan dengan koneksi *bandwith* yang digunakan antar kedua jaringan sebesar 128 kbps, client 2 mengirim data pada port 8000 yang ditujukan ke *VoIP server* pada port 18468. Pada proses komunikasi ini tidak terdapat paket *loss*. Hal ini kemungkinan terjadi karena paket data *VoIP* berebut jatah *bandwith* yang ada dengan paket data lainnya. Dalam kasus ini paket data yang berupa *VoIP* tersebut berebut data dengan paket http karena sistem dibebani oleh perangkat ip camera yang dibrowsing oleh salah satu pc di *network* 192.168.1 .

4.5.10. Pengukuran *Jitter* Pada Ujicoba Koneksi 128 Kbps

Dari adanya koneksi komunikasi pada SIP client 1 dan SIP client 2 didapatkan statistik *jitter* pada saat mentransmisikan paket data selama terjadinya komunikasi, sebagai berikut :

Tabel IV.11 *Jitter* pada ujicoba koneksi 128 kbps tanpa *priority queuing*

Ip sumber	Port asal	Ip tujuan	Port tujuan	Paket dikirim	*Jitter* (ms)
192.168.1.100	8000	192.168.1.2	12080	12279	117,5
192.168.2.100	8000	192.168.1.2	18468	11854	2,86
192.168.1.2	18468	192.168.2.100	8000	6140	2,41
192.168.1.2	12080	192.168.1.100	8000	5930	239,37
Rata – rata *jitter*					90,5

Pada tabel IV.11 bisa diketahui dari seluruh total pengiriman paket pada ujicoba komunikasi *VoIP* dengan koneksi *backbone* 128 kbps, terdapat *jitter* yang

berasal dari 4 port transmisi yang bervariasi dengan rata-rata *jitter* sebesar 90,5 ms. Dimana pada client 1 mengalami *jitter* 117,5 untuk transmisi ke *server*. Dari *server* ke client 1 terdapat *jitter* sebesar 239,4 ms. sedangkan pada client 2 mengalami *jitter* 2,86 ms untuk transmisi ke *server*. Dari *server* ke client 2 terdapat *jitter* sebesar 2,41. Berdasarkan tabel referensi yang ada di atas maka dapat diambil kesimpulan *jitter* yang dihasilkan termasuk sangat buruk sekali dan tidak direkomendasikan untuk diaplikasikan dalam sistem *VoIP*.

4.5.11. Pengukuran *Bandwith* Pada Ujicoba Koneksi 128 Kbps

Pengukuran *bandwidth* pada ujicoba dengan *bandwith* interkoneksi sebesar 128 kbps, dilakukan dengan cara melakukan proses komunikasi suara antara kedua SIP client. Berikut ini adalah besarnya *bandwidth* yang dibutuhkan saat terjadi ujicoba kominikasi dengan *bandwith* 128 kbps tanpa menggunakan *QoS priority* queuing.

Tabel IV.12 Pengukuran *Bandwidth* pada ujicoba dengan koneksi 128 kbps

Ip asal	port asal	Ip tujuan	Port tujuan	Paket dikirim	*bandwith*	*Packet loss*
192.168.1.100	8000	192.168.1.2	12080	12279	52,24	6222
192.168.2.100	8000	192.168.1.2	18468	11854	104,74	0
192.168.1.2	18468	192.168.2.100	8000	6140	83.63	1
192.168.1.2	12080	192.168.1.100	8000	5930	41,73	2
TOTAL				36203	282,34	6225
RATA-RATA				9050,75	70,3	1556,3

Dari tabel IV.12 bisa diketahui untuk melakukan proses komunikasi, masing –masing client akan mengirim paket ke *VoIP server* untuk kemudian oleh *VoIP server* paket tersebut diteruskan ke client yang dituju. *bandwidth* yang dibutuhkan saat melakukan komunikasi antara SIP client 1 ke *server* sebesar 52,24 kbps. Sedangkan dari SIP client 2 ke *server bandwith* yang dibutuhkan sebesar 104,74 kbps. *VOIP server* akan meneruskan paket yang diterima dari sip

client 2 ke sip client 1 sebesar 41,73 kbps, sedangkan untuk dari sip client 2 ke sip client 1 sbesar 83,63 kbps. Dari data diatas dapat kita lihat *bandwith* rata-rata yang dibutuhkan untuk satu kanal komunikasi *VoIP* adalah sebesar 70,3 kbps yang didapat dari rata-rata paket yang dikirim dan diterima oleh sip client.

4.5.12. Pengukuran MOS Score Pada Ujicoba Koneksi 128 Kbps

Kualitas suara pada saat ujicoba dengan *bandwith backbone* 128 kbps pada client 1 dan client 2, berdasarkan data dari commview didapatkan hasil penilaian kualitas suara atau MOS sebagai berikut :

Tabel IV.13 Pengukuran *MOS* pada ujicoba dengan koneksi 128 kbps

Ip asal	port asal	Ip tujuan	Port tujuan	Paket dikirim	*Packet loss*	MOS score
192.168.1.100	8000	192.168.1.2	12080	12279	1	1,1
192.168.2.100	8000	192.168.1.2	18468	11854	422	4,4
192.168.1.2	18468	192.168.2.100	8000	6140	1	4,4
192.168.1.2	12080	192.168.1.100	8000	5930	3	2,7
RATA-RATA				10654,6	106,75	3,15

Dari tabel IV.13 diatas dapat diketahui bahwa kualitas suara dan gambar yang dihasilkan saat terjadi koneksi komunikasi antara client 1 dan client 2 rata-rata nilai MOS score yang didapatkan adalah sebesar 3,15. Sehingga dapat diambil kesiimpulan MOS dari nilai tersebut dapat dikatakan kualitas suara yang dihasilkan adalah “buruk” untuk diaplikasikan.

4.6 Pengukuran Pada Ujicoba Dengan Menerapkan *Priority* Queueing

Setelah dilakukan pengukuran pada koneksi *VoIP* yang terjadi antara SIP client 1 yang memiliki IP. 192.168.1.100 dengan SIP client 2 yang memiliki IP. 192.168.2.100 dengan diterapkan *priority* queueing yang diset sebagai berikut :

a. Trafik UDP untuk RTP port range antara 8000 samapi dengan 20000, akan mendapatkan jatah 50 % dari *bandwith* yang ada.

b. Trafik HTTP mendapatkan 30% dari *bandwith*.

c. Trafik yang lainnya mendapatkan 20% dari *bandwith* .

Dari hasil pengujian didapatkan hasil analisa dan pembahasannya sebagai berikut:

4.6.1. Pengamatan Paket *Loss* Pada Ujicoba Koneksi 512 Kbps

Dari pengukuran paket *loss* pada koneksi dengan konfigurasi limitasi *bandwith* 512 kbps setelah diterapkan *priority* queuing berdasarkan analisa data dari commview didapatkan statistik :

Tabel IV.14 Jumlah Paket *Loss* pada ujicoba koneksi 512 kbps dengan *QoS*

Ip sumber	Port asal	Ip tujuan	Port tujuan	Paket dikirim	Paket *loss*	Prosentase %
192.168.1.100	8000	192.168.1.2	19894	12547	0	0 %
192.168.2.100	8000	192.168.1.2	14962	12645	0	0%
192.168.1.2	19894	192.168.2.100	8000	6357	0	0 %
192.168.1.2	14962	192.168.1.100	8000	6240	0	0 %

Berdasarkan tabel 4.14 diatas dapat diketahui bahwa pada saat ujicoba koneksi dengan limitasi *bandwith* 512 kbps antara client 1 dan *VoIP server* yang berfungsi sebagai *VoIP* gateway tidak ada paket yang hilang. Pada client 1 saat mengirim informasi suara menggunkan port 8000 yang ditujukan ke *VoIP server* pada port 19894. Total paket yang dikirim sebanyak 12547 paket dantidak ada satu pun paket yang hilang.

Pada client 2 yang sudah terhubung ke luar jaringan dengan koneksi *bandwith* yang digunakan antar kedua jaringan sebesar 512 kbps, client 2

mengirim data pada port 8000 yang ditujukan ke *VoIP server* pada port 14962.Pada proses komunikasi ini tidak terdapat paket *loss* juga. Walaupun sistem telah dibebani oleh trafik dari ip camera tetapi karena trafik yang berupa rtp lebih diprioritaskan daripada trafik yang lainnya, maka trafik yang lain akan mengalami penyesuaian dengan jatah *bandwith* yang dimilikinya.

4.6.2. Pengukuran *Jitter* Pada Ujicoba Koneksi 512 Kbps

Dari adanya koneksi komunikasi pada SIP client 1 dan SIP client 2 didapatkan statistik *jitter* pada saat mentransmisikan paket data selama terjadinya komunikasi, sebagai berikut :

Tabel IV.15 *jitter* pada ujicoba koeneksi 512 kbps dengan *priority* queuing

Ip sumber	Port asal	Ip tujuan	Port tujuan	Paket dikirim	*Jitter* (ms)
192.168.1.100	8000	192.168.1.2	19894	12547	3,05
192.168.2.100	8000	192.168.1.2	14962	12645	17,5
192.168.1.2	19894	192.168.2.100	8000	6357	1,91
192.168.1.2	14962	192.168.1.100	8000	6240	20,29
Rata – rata *jitter*					10.7

Pada tabel IV.15 bisa diketahui dari seluruh total pengiriman paket pada ujicoba komunikasi *VoIP* dengan koneksi *backbone* 512 kbps, terdapat *jitter* yang berasal dari 4 port transmisi yang bervariasi dengan rata-rata *jitter* sebesar 10,7 ms. Dimana pada client 1 mengalami *jitter* 3,05 untuk transmisi ke *server*. Dari *server* ke client 1 terdapat *jitter* sebesar 20,29 ms. sedangkan pada client 2 mengalami *jitter* 17,5 ms untuk transmisi ke *server*. Dari *server* ke client 2 terdapat *jitter* sebesar 1,91. Berdasarkan tabel referensi yang ada di atas maka dapat diambil kesimpulan *jitter* yang dihasilkan termasuk dalam kategori bagus

4.6.3. Pengukuran *Bandwith* Pada Ujicoba Koneksi 512 Kbps Dengan *QoS*

Pengukuran *bandwidth* pada ujicoba dengan *bandwith* interkoneksi sebesar 512 kbps dengan penerapan *priority* queuing dari hasil pengujian didapatkan statistik sebagai berikut :

Tabel IV.16 Pengukuran *Bandwidth* pada ujicoba dengan koneksi 512 kbps

Ip asal	port asal	Ip tujuan	Port tujuan	Paket dikirim	*bandwith*	*Packet loss*
192.168.1.100	8000	192.168.1.2	19894	12547	104.73	0
192.168.2.100	8000	192.168.1.2	14962	12645	101.12	0
192.168.1.2	14962	192.168.2.100	8000	6357	83.65	0
192.168.1.2	12080	192.168.1.100	8000	6240	80.79	0
TOTAL				37789	370,29	0
RATA-RATA				9447,25	92,57	0

Dari tabel IV.16 bisa diketahui untuk melakukan proses komunikasi, masing –masing client akan mengirim paket ke *VoIP server* untuk kemudian oleh *VoIP server* paket tersebut diteruskan ke client yang dituju. *bandwidth* yang dibutuhkan saat melakukan komunikasi antara SIP client 1 ke *server* sebesar 104,73 kbps. Sedangkan dari SIP client 2 ke *server bandwith* yang dibutuhkan sebesar 101,12 kbps. *VOIP server* akan meneruskan paket yagn diterima dari sip client 2 ke sip client 1 sebesar 80,79 kbps, sedangkan untuk dari sip client 2 ke sip client 1 sbesar 83,65 kbps. Dari data diatas dapat kita lihat *bandwith* rata-rata yang dibutuhkan untuk satu kanal komunikasi *VoIP* adalah sebesar 92,57 kbps yang didapat dari rata-rata paket yang dikirim dan diterima oleh sip client.

4.6.4. Pengukuran MOS Score Pada Ujicoba Koneksi 512 Kbps

Kualitas suara pada saat ujicoba dengan *bandwith backbone* 512 kbps pada client 1 dan client 2, berdasarkan data dari commview didapatkan hasil penilaian kualitas suara atau MOS sebagai berikut :

Tabel IV.17 Pengukuran *MOS* pada ujicoba dengan koneksi 512 kbps

Ip asal	port asal	Ip tujuan	Port tujuan	Paket dikirim	*Packet loss*	MOS score
192.168.1.100	8000	192.168.1.2	19894	12547	0	4,4
192.168.2.100	8000	192.168.1.2	14962	12645	0	4,4
192.168.1.2	14962	192.168.2.100	8000	6357	0	4,4
192.168.1.2	12080	192.168.1.100	8000	6240	0	4,4
RATA-RATA				9447,25	0	4,2

Dari tabel IV.17 diatas dapat diketahui bahwa kualitas suara dan gambar yang dihasilkan saat terjadi koneksi komunikasi antara client 1 dan client 2 rata-rata nilai MOS score yang didapatkan adalah sebesar 4,4. Sehingga dapat diambil kesiimpulan MOS dari nilai tersebut dapat dikatakan kualitas suara yang dihasilkan adalah “Sangat Baik” untuk diaplikasikan.

4.6.5. Pengamatan Paket *Loss* Pada Ujicoba Koneksi 256 Kbps

Dari pengukuran paket *loss* pada koneksi dengan konfigurasi limitasi *bandwith* 256 kbps setelah diterapkan *priority* queuing berdasarkan analisa data dari commview didapatkan statistik :

Tabel IV.18 Jumlah Paket *Loss* pada ujicoba koneksi 256 kbps dengan *QoS*

Ip sumber	Port asal	Ip tujuan	Port tujuan	Paket dikirim	Paket *loss*	Prosentase %
192.168.1.100	8000	192.168.1.2	19894	12547	1	0 %
192.168.2.100	8000	192.168.1.2	14962	12645	62	0,5 %
192.168.1.2	19894	192.168.2.100	8000	6357	0	0 %
192.168.1.2	14962	192.168.1.100	8000	6240	0	0 %

Berdasarkan tabel IV.18 diatas dapat diketahui bahwa pada saat ujicoba

koneksi dengan limitasi *bandwith* 256 kbps antara client 1 dan *VoIP server* yang berfungsi sebagai *VoIP* gateway hampir tidak ada paket yang hilang. Hal ini disebabkan karena kedua *node* ini masih berada dalam satu jaringan lokal sehingga komunikasi suara dapat berlangsung dengan sangat baik tanpa ada suara yang hilang. Pada client 1 saat mengirim informasi suara menggunakan port 8000 yang ditujukan ke *VoIP server* pada port 19894. Total paket yang dikirim sebanyak 12547 paket sedangkan paket yang hilang hanya 1 paket saja.

Pada client 2 yang sudah terhubung ke luar jaringan dengan koneksi *bandwith* yang digunakan antar kedua jaringan sebesar 256 kbps, client 2 mengirim data pada port 8000 yang ditujukan ke *VoIP server* pada port 19894. Pada proses komunikasi ini terdapat paket *loss* sebanyak 62 paket atau sebesar 0,5%.

4.6.6. Pengukuran *Jitter* Pada Ujicoba Koneksi 256 Kbps

Dari adanya koneksi komunikasi pada SIP client 1 dan SIP client 2 didapatkan statistik *jitter* pada saat mentransmisikan paket data selama terjadinya komunikasi, sebagai berikut :

Tabel IV.19 *jitter* pada ujicoba koneksi 256 kbps dengan *priority* queueing

Ip sumber	Port asal	Ip tujuan	Port tujuan	Paket dikirim	*Jitter* (ms)
192.168.1.100	8000	192.168.1.2	19894	12547	3,05
192.168.2.100	8000	192.168.1.2	14962	12645	23,01
192.168.1.2	19894	192.168.2.100	8000	6357	10,33
192.168.1.2	14962	192.168.1.100	8000	6240	25,58
Rata – rata *jitter*					15,5

Pada tabel IV.19 bisa diketahui dari seluruh total pengiriman paket pada ujicoba komunikasi *VoIP* dengan koneksi *backbone* 256 kbps, terdapat *jitter* yang berasal dari 4 port transmisi yang bervariasi dengan rata-rata *jitter* sebesar 15,5 ms. Dimana pada client 1 mengalami *jitter* 3,05 untuk transmisi ke *server*. Dari

server ke client 1 terdapat *jitter* sebesar 25,58 ms. sedangkan pada client 2 mengalami *jitter* 23,01 ms untuk transmisi ke *server*. Dari *server* ke client 2 terdapat *jitter* sebesar 10,33. Berdasarkan tabel referensi yang ada di atas maka dapat diambil kesimpulan *jitter* yang dihasilkan termasuk dalam kategori cukup.

4.6.7. Pengukuran *Bandwith* Pada Ujicoba Koneksi 256 Kbps

Pengukuran *bandwidth* pada ujicoba dengan *bandwith* interkoneksi sebesar 256 kbps setelah diteapkan *priority* queuing didapatkan statistik sebagai berikut :

Tabel IV.20 Pengukuran *Bandwidth* pada ujicoba dengan koneksi 256 kbps

Ip asal	port asal	Ip tujuan	Port tujuan	Paket dikirim	*bandwith*	*Packet loss*
192.168.1.100	8000	192.168.1.2	19894	12547	104.76	1
192.168.2.100	8000	192.168.1.2	14962	12645	104,2	62
192.168.1.2	14962	192.168.2.100	8000	6357	83.67	0
192.168.1.2	19894	192.168.1.100	8000	6240	83,2	0
TOTAL				36203	370,79	63
RATA-RATA				9050,75	92,57	15,75

Dari tabel IV.20 bisa diketahui *bandwidth* yang dibutuhkan saat melakukan komunikasi antara SIP client 1 ke *server* sebesar 104,76 kbps. Sedangkan dari SIP client 2 ke *server bandwith* yang dibutuhkan sebesar 104,2 kbps. *VOIP server* akan meneruskan paket yagn diterima dari sip client 2 ke sip client 1 sebesar 83,67 kbps, sedangkan untuk dari sip client 2 ke sip client 1 sbesar 83,2 kbps. Dari data diatas dapat kita lihat *bandwith* rata-rata yang dibutuhkan untuk satu kanal komunikasi *VoIP* adalah sebesar 92,57 kbps yang didapat dari rata-rata paket yang dikirim dan diterima oleh sip client.

4.6.8. Pengukuran MOS Score Pada Ujicoba Koneksi 256 Kbps

Kualitas suara pada saat ujicoba dengan *bandwith backbone* 256 kbps pada client 1 dan client 2, berdasarkan data dari commview didapatkan hasil penilaian kualitas suara atau MOS sebagai berikut :

Tabel IV.20 Pengukuran *MOS* pada ujicoba dengan koneksi 256 kbps

Ip asal	port asal	Ip tujuan	Port tujuan	Paket dikirim	*Packet loss*	MOS score
192.168.1.100	8000	192.168.1.2	19894	12547	1	4,4
192.168.2.100	8000	192.168.1.2	14962	12645	62	4,3
192.168.1.2	14962	192.168.2.100	8000	6357	0	4,4
192.168.1.2	19894	192.168.1.100	8000	6240	0	4,4
RATA-RATA				10654,6	15,75	4,38

Dari tabel IV.20 diatas dapat diketahui bahwa kualitas suara dan gambar yang dihasilkan saat terjadi koneksi komunikasi antara client 1 dan client 2 rata-rata nilai MOS score yang didapatkan adalah sebesar 4,38. Sehingga dapat diambil kesiimpulan MOS dari nilai tersebut dapat dikatakan kualitas suara yang dihasilkan adalah “sangat baik” untuk diaplikasikan.

4.6.9. Pengamatan Paket *Loss* Pada Ujicoba Koneksi 128 Kbps

Dari pengukuran paket *loss* pada koneksi dengan konfigurasi limitasi *bandwith* 128 kbps setelah diterapkan *priority* queuing berdasarkan analisa data dari commview didapatkan statistik :

Tabel IV.21 Jumlah paket *loss* pada ujicoba koneksi 128 kbps dengan *QoS*

Ip sumber	Port asal	Ip tujuan	Port tujuan	Paket dikirim	Paket *loss*	Prosentase %
192.168.1.100	8000	192.168.1.2	12080	13841	4	0 %
192.168.2.100	8000	192.168.1.2	18468	13830	1864	13,5 %
192.168.1.2	18468	192.168.2.100	8000	6919	3	0 %
192.168.1.2	12080	192.168.1.100	8000	5983	6	0,1 %

Berdasarkan tabel IV.21 diatas dapat diketahui bahwa pada saat ujicoba koneksi dengan limitasi *bandwith* 128 kbps antara client 1 dan *VoIP server* yang berfungsi sebagai *VoIP* gateway terdapat banyak paket yang hilang.. Pada client 1 saat mengirim informasi suara menggunakan port 8000 yang ditujukan ke *VoIP server* pada port 12080. Total paket yang dikirim sebanyak 13841 paket sedangkan paket yang hilang sebanyak 4 paket.

Pada client 2 yang sudah terhubung ke luar jaringan dengan koneksi *bandwith* yang digunakan antar kedua jaringan sebesar 128 kbps, client 2 mengirim data pada port 8000 yang ditujukan ke *VoIP server* pada port 18468. Pada proses komunikasi ini terdapat paket *loss* yagn tinggi. Yaitu sebesar 1864 paket atau 13,5%

4.6.10. Pengukuran *Jitter* Pada Ujicoba Koneksi 128 Kbps

Dari adanya koneksi komunikasi pada SIP client 1 dan SIP client 2 didapatkan statistik *jitter* pada saat mentransmisikan paket data selama terjadinya komunikasi, sebagai berikut :

Tabel IV.22 *Jitter* pada ujicoba koneksi 128 kbps dengan *priority queuing*

Ip sumber	Port asal	Ip tujuan	Port tujuan	Paket dikirim	*Jitter* (ms)
192.168.1.100	8000	192.168.1.2	12080	13841	2,03
192.168.2.100	8000	192.168.1.2	18468	13830	36,23
192.168.1.2	18468	192.168.2.100	8000	6919	2,4
192.168.1.2	12080	192.168.1.100	8000	5983	54,33
Rata – rata *jitter*					23,75

Pada tabel IV.22 bisa diketahui dari seluruh total pengiriman paket pada ujicoba komunikasi *VoIP* dengan koneksi *backbone* 128 kbps, terdapat *jitter* yang berasal dari 4 port transmisi yang bervariasi dengan rata-rata *jitter* sebesar 94,99 ms. Dimana pada client 1 mengalami *jitter* 2,03ms untuk transmisi ke *server*. Dari *server* ke client 1 terdapat *jitter* sebesar 54,33 ms. sedangkan pada client 2 mengalami *jitter* 36,23 ms untuk transmisi ke *server*. Dari *server* ke client 2 terdapat *jitter* sebesar 2,4 ms. Berdasarkan tabel referensi yang ada di atas maka dapat diambil kesimpulan *jitter* yang dihasilkan termasuk dalam kategori buruk dan tidak direkomendasikan untuk diaplikasikan dalam sistem *VoIP*.

4.6.11. Pengukuran *Bandwith* Pada Ujicoba Koneksi 128 Kbps

Pengukuran *bandwidth* pada ujicoba dengan *bandwith* interkoneksi sebesar 128 kbps, dilakukan dengan cara melakukan proses komunikasi suara antara kedua SIP client. Berikut ini adalah besarnya *bandwidth* yang dibutuhkan saat terjadi ujicoba kominikasi dengan *bandwith* 128 kbps tanpa menggunakan *QoS priority* queuing.

Tabel IV.23 Pengukuran *Bandwidth* pada ujicoba dengan koneksi 128 kbps

Ip asal	port asal	Ip tujuan	Port tujuan	Paket dikirim	*bandwith*	*Packet loss*
192.168.1.100	8000	192.168.1.2	12080	12279	52,24	6222
192.168.2.100	8000	192.168.1.2	18468	11854	104,74	0
192.168.1.2	18468	192.168.2.100	8000	6140	83.63	1
192.168.1.2	12080	192.168.1.100	8000	5930	41,73	2
TOTAL				36203	282,34	6225
RATA-RATA				9050,75	70,3	1556,3

Dari tabel IV.23 bisa diketahui untuk melakukan proses komunikasi, masing –masing client akan mengirim paket ke *VoIP server* untuk kemudian oleh *VoIP server* paket tersebut diteruskan ke client yang dituju. *bandwidth* yang dibutuhkan saat melakukan komunikasi antara SIP client 1 ke *server* sebesar 52,24 kbps. Sedangkan dari SIP client 2 ke *server bandwith* yang dibutuhkan sebesar 104,74 kbps. *VOIP server* akan meneruskan paket yang diterima dari sip client 2 ke sip client 1 sebesar 41,73 kbps, sedangkan untuk dari sip client 2 ke sip client 1 sbesar 83,63 kbps. Dari data diatas dapat kita lihat *bandwith* rata-rata yang dibutuhkan untuk satu kanal komunikasi *VoIP* adalah sebesar 70,3 kbps yang didapat dari rata-rata paket yang dikirim dan diterima oleh sip client.

4.6.12. Pengukuran MOS Score Pada Ujicoba Koneksi 128 Kbps

Kualitas suara pada saat ujicoba dengan *bandwith backbone* 128 kbps pada client 1 dan client 2, berdasarkan data dari commview didapatkan hasil penilaian kualitas suara atau MOS sebagai berikut :

Tabel IV.24 Pengukuran MOS pada ujicoba dengan koneksi 128 kbps

Ip asal	port asal	Ip tujuan	Port tujuan	Paket dikirim	*Packet loss*	MOS score
192.168.1.100	8000	192.168.1.2	12080	12279	4	4,4
192.168.2.100	8000	192.168.1.2	18468	11854	1864	2
192.168.1.2	18468	192.168.2.100	8000	6140	3	4,4
192.168.1.2	12080	192.168.1.100	8000	5930	6	4,4
RATA-RATA						3,8

Dari tabel IV.24 diatas dapat diketahui bahwa kualitas suara dan gambar yang dihasilkan saat terjadi koneksi komunikasi antara client 1 dan client 2 pada saat transmisi data dari client 2 ke *server* mendapatkan nilai MOS 2, Sehingga dapat diambil kesiimpulan MOS dari nilai tersebut dapat dikatakan kualitas suara yang dihasilkan adalah “sangat buruk” untuk diaplikasikan.

4.7 Hasil Pengujian Dengan Menggunakan 2 Kanal Komunikasi

sebagai bahan perbandingan sistem akan diuji dengan melakukan dua kanal koneksi komunikasi yang hasilnya akan dibandingkan dengan performansi di dalam komunikasi satu kanal.

4.7.1. Pengamatan Paket *Loss* Pada Ujicoba Koneksi 512 Kbps

Dari pengukuran paket *loss* pada koneksi dengan konfigurasi limitasi *bandwith* 512 kbps sebelum diterapkan *priority* queuing berdasarkan analisa data dari commview didapatkan statistik :

Tabel IV.25 Paket *Loss* pada ujicoba koneksi 512 kbps tanpa *QoS* untuk dua kanal komunikasi

Ip sumber	Port asal	Ip tujuan	Port tujuan	Paket dikirim	Paket *loss*	Prosentase %
192.168.2.200	8000	192.168.1.2	16990	4648	89	1,9
192.168.2.100	8000	192.168.1.2	19396	4676	59	1,2
192.168.1.2	19074	192.168.1.5	36036	2326	3	0,1
192.168.1.5	36036	192.168.1.2	19074	2368	0	0
192.168.1.2	16990	192.168.2.200	8000	2260	25	1
192.168.1.2	19396	192.168.2.100	8000	2372	23	1

Berdasarkan tabel IV.25 diatas dapat diketahui bahwa pada saat ujicoba koneksi dengan limitasi *bandwith* 512 kbps dimana terdapat dua kanal komunikasi, ada beberapa paket yang hilang yangdikirimkan oleh 2 client dari ip jaringan 192.168.2.X . tetapi jumlah paket yang hilang ini tidak terlalu mempengaruhi kepada beban system

4.7.2. Pengukuran *Jitter* Pada Ujicoba Koneksi 512 Kbps

Dari adanya proses komunikasi antara empat SIP client yagn masing-masing mengirimkan paket-paket RTP didapatkan statistik *jitter* pada saat mentransmisikan paket data selama terjadinya komunikasi, sebagai berikut :

Tabel IV.26 *jitter* pada ujicoba koneksi 512 kbps tanpa *QoS* untuk dua kanal komunikasi

Ip sumber	Port asal	Ip tujuan	Port tujuan	Paket dikirim	*Jitter* (ms)
192.168.2.200	8000	192.168.1.2	16990	4648	17,35
192.168.2.100	8000	192.168.1.2	19396	4676	18,13
192.168.1.2	19074	192.168.1.5	36036	2326	26,36
192.168.1.5	36036	192.168.1.2	19074	2368	9,73
192.168.1.2	16990	192.168.2.200	8000	2260	9,72
192.168.1.2	19396	192.168.2.100	8000	2372	9,44

Pada tabel IV.26 bisa diketahui dari seluruh total pengiriman paket pada ujicoba komunikasi *VoIP* dengan koneksi *backbone* 512 kbps, terdapat *jitter* yang berasal dari 6 port transmisi. Dimana pada client 1 mengalami *jitter* 17,35 ms untuk transmisi ke *server*. Dari *server* ke client 1 terdapat *jitter* sebesar 9,72 ms. sedangkan pada client 2 mengalami *jitter* 18,13 ms untuk transmisi ke *server*. Dari *server* ke client 2 terdapat *jitter* sebesar 9,44 ms. Pada client 3 terdapat *jitter* sebesar 9,73 ms untuk transmisi ke *server* dan 26,36 untuk sebaliknya. Berdasarkan tabel referensi yang ada di atas maka dapat diambil kesimpulan *jitter* yang dihasilkan termasuk dalam kategori bagus.

4.7.3. Pengukuran *Bandwith* Pada Ujicoba Koneksi 512 Kbps

Pengukuran *bandwidth* pada ujicoba dengan *bandwith* interkoneksi sebesar 512 kbps, dilakukan dengan cara melakukan proses komunikasi suara antara kedua SIP client, kemudian didapatkan paket-paket informasi yang dikirim dan dari total paket yang dikirim dengan menggunakan commview dapat diketahui berapa besar kapasitas dari masing-masing paket, setelah itu nilai rata-rata tiap pengiriman bisa diketahui besarnya *bandwidth* yang dibutuhkan. Berikut ini adalah besarnya *bandwidth* yang dibutuhkan saat terjadi ujicoba kominikasi 2 dua kanal dengan *bandwith* 512 kbps tanpa menggunakan *QoS priority* queuing.

Tabel IV.27 Pengukuran *Bandwidth* untuk dua kanal komunikasi

Ip sumber	Port asal	Ip tujuan	Port tujuan	Paket dikirim	Bandwidth rata-rata
192.168.2.200	8000	192.168.1.2	16990	4648	102,76
192.168.2.100	8000	192.168.1.2	19396	4676	103,41
192.168.1.2	19074	192.168.1.5	36036	2326	82,14
192.168.1.5	36036	192.168.1.2	19074	2368	83,64
192.168.1.2	16990	192.168.2.200	8000	2260	83,35
192.168.1.2	19396	192.168.2.100	8000	2372	100,19

Dari tabel IV.27 bisa diketahui untuk melakukan proses komunikasi, masing –masing client akan mengirim paket ke *VoIP server* untuk kemudian oleh *VoIP server* paket tersebut diteruskan ke client yang dituju. *bandwidth* yang dibutuhkan saat melakukan komunikasi antara SIP client 1 ke *server* sebesar 102,76 kbps. Sedangkan dari SIP client 2 ke *server bandwith* yang dibutuhkan sebesar 103,41 kbps. *VOIP server* akan meneruskan paket yagn diterima dari sip client 2 ke sip client 1 sebesar 83,35 kbps.

4.7.4. Pengukuran MOS Score Pada Ujicoba Koneksi 512 Kbps

Kualitas suara pada saat ujicoba dengan *bandwith backbone* 512 kbps pada client 1 dan client 2, berdasarkan data dari commview didapatkan hasil penilaian kualitas suara atau MOS sebagai berikut :

Tabel IV.28 Pengukuran *MOS* untuk dua kanal komunikasi

Ip sumber	Port asal	Ip tujuan	Port tujuan	Paket dikirim	MOS
192.168.2.200	8000	192.168.1.2	16990	4648	4,0
192.168.2.100	8000	192.168.1.2	19396	4676	4,1
192.168.1.2	19074	192.168.1.5	36036	2326	4,4
192.168.1.5	36036	192.168.1.2	19074	2368	4,4
192.168.1.2	16990	192.168.2.200	8000	2260	4,2
192.168.1.2	19396	192.168.2.100	8000	2372	4,2

Dari tabel IV.28 diatas dapat diketahui bahwa kualitas suara dan gambar yang dihasilkan saat terjadi koneksi komunikasi antara client 1 dan client 2 rata-rata nilai MOS score yang didapatkan adalah sebesar 4,2. Sehingga dapat diambil kesiimpulan MOS dari nilai tersebut dapat dikatakan kualitas suara yang dihasilkan adalah “Sangat Baik” untuk diaplikasikan.

4.7.5. Pengamatan Paket *Loss* Pada Ujicoba Koneksi 256 Kbps

Dari hasil pengujian untuk mengamati paket *loss* pada koneksi dengan konfigurasi limitasi *bandwith* 256 kbps pada saat sebelum diterapkan *priority* queuing berdasarkan analisa data dari commview didapatkan statistik sebagai berikut :

Tabel IV.29 Jumlah Paket *Loss* pada ujicoba koneksi 256 kbps tanpa *QoS*

Ip sumber	Port asal	Ip tujuan	Port tujuan	Paket dikirim	Paket *loss*	Prosentase %
192.168.2.200	8000	192.168.1.2	16990	4803	76	1,6
192.168.2.100	8000	192.168.1.2	19396	4782	96	2
192.168.1.2	19074	192.168.1.5	36036	2402	12	0,5
192.168.1.5	36036	192.168.1.2	19074	2437	0	0
192.168.1.2	16990	192.168.2.200	8000	2437	1	0,6
192.168.1.2	19396	192.168.2.100	8000	2433	14	1

Berdasarkan tabel IV.29 diatas dapat diketahui bahwa pada saat ujicoba koneksi dengan limitasi *bandwith* 256 kbps dimana terdapat dua kanal komunikasi, ada beberapa paket yang hilang yang dikirimkan oleh 2 client dari ip jaringan 192.168.2.X . dapat dilihat pada tabel diatas total paket *loss* yang ada cukup tinggi karena sudah melebihi 1 % hal ini akan sangat mempengaruhi kualitas suara yang dihasilkan. Dimana suara yagn dihasilkan akan menjadi terputus-putus .

4.7.6. Pengukuran *Jitter* Pada Ujicoba Koneksi 512 Kbps

Dari adanya koneksi komunikasi pada SIP client 1 dan SIP client 2 didapatkan statistik *jitter* pada saat mentransmisikan paket data selama terjadinya komunikasi, sebagai berikut :

Tabel IV.30 *jitter* pada ujicoba dua kanal *VoIP* koneksi 512 kbps tanpa *QoS*

Ip sumber	Port asal	Ip tujuan	Port tujuan	Paket dikirim	*Jitter* (ms)
192.168.2.200	8000	192.168.1.2	16990	4803	23,3
192.168.2.100	8000	192.168.1.2	19396	4782	20,54
192.168.1.2	19074	192.168.1.5	36036	2402	25,18
192.168.1.5	36036	192.168.1.2	19074	2437	9,41
192.168.1.2	16990	192.168.2.200	8000	2437	9,44
192.168.1.2	19396	192.168.2.100	8000	2433	2,2

Pada tabel IV.30 bisa diketahui dari seluruh total pengiriman paket pada ujicoba komunikasi *VoIP* dengan koneksi *backbone* 256 kbps, terdapat *jitter* yang berasal dari 6 port transmisi. Dimana pada client 1 mengalami *jitter* 23,3 ms untuk transmisi ke *server*. Dari *server* ke client 1 terdapat *jitter* sebesar 9,44 ms. sedangkan pada client 2 mengalami *jitter* 20,54 ms untuk transmisi ke *server*. Dari *server* ke client 2 terdapat *jitter* sebesar 2,2 ms. Pada client 3 terdapat *jitter* sebesar 9,41 ms untuk transmisi ke *server* dan 25,18 untuk sebaliknya. Berdasarkan tabel referensi yang ada di atas maka dapat diambil kesimpulan *jitter* yang dihasilkan termasuk dalam kategori cukup

4.7.7. Pengukuran *Bandwith* Pada Ujicoba Koneksi 256 Kbps

Pengukuran *bandwidth* pada ujicoba dengan *bandwith* interkoneksi sebesar 256 kbps, dilakukan dengan cara melakukan proses komunikasi suara antara kedua SIP client, kemudian didapatkan paket-paket informasi yang dikirim

dan dari total paket yang dikirim dengan menggunakan commview dapat diketahui berapa besar kapasitas dari masing-masing paket, setelah itu nilai rata-rata tiap pengiriman bisa diketahui besarnya *bandwidth* yang dibutuhkan. Berikut ini adalah besarnya *bandwidth* yang dibutuhkan saat terjadi ujicoba kominikasi 2 dua kanal dengan *bandwith* 256 kbps tanpa menggunakan *QoS priority* queuing.

Tabel IV.31 Pengukuran b*andwidth* pada ujicoba dengan koneksi 256 kbps

Ip sumber	Port asal	Ip tujuan	Port tujuan	Paket dikirim	Bandwidth rata-rata
192.168.2.200	8000	192.168.1.2	16990	4648	103,13
192.168.2.100	8000	192.168.1.2	19396	4676	102,70
192.168.1.2	19074	192.168.1.5	36036	2326	82,35
192.168.1.5	36036	192.168.1.2	19074	2368	83,63
192.168.1.2	16990	192.168.2.200	8000	2260	83, 63
192.168.1.2	19396	192.168.2.100	8000	2372	83,44

Dari tabel IV.31 bisa diketahui untuk melakukan proses komunikasi, masing –masing client akan mengirim paket ke *VoIP server* untuk kemudian oleh *VoIP server* paket tersebut diteruskan ke client yang dituju. *bandwidth* yang dibutuhkan saat melakukan komunikasi antara SIP client 1 ke *server* sebesar 102,76 kbps. Sedangkan dari SIP client 2 ke *server bandwith* yang dibutuhkan sebesar 103,41 kbps. *VOIP server* akan meneruskan paket yagn diterima dari sip client 2 ke sip client 1 sebesar 83,35 kbps, sedangkan untuk dari sip client 2 ke sip client 1 sbesar 100,19 kbps.

4.7.8. Pengukuran MOS Score Pada Ujicoba Koneksi 256 Kbps

Kualitas suara pada saat ujicoba dengan *bandwith backbone* 256 kbps pada client 1 dan client 2, berdasarkan data dari commview didapatkan hasil penilaian kualitas suara atau MOS sebagai berikut :

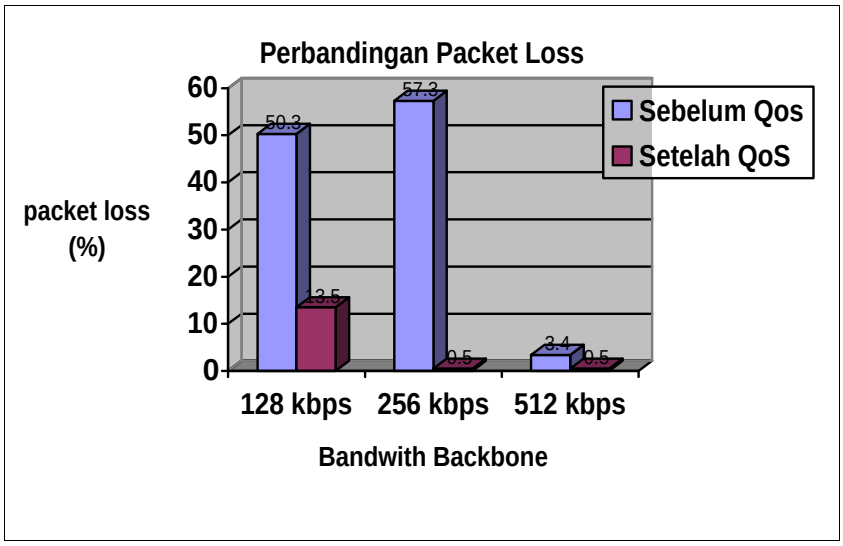
Tabel 4.32 *MOS* pada ujicoba dua kanal *VoIP* dengan koneksi 256 kbps

Ip sumber	Port asal	Ip tujuan	Port tujuan	Paket dikirim	MOS
192.168.2.200	8000	192.168.1.2	16990	4648	4,0
192.168.2.100	8000	192.168.1.2	19396	4676	3,9
192.168.1.2	19074	192.168.1.5	36036	2326	4,3
192.168.1.5	36036	192.168.1.2	19074	2368	4,4
192.168.1.2	16990	192.168.2.200	8000	2260	4,4
192.168.1.2	19396	192.168.2.100	8000	2372	4,3

Dari tabel IV.32 diatas dapat diketahui bahwa kualitas suara dalam satuan MOS paling rendah terjadi pada proses komunikasi antara client 2 dengan *server* dimana terdapat nilai MOS sebesar 3,9.

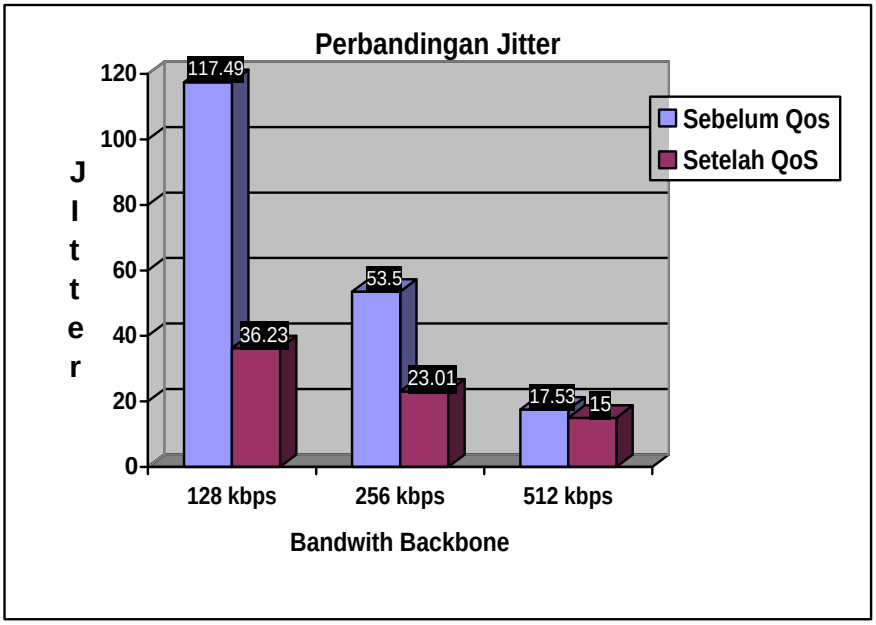
4.8 Perbandingan Performansi Jaringan

Berdasarkan pada hasil pengujian yang dilakukan dengan menggunakan beberapa skenario pengujian. Kita dapat menganalisa data-data yagn dihasilkan oleh program commview tersebut. Untuk memudahkan pembaca, hasil penelitian ini penulis sajikan dalam bentuk grafik sebagai berikut :



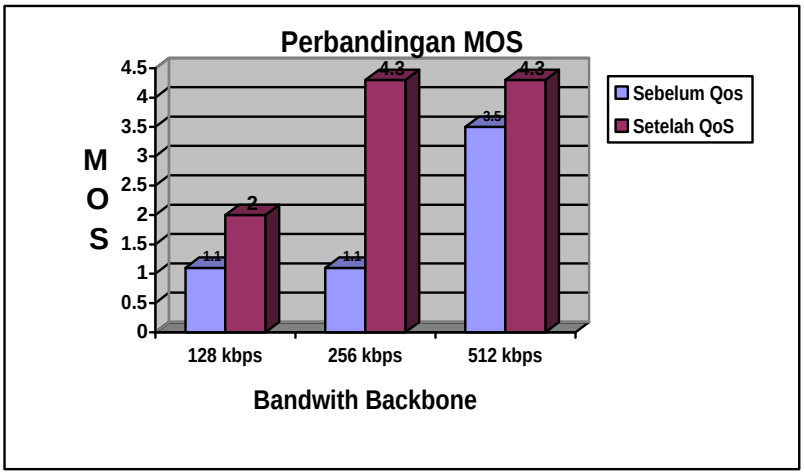
Gambar IV.18 Grafik perbandingan *packet loss*

Dari grafik diatas dapat diketahui, jumlah paket yang hilang pada saat dilakukan ujicoba sebelum diterapkan *priority queuing* pada koneksi *backbone* 128 kbps dan 256 kbps sangat banyak sekali, tetapi setelah diimplementasikan *priority queuing* paket yang hilang tersebut berkurang sangat jauh. Berdasarkan grafik diatas pengurangan tertinggi terjadi pada saat ujicoba pada koneksi *backbone* 256 kbps dimana sebelum diterapkan *QoS* tedapat paket yang hilang sebesar 50,3%, tetapi setelah diterpakan *QoS* terjadi penurunan julah paket yang hilang menjadi sebesar 0,5 %. Dampak dari penurunan paket yang hilang ini terasa sekali, karena suara yaagn dihasilkan menjadi lebih bagus dan dapat terdengar dengan jelas dibandingkan sebelummnya.



Gambar IV.19 Grafik perbandingan *Jitter*

Dari grafik diatas dapat diketahui parameter *jitter* mengalami penurunan yang cukup signifikan. Pada saat dilakukan ujicoba sebelum diterapkan *priority queuing* sangat banyak sekali, tetapi setelah diimplementasikan *priority queuing jitter* berkurang sangat jauh. Dapat dilihat pada grafik diatas pada saat proses komunikasi dilakukan pada koenksi backboen 128 kbps terdapat *jitter* sebesar 117,49 tetapi setelah diimplementasikan priority queing mengalami pengurangn menjadi hanya sebesar 36,23. Pengurangan tertinggi terjadi pada saat ujicoba pada koneksi *backbone* 128 kbps dimana terjadi penurunan hampir 300% lebih.



Gambar IV.20 Grafik perbandingan MOS

Dari grafik diatas dapat diketahui hasil perolehan nilai MOS meningkat jauh setelah diimplementasikan *priority queuing*. Kenaikan tertinggi terdapat pada saat ujicoba pada koneksi *backbone* 256 kbps, dimana terjadi kenaikan nilai MOS dari yang semula 1,1 menjadi 4,3 nilai in sudah sangat bagus jika kita implementasikan *VoIP* menggunakan codec G.711.

Dari hasil uji coba diatas dapat diambil kesimpulan bahwa implementasi *priority queuing* di dalam sebuah jaringan sistem *VoIP* sangat efektif untuk membantu meningkatkan kualitas sambungan komunikasi dalam sistem *VoIP*. Dengan diaplikasikannya metode ini dampak yang dihasilkan sangat besar sekali, dimana seperti terlihat pada grafik diatas terdapat pengurangan *packet loss* yang cukup signifikan jika dibandingkan dengan sebelum diterapkan metode ini, hasil pengurangan *packet loss* ini berdampak pula ke pengurangan *delay*/*jitter* sehinggga kualitas suara yang dihasilkan termasuk pada kategori bagus.

4.9 Studi Kasus Implementasi *VoIP* Pada Suatu Perusahaan.

Telepon internet pada dasarnya beroperasi menggunakan jaringan komputer berbasis internet dengan menggunakan protokol TCP/IP. Oleh karenanya, *VoIP* dapat dioperasikan dengan menggunakan jaringan internet, tapi dapat pula dioperasikan secara internal di LAN. Adapun kebutuhan minimal perangkat yang harus tersedia jika kita akan mengimplementasikan *VoIP* pada suatu perusahaan adalah sebagai berikut :

a. *VoIP server*

   Kita dapat menggunakan Asterisk sebagai *VoIP server* karena asterisk merupakan paket distro *VoIP* yang handal dan open source.

b. Komputer yang terhubung ke jaringan LAN yangdilengkapi dengan kartu suara dan headset untuk komunikasi antar klien. Jika ada dana lebih baiknya gunakanlah ip phone, karena dalam penggunaanya lebih irit listrik. Beberapa ip phone mempunyai kemampuan Wi-Fi sehingga bisa langsung digunakan seperti layaknya ponsel di sebuah jaringan hotspot.

c. Software klien

   Agar bisa menjadi user dari Asterisk kita harus menginstall software klien *VoIP* yang berbasis SIP seperti X-Lite, SJ-Phone dan lain sebagainya yang bisa kita unduh secara gratis

d. Media gateway

   Perangkat ini dibutuhkan jika kita ingin mengabungkan sistem *VoIP* kita dengan PSTN telkom, sehingga user *VoIP* dapat dipakai untuk melakukan panggilan ke jaringan telepon analog, GSM dan CDMA.

e. Jaringan LAN

Paket-paket *VoIP* yang berupa trafik RTP akan dilewatkan pada media transmisi yang berupa jaringan LAN. Dalam penerapanya sebaiknya digunakan peralatan-peralatan jaringan yang sudah mendukung gigabit *ethernet* agar sistem bisa dikembangkan lagi jika ada kebutuhan penambahan aplikasi seperti video conference dsb. Gunakanlah *managed switch* agar *network administrator* lebih mudah menerapkan kebijakan-kebijakan yang berkaitan dengan performansi jaringan.

4.9.1. Kebutuhan Hardware Bagi Asterisk IP PBX

Pertanyaan yang sering muncul jika kita ingin mengimplementasikan asterisk adalah seberapa besar jumlah panggilan serempak yang akan ditangani oleh asterisk. Jawaban dari pertanyaan diatas sangat kompleks sekali karena tergantung dari beberapa faktor, tetapi secara umum asterisk membutuhkan sekitar 30Mhz kemampuan CPU untuk setiap kanal komunikasi yang aktif dengan asumsi bahwa semua panggilan menggunakan codec G.711.

4.9.2. Perencanaan *VoIP*

Beberapa hal yang harus kita perhitungkan dalam perencanaan *VoIP* adalah :

a. Kualitas suara
b. *Bandwith* yang dibutuhkan
c. Konsekuensi *delay*/ Jittter

4.9.2.1. Memperhitungkan Kualitas Suara

Kualitas suara menjadi penting, karena ini akan sangat mempengaruhi

penerimaan suara di lawan bicara kita. Satuan kualitas suara yang digunakan yaitu menggunakan MOS *(mean opinion score)* . Kualitas suara sangat tergantung pada dua hal, yaitu :

a. Teknik kompresi yang digunakan

   Semakin bagus tingkat kompresi yang digunakan maka nilai MOS nya semakin berkurang (lihat tabel MOS di BAB II) . jika bandwtih tersedia gunakanlah codec G.711, karena codec ini sangat bagus nilai MOS nya sehingga kualitas suara yang dihasilkan sangat jernih.

b. Banyaknya paket *loss* di jaringan

   Efek dari paket *loss* di jaringan (karena berbagai hal) akan menyebabkan MOS dan R-Factor turun drastis setelah 5% paket *loss* seperti tabel dibawah, disinilah pentingny asebuah manajemen jaringan yang handal (*QoS*), seorang *network administrator* harus dapat mengetahui trafik mana yang harus didahulukan agar tidak terjadi paket *loss* pada paket *VoIP* yang banyak.

Tabel 4.3 Standarisasi *packet loss*

Codec	Frame rate	*Packet Loss*	MOS
G.729	20ms	0%	4,1
G.729	20ms	5%	3,3
G.729	20ms	10%	2,7
G.729	20ms	20%	1,9

4.9.2.2. Menghitung Kebutuhan *Bandwith*

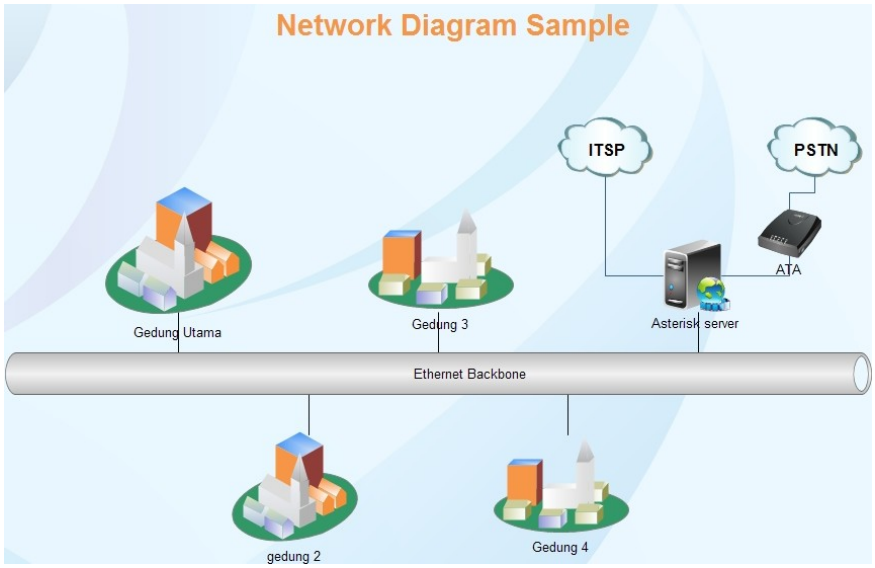
Untuk memperoleh perkiraan penggunaan *bandwith* kita bisa menggetahuinya dengan mengetaui situs web asterisk guru di http://www.asteriskguru.org lalu cari link *bandwith* calculator. Dengan tools *bandwith* calculator ini kita dapat menghitung *bandwith* yang dibutuhkan untuk berbagai codec yang kita gunakan untuk jumlah panggilan tertentu. Hasilnya akan diperlihatkan untuk outgoing *bandwith* maupun incoming *bandwith*. Secara umum untuk codec G.711 dibutuhkan *bandwith* sebesar 82 kbps per satu kanal komunikasi, jika kita ingin menyediakan 10 kanal, maka tinggal mengalikan saja kebutuhan *bandwith* tersbut dengan jumlah panggilan serentak yang ingin dicapai.dalam hal ini 82 x 10 = 820 kbps.

4.9.2.3. Konsekuensi *Delay*/*Jitter*

Seperti kita ketahui semakin tinggi *delay*/*jitter* maka kualitas suara yang dihasilkan akan semakin berkurang. Jika *bandwith*nya cukup gunakanlah codec G.711, karena codec ini tingkat kompresinya sangat minim. Semakin tinggi tingkat kompresinya maka *delay* yang dihasilkan juga akan semakin tinggi.

4.9.3. Konfigurasi Jaringan

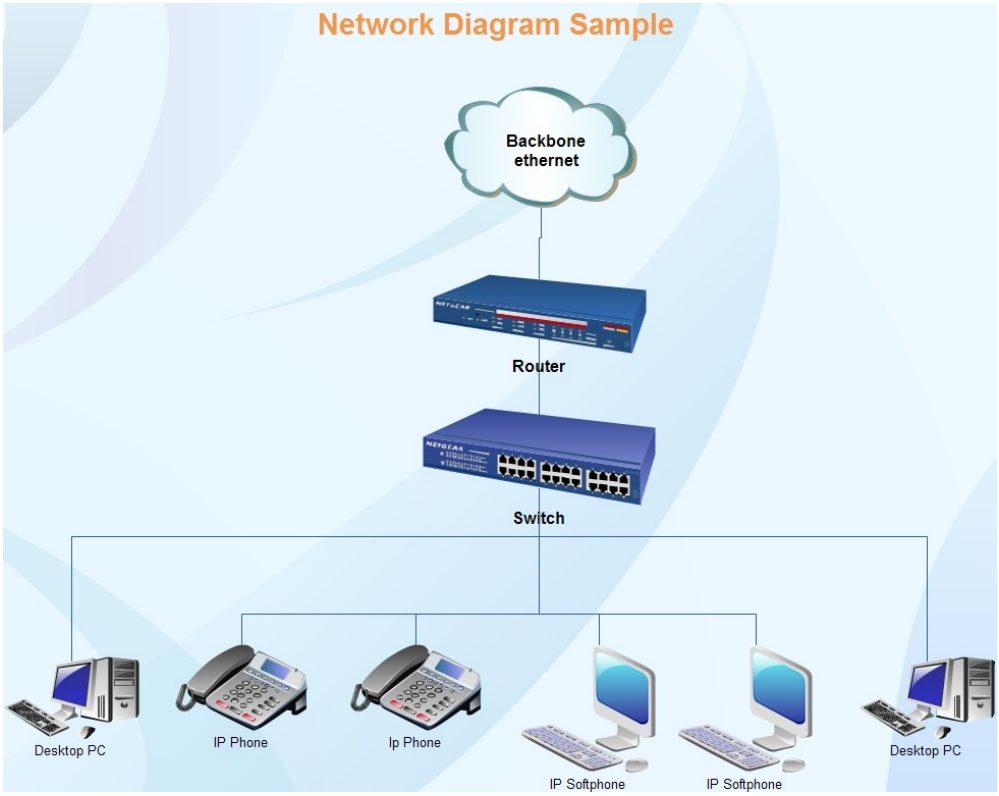
Sebagai contoh penulis gambarkan konfigurasi jaringan untuk diimplementasikan sistem *VoIP* seperti pada gambar berikut ini :



Gambar IV.21 Konfigurasi jaringan *backbone*

Pada gambar diatas dapat terlihat bahwa semua gedung terhubung dalam satu jaringan *Ethernet*/LAN yang sudah terdapat *VoIP server* didalamnya. Jaringan *Ethernet* ini akan digunakan untuk transmisi semua paket trafik data baik yang berupa data, *voice* ataupun video. Pada masing-masing gedung terdapat beberapa user dari asterisk *VoIP server* baik yang berupa *ip phone* ataupun *softphone.* Masing-masing user ini akan mendapatkan nomor elektronik yang berguna unutkk mewakili user tersebut pada saat terjadi sebuah panggilan. Masing-masing user ini akan mendapatkan empat digit nomor untuk bisa dipanggil oleh user yang lainnya. Agar dapat terhubung ke jaringan telephone analog PSTN (*Public switched telephone Network*) system ini kita pasangi sebuah *ATA (analog telephone adapter*) agar dari masing-masing user dapat melakukan panggilan ke nomor telepon rumah, GSM ataupun CDMA. Selain itu system juga akan dikoneksikan dengan ITSP (*Internet telphoni service provider*). ITSP ini berperan sebagai penyedia layanan internet telephoni agar kita bias terubung ke

server-server VoIP yang berada di jaringan WAN seperti skype atau *VoIP*rakyat. Adapun konfigurasi jaringan yang ada pada masing-masing gedung adalah sebagai berikut:



Gambar IV.22 Konfigurasi Jaringan Internal Gedung

Pada gambar IV.22 dapat kita lihat semua user dari Asterisk *VoIP server* terhubung pada sebuah *switch* utama yagn melayani komunikasi data dan *voice*. *switch* ini terkoneksi dengan *router* yang terhubung pada jaringan *backbone* yang ada di perusahaan tersebut. Untuk *router* ini kita bisa menggunakan sebuah pc yang kita install Mikrotik *router*OS agar kita bias menerapkan kebijakan-kebijakan yang berkaitan dengan performansi jaringan seperti *QoS*, *bandwith* manajemen dan lain sebagainya.Pada sisi *router* ini kita akan terapkan *priority*

queuing agar trafik yang berupa paket-paket RTP dapat dilewatkan terlbih dahulu dan mendapat prioritas utama untuk dikirimkan oleh *router* ke *node* jaringan yang selanjutnya yaitu *backbone Ethernet*.