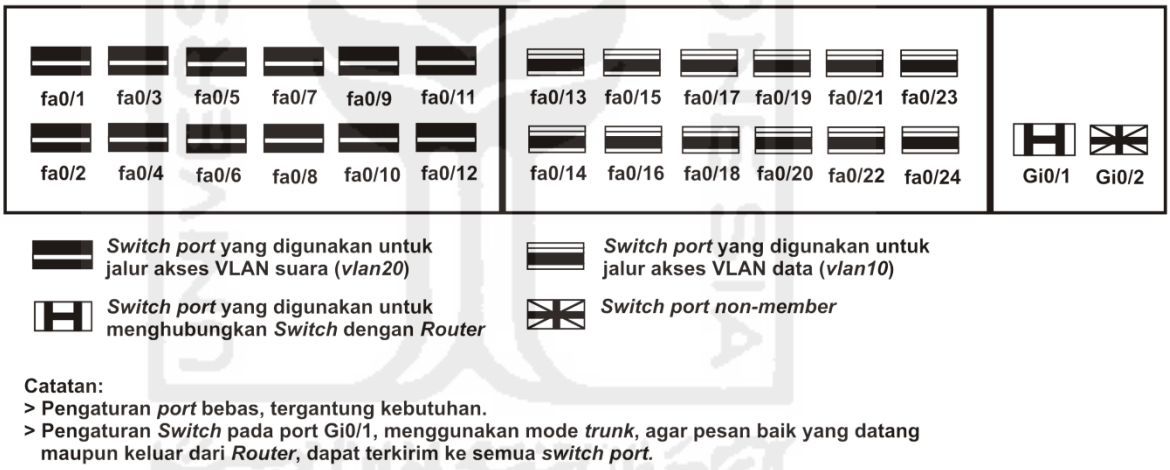
3.4.3 Konfigurasi Segmentasi VLAN Data dan Suara

Konfigurasi segmentasi VLAN data dan suara, dilakukan dengan memberikan pelabelan pada masing-masing *switch port*, yakni label "*DATA*" untuk akses VLAN data dan label "*VOICE*" untuk akses VLAN suara. Langkah ini dilakukan, karena berdasarkan pernyataan Dosen Universitas Jenderal Soedirman, Azis Wisnu Widhi N., mengatakan bahwa notabene baik paket data maupun suara, sama-sama diperlakukan sebagai data dalam komunikasi IP. Oleh karena itu, agar sistem dapat mengetahui mana jalur yang digunakan sebagai akses data dan mana yang digunakan sebagai jalur akses suara, maka perlu dibedakan berdasarkan *protokol* dan *port* yang digunakan. Untuk memenuhi kriteria tersebut, maka penulis melakukan pengaturan *switch port* seperti pada gambar berikut.

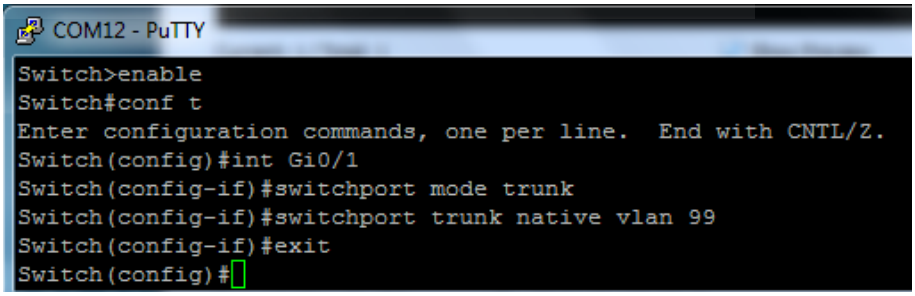


Gambar 3.39 Pengaturan *Switch Port*

Selain melalui pelabelan di atas, paket data dan suara juga perlu dibedakan berdasarkan prioritas, dimana jalur yang digunakan untuk akses suara, memiliki prioritas lebih tinggi daripada jalur data. Hal ini dikarenakan, penggunaan jalur suara bersifat *real time*, sehingga membutuhkan akses yang lebih cepat. Berdasarkan artikel yang dipublikasikan media *Ciscopress.com* (2011), diketahui bahwa, dalam menghubungkan *Cisco IP Phones* ke *Switch*, maka harus mengaktifkan juga *spanning-tree portfast* pada konfigurasi VLAN. Dengan menambahkan pengaturan ini, maka *IP Phones* dapat melakukan *booting* dengan cepat dan dapat dengan segera meminta alamat DHCP, meskipun *port* khas oleh *spanning-tree* belum diaktifkan.

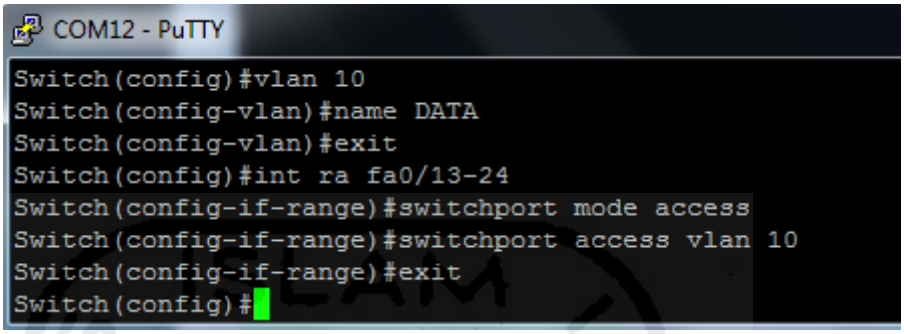
Tujuan dari penerapan segmentasi VLAN ini, untuk menjaga lalu lintas data dan suara dari berbagai ancaman *data network*. Dengan segmentasi ini, serangan apapun yang menuju ke *data network*, tidak akan mempengaruhi proses transmisi data, yang berlangsung di jaringan tersebut (Yusro, 2009). Kemudian untuk konfigurasi dari segmentasi VLAN, akan dijelaskan pada tahapan berikut.

1) Melakukan pengaturan *mode trunk*, pada *switch port* yang terhubung ke *Router*. *Mode trunk* digunakan untuk menghubungkan beberapa VLAN yang berada pada *Switch* berbeda, ke dalam satu segmen yang sama. Kemudian *mode trunk* tersebut, diaktifkan dengan menggunakan VLAN asli (*native* VLAN), agar dapat mendukung tag 802.1Q, dimana ia digunakan agar semua VLAN yang terkonfigurasi, dapat beroperasi di lingkungan apapun.



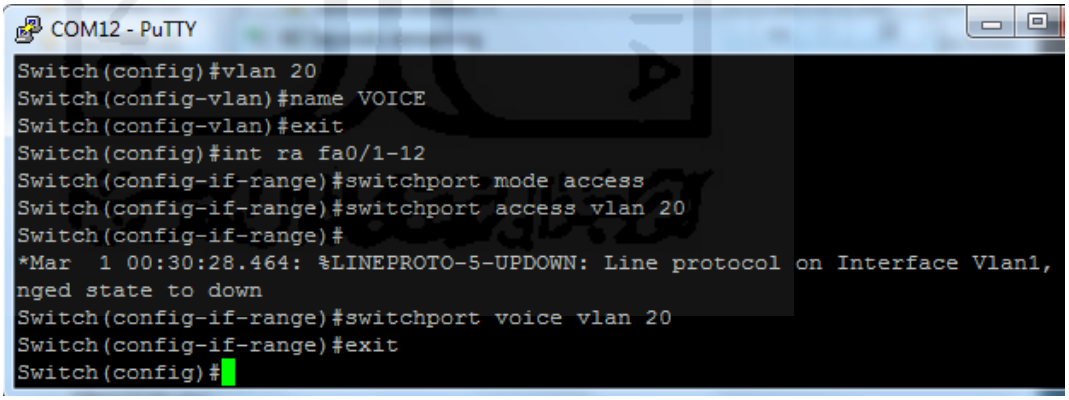
Gambar 3.40 Pengaturan *Switch Port Mode Trunk*

2) Melakukan pengaturan segmentasi *vlan 10*, sebagai jalur akses data. Hal ini ditandai, dengan perintah *mode access* dan *access vlan 10* pada *Switch Cisco*, yang diletakkan pada *port* fa0/13-24.



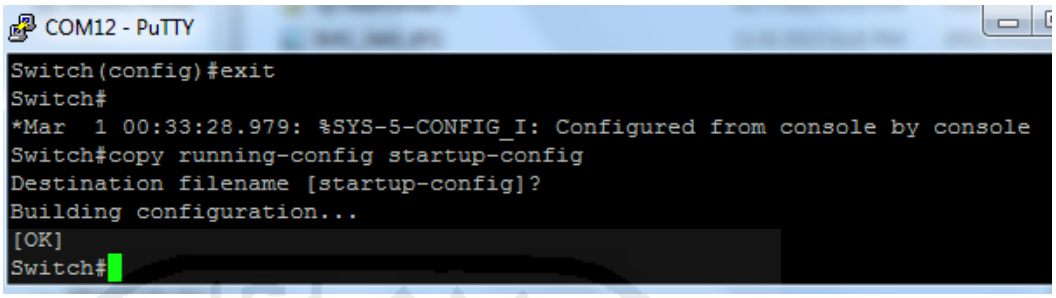
Gambar 3.41 Pengaturan *Switch Port Mode Access*

3) Melakukan pengaturan segmentasi *vlan 20*, sebagai jalur akses suara. Hal ini ditandai, dengan perintah *mode access*, *access vlan 20*, dan *voice vlan 20* pada *Switch Cisco*, yang diletakkan pada *port* fa0/1-12. Dengan memasukkan perintah *switchport voice <VLAN ID>*, maka secara otomatis, akan muncul pengaturan *spanning-tree portfast* pada sistem *Switch* untuk *vlan 20*.



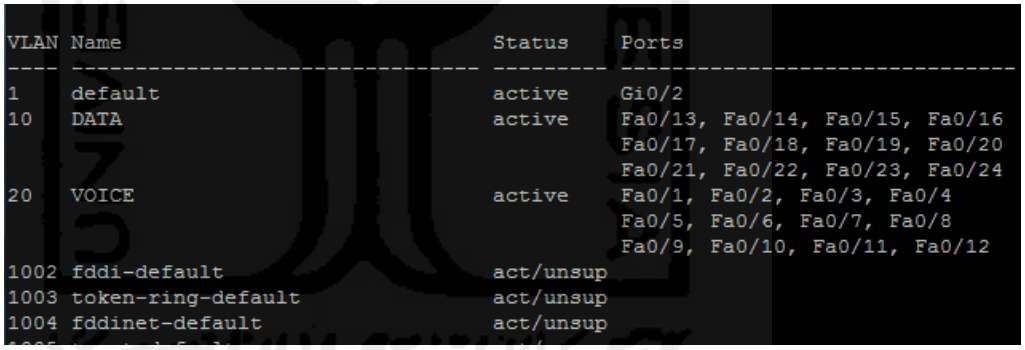
Gambar 3.42 Pengaturan *Switch Port Mode Voice*

4) Menyimpan pengaturan segmentasi VLAN dengan perintah #*copy running-config startup-config.*



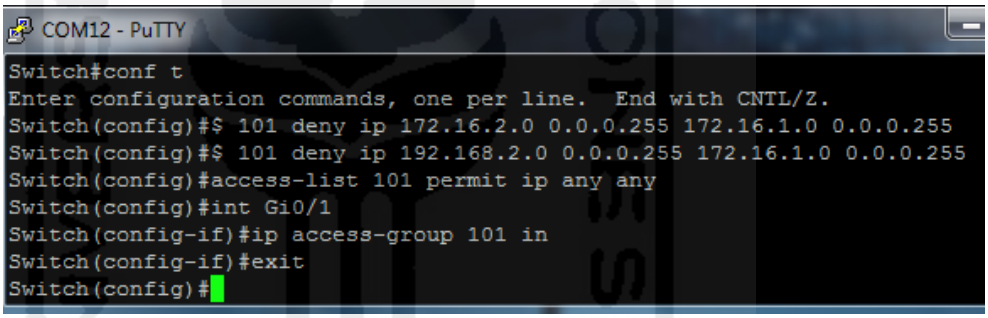
Gambar 3.43 Cara Menyimpan Konfigurasi VLAN

5) Hasil pengaturan VLAN dapat dilihat melalui perintah #*show vlan*. Dari informasi yang ditampilkan, tidak menunjukkan lokasi *vlan 99* berada. Hal ini terjadi karena, *vlan 99* telah diatur agar memiliki fungsi *trunk*, sehingga *port* yang digunakan VLAN tersebut (*port* Gi0/1), tidak tampak pada jajaran VLAN.



Gambar 3.44 Melihat Hasil Pengaturan VLAN melalui Perintah *Show Vlan*

6) Setelah melakukan pengaturan segmentasi VLAN, maka langkah selanjutnya adalah membuat aturan *firewall*. Hal ini dikarenakan, pengaturan segmentasi VLAN belum cukup untuk mengamankan jaringan VoIP pada *layer* 2 (*data link*). Sampai pada tahap ini, kedua jaringan (*vlan10 dan vlan20*) masih dapat melakukan pengiriman paket ICMP satu dengan yang lain, sedangkan tujuan dari percobaan praktikum ini adalah bagaimana agar kedua jaringan, tidak saling terhubung kecuali mereka dalam satu segmen yang sama dan melalui pengaturan tertentu yang diarahkan melalui *firewall*. Penambahan aturan *firewall* ini, penulis terapkan pada perangkat *Cisco Catalyst Switches* 2960 *Series*, karena perangkat tersebut juga memiliki kemampuan untuk melakukan *routing table* pada *layer* 3 (*network*).



Gambar 3.45 Penambahan Aturan *Firewall* pada *Switch Cisco*

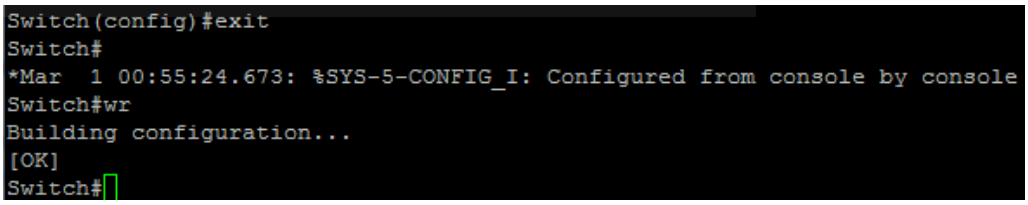
Berikut keterangan dari pembuatan *firewall* di atas:

- Sintak yang tertutup pada baris tiga dan empat, berisi perintah *access-list*, sehingga aturan dari *firewall* tersebut, lengkapnya adalah

  #access-list [nomor ACL] permit/deny [tipe paket yang akan diakses] [IP sumber] [wildcard sumber] [IP tujuan] [wildcard tujuan].

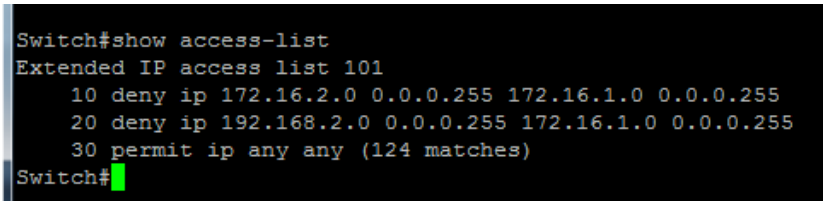
- Nomor ACL yang dimaksud pada aturan di atas adalah *Access Control List* yang digunakan untuk menyaring paket apapun yang lewat, sesuai dengan tipe paket yang dipilih. Nomor ACL yang digunakan pada pengerjaan tugas akhir ini adalah tipe *extended*, yang memiliki *range* 100-199. Nomor ACL ini, dapat menyaring paket berdasarkan IP *address*, *port*, dsbnya. Berbeda dengan tipe *standard* (1-99), yang hanya dapat menyaring paket berdasarkan IP *address* (Alberghetti, 2013).
- Secara garis besar, maksud dari pembuatan aturan *firewall* di atas adalah, upaya untuk menutup akses pengiriman paket IP yang bersumber dari jaringan 172.16.2.0/24 (*vlan20*) dan jaringan 192.168.2.0/24 (yang berada di luar *server Router*), agar kedua jaringan tidak dapat melakukan komunikasi dengan jaringan 172.16.1.0/24 (*vlan10*). Pengaturan *firewall* ini, dilakukan agar masing-masing jalur, tidak mengalami interferensi. Pengaturan tersebut sengaja dijalankan pada *layer* 3 *Switch Cisco*, agar tidak mengganggu aktivitas *routing table* yang berjalan di *layer* 3 *server Router*. Penutupan jalur yang dilakukan dari luar jaringan *vlan10*, memberikan keuntungan bagi *user* pada jaringan tersebut, untuk dapat berkomunikasi dengan *user* lain, asal berada dalam satu segmen yang sama, meskipun berbeda jaringan.

7) Hasil pengaturan *firewall*, kemudian disimpan dengan perintah #*wr*.



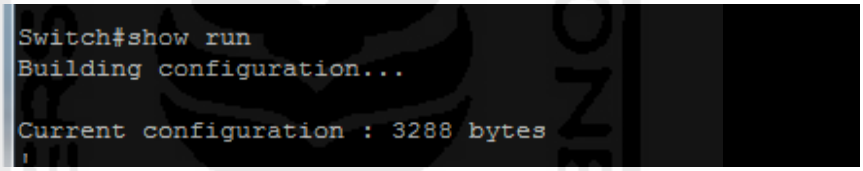
Gambar 3.46 Cara Menyimpan Hasil Pengaturan *Firewall* pada *Switch Cisco*

8) Hasil pengaturan *firewall* pada *Switch Cisco*, dapat dilihat melalui perintah *#show access-list*.



Gambar 3.47 Melihat Hasil Pengaturan *Firewall*

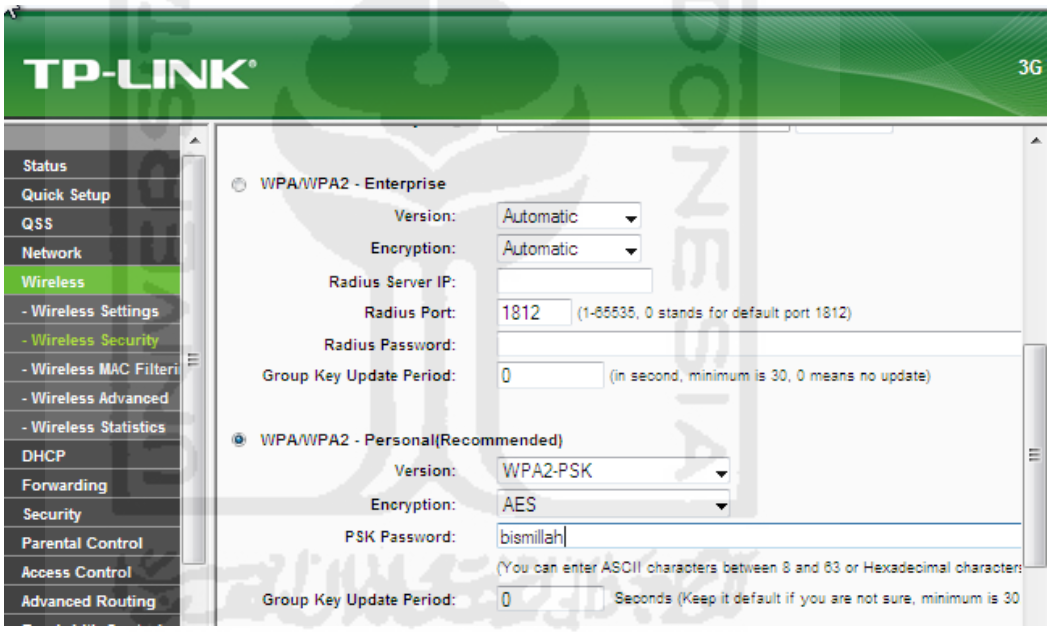
9) Untuk melihat hasil pengaturan yang lebih detail, dari konfigurasi VLAN yang telah dilakukan sebelumnya, dapat diperoleh melalui perintah *#show run*. Hasil konfigurasi dari pengaturan VLAN tersebut , lebih lanjut dapat dilihat dalam Lampiran I.



Gambar 3.48 Menampilkan Informasi Detail terhadap Setiap Langkah Konfigurasi *Switch Cisco*

3.4.4 Konfigurasi *Access Point TP Link*

Access Point digunakan sebagai media untuk mendukung distribusi alamat IP, yang telah diatur di dalam *server Router*. Dalam penggunaannya, terdapat bagian yang perlu diatur untuk meningkatkan keamanan WiFi (*Wireless Fidelity*) itu sendiri. Pengaturan tersebut dapat dilakukan dengan menggunakan keamanan versi *WPA2-PSK*, dimana ia menggunakan enkripsi AES (*Advanced Encryption Standard*) untuk melindungi kerahasiaan data. Enkripsi tersebut memiliki panjang *key* 8-63 bit, sehingga lebih lama tertembus pertahanannya daripada enkripsi TKIP (*Temporal Key Integrity Protocol*), yang memiliki banyak kelemahan (Yasir, 2014).



Gambar 3.49 Mengatur Keamanan WiFi

3.4.5 Konfigurasi *Server* VoIP

Dalam melakukan konfigurasi *server* VoIP, perlu menyediakan terlebih dahulu *package* yang dibutuhkan, untuk mendukung kelancaran dan kestabilan kinerja *Server* VoIP. Berikut tahapan konfigurasi *server* VoIP, pada Sistem Operasi *Ubuntu* 13.10, beserta penjelasannya.

1) Melakukan *instalasi* terhadap *package* berikut, untuk mendukung kelancaran dan kestabilan kinerja *server* VoIP.
   - *Asterisk*, berfungsi sebagai *file* utama dalam instalasi VoIP.
   - *Asterisk Addons,* dibutuhkan ketika ingin menambahkan kemampuan *Asterisk,* seperti instalasi *database MySQL* atau mp3.
   - *Asterisk Sound, package* yang berfungsi mendukung kemampuan suara pada *server* VoIP.
   - *Libpri,* digunakan untuk *primary rate,* yakni standar *interface* telekomunikasi, yang digunakan untuk membawa beberapa suara dan transmisi data antar jaringan dan pengguna.
   - *Libss7,* berfungsi menghindari interferensi antar jaringan, dengan menggunakan sistem *point to point,* yang berjalan pada lapisan *data link.*
   - *Dahdi,* berfungsi sebagai *timing* pada *server* VoIP.
   - *Mpg123,* mendukung *music on hold* sebagai *media player*.

2) Melakukan pengaturan terhadap *file iax.conf,* untuk mendaftarkan *user* agar tergabung dalam jaringan VoIP. *File* tersebut, terletak dalam *folder /etc/asterisk*. Untuk membukanya, dapat dilakukan dengan mengetikkan perintah *#nano iax.conf,* pada *folder* tersebut. Berikut perintah yang dapat ditambahkan di dalam baris terakhir dari *file iax.conf* pada Gambar 3.50.

```
bindaddr=0.0.0.0
disallow=all
allow=gsm
context=default ;default context for incoming call

;BEGIN 11523034
[11523034]
type=friend
username=11523034
secret=coba123
host=dynamic
callerid=Metri <11523034>
context=users
dtmfmode=rfc2833
nat=yes
canreinvite=no
;END 11523034

;BEGIN 11523046
[11523046]
type=friend
username=11523046
secret=coba123
host=dynamic
callerid=Puja <11523046>
context=users
dtmfmode=rfc2833
nat=yes
canreinvite=no
;END 11523046

;BEGIN 11523207
[11523207]
type=friend
username=11523207
secret=coba123
host=dynamic
callerid=Lulu <11523207>
context=users
dtmfmode=rfc2833
nat=yes
canreinvite=no
;END 11523207

;BEGIN 11523261
[11523261]
type=friend
username=11523261
secret=coba123
host=dynamic
callerid=Prisma <11523261>
context=users
dtmfmode=rfc2833
nat=yes
canreinvite=no
;END 11523261
```

Gambar 3.50 Pendaftaran *User* VoIP melalui *File Iax.conf*

3) Melakukan pengaturan terhadap *file extensions.conf*, untuk memasukkan perintah tambahan sebagai aturan VoIP. *File* ini juga terletak dalam *folder /etc/asterisk.* Untuk membukanya, dapat dilakukan dengan mengetikkan perintah *#nano extensions.conf*, pada *folder* tersebut. Berikut perintah yang dapat ditambahkan di dalam baris terakhir dari *file extensions.conf* pada Gambar 3.51. Pengaturan tersebut dalam tugas akhir ini, masih dilakukan secara manual, sehingga belum fleksibel digunakan untuk menambahkan aturan panggilan secara otomatis.

```
; All Setting
[users]
exten => 11523034,1,Answer
exten => 11523034,n,Log(NOTICE,"11523034 ACCOUNT")
exten => 11523034,n,Dial(IAX2/11523034,120,Tt)
exten => 11523034,n,Dial(IAX2/11523034)
exten => 11523034,n,Hangup()
exten => 11523034,1,Goto(11523046,1)
exten => 11523034,1,Goto(11523207,1)
exten => 11523034,1,Goto(11523261,1)

exten => 11523046,1,Answer
exten => 11523046,n,Log(NOTICE,"11523046 ACCOUNT")
exten => 11523046,n,Dial(IAX2/11523046,120,Tt)
exten => 11523046,n,Dial(IAX2/11523046)
exten => 11523046,n,Hangup()
exten => 11523046,1,Goto(11523034,1)
exten => 11523046,1,Goto(11523207,1)
exten => 11523046,1,Goto(11523261,1)

exten => 11523207,1,Answer
exten => 11523207,n,Log(NOTICE,"11523207 ACCOUNT")
exten => 11523207,n,Dial(IAX2/11523207,120,Tt)
exten => 11523207,n,Dial(IAX2/11523207)
exten => 11523207,n,Hangup()
exten => 11523207,1,Goto(11523034,1)
exten => 11523207,1,Goto(11523046,1)
exten => 11523207,1,Goto(11523261,1)

exten => 11523261,1,Answer
exten => 11523261,n,Log(NOTICE,"11523261 ACCOUNT")
exten => 11523261,n,Dial(IAX2/11523261,120,Tt)
exten => 11523261,n,Dial(IAX2/11523261)
exten => 11523261,n,Hangup()
exten => 11523261,1,Goto(11523034,1)
exten => 11523261,1,Goto(11523046,1)
exten => 11523261,1,Goto(11523207,1)
```

Gambar 3.51 *Input* Pengaturan Panggilan melalui *File Extentions.conf*

4) Mengaktifkan *Ethernet eth0* untuk s*etting Static* pada *IPAddress server* VoIP. Fungsi dari langkah ini adalah untuk memudahkan klien, dalam mencari alamat *host server* VoIP. Karena *server* VoIP diletakkan pada jaringan VLAN suara (*vlan20*), maka pengaturan *IPAddress* berada dalam jaringan 172.16.2.0/24.

```
auto eth0
iface eth0 inet static
address 172.16.2.3
netmask 255.255.255.0
gateway 172.16.2.1
```

Gambar 3.52 Pengaturan Alamat IP *Server* VoIP

5) Mengecek *interface Dahdi*, dengan perintah *#/etc/init.d/dahdi restart*, dimana modul-modul *dahdi* berfungsi sebagai *timing* pada *server* VoIP.

Gambar 3.53 *Interface Dahdi*

6) Menjalankan VoIP *Asterisk* dengan perintah *#asterisk –stop* terlebih dahulu, baru kemudian perinta *#asterisk -r*. Berikut tampilan VoIP *Asterisk* pada Gambar 3.54, yang berhasil dijalankan.

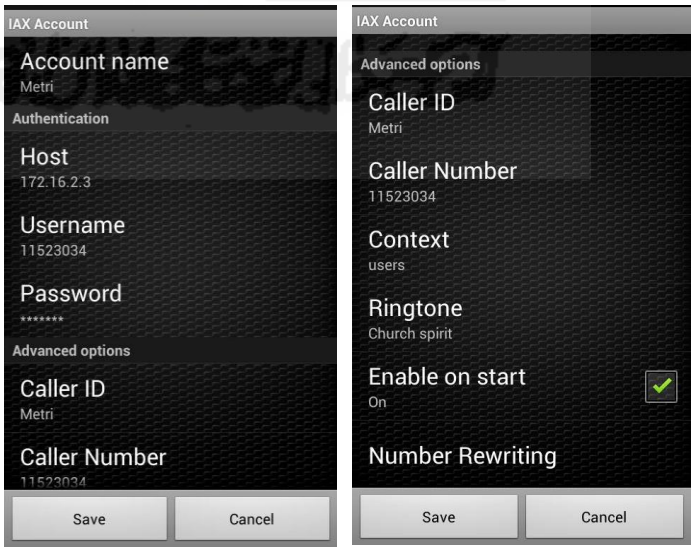
Gambar 3.54 Perintah Menjalankan VoIP *Asterisk*

3.4.6 Pengaturan Akun *Client* VoIP pada Aplikasi *Zoiper*

Berikut tahapan pengaturan akun klien pada aplikasi *zoiper*.

a. Pendaftaran Klien

1) Dalam pengerjaan tugas akhir ini, penulis memilih "*Manual Configuration*" dengan menggunakan protokol *signaling* "IAX". Berikut tata cara pengisian identitas *user* pada Gambar 3.55, yang telah terdaftar di dalam *server* VoIP.



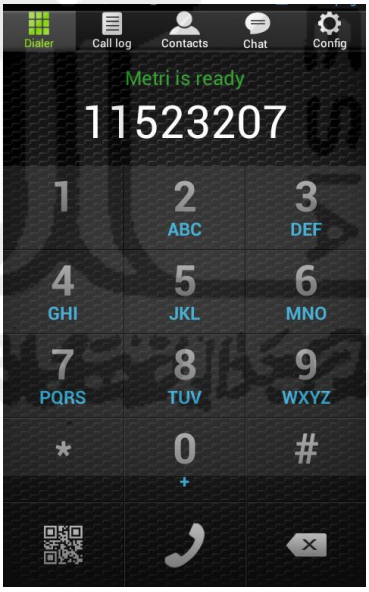
Gambar 3.55 Cara Pengisian Akun

2) Di menu paling bawah, terdapat pengaturan *codec*. Pilih *codec* "GSM", karena ia memiliki kualitas yang sama bagus dengan *codec* G.729, yang merupakan *codec* versi berbayar.

Gambar 3.56 Penentuan *Codec*

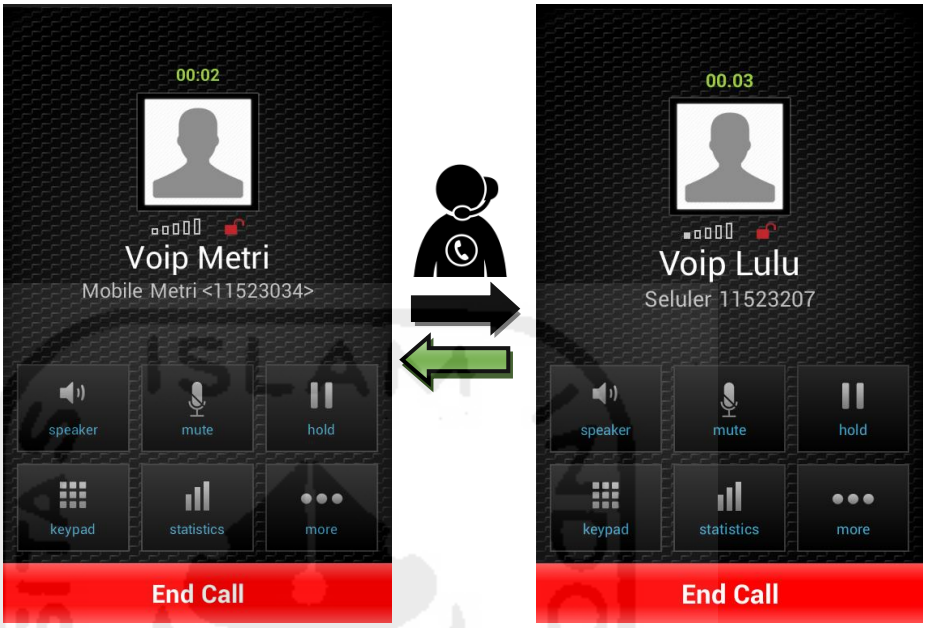
b. Uji Coba Penggunaan *server* VoIP pada *client user*.

1) Menjalankan terlebih dahulu VoIP *server Asterisk,* dengan perintah *#asterisk -r*.
2) Melakukan panggilan ke nomor yang telah terdaftar di *iax.conf*.



Gambar 3.57 *Calling User* lain

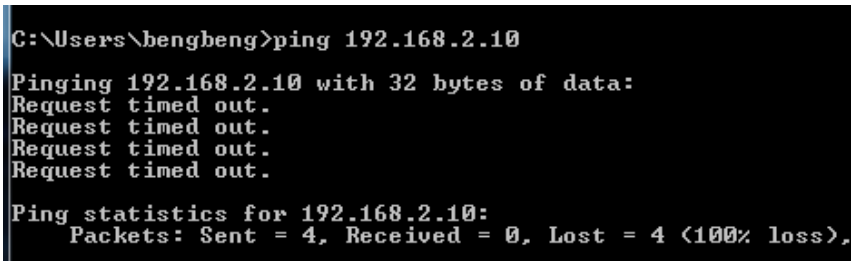
3) Panggilan terhubung ke *user* tujuan.



Gambar 3.58 Panggilan Terhubung ke *User* Tujuan

3.5. Pengecekan Jaringan Hasil Konfigurasi

Setelah melakukan konfigurasi untuk pembangunan sistem, maka selanjutnya akan dilakukan pengecekan terhadap kondisi jaringan yang ada. Topologi jaringan yang digunakan, dapat dilihat pada Gambar 3.4. Hal ini dilakukan, untuk mengetahui apakah jaringan telah berjalan sesuai yang diharapkan atau tidak. Berikut rangkaian hasil pengecekan yang dilakukan dari tiga sisi klien, yakni *user client Router1* yang berada **di luar** jaringan *server Router*, *client vlan10* dan *client vlan20* yang berada pada jaringan *server Router*. Informasi lebih lengkapnya terkait tahapan pengecekan jaringan hasil konfigurasi, terdapat pada Lampiran II. Tahapan dilakukan hingga pengecekan paket ICMP dari *client vlan10* (172.16.1.2) ke alamat *user client Router1*, selesai dilakukan.



Gambar 3.59 Pengecekan dari *Client Vlan10* (172.16.1.2) ke Alamat *User Client Router1*

Hasil pengecekan jaringan pada ketiga sisi klien yang telah ditunjukkan pada Lampiran II, menunjukkan bahwa hasil konfigurasi telah sesuai dengan konsep awal topologi VoIP, yang telah terbentuk pada tahap sebelumnya. Hal ini dibuktikan dengan, tertutupnya akses menuju *gateway vlan10* dari *user* lain, dimana aturan tersebut ditentukan melalui aturan *firewall*. Meskipun demikian, *user* yang berada dalam jaringan *vlan10*, tetap dapat berkomunikasi dengan *user* lain yang berada pada jaringan berbeda, asalkan dalam satu segmen (VLAN) yang sama. Hal ini dikarenakan, jalur akses dari *user client vlan10* ke *Router*, tetap dibiarkan terbuka, kecuali yang berasal dari dan menuju ke jaringan 192.168.2.0/24 (*ether1* pada *client Router1*) dan 172.16.2.0/24 (*vlan20* pada *server Router*).

3.6. Perintah Pengujian VoIP

Setelah melakukan pengecekan terhadap jaringan, maka tahap selanjutnya adalah melakukan pengujian terhadap *Quality of Service* dan keamanan jaringan. Pengujian ini dilakukan, untuk membantu tahap pengambilan data, agar penulis memperoleh perbedaan, dari penggunaan metode MPLS-VPN dan segmentasi VLAN (yang disertai pengaturan *firewall*), sebelum dan sesudah diterapkan. Berikut penjabaran kedua pengujian tersebut.

3.6.1 Pengujian *Quality of Service*

Pengujian ini dilakukan, untuk mengetahui kualitas transmisi data yang berlangsung dalam sistem. Parameter yang digunakan dalam pengujian ini adalah *throughput, packet loss, delay,* dan *jitter*. Parameter tersebut diukur secara manual dengan menggunakan *software wireshark*, dimana *software* tersebut berfungsi melakukan *monitoring* terhadap jaringan yang sedang berjalan. Besar *bandwidth* yang digunakan untuk pengujian ini adalah 100 kbps, 512 kbps, 1 Mbps, dan 3 Mbps. Berikut tata cara pengujian *Quality of Service* pada masing-masing parameter, yang dimulai dengan melakukan *capturing file*, untuk menangkap paket data dan suara yang lewat, agar dapat dilakukan perhitungan di tahap selanjutnya.

a) *Capturing File*

Untuk dapat melakukan perhitungan *throughput, packet loss, delay, dan jitter*, maka perlu dilakukan proses *capture file* terlebih dahulu, agar diperoleh besar nilai *variable* yang dibutuhkan. Dalam melakukan *capture file* ini, ada beberapa batasan yang perlu diperhatikan, untuk meminimalisir kesalahan pengujian VoIP, yakni lama pengamatan yang dilakukan berapa detik, aktivitas panggilan mulai dilakukan pada detik ke berapa, kemudian jenis paket yang tersaring apa saja. Hal ini perlu dilakukan karena, perbedaan selang waktu panggilan, dapat memengaruhi besar volume panggilan, dan pada akhirnya memengaruhi pula hasil akhir dari perhitungan parameter VoIP.

Untuk memenuhi kualifikasi di atas, maka dalam pengujian ini, penulis menggunakan durasi pengamatan selama 200 detik, aktivitas panggilan dilakukan sebelum pengamatan dimulai, dan paket yang tersaring adalah IAX2 (*port* 4569) dan ICMP.

Berikut contoh perintah yang digunakan pada Gambar 3.60, untuk melakukan *capturing file* pada jaringan VoIP, yang menggunakan pengaturan *routing table* biasa.

Gambar 3.60 Perintah *Capturing File*

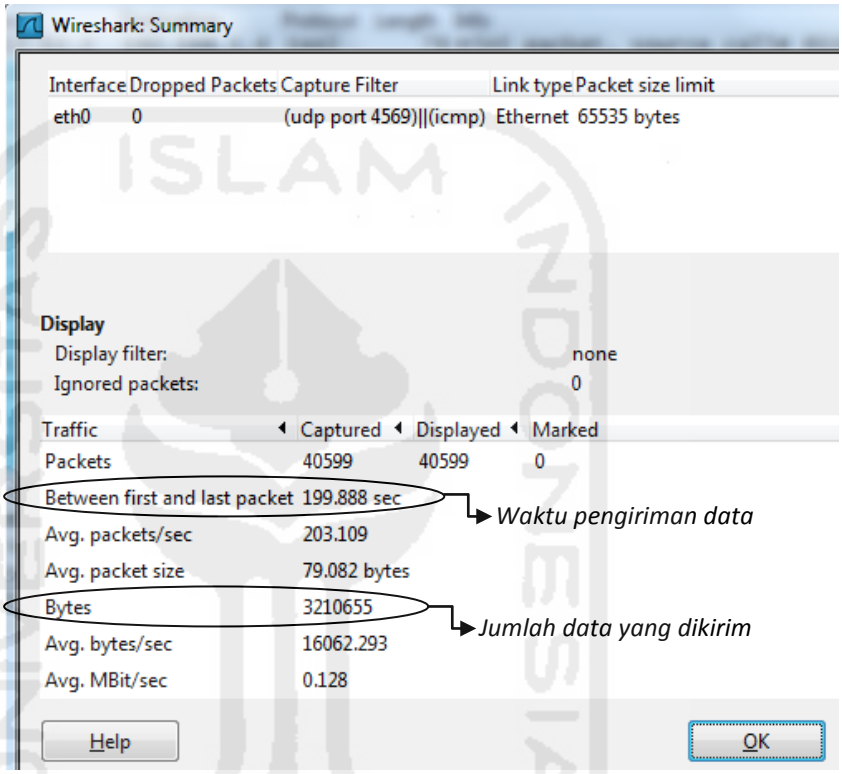
b) Perhitungan *Throughput*

Throughput digunakan, untuk menghitung waktu sebenarnya dari aktivitas *download* yang berjalan. Parameter ini digunakan untuk mengukur kualitas jaringan, dengan satuan *kilobit per second* (kbps). Semakin tinggi nilai *throughput* yang diperoleh, maka nilai *delay* akan semakin rendah, sehingga kualitas jaringan menjadi lebih baik (Anggita dkk, 2012). Berikut Rumus 3.1, yang digunakan untuk melakukan perhitungan *throughput*.

Rumus 3.1 *throughput* (Agustia, 2011)**:**

$$Throughput = \frac{\text{Jumlah data yang dikirim (bytes)}}{\text{Waktu pengiriman data (sec)}}$$

Berikut letak *variable* data yang digunakan pada Gambar 3.61, untuk melakukan perhitungan *throughput* pada Rumus 3.1. Contoh data pada Gambar 3.61, diperoleh dari aktivitas *capture file*, yang dilakukan pada tahap sebelumnya, yang dibuka dengan menggunakan aplikasi *wireshark*, di dalam menu "*Statistics*> *Summary*".



Gambar 3.61 Letak Variabel *Throughput*

c) Perhitungan *Packet Loss*

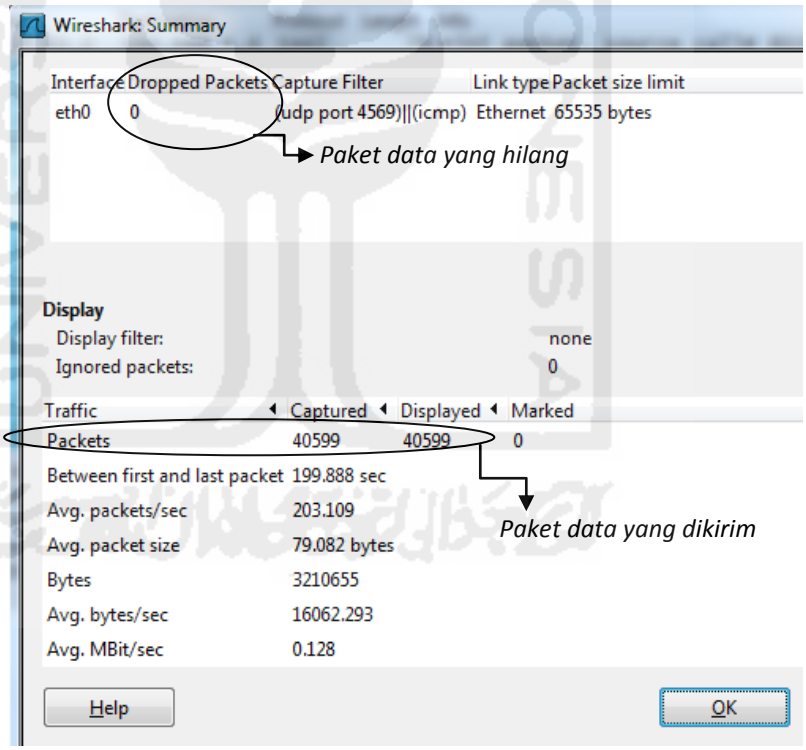
Packet loss adalah parameter yang digunakan untuk menghitung paket data yang hilang ketika proses transmisi terjadi. Parameter ini, memberikan pengaruh yang besar terhadap *IP Telephony*, dimana apabila terjadi *packet loss* dalam jumlah tertentu, akan menyebabkan interkoneksi TCP menjadi melambat. Satuan yang digunakannya adalah persen (%). Berikut Rumus 3.2, yang digunakan untuk melakukan perhitungan *packet loss*.

$$\text{Paket data yang diterima} = \text{Paket data yang dikirim} - \text{Paket data yang hilang}$$

Rumus 3.2 ***packet loss*** (Agustia, 2011)**:**

$$Packet\ loss = \left(\frac{\text{Paket data yang dikirim} - \text{Paket data yang diterima}}{\text{Paket data yang dikirim}}\right) x100\%$$

Berikut letak *variable* data yang dibutuhkan pada Gambar 3.62, untuk melakukan perhitungan *packet loss* pada Rumus 3.2.



Gambar 3.62 Letak Variabel *Packet Loss*

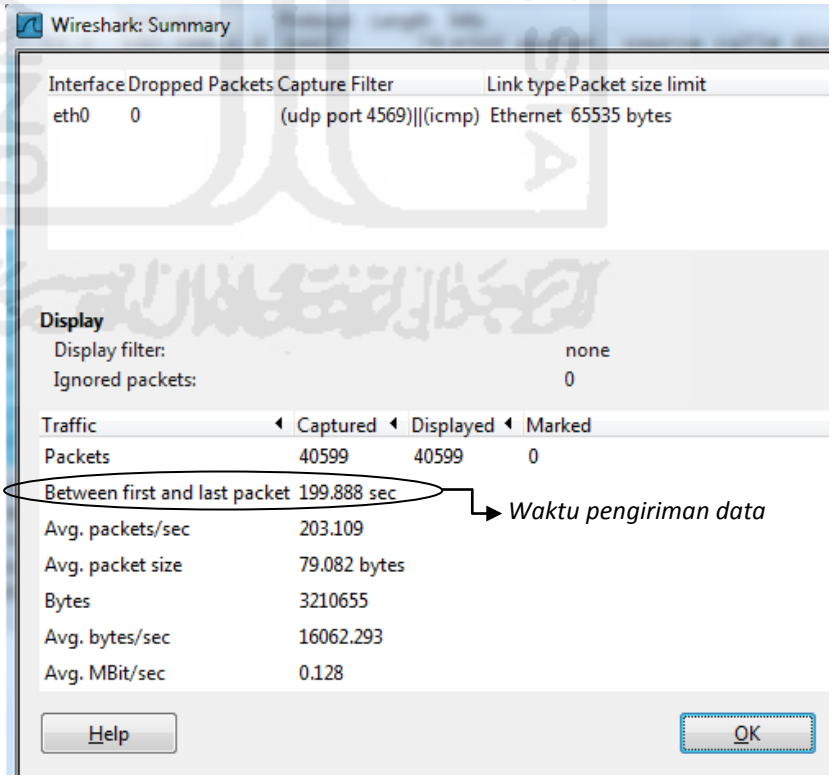
d) Perhitungan *Delay*

Delay adalah parameter waktu yang dibutuhkan sebuah paket, dari saat paket tersebut dikirim sampai diterima. Satuan yang digunakan adalah *milliseconds* (ms). Parameter ini penting digunakan untuk menentukan kualitas VoIP. Semakin besar *delay*, berarti semakin rendah kualitas VoIP yang dihasilkan. Berikut Rumus 3.3, yang digunakan untuk melakukan perhitungan *delay*.

Rumus 3.3 *delay* (Agustia, 2011)**:**

$$Delay = \frac{\text{Waktu pengiriman data (sec)}}{\text{Paket data yang diterima}}$$

Berikut letak *variable* data yang digunakan pada Gambar 3.63, untuk melakukan perhitungan *delay* pada Rumus 3.3 di atas. Untuk *variable* "Paket data yang diterima", cara perolehan hasilnya, telah dijelaskan pada perhitungan *packet loss*.



Gambar 3.63 Letak Variabel *Delay*

e) Perhitungan *Jitter*

Jitter adalah parameter QoS yang digunakan, untuk menghitung perbedaan waktu kirim dan sampainya paket data ke tujuan. Satuan yang digunakan adalah *milliseconds* (ms). Untuk meminimalkan nilai *jitter* ini, diusahakan pengiriman paket dilakukan melalui jalur yang sama, dan jangan sampai terjadi *packet loss*. Berikut rumus-rumus yang digunakan, untuk menghitung nilai *jitter*.

1) Mencari nilai *delay*

Pencarian nilai *delay,* diperoleh dari paket yang tertangkap. Pencarian ini, digunakan untuk memperoleh nilai "*Total variasi delay*", yang digunakan pada langkah selanjutnya. Rumus yang digunakan untuk pencarian *delay* ini, berbeda dengan rumus pencarian *delay* pada langkah sebelumnya. Pencarian *delay* pada tahapan ini, menggunakan informasi data yang diperoleh dari paket ICMP (*Internet Control Message Protocol*), yakni paket yang digunakan untuk memberikan laporan terkait kondisi jalur komunikasi, antara alamat IP asal dengan alamat IP tujuan.

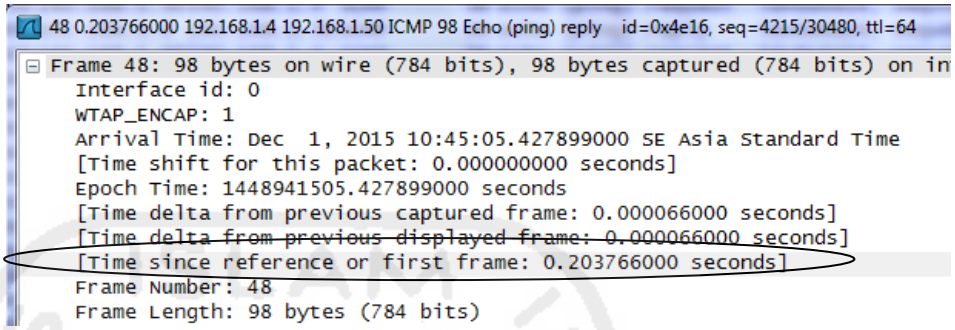
Berikut Rumus 3.4, yang digunakan untuk mencari nilai *delay* (Kristianto, 2012)**.** Banyaknya *delay* yang dibutuhkan, cukup hingga tiga kali perhitungan (s/d n=3).

Rumus 3.4 *delay* untuk perhitungan *jitter*:

$$\boldsymbol{Delay}_{(1,2,3)} = Time\ since\ reference\ or\ first\ frame\ (Reply) - Time\ since\ reference\ or\ first\ frame\ (Request)$$

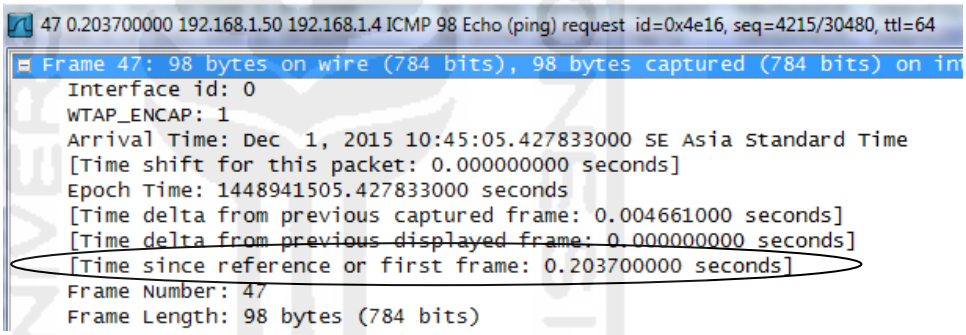
Berikut letak *variable* yang dibutuhkan pada Gambar 3.64 dan 3.65, untuk menghitung nilai *delay* pada Rumus 3.4.

a) Letak nilai *variable reply* berada.



Gambar 3.64 Letak Variabel *Reply*

b) Letak nilai *variable request* berada.



Gambar 3.65 Letak Variabel *Request*

2) Pencarian total variasi *delay*

Pencarian total variasi *delay*, digunakan untuk memperoleh hasil penjumlahan dari varian *delay*, yang perhitungannya telah dilakukan pada langkah sebelumnya. Berikut Rumus 3.5, total variasi *delay* untuk perhitungan *jitter*.

Rumus 3.5 total variasi *delay* untuk perhitungan *jitter*:

$$\text{Total variasi } delay =$$

$$(Delay_{(2)} - Delay_{(1)}) + (Delay_{(3)} - Delay_{(2)}) + \cdots + (Delay_{(n)} - Delay_{(n-1)})$$

3) Tahap akhir perhitungan *jitter*

Pada tahap ini, hasil penjumlahan dari *variable* "*Total variasi delay*", dijadikan dalam bentuk nilai mutlak (Matualaga dkk, 2014). Hasil nilai dari *variable* tersebut, kemudian dibagi dengan *variable* "*Total paket data yang diterima*", dimana *variable* tersebut telah dilakukan pengurangan nilai satu sebelumnya. Untuk mengetahui asal nilai dari *variable* "*Total paket data yang diterima*", dapat dilihat pada rumus perhitungan *packet loss*. Dari data yang diperoleh, berikut Rumus 3.6, perhitungan nilai total *jitter* keseluruhan.

Rumus 3.6 total *jitter* (Agustia, 2011)**:**

$$Jitter = \frac{|\text{Total variasi } delay|}{\text{Total paket data yang diterima} - 1}$$

3.6.2 Pengujian Keamanan Jaringan

Pengujian ini dilakukan, untuk mengetahui tingkat keamanan jaringan, dalam menghadapi beberapa serangan *cybercrime*, yakni peracunan jaringan (ARP *poisoning*), pencurian *Voice VLAN ID* (VLAN *hopping*), penyadapan (*eavesdropping*), peniruan IP (IP *spoofing*), dan aktivitas pelumpuhan target dengan mengirimkan banyak paket ICMP ke IP target (*ping flooding*).

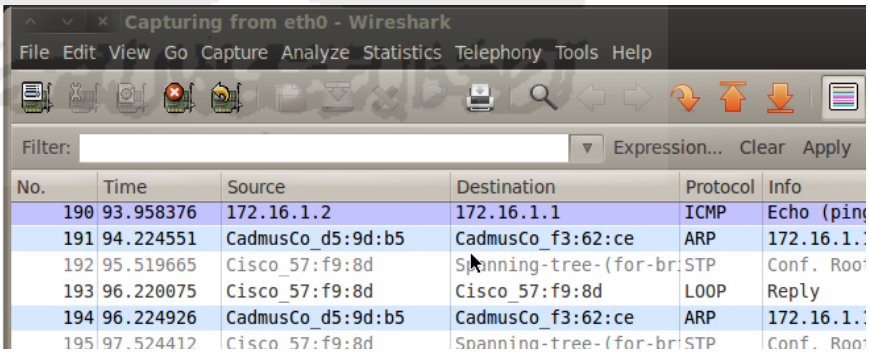
Dalam pengujian ini, penulis tidak membatasi jumlah kecepatan *bandwidth*, karena tahap ini lebih fokus membahas masalah keamanan sistem. Berikut beberapa teknik serangan *cybercrime* yang digunakan.

a) ARP *Poisoning*

ARP *Poisoning* adalah serangan *cybercrime* yang dijalankan di dalam *layer* 2 (*data link*), dilakukan dengan cara, mengarahkan lalu lintas data dari klien target ke komputer penyerang. Hal ini dapat terjadi karena, komputer penyerang memberi tahu kepada klien, bahwa MAC *address* komputer penyerang adalah MAC *address* dari komputer *server*, sehingga klien dapat dikelabui dari teknik ini. Jika cara ini berhasil, maka penyerang dapat memasuki jaringan klien target, untuk mengirimkan paket yang berisi informasi palsu ke klien tersebut.

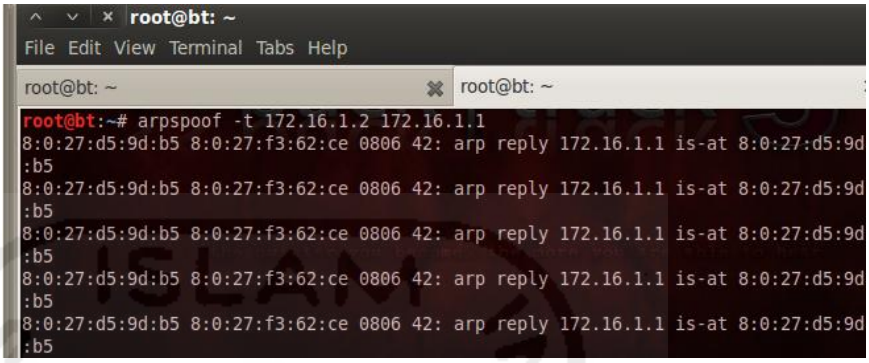
Langkah-langkah yang ditampilkan pada tahap ini, adalah hasil dari pengujian dengan menggunakan topologi pada Gambar 3.4, dimana pengujian dilakukan dari jaringan VLAN 10 (data), menuju klien yang juga berada pada jaringan VLAN 10. Berikut langkah-langkah yang perlu dilakukan, untuk melakukan ARP *poisoning.*

1) Melakukan *capturing file*, melalui aplikasi *wireshark*, untuk memperoleh informasi *user* yang terhubung dan IP *gateway* yang digunakannya berapa. Berikut tampilan *capturing file* dan perintah ARP *poisoning*, dari jaringan VLAN 10 (data), pada kondisi sistem menggunakan kedua metode, yakni MPLS-VPN dan segmentasi VLAN-*Firewall.*



Gambar 3.66 *Capturing File* untuk Mencari Informasi Target yang Terhubung

2) Memasukkan perintah *#arpspoof [-t target] [IP target] [IP gateway]*, dalam terminal *Backtrack*, untuk mulai melakukan ARP *poisoning*.



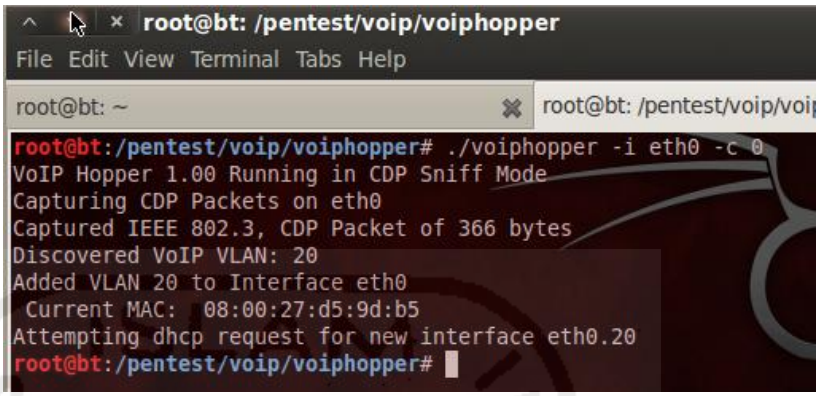
Gambar 3.67 Perintah ARP *Poisoning*

b) VLAN *Hopping*

VLAN *hopping* adalah jenis serangan yang digunakan untuk melakukan *discovery packet* dan pencurian VVID (*Voice* VLAN ID), yang berlangsung di *layer* 2 (*data link*) (Mason, 2011). Teknik penyerangan ini dilakukan, karena dewasa ini banyak perusahaan biasa menjalankan komunikasi VoIP, dengan menggunakan jalur VLAN khusus. Untuk dapat melakukan serangan VLAN *hopping*, dapat digunakan perintah *#./voiphopper –i [interface] –c {0|1|2}*. Dengan perintah tersebut, maka akan diperoleh informasi VVID yang ada di dalam jaringan.

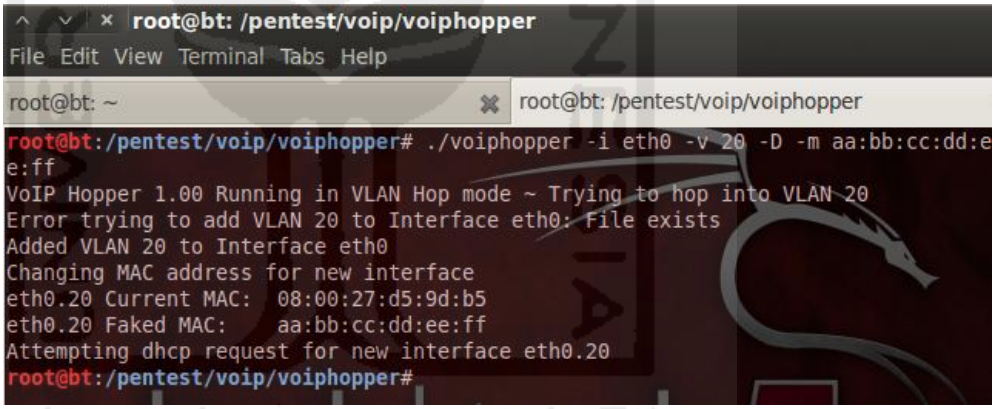
Jika langkah di atas berhasil, maka perintah dapat dilanjutkan ke serangan MAC VLAN. Perintah yang digunakan untuk melakukan serangan tersebut, yakni *#./voiphopper –i [interface] -v [VVID] –D –m [MAC palsu]*. Tujuan dari serangan MAC VLAN tersebut, digunakan untuk menyembunyikan MAC *address* asli penyerang. Berikut tampilan perintah VLAN *hopping*, dari jaringan VLAN 20 (suara) menuju klien yang juga berada pada jaringan VLAN 20. Topologi yang digunakan untuk menampilkan tahapan ini adalah topologi pada Gambar 3.4.

1) Tampilan terminal *Backtrack* ketika perintah *discovery packet* VLAN *hopping* berhasil dijalankan.



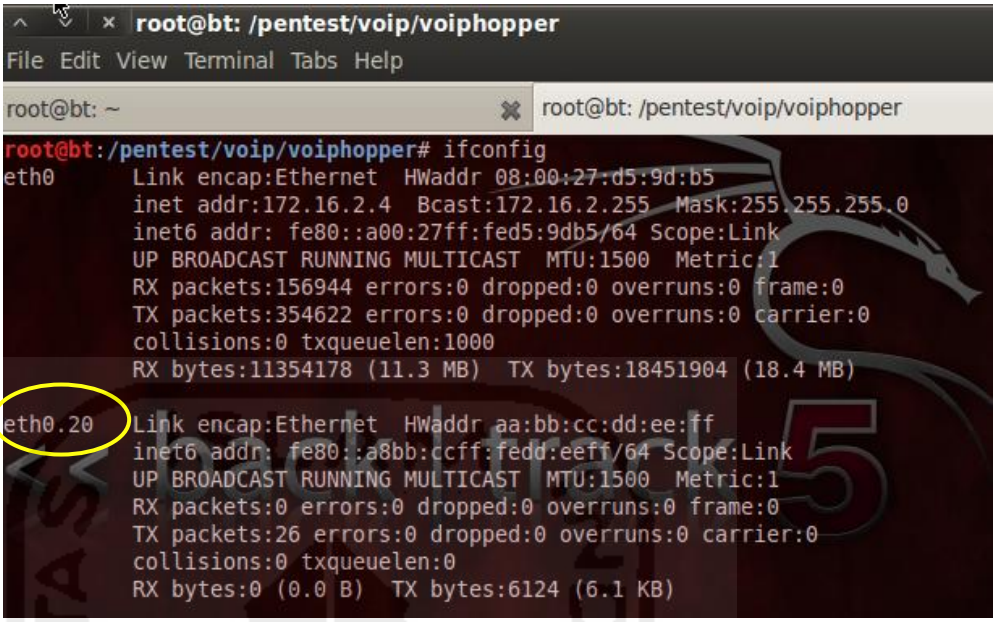
Gambar 3.68 *Discovery Packet* VLAN *Hopping*

2) Tampilan terminal *Backtrack* ketika perintah serangan MAC *address* VLAN berhasil dijalankan.



Gambar 3.69 Serangan MAC *Address* VLAN

3) Melakukan pengecekan terhadap *interface* jaringan, apakah serangan MAC *address* VLAN pada tahap sebelumnya, benar-benar telah berhasil dilakukan atau belum.



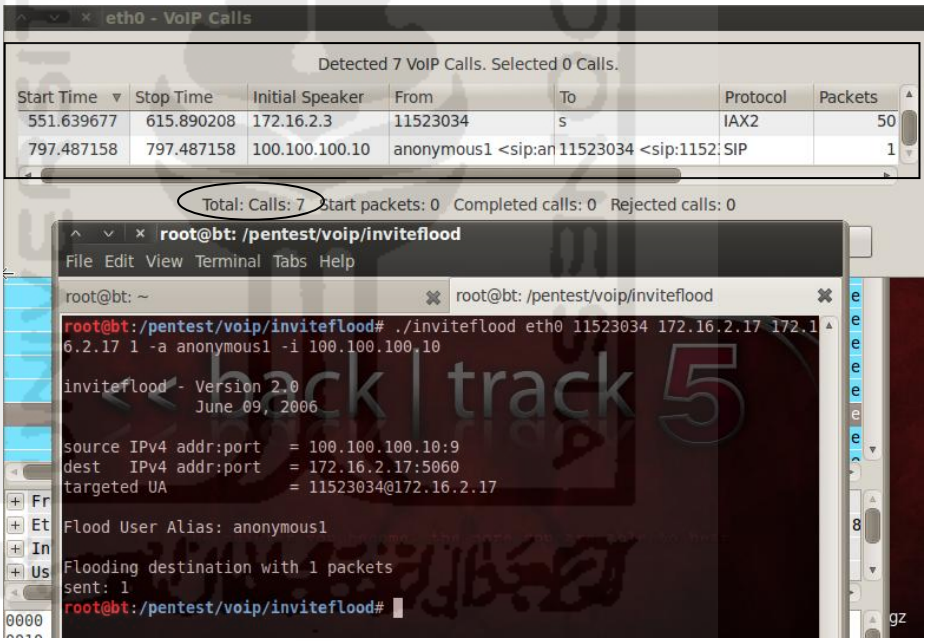
Gambar 3.70 Perubahan *Ethernet*, setelah VLAN *Hopping* Dijalankan

c) IP *Spoofing*

Teknik serangan ini bertujuan untuk mengelabui komputer korban, dengan memalsukan alamat IP sumber, agar identitas asli penyerang tidak diketahui (Satria, 2014). Teknik ini berjalan pada pada *layer* 3 (*network*), dengan meniru koneksi lain, dalam hal ini contohnya adalah koneksi VoIP, dimana protokol yang digunakan dapat berupa SIP atau IAX2. Untuk dapat melakukan serangan IP *Spoofing*, dapat digunakan perintah *#./inviteflood [interface] [user target] [domain target] [IP target] [number packet] –a [alias] –i [IP address palsu] –D [port yang digunakan]*.

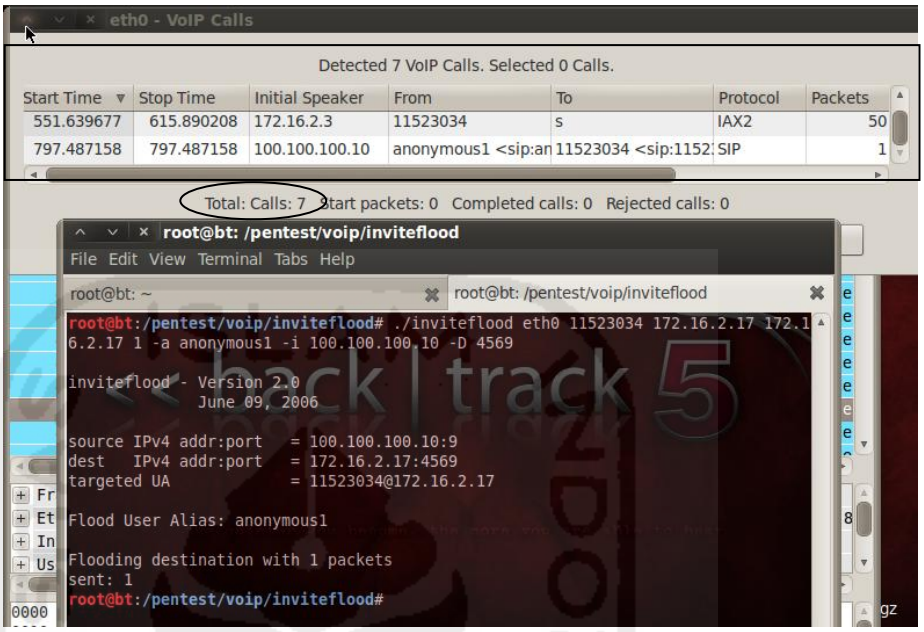
Berikut tampilan perintah IP *spoofing*, dari jaringan VLAN 20 (suara) menuju klien yang juga berada pada jaringan VLAN 20. Topologi jaringan yang digunakan untuk menampilkan tahapan ini, adalah topologi yang terdapat pada Gambar 3.4. Untuk membuktikan apakah perintah IP *spoofing* dapat berjalan dengan baik atau tidak pada sistem, maka pengujian akan dilakukan terlebih dahulu dengan menggunakan protokol SIP secara *default*, baru setelah itu dilanjutkan dengan menggunakan protokol IAX2 (4569), untuk mengetahui perbedaannya.

1) Tampilan terminal *Backtrack* ketika perintah IP *spoofing* dengan menggunakan protokol SIP secara *default*, berhasil dijalankan.



Gambar 3.71 IP *Spoofing* dengan Menggunakan Protokol SIP secara *Default*

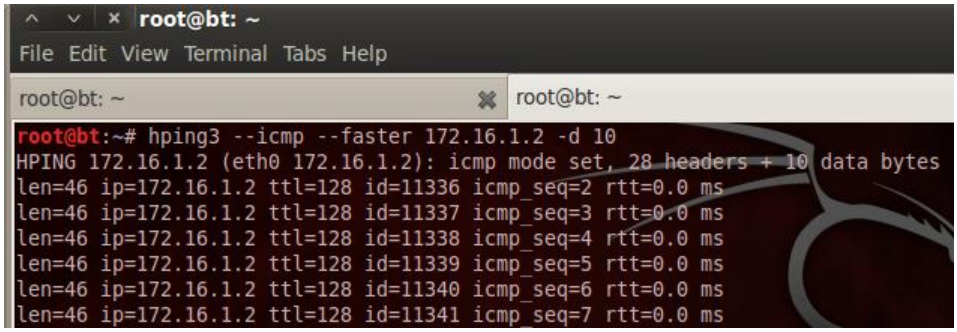
2) Tampilan terminal *Backtrack* ketika perintah IP *spoofing* dengan menggunakan protokol IAX2 (4569), gagal dijalankan.



Gambar 3.72 IP *Spoofing* dengan Menggunakan Protokol IAX2 (4569)

d) *Ping Flooding*

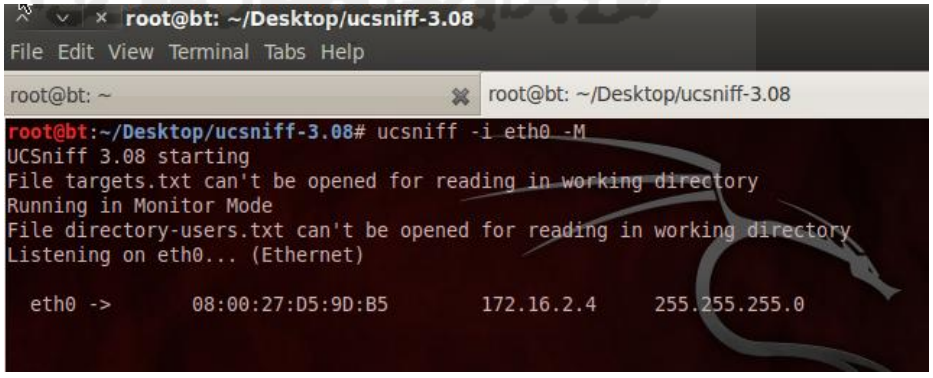
Teknik serangan ini berjalan pada *layer* 3 (*network*). Cara kerjanya adalah dengan membanjiri sistem target, dengan mengirim paket ICMP dalam jumlah tak terbatas. Teknik ini dapat membuat sistem target *down*, sebab sistem tidak dapat menampung lagi paket-paket yang dikirim penyerang. Dengan dampak tersebut, teknik ini digunakan untuk menguji kualitas sistem, apakah sistem akan tetap berjalan dengan normal, ketika diberikan beban yang berat atau justru sebaliknya. Untuk dapat melakukan serangan *ping flooding*, dapat digunakan perintah *#hping3 --icmp --faster [IP target] –d [data size]*. Berikut tampilan perintah *ping flooding*, dari VLAN 10 (data) menuju klien VLAN 10 pada Gambar 3.73, dengan menggunakan topologi jaringan yang terdapat pada Gambar 3.4.



Gambar 3.73 Perintah *Ping Flooding*

e) *Eavesdropping*

Serangan ini dilakukan pada *physical layer*, dengan cara menangkap paket data apa pun yang mengalir di dalam jaringan VoIP. Paket data yang tertangkap, dapat berupa *file audio, file monitoring*, alamat IP dan nomor *extension user* VoIP, yang berasal dari protokol *transport* RTP atau IAX2. Dengan menggunakan protokol tersebut, penyerang dapat melakukan penyadapan pada komunikasi, yang berlangsung di jaringan VoIP. Untuk dapat melakukan serangan *eavesdropping*, dapat digunakan perintah *#ucsniff –i [interface] –M*. Berikut tampilan perintah serangan tersebut, yang dilakukan dari jaringan VLAN 20 (suara) pada Gambar 3.74. Langkah pengujian yang ditampilkan pada tahapan ini, menggunakan topologi yang terdapat pada Gambar 3.4.



Gambar 3.74 Perintah *Eavesdropping*

3.7. Metode Analisis

Metode yang digunakan sebagai tahapan untuk melakukan analisis data. Tahapan tersebut mencakup statistik deskriptif, statistik induktif, dan hipotesis. Metode analisis ini, adalah merupakan kelanjutan dari tahap pengambilan data sebelumnya. Berikut penjabaran ketiga tahapan metode analisis tersebut.

3.7.1 Statistik Deskriptif

Metode yang bersifat memberikan gambaran, terhadap hasil uji coba yang telah dilakukan. Metode statistik deskriptif ini, berkaitan dengan aktivitas pengumpulan, peringkasan, dan penyajian data, sehingga data yang ditampilkan, dapat memberikan informasi secara lebih efektif dan mudah dipahami. Contoh model presentasi yang dilakukan dengan metode ini, adalah dengan menampilkan data berupa tabel, histogram, atau *pie chart*.

3.7.2 Statistik Induktif

Metode yang berkaitan dengan analisis data, yang pada akhirnya bertujuan untuk menarik kesimpulan. Hasil analisis metode tersebut, digunakan untuk mengetahui perbedaan yang diperoleh dari, pengujian penggunaan metode MPLS-VPN dengan segmentasi VLAN-*firewall*. Contoh model pengujian yang digunakan dalam metode ini, antara lain penggunaan statistik, pengujian hipotesis, maupun regresi dan korelasi.

3.7.3 Hipotesis

Pada pengerjaan tugas akhir ini, penulis membuat dugaan sementara, bahwa dengan menerapkan metode segmentasi VLAN pada jaringan MPLS-VPN, yang disertai dengan pengaturan *firewall*, maka serangan apapun menuju *data network*, tidak akan mengganggu lalu lintas data dan suara. Dengan demikian penggabungan metode tersebut, diharapkan dapat meningkatkan jaringan VoIP, dari segi kualitas transmisi dan keamanan data.