

# SKRIPSI

## DESAIN DAN IMPLEMENTASI APLIKASI MONITORING TRAFFIC PADA JARINGAN VOIP



Disusun Oleh

**ARIF NARENDRA**

**07. 12. 519**



**JURUSAN TEKNIK ELEKTRO S-1  
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL MALANG  
2011**

CONTENTS

CONTENTS (CONTENTS) (CONTENTS) (CONTENTS) (CONTENTS) (CONTENTS)

CONTENTS  
CONTENTS  
CONTENTS

CONTENTS (CONTENTS) (CONTENTS) (CONTENTS) (CONTENTS) (CONTENTS)

**LEMBAR PERSETUJUAN**

**DESAIN DAN IMPLEMENTASI APLIKASI MONITORING  
TRAFFIC PADA JARINGAN VOIP**

**SKRIPSI**

*Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh  
Gelar Sarjana Teknik Komputer dan Informatika Strata Satu (S-1)*

**Disusun oleh :**

**ARIF NARENDRA**

**07. 12. 519**

**Mengetahui,**

**Ketua Jurusan Teknik Elektro S-1**



**Ir. Yusuf Ismail Nakhoda, MT**  
**NIP.Y.1018800189**

**Diperiksa dan Disetujui**

**Dosen Pembimbing I**

**Dr. Eng. Aryuanto, ST, MT.**  
**NIP.Y.1030800417**

**Dosen Pembimbing II**

**Sotyohadi, ST.**  
**NIP.Y.1039700309**

**JURUSAN TEKNIK ELEKTRO S-1  
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL MALANG**

**2011**

## ABSTRAK

**Nama : Arif Narendra, NIM : 07.12.519, Jurusan : Teknik Komputer dan Informatika S-1, Fakultas Teknologi Industri, Institut Teknologi Nasional Malang, Judul Skripsi : " DESAIN DAN IMPLEMENTASI APLIKASI MONITORING TRAFFIC PADA JARINGAN VOIP"**

**Dosen Pembimbing : I Dr. Eng. Aryuanto,ST, MT.**

**II Sotyohadi,ST.**

---

*Perkembangan teknologi informasi menyebabkan semakin banyaknya pengguna Telekomunikasi dan jaringan komputer, hal ini menuntut para administrator jaringan untuk meningkatkan pengelolaan pada jaringan tersebut. Voice Over Internet Protocol (VoIP) merupakan terobosan baru dibidang telekomunikasi yang mengusung teknologi IP (internet protocol) sebagai media penghantaran informasi. Ini berarti, informasi yang berupa suara (sinyal analog) akan dirubah menjadi sinyal digital untuk dapat dilewatkan jaringan IP. Untuk dapat mengetahui alur komunikasi pada jaringan VoIP, maka diimplementasikan sebuah client yang dapat melakukan panggilan ke tujuan melalui server VoIP(IP PBX). Sedangkan aplikasi Monitoring Traffic pada jaringan VoIP dibuat untuk memudahkan administrator dan user untuk memonitoring traffic yang terjadi pada jaringan VoIP tersebut, Dengan Tanpa Membuka atau melihat keseluruhan fitur Aplikasi Trixbox melalui Webmin yang tersedia terlebih dahulu.*

***Kata kunci : Jaringan VoIP, aplikasi Monitoring traffic,IP PBX.***

## KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, atas karunia yang telah dilimpahkanNya sehingga penulis dapat menyelesaikan tugas akhir dengan judul *DESAIN DAN IMPLEMENTASI APLIKASI MONITORING TRAFFIC PADA JARINGAN VOIP*.

Selanjutnya pada kesempatan ini penulis juga menyampaikan rasa terimakasih yang sebesar-besarnya kepada pihak-pihak yang telah banyak membantu penulis selama penyusunan tugas akhir, diantaranya :

1. Bapak Ir. Yusuf Ismail Nahkoda, MT selaku Ketua Jurusan Teknik Elektro S-1 ITN Malang.
2. Bapak Dr. Aryunto Soetedjo, ST, MT selaku Sekertaris Jurusan Teknik Elektro S-1 ITN Malang dan pengusul serta penyedia ruang Skripsi.
3. Bapak Dr. Aryunto Soetedjo, ST, MT selaku Dosen Pembimbing I
4. Bapak Sotyohadi,ST selaku Dosen Pembimbing II
5. Bapak Ahmad Faisol, selaku Dosen Wali.
6. Kedua orang tua yang telah memberikan dukungan untuk selalu berdoa, berusaha dan nasehat yang telah diberikan sampai saat ini.
7. Seluruh dosen dan pegawai ITN Kampus 2 Malang.
8. Semua teman-teman Laboratorium Jaringan Komputer dan CISCO.
9. Semua pihak yang telah membantu penulis dalam menyelesaikan skripsi ini yang tidak bisa penulis sebutkan satu persatu..

Tentunya laporan tugas akhir ini masih memiliki banyak kekurangan dan kelemahan dalam penyusunannya. Oleh karena itu penulis mengharapkan kritik dan saran yang membangun demi kesempurnaan penyusunan tugas akhir ini. Besar harapan penulis laporan tugas akhir ini dapat bermanfaat bagi semua pihak.

Malang, Agustus 2011

Penulis

## DAFTAR ISI

<b>HALAMAN JUDUL</b> .....	i
<b>HALAMAN PERSETUJUAN</b> .....	ii
<b>ABSTRAK</b> .....	iii
<b>KATA PENGANTAR</b> .....	iv
<b>DAFTAR ISI</b> .....	vi
<b>DAFTAR GAMBAR</b> .....	x
<b>DAFTAR TABEL</b> .....	xiii

### **BAB I PENDAHULUAN**

1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Tujuan Penelitian.....	3
1.4 Batasan Masalah.....	3
1.5 Metode Penelitian.....	3
1.6 Sistematika Penulisan.....	4

### **BAB II LANDASAN TEORI**

2.1 Konsep Telekomunikasi .....	6
2.2 Jaringan Komputer .....	6
2.2.1 Jenis-jenis Jaringan Komputer .....	7
2.2.2 Intranet .....	8
2.2.3 Media Implementasi Jaringan Komputer .....	8
2.2.4 Topologi Jaringan Komputer .....	16
2.2.5 OSI Layer .....	21
2.2.6 Hubungan Komunikasi Data dan Voice .....	26
2.3 VoIP (Voice Over Internet Protocol) .....	27
2.3.1 Konsep Dasar VoIP .....	27
2.3.2 Latar Belakang Teknologi VoIP .....	28

2.3.3 Kelebihan VoIP .....	28
2.3.4 Kekurangan VoIP .....	29
2.3.5 Cara Kerja VoIP .....	30
2.3.6 Protokol Penunjang VoIP .....	30
2.3.7 Format Paket VoIP .....	32
2.3.8 Komponen VoIP .....	33
2.3.9 Codec.....	35
2.4 SIP .....	36
2.4.1 Fungsi Session Initiation Protocol (SIP) .....	36
2.4.2 Susunan Protokol SIP .....	37
2.4.3 Komponen SIP .....	38
2.4.4 SIP Request .....	40
2.4.5 SIP Response.....	41
2.4.6 SIP Header.....	41
2.4.7 Cara Kerja Protokol SIP .....	42
2.5 SNMP .....	43
2.5.1 Konsep Dasar SNMP .....	44
2.5.2 Arsitektur SNMP .....	46
2.5.3 Entitas SNMP .....	47
2.6 Sistem Operasi Linux .....	49
2.6.1 Linux Trixbox .....	49
2.6.2 Fitur Linux Trixbox.....	50
2.7 Pemrograman Borland Delphi 7 .....	50

### **BAB III ANALISA DAN DESAIN SISTEM**

3.1 Analisa Sistem.....	52
3.1.1 Perintah yang Digunakan Untuk Mengakses Agent .....	53
3.1.2 Mekanisme Pendefinisian MIB .....	53
3.2 Desain Sistem .....	55
3.2.1 Desain Aplikasi Monitoring Traffic .....	57
3.2.1.1 Desain Proses Monitoring Traffic .....	57



3.2.1.2 Desain Proses Request Nilai Traffic .....	58
3.2.1.3 Desain Proses Konversi Nilai.....	58
3.2.1.4 Desain Tampilan Grafik .....	58
3.2.2 Desain Menu PC Information .....	59
3.2.2.1 Request Informasi Software Yang Terinstall .....	59
3.2.2.2 Menampilkan Informasi Software Yang Terinstall.....	59
3.2.3 Desain Menu Trace SNMP .....	60
3.2.3.1 Request Nilai Variabel OID .....	60
3.2.3.2 Menampilkan Informasi OID .....	60
3.2.4 Diagram Blok .....	60
3.2.5 Flowchart.....	62

## **BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM**

4.1 Implementasi Sistem .....	66
4.2 Aplikasi Monitoring Traffic Jaringan VoIP .....	67
4.2.1 Instalasi Server VoIP.....	67
4.2.1.1 Alat dan Bahan .....	67
4.2.1.2 Instalasi Linux .....	67
4.2.2 Konfigurasi Server .....	73
4.2.2.1 Instalasi Modul .....	74
4.2.2.2 Konfigurasi SIP Extension .....	76
4.2.2.3 Tampilan Menu Aplikasi Monitoring Traffic .....	78
4.3 Implementasi Jaringan VoIP dan Pengujian Sistem .....	79
4.4 Pengujian Berdasarkan Topologi Jaringan VoIP Yang Sederhana.	81
4.4.1 Pengujian Pada Saat Terjadi Komunikasi Antara Extension User 101 dengan Extension User 102 .....	81
4.4.2 Pengujian Pada Saat Terjadi Komunikasi Antara Extension User 101 dengan Extension User 102 dan Extension User 201 dengan Extension User 204.....	83
4.4.3 Pengujian Pada Saat Terjadi Komunikasi Antara Extension	

User 101 dengan Extension User 102 ,Extension User 201 dengan Extension User 204 dan Extension User 201 dengan Extension User 204 .....	85
4.4.4 Penyimpanan Log Trafic Ke Database.....	87
4.4.4.1 Koneksi Database Dari Delphi 7 ke Ms Access.....	87
4.4.4.2 Koneksi Melalui ODBC Data Administrator .....	88

## **BAB V PENUTUP**

5.1 Kesimpulan.....	90
5.2 Saran.....	91

## **DAFTAR PUSTAKA**

## DAFTAR GAMBAR

Gambar 2.1 Sistem Telekomunikasi .....	6
Gambar 2.2 Kabel UTP .....	10
Gambar 2.3 Kabel STP.....	13
Gambar 2.4 Kabel Coaxial .....	14
Gambar 2.5 <i>Network Interface Card</i> .....	16
Gambar 2.6 Topologi Jaringan Bintang .....	17
Gambar 2.7 Topologi Jaringan Cincin .....	19
Gambar 2.8 Topologi Jaringan Pohon.....	20
Gambar 2.9 Topologi Jaringan Bus.....	21
Gambar 2.10 OSI Layer .....	21
Gambar 2.11 Gambaran VoIP .....	27
Gambar 2.12 Cara Kerja VoIP .....	30
Gambar 2.13 Format Paket VoIP .....	33
Gambar 2.14 Cara Kerja Protokol SIP .....	42
Gambar 2.15 Alur Paket SNMP .....	46
Gambar 2.16 Manajer SNMP Tradisional.....	48
Gambar 3.1 Desain Sistem .....	56
Gambar 3.2 Desain Menu Aplikasi <i>Monitoring Traffic</i> Jaringan VoIP .....	57
Gambar 3.3 Desain Tampilan Grafik .....	58
Gambar 3.4 Desain Tampilan Menu <i>PC Information</i> .....	59
Gambar 3.5 Diagram Blok Sistem .....	61
Gambar 3.7 Flowchart Komunikasi <i>Antar User</i> .....	63
Gambar 3.8 Flowchart Aplikasi <i>Monitoring Traffic</i> .....	64
Gambar 4.1 Tampilan awal <i>booting</i> Trixbox .....	68
Gambar 4.2 Menu <i>Selecting Keyboard</i> .....	68
Gambar 4.3 Menu <i>Selecting Time Zone</i> .....	69
Gambar 4.4 Menu <i>Setting password root</i> .....	69
Gambar 4.5 Tampilan <i>install starting</i> .....	70

Gambar 4.6 Proses Instalasi .....	70
Gambar 4.7 Proses <i>installing bootloader</i> .....	70
Gambar 4.8 Tampilan <i>Bootloader Linux Trixbox</i> .....	71
Gambar 4.9 Tampilan <i>Start Up Linux</i> .....	71
Gambar 4.10 Konfigurasi <i>Network</i> .....	72
Gambar 4.11 <i>Edit Device</i> .....	72
Gambar 4.12 Pemilihan Ethernet .....	73
Gambar 4.13 Pemberian alamat IP .....	73
Gambar 4.14 Tampilan <i>address</i> bar pada browser .....	74
Gambar 4.15 Tampilan awal web konfigurasi trixbox.....	74
Gambar 4.16 Kotak dialog <i>login system administration</i> .....	75
Gambar 4.17 Halaman <i>Configuration and Administration</i> .....	75
Gambar 4.18 Tampilan Halaman PBX Setting .....	76
Gambar 4.19 Halaman <i>Add Extension</i> .....	76
Gambar 4.20 Penambahan <i>User Extension</i> .....	77
Gambar 4.21 Pengisian <i>Screet</i> .....	77
Gambar 4.202 <i>Submit dan Apply Configuration Changes</i> .....	77
Gambar 4.23 <i>Contimue With Reload</i> .....	78
Gambar 4.24 Menu Aplikasi <i>Monitoring Traffic</i> .....	78
Gambar 4.25 Tampilan <i>Monitoring Traffic</i> Jaringan VoIP .....	79
Gambar 4.26 Desain Jaringan VoIP .....	79
Gambar 4.27 Hasil <i>Monitoring Traffic</i> .....	80
Gambar 4.28 Desain Pengujian Jaringan VoIP .....	81
Gambar 4.29 Pengujian komunikasi antara <i>extension user 101</i> dengan <i>extension user 102</i> .....	82
Gambar 4.30 Pengujian komunikasi antara <i>extension user 101</i> dengan <i>extension user 102</i> dengan menggunakan STG .....	83
Gambar 4.31 Pengujian komunikasi antara <i>extension user 101</i> dengan <i>extension user 102</i> dan <i>extension user 201</i> dengan <i>extension user 204</i> .....	84
Gambar 4.32 Pengujian komunikasi antara <i>extension user 101</i> dengan <i>extension user 102</i> dan <i>extension user 201</i> dengan <i>extension user 204</i>	

dengan menggunakan STG .....	84
Gambar 4.33 Pengujian komunikasi antara <i>extension user</i> 101 dengan <i>extension user</i> 102 , <i>extension user</i> 201 dengan <i>extension user</i> 204 dan , <i>extension user</i> 202 dengan <i>extension user</i> 203.....	85
Gambar 4.34 Pengujian komunikasi antara <i>extension user</i> 101 dengan <i>extension user</i> 102 , <i>extension user</i> 201 dengan <i>extension user</i> 204 dan , <i>extension user</i> 202 dengan <i>extension user</i> 203 dengan menggunakan STG.....	86
Gambar 4.35 Tabel <i>Traffic</i> .....	87
Gambar 4.36 <i>Setting</i> ODBC.....	88
Gambar 4.37 Hasil Log <i>Traffic</i> Pada Database.....	89

## DAFTAR TABEL

Tabel 2.1 Kategori Kabel UTP.....	9
Tabel 2.1 OSI Layer .....	23
Tabel 2.3 Ukuran header pada kompersi G.729 .....	33
Tabel 2.4 Response SIP .....	41
Tabel 3.1 Jenis pesan SNMP .....	52
Tabel 3.2 Kelompok Object Internet MIB .....	54
Tabel 4.1 Spesifikasi Perlengkapan Implementasi.....	66
Tabel 4.2 Perbandingan hasil pengujian Traffic .....	86

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Seiring dengan perkembangan zaman, hampir semua instansi di dunia telah memanfaatkan teknologi jaringan sebagai pendukung dari perkembangan teknologi informasi yang mereka gunakan. Hal ini dialami instansi baik berskala kecil maupun skala besar. Perkembangan ini sangat terasa pada saat memasuki era globalisasi, dimana teknologi informasi berkembang dengan cepat dan pesat, membawa dunia bisnis memasuki suatu babak baru.

Komunikasi yang dulunya dilakukan secara tradisional dan manual, kini mulai bergeser pada proses otomatisasi. Komunikasi yang memerlukan waktu berhari-hari, kini dapat diproses dengan cepat. Hal ini menjadi era baru dalam dunia komunikasi. Telekomunikasi dan jaringan komputer telah berdampak sangat luas dalam bisnis, teknologi dan gaya hidup manusia saat ini. Teknologi-teknologi baru di bidang telekomunikasi dan jaringan komputer terus bermunculan dengan konsep-konsep baru. Semuanya untuk mencapai satu tujuan yaitu mempermudah kehidupan manusia, khususnya dalam mempermudah pertukaran data dan informasi, serta penyebarannya yang tak terbatas lagi pada ruang dan waktu. Saat ini sarana telekomunikasi menjadi “penyedia jalan” bagi jaringan komputer global (*internet*). Pengintegrasian teknologi analog dan digital yang selama ini berkembang pesat telah sampai pada suatu titik balik, dimana selama ini teknologi digital yang menumpang pada saluran telekomunikasi analog (internet lewat PSTN) akan digantikan dengan teknologi baru yang sepenuhnya digital.

Jenuhnya teknologi komunikasi analog (berbasis *switching*) yang terbentur pada mahalnya biaya, dijawab dengan teknologi digital (berbasis *packet*) yang lebih murah biayanya, khususnya biaya untuk sambungan perkomunikasinya. Konsep ini sering disebut *Next Generation Network* (NGN) yaitu menggantikan infrastruktur analog (*switching based*) dengan infrastruktur baru yang berbasis *packet*. VoIP Server merupakan salah satu solusi komunikasi digital yang telah sepenuhnya mengadopsi teknologi komunikasi digital berbasis IP (*packet*) mulai dari komputer, headphones, handset sampai peralatan *switching*, *gateway* dan

peralatan jaringan komputer lainnya. Teknologi ini pada dasarnya kedepannya ingin menggantikan sistem PABX yang pada skala besar digunakan pada jaringan telepon PSTN konvensional. Dengan kata lain dapat dikatakan bahwa VoIP Server merupakan IP PBX yang adalah sebuah sistem yang mempunyai fungsi utama menyediakan layanan VoIP (*Voice Over IP*) mulai dari *registrasi user, call routing, call conference, interactive voice response, call forwarding, caller id, voice mail* dan sebagainya. Dalam sebuah jaringan VoIP, selain terdapat IP PBX, juga terdapat beberapa *client* yang dapat saling berkomunikasi dengan baik dengan perantara IP PBX ini.

Prinsip kerja dari sistem layanan VoIP yaitu *Client-client* yang terhubung dalam sistem tersebut mempunyai nomor IP Address sendiri. Untuk biasa berkomunikasi antar *client*, maka masing-masing *client* harus ter-registri di IP PBX. Setelah diregistrasi, setiap *client* akan mendapat nomor *user (user account)*. Sebuah *client*, jika ingin berkomunikasi dengan *client* lain harus men-dial nomor *user* dari *client* tujuan sesuai dengan nomor registrasi yang diberikan oleh IP PBX. Komunikasi antar *client* ini selalu dimonitor oleh IP PBX.

Hal-hal diatas mendorong saya untuk menjadi salah satu pengembang pengimplementasian sebuah teknologi yang memanfaatkan IP *Communications*, khususnya fitur IP PBX, yang telah didukung oleh perangkat jaringan mereka sekarang, guna mentransmisikan lalu lintas data, suara dan video melalui satu jaringan infrastruktur ini. Tentunya yang secara nyata akan membantu mengoptimalkan kinerja komunikasi dalam suatu instansi yang berdampak pada penghematan biaya operasional telekomunikasi.

Pada saat ini para administrator dalam memonitoring *traffic* jaringan VoIP masih memanfaatkan webmin yang tersedia pada fitur aplikasi server IP PBX. Sehingga dalam penelitian ini akan dibuat suatu aplikasi monitoring *traffic* berbasis GUI (*Graphic User Interface*) yang memudahkan para *administrator* dan *user* lainnya untuk melihat kondisi lalu lintas pada jaringannya dan mengoptimalkan pekerjaan yang menjadi tanggung jawabnya.



## 1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, telah diambil permasalahan yang akan dibahas dalam penelitian ini adalah bagaimana membuat suatu Aplikasi Monitoring *Traffic* Pada Jaringan VoIP .

## 1.3 Tujuan Penelitian

Adapun tujuan yang diharapkan dalam penelitian ini adalah dapat membuat suatu aplikasi yang diperuntukkan sebagai pemantau lalu lintas komunikasi data pada jaringan VoIP, sehingga memudahkan administrator dan user lain untuk memonitoring adanya komunikasi, Dengan tanpa membuka atau melihat keseluruhan fitur Aplikasi pada Server IP PBX melalui Webmin yang tersedia terlebih dahulu.

## 1.4 Batasan Masalah

Agar permasalahan mengarah sesuai dengan dengan tujuan yang diharapkan, maka pembahasan dibatasi oleh hal – hal sebagai berikut:

1. Aplikasi *Monitoring traffic Pada Jaringan VoIP* ini dibuat pada aplikasi berbasis windows.
2. Aplikasi *Monitoring traffic Pada Jaringan VoIP* ini tidak menangani aplikasi lain selain *monitoring traffic* pada server IP PBX ( *TRIXBOX* ).
3. Tidak membahas mengenai Codec (*CoderDecoder*) dan pembuatan Extension dalam Aplikasi Trixbox.
4. Implementasi VoIP menggunakan protokol SIP (*Session Initiation Protocol*).
5. Implementasi *request data traffic* menggunakan protokol SNMP ( *Simple Network Management Protocol*).

## 1.5 Metode Penelitian

Adapun metode penelitian yang digunakan adalah sebagai berikut:

1. Studi literatur

Pengumpulan data yang dilakukan dengan mencari bahan-bahan kepustakaan dan referensi dari berbagai sumber sebagai landasan teori yang ada hubungannya dengan permasalahan yang dijadikan objek penelitian.

## 2. Analisa Kebutuhan Sistem

Data dan informasi yang telah diperoleh akan dianalisa agar didapatkan kerangka global yang bertujuan untuk mendefinisikan kebutuhan sistem di mana nantinya akan digunakan sebagai acuan perancangan sistem.

## 3. Perancangan dan Implementasi

Berdasarkan data dan informasi yang telah diperoleh serta analisa kebutuhan untuk membangun sistem ini, akan dibuat rancangan kerangka global yang menggambarkan mekanisme dari sistem yang akan dibuat dan diimplementasikan kedalam system.

## 4. Eksperimen dan Evaluasi

Pada tahap ini, sistem yang telah selesai dibuat akan diuji coba, yaitu pengujian berdasarkan fungsionalitas program, dan akan dilakukan koreksi dan penyempurnaan program jika diperlukan.

### **1.6 Sistematika Penulisan**

Untuk mempermudah dan memahami pembahasan penulisan skripsi ini, maka sistematika penulisan disusun sebagai berikut :

**Bab I : Pendahuluan**

Berisi Latar Belakang, Rumusan Masalah, Tujuan Penelitian, Pembatasan Permasalahan, Metode Penelitian dan Sistematika Penulisan.

**Bab II : Tinjauan Pustaka**

Berisi tentang landasan teori mengenai permasalahan yang berhubungan dengan penelitian yang dilakukan.

**Bab III : Perancangan dan Analisa Sistem**

Dalam bab ini berisi mengenai analisa kebutuhan sistem baik software maupun hardware yang diperlukan untuk membuat

kerangka global yang menggambarkan mekanisme dari sistem yang akan dibuat. .

**Bab IV : Pembuatan dan Pengujian Sistem**

Berisi tentang implementasi dari perancangan sistem yang telah dibuat serta pengujian terhadap sistem tersebut.

**Bab V : Penutup**

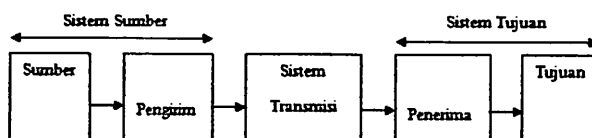
Merupakan bab terakhir yang memuat intisari dari hasil pembahasan yang berisikan kesimpulan dan saran yang dapat digunakan sebagai pertimbangan untuk pengembangan penulisan selanjutnya.

## BAB II

### LANDASAN TEORI

#### 2.1 Konsep Telekomunikasi

Kegunaan dasar dari sistem telekomunikasi adalah menjalankan pertukaran data antara dua pihak. Elemen-elemen yang ada didalamnya dapat digambarkan sebagai berikut:



Gambar 2.1. Sistem Komunikasi

1. **Sumber** (*source*): Membangkitkan data atau informasi yang akan ditransmisikan, contoh: telepon dan PC.
2. **Pengirim** (*transmitter*): Data atau informasi yang dibangkitkan oleh sistem sumber tidak ditransmisikan secara langsung dalam bentuk aslinya. Sebuah *transmitter* cukup memindahkan dan menandai informasi dengan cara yang sama seperti menghasilkan sinyal-sinyal elektromagnetik yang dapat ditransmisikan melewati beberapa sistem transmisi berurutan.
3. **Sistem Transmisi** (*transmission system*): Merupakan jalur transmisi tunggal (*single transmission line*) atau merupakan jaringan kompleks (*complex network*) yang menghubungkan sumber dan tujuan.
4. **Penerima** (*receiver*): Berfungsi menerima sinyal dari sistem transmisi dan menggabungkannya ke dalam bentuk tertentu yang dapat ditangkap dan dimengerti oleh tujuan (*destination*).
5. **Tujuan** (*destination*): Menangkap data yang dihasilkan oleh *receiver*.

#### 2.2 Jaringan Komputer

Jaringan komputer adalah sebuah kumpulan komputer, printer dan peralatan lainnya yang terhubung. Informasi dan data bergerak melalui kabel-kabel sehingga memungkinkan pengguna jaringan komputer dapat saling bertukar dokumen dan data, mencetak pada printer yang sama dan bersama sama

menggunakan *hardware software* yang terhubung dengan jaringan. Tiap komputer, printer atau periferal yang terhubung dengan jaringan disebut node. Sebuah jaringan komputer dapat memiliki dua, puluhan, ribuan atau bahkan jutaan node. Sebuah jaringan biasanya terdiri dari dua atau lebih komputer yang saling berhubungan diantara satu dengan yang lain, dan saling berbagi sumber daya misalnya CDROM, Printer, pertukaran file, atau memungkinkan untuk saling berkomunikasi secara elektronik. Komputer yang terhubung tersebut, dimungkinkan berhubungan dengan media kabel, saluran telepon, gelombang radio, satelit, atau sinar infra merah.

### **2.2.1 Jenis-jenis Jaringan Komputer**

Ada 3 macam jenis Jaringan, yaitu :

#### **a. *Local Area Network (LAN)***

*Local Area Network* biasa disingkat LAN adalah jaringan komputer yang jaringannya hanya mencakup wilayah kecil seperti jaringan komputer kampus, gedung, kantor, dalam rumah, sekolah atau yang lebih kecil. Saat ini, kebanyakan LAN berbasis pada teknologi IEEE 802.3 Ethernet menggunakan perangkat switch, yang mempunyai kecepatan transfer data 10, 100, atau 1000 Mbit/s. Selain teknologi Ethernet, saat ini teknologi 802.11b (atau biasa disebut Wi-fi) juga sering digunakan untuk membentuk LAN. Tempat-tempat yang menyediakan koneksi LAN dengan teknologi Wi-fi biasa disebut hotspot.

Pada sebuah LAN, setiap node atau komputer mempunyai daya komputasi sendiri, berbeda dengan konsep dump terminal. Setiap komputer juga dapat mengakses sumber daya yang ada di LAN sesuai dengan hak akses yang telah diatur. Sumber daya tersebut dapat berupa data atau perangkat seperti printer. Pada LAN, seorang pengguna juga dapat berkomunikasi dengan pengguna yang lain dengan menggunakan aplikasi yang sesuai.

**b. Metropolitan Area Network (MAN)**

Metropolitan area network atau disingkat dengan MAN. Suatu jaringan dalam suatu kota dengan transfer data berkecepatan tinggi, yang menghubungkan berbagai lokasi seperti kampus, perkantoran, pemerintahan, dan sebagainya. Jaringan MAN adalah gabungan dari beberapa LAN. Jangkauan dari MAN ini antar 10 hingga 50 km, MAN ini merupakan jaringan yang tepat untuk membangun jaringan antar kantor-kantor dalam satu kota antara pabrik/instansi dan kantor pusat yang berada dalam jangkauannya.

**c. Wide Area Network (WAN)**

WAN (*Wide Area Network*) merupakan jaringan komputer yang mencakup area yang besar sebagai contoh yaitu jaringan komputer antar wilayah, kota atau bahkan negara, atau dapat didefinisikan juga sebagai jaringan komputer yang membutuhkan router dan saluran komunikasi publik.

WAN digunakan untuk menghubungkan jaringan lokal yang satu dengan jaringan lokal yang lain, sehingga pengguna atau komputer di lokasi yang satu dapat berkomunikasi dengan pengguna dan komputer di lokasi yang lain.

### **2.2.2 Intranet**

Intranet adalah konsep LAN yang mengadopsi teknologi Internet dan mulai diperkenalkan pada akhir tahun 1995. Atau bisa dikatakan intranet adalah LAN yang menggunakan standar komunikasi dan segala fasilitas Internet, diibaratkan berinternet dalam lingkungan lokal. umumnya juga terkoneksi ke internet sehingga memungkinkan pertukaran informasi dan data dengan jaringan intranet lainnya (*internetworking*) melalui *backbone* internet.

### **2.2.3 Media Implementasi Jaringan Komputer**

Media ini terdiri dari beberapa bagian yaitu :

**a) Kabel Twisted Pair**

Kabel *Twisted pair* (pasangan berpilin) adalah sebuah bentuk kabel di mana dua konduktor digabungkan dengan tujuan untuk mengurangi atau meniadakan interferensi elektromagnetik dari luar seperti radiasi elektromagnetik dari kabel *unshielded twisted pair (UTP) cables*, dan crosstalk di antara pasangan kabel yang berdekatan.

- ***Unshielded twisted-pair***

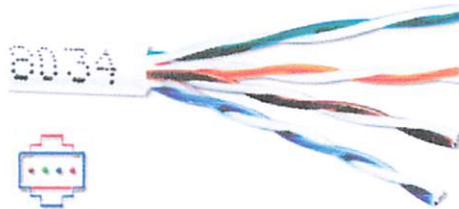
*Unshielded twisted-pair* (disingkat UTP) adalah sebuah jenis kabel jaringan yang menggunakan bahan dasar tembaga, yang tidak dilengkapi dengan *shield* internal. UTP merupakan jenis kabel yang paling umum yang sering digunakan di dalam jaringan lokal (LAN), karena memang harganya yang rendah, fleksibel dan kinerja yang ditunjukkannya relatif bagus. Dalam kabel UTP, terdapat insulasi satu lapis yang melindungi kabel dari ketegangan fisik atau kerusakan tapi, tidak seperti kabel *Shielded Twisted-pair (STP)*, insulasi tersebut tidak melindungi kabel dari interferensi elektromagnetik.

Kabel UTP memiliki impedansi kira-kira 100 Ohm dan tersedia dalam beberapa kategori yang ditentukan dari kemampuan transmisi data yang dimilikinya seperti tertulis dalam tabel berikut.

Tabel 2.1  
Kategori Kabel UTP

Kategori	Kegunaan
Category 1 (Cat1)	Kualitas suara analog
Category 2 (Cat2)	Transmisi suara digital hingga 4 megabit per detik
Category 3 (Cat3)	Transmisi data digital hingga 10 megabit per detik
Category 4 (Cat4)	Transmisi data digital hingga 16 megabit per detik
Category 5 (Cat5)	Transmisi data digital hingga 100 megabit per detik
Enhanced Category 5 (Cat5e)	Transmisi data digital hingga 250 megabit per detik
Category 6 (Cat6)	
Category 7 (Cat7)	

Di antara semua kabel di atas, kabel *Enhanced Category 5* (Cat5e) dan *Category 5* (Cat5) merupakan kabel UTP yang paling populer yang banyak digunakan dalam jaringan berbasis teknologi Ethernet.



Gambar 2.2 Kabel UTP

#### ❖ **Kategori 1**

Kabel UTP *Category 1* (Cat1) adalah kabel UTP dengan kualitas transmisi terendah, yang didesain untuk mendukung komunikasi suara analog saja. Kabel Cat1 digunakan sebelum tahun 1983 untuk menghubungkan telepon analog Plain Old Telephone Service (POTS). Karakteristik kelistrikan dari kabel Cat1 membuatnya kurang sesuai untuk digunakan sebagai kabel untuk mentransmisikan data digital di dalam jaringan komputer, dan karena itulah tidak pernah digunakan untuk tujuan tersebut.

#### ❖ **Kategori 2**

Kabel UTP *Category 2* (Cat2) adalah kabel UTP dengan kualitas transmisi yang lebih baik dibandingkan dengan kabel UTP *Category 1* (Cat1), yang didesain untuk mendukung komunikasi data dan suara digital. Kabel ini dapat mentransmisikan data hingga 4 megabit per detik. Seringnya, kabel ini digunakan untuk menghubungkan node-node dalam jaringan dengan teknologi Token Ring dari IBM. Karakteristik kelistrikan dari kabel Cat2 kurang cocok jika digunakan sebagai kabel jaringan masa kini.



Gunakanlah kabel yang memiliki kinerja tinggi seperti Category 3, Category 4, atau Category 5.

❖ **Kategori 3**

Kabel UTP *Category 3 (Cat3)* adalah kabel UTP dengan kualitas transmisi yang lebih baik dibandingkan dengan kabel UTP Category 2 (Cat2), yang didesain untuk mendukung komunikasi data dan suara pada kecepatan hingga 10 megabit per detik. Kabel UTP Cat3 menggunakan kawat-kawat tembaga 24-gauge dalam konfigurasi 4 pasang kawat yang dipilin (*twisted-pair*) yang dilindungi oleh insulasi. Cat3 merupakan kabel yang memiliki kemampuan terendah (jika dilihat dari perkembangan teknologi Ethernet), karena memang hanya mendukung jaringan 10BaseT saja. Seringnya, kabel jenis ini digunakan oleh jaringan IBM Token Ring yang berkecepatan 4 megabit per detik, sebagai pengganti Cat2.

❖ **Kategori 4**

Kabel UTP *Category 4 (Cat4)* adalah kabel UTP dengan kualitas transmisi yang lebih baik dibandingkan dengan kabel UTP Category 3 (Cat3), yang didesain untuk mendukung komunikasi data dan suara hingga kecepatan 16 megabit per detik. Kabel ini menggunakan kawat tembaga 22-gauge atau 24-gauge dalam konfigurasi empat pasang kawat yang dipilin (*twisted pair*) yang dilindungi oleh insulasi. Kabel ini dapat mendukung jaringan Ethernet 10BaseT, tapi seringnya digunakan pada jaringan IBM Token Ring 16 megabit per detik.

❖ **Kategori 5**

Kabel UTP Category 5 (Cat5) adalah kabel dengan kualitas transmisi yang jauh lebih baik dibandingkan dengan kabel UTP Category 4 (Cat4), yang didesain untuk

mendukung komunikasi data serta suara pada kecepatan hingga 100 megabit per detik. Kabel ini menggunakan kawat tembaga dalam konfigurasi empat pasang kawat yang dipilin (*twisted pair*) yang dilindungi oleh insulasi. Kabel ini telah distandardisasi oleh Electronic Industries Alliance (EIA) dan Telecommunication Industry Association (TIA).

Kabel Cat5 dapat mendukung jaringan Ethernet (10BaseT), Fast Ethernet (100BaseT), hingga Gigabit Ethernet (1000BaseT). Kabel ini adalah kabel paling populer, mengingat kabel serat optik yang lebih baik harganya hampir dua kali lipat lebih mahal dibandingkan dengan kabel Cat5. Karena memiliki karakteristik kelistrikan yang lebih baik, kabel Cat5 adalah kabel yang disarankan untuk semua instalasi jaringan.

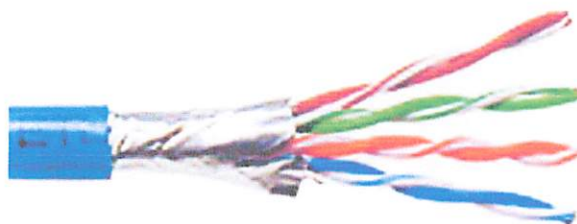
❖ ***Enhanced Category***

Kabel ini merupakan versi perbaikan dari kabel UTP Cat5, yang menawarkan kemampuan yang lebih baik dibandingkan dengan Cat5 biasa. Kabel ini mampu mendukung frekuensi hingga 250 MHz, yang direkomendasikan untuk penggunaan dalam jaringan Gigabit Ethernet, meskipun penggunaan kabel UTP Category 6 lebih disarankan untuk mencapai kinerja tertinggi.

• ***Shielded twisted pair***

Kabel STP mengkombinasikan teknik-teknik perlindungan dan antisipasi tekukan kabel. STP yang peruntukan bagi instalasi jaringan ethernet, memiliki resistansi atas interferensi elektromagnetik dan frekuensi radio tanpa perlu meningkatkan ukuran fisik kabel. Kabel Shielded Twister-Pair nyaris memiliki kelebihan dan kekurangan yang sama dengan kabel UTP. Satu hal keunggulan STP adalah jaminan proteksi jaringan dari

interferensi-interferensi eksternal, sayangnya STP sedikit lebih mahal dibandingkan UTP.



Gambar 2.3 Kabel STP

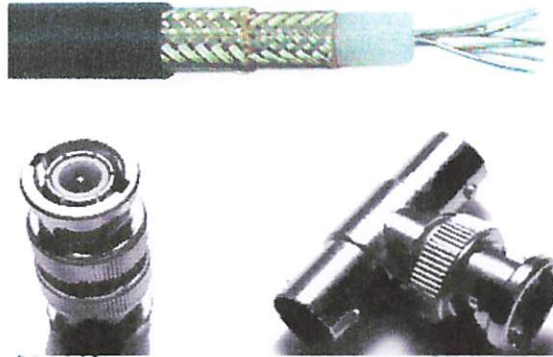
Tidak seperti kabel coaxial, lapisan pelindung kabel STP bukan bagian dari sirkuit data, karena itu perlu diground pada setiap ujungnya. Pada prakteknya, melakukan ground STP memerlukan kejelian. Jika terjadi ketidaktepatan, dapat menjadi sumber masalah karena bisa menyebabkan pelindung bekerja sebagai layaknya sebuah antenna; menghisap sinyal-sinyal elektrik dari kawat-kawat dan sumber-sumber listrik lain disekitarnya. Kabel STP tidak dapat dipakai dengan jarak lebih jauh sebagaimana media-media lain (seperti kabel coaxial) tanpa bantuan *device* penguat (*repeater*). Karakteristik kabel STP sebagai berikut :

- ✓ Kecepatan dan keluaran: 10-100 Mbps.
- ✓ Biaya rata-rata per node: sedikit mahal dibandingkan UTP dan coaxial.
- ✓ Media dan ukuran konektor: medium.
- ✓ Panjang kabel maksimum yang diizinkan : 100m (pendek).

#### **b) Kabel Coaxial**

Kabel coaxial atau populer disebut “coax” terdiri atas konduktor silindris melingkar, yang mengelilingi sebuah kabel tembaga inti yang konduktif. Untuk LAN, kabel coaxial menawarkan beberapa keunggulan. Diantaranya dapat dijalankan dengan tanpa banyak membutuhkan bantuan repeater sebagai penguat untuk komunikasi jarak jauh diantara node network, dibandingkan kabel STP atau UTP. *Repeater* juga dapat diikutsertakan untuk

meregenerasi sinyal-sinyal dalam jaringan coaxial sehingga dalam instalasi network cukup jauh dapat semakin optimal. Kabel coaxial juga jauh lebih murah dibanding Fiber Optic, coaxial merupakan teknologi yang sudah lama dikenal. Digunakan dalam berbagai tipe komunikasi data sejak bertahun-tahun, baik di jaringan rumah, kampus, maupun perusahaan.



Gambar 2.4 Kabel Coaxial

- ✓ Kecepatan dan keluaran: 10 -100 Mbps.
- ✓ Biaya rata-rata per node: murah.
- ✓ Media dan ukuran konektor: medium.
- ✓ Panjang kabel maksimum: 200m (disarankan 180m) untuk *thin-coaxial* dan 500m untuk *thick-coaxial*.

Saat bekerja dengan kabel, penting bagi kita untuk mempertimbangkan ukurannya; seperti ketebalan, diameter, pertambahan kabel sehingga akan menjadi pertimbangan atas kesulitan saat instalasi dilapangan. Kita juga harus ingat bahwa kabel akan mengalami tarikan-tarikan dan tekukan di dalam pipa. Kabel coaxial datang dalam beragam ukuran. Diameter terbesar diperuntukkan sebagai backbone Ethernet karena secara historis memiliki ketahanan transmisi dan daya tolak interferensi yang lebih besar. Tipe kabel coaxial ini sering disebut dengan thicknet, namun dewasa ini sudah banyak ditinggalkan. Kabel coaxial lebih mahal saat diinstal dibandingkan kabel twisted-pair.

**c) *Fiber Optic***

Kabel fiber optic merupakan media networking yang mampu digunakan untuk transmisi-transmisi modulasi. Jika dibandingkan media-media lain, fiber optic memiliki harga lebih mahal, tetapi cukup tahan terhadap interferensi elektromagnetis dan mampu beroperasi dengan kecepatan dan kapasitas data yang tinggi. Kabel fiber optic dapat mentransmisikan puluhan juta bit digital perdetik pada link kabel optic yang beroperasi dalam sebuah jaringan komersial. Ini sudah cukup untuk mengantarkan ribuan panggilan telepon.

Beberapa keuntungan kabel fiber optic:

- ✓ Kecepatan: jaringan-jaringan fiber optic beroperasi pada kecepatan tinggi, mencapai gigabits per second
- ✓ Bandwidth: fiber optic mampu membawa paket-paket dengan kapasitas besar.
- ✓ Distance: sinyal-sinyal dapat ditransmisikan lebih jauh tanpa memerlukan perlakuan “refresh” atau “diperkuat”.
- ✓ Resistance: daya tahan kuat terhadap imbas elektromagnetik yang dihasilkan perangkat-perangkat elektronik seperti radio, motor, atau bahkan kabel-kabel transmisi lain di sekelilingnya.

**d) *Network Interface Card (NIC)***

NIC adalah sebuah kartu yang berfungsi sebagai jembatan dari komputer ke sebuah jaringan komputer. Jenis NIC yang beredar, terbagi menjadi dua jenis, yakni NIC yang bersifat fisik, dan NIC yang bersifat logis. Contoh NIC yang bersifat fisik adalah NIC Ethernet, Token Ring, dan lainnya. Sementara NIC yang bersifat logis adalah loopback adapter dan Dial-up Adapter. Disebut juga sebagai Network Adapter. Setiap jenis NIC diberi nomor alamat yang disebut sebagai MAC address, yang dapat bersifat statis atau dapat diubah oleh pengguna.



Gambar 2.5 Network Interface Card

#### 2.2.4 Topologi Jaringan Komputer

Topologi menggambarkan struktur dari suatu jaringan atau bagaimana sebuah jaringan didesain. Pola ini sangat erat kaitannya dengan metode access dan media pengiriman yang digunakan. Topologi yang ada sangatlah tergantung dengan letak geografis dari masing-masing terminal, kualitas kontrol yang dibutuhkan dalam komunikasi ataupun penyampaian pesan, serta kecepatan dari pengiriman data. Dalam definisi topologi terbagi menjadi dua, yaitu topologi fisik (physical topology) yang menunjukkan posisi pemasangan kabel secara fisik dan topologi logik (logical topology) yang menunjukkan bagaimana suatu media diakses oleh host.

Adapun topologi fisik yang umum digunakan dalam membangun sebuah jaringan adalah :

##### 1. *Point to Point (Titik ke-Titik)*

Point to Point merupakan jaringan kerja yang paling sederhana tetapi dapat digunakan secara luas. Dalam hal ini, kedua simpul mempunyai kedudukan yang setingkat, sehingga simpul manapun dapat memulai dan mengendalikan hubungan dalam jaringan tersebut. Data dikirim dari satu simpul langsung kesimpul lainnya sebagai penerima, misalnya antara terminal dengan CPU.

##### 2. *Star Network (Jaringan Bintang)*

Dalam konfigurasi bintang, beberapa peralatan yang ada akan dihubungkan kedalam satu pusat kontrol. Kontrol yang ada akan dipusatkan pada satu titik, seperti misalnya mengatur beban kerja serta pengaturan sumber daya yang ada. Semua link harus berhubungan dengan pusat apabila ingin menyalurkan data kesimpul lainnya yang dituju. Dalam hal ini, bila pusat mengalami gangguan, maka semua

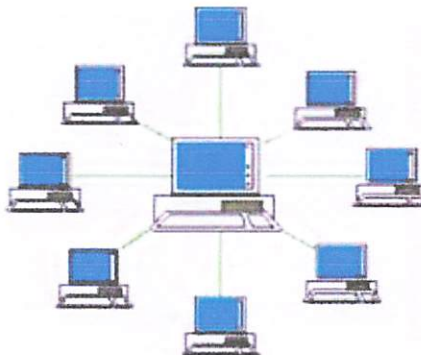
terminal juga akan terganggu. Model jaringan bintang ini relatif sangat sederhana, sehingga banyak digunakan oleh pihak perbankan yang biasanya mempunyai banyak kantor cabang yang tersebar diberbagai lokasi. Dengan adanya konfigurasi bintang ini, maka segala macam kegiatan yang ada di-kantor cabang dapatlah dikontrol dan dikoordinasikan dengan baik.

Kelebihan :

- Kerusakan pada satu saluran hanya akan mempengaruhi jaringan pada saluran tersebut dan stasiun yang terpaut.
- Tingkat keamanan termasuk tinggi.
- Penambahan dan pengurangan station dapat dilakukan dengan mudah.

Kekurangan :

- Jika node tengah mengalami kerusakan, maka maka seluruh jaringan akan terhenti.



Gambar 2.6 Topologi jaringan bintang

### 3. *Ring Networks (Jaringan Cincin)*

Pada jaringan komputer ini terdapat beberapa peralatan saling dihubungkan satu dengan lainnya dan pada akhirnya akan membentuk bagan seperti halnya sebuah cincin. Jaringan cincin tidak memiliki suatu titik yang bertindak sebagai pusat ataupun pengatur lalu lintas data, semua simpul mempunyai tingkatan yang sama. Data yang dikirim akan berjalan melewati beberapa simpul sehingga sampai pada simpul yang dituju. Dalam menyampaikan data, jaringan bisa bergerak dalam

satu ataupun dua arah. Walaupun demikian, data yang ada tetap bergerak satu arah dalam satu saat. Pertama, pesan yang ada akan disampaikan dari titik ketitik lainnya dalam satu arah. Apabila ditemui kegagalan, misalnya terdapat kerusakan pada peralatan yang ada, maka data yang ada akan dikirim dengan cara kedua, yaitu pesan kemudian ditransmisikan dalam arah yang berlawanan, dan pada akhirnya bisa berakhir pada tempat yang dituju.

Konfigurasi semacam ini relative lebih mahal apabila dibanding dengan konfigurasi jaringan bintang. Hal ini disebabkan, setiap simpul yang ada akan bertindak sebagai komputer yang akan mengatasi setiap aplikasi yang dihadapinya, serta harus mampu membagi sumber daya yang dimilikinya pada jaringan yang ada. Disamping itu, sistem ini lebih sesuai digunakan untuk sistem yang tidak terpusat (*decentralized-system*), dimana tidak diperlukan adanya suatu prioritas tertentu.

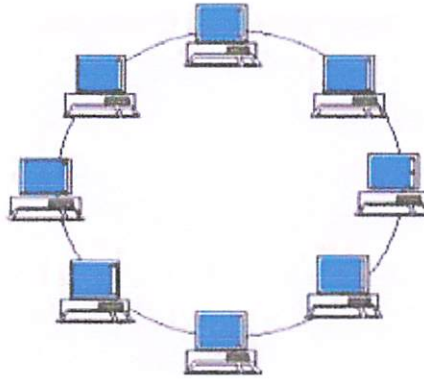
**Kelebihan :**

- Tidak ada komputer yang memonopoli jaringan, karena setiap komputer mempunyai hak akses yang sama terhadap *token*.
- Data mengalir dalam satu arah sehingga terjadinya *collision* dapat dihindarkan.

**Kekurangan :**

- Apabila ada satu komputer dalam *ring* yang gagal berfungsi, maka akan mempengaruhi keseluruhan jaringan.
- Sulit untuk mengatasi kerusakan di jaringan yang menggunakan topologi *ring*.
- Menambah atau mengurangi komputer akan mengacaukan jaringan.
- Sulit untuk melakukan konfigurasi ulang.





Gambar 2.7 Topologi jaringan cincin

#### 4. *Tree Network (Jaringan Pohon)*

Pada jaringan pohon, terdapat beberapa tingkatan simpul (node). Pusat atau simpul yang lebih tinggi tingkatannya, dapat mengatur simpul lain yang lebih rendah tingkatannya. Data yang dikirim perlu melalui simpul pusat terlebih dahulu. Misalnya untuk bergerak dari komputer dengan node-3 ke komputer node-7 seperti halnya pada gambar, data yang ada harus melewati node-3, 5 dan node-6 sebelum berakhir pada node-7. Keunggulan jaringan model pohon seperti ini adalah, dapat terbentuknya suatu kelompok yang dibutuhkan pada setiap saat.

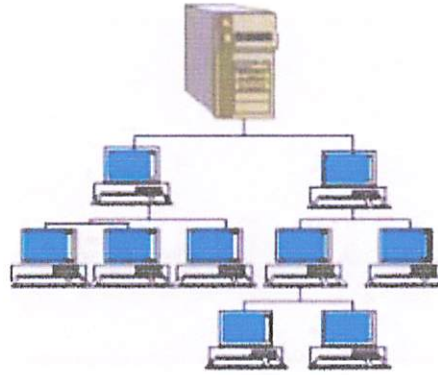
Sebagai contoh, perusahaan dapat membentuk kelompok yang terdiri atas terminal pembukuan, serta pada kelompok lain dibentuk untuk terminal penjualan. Adapun kelemahannya adalah, apabila simpul yang lebih tinggi kemudian tidak berfungsi, maka kelompok lainnya yang berada dibawahnya akhirnya juga menjadi tidak efektif. Cara kerja jaringan pohon ini relatif menjadi lambat.

Kelebihan :

- Data terpusat secara hirarki sehingga manajemen data lebih baik dan mudah

Kekurangan :

- Komputer di bawahnya tidak dapat dioperasikan apabila jalur pada komputer tingkat atasnya terputus



Gambar 2.8 Topologi jaringan pohon

### 5. *Topologi Bus*

Pada topologi bus semua komputer (terminal) dihubungkan pada jalur data (bus) yang berbentuk garis lurus (linear). Sehingga, data yang dikirim akan melalui semua terminal pada jalur tersebut. Jika alamat data tersebut sesuai dengan alamat yang dilalui, maka data tersebut akan diterima dan diproses. Namun, jika alamat tidak sesuai, maka data akan diabaikan oleh terminal yang dilalui dan pencarian alamat akan diteruskan hingga ditemukan alamat yang sesuai.

Kelebihan :

- Hemat kabel
- Tidak membutuhkan kendali pusat
- Penambahan dan pengurangan terminal dapat dilakukan tanpa mengganggu operasi yang berjalan
- Layout kabel sederhana

Kekurangan :

- Kecepatan akan menurun bila jumlah user (pemakai) bertambah
- Jika terjadi gangguan atau kerusakan pada salah satu lokasi (titik) dalam jaringan maka hal itu akan mempengaruhi jaringan secara keseluruhan.

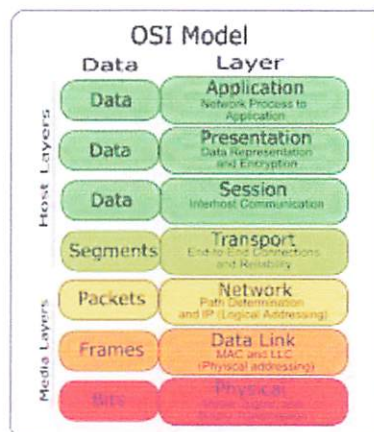


Gambar 2.9 Topologi jaringan bus

### 2.2.5 OSI Layer

Model referensi jaringan terbuka OSI atau *OSI Reference Model for open networking* adalah sebuah model arsitektural jaringan yang dikembangkan oleh badan International Organization for Standardization (ISO) di Eropa pada tahun 1977. OSI sendiri merupakan singkatan dari *Open System Interconnection*. Model ini disebut juga dengan model "Model tujuh lapis OSI" (*OSI seven layer model*).

Sebelum munculnya model referensi OSI, sistem jaringan komputer sangat tergantung kepada pemasok (*vendor*). OSI berupaya membentuk standar umum jaringan komputer untuk menunjang interoperabilitas antar pemasok yang berbeda. Dalam suatu jaringan yang besar biasanya terdapat banyak protokol jaringan yang berbeda. Tidak adanya suatu protokol yang sama, membuat banyak perangkat tidak bisa saling berkomunikasi.



Gambar 2.10 OSI Layer

Model referensi ini pada awalnya ditujukan sebagai basis untuk mengembangkan protokol-protokol jaringan, meski pada kenyataannya inisiatif ini mengalami kegagalan. Kegagalan itu disebabkan oleh beberapa faktor berikut:

- Standar model referensi ini, jika dibandingkan dengan model referensi DARPA (Model Internet) yang dikembangkan oleh Internet Engineering Task Force (IETF), sangat berdekatan. Model DARPA adalah model basis protokol TCP/IP yang populer digunakan.
- Model referensi ini dianggap sangat kompleks. Beberapa fungsi (seperti halnya metode komunikasi *connectionless*) dianggap kurang bagus, sementara fungsi lainnya (seperti *flow control* dan koreksi kesalahan) diulang-ulang pada beberapa lapisan.
- Pertumbuhan Internet dan protokol TCP/IP (sebuah protokol jaringan dunia nyata) membuat OSI Reference Model menjadi kurang diminati.

Pemerintah Amerika Serikat mencoba untuk mendukung protokol OSI Reference Model dalam solusi jaringan pemerintah pada tahun 1980-an, dengan mengimplementasikan beberapa standar yang disebut dengan ***Government Open Systems Interconnection Profile*** (GOSIP). Meski demikian, usaha ini akhirnya ditinggalkan pada tahun 1995, dan implementasi jaringan yang menggunakan *OSI Reference model* jarang dijumpai di luar Eropa.

*OSI Reference Model* pun akhirnya dilihat sebagai sebuah model ideal dari koneksi logis yang harus terjadi agar komunikasi data dalam jaringan dapat berlangsung. Beberapa protokol yang digunakan dalam dunia nyata, semacam TCP/IP dan IBM Systems Network Architecture (SNA) memetakan tumpukan protokol (*protocol stack*) mereka ke *OSI Reference Model*. *OSI Reference Model* pun digunakan sebagai titik awal untuk mempelajari bagaimana beberapa protokol jaringan di dalam sebuah kumpulan protokol dapat berfungsi dan berinteraksi.

OSI Reference Model memiliki tujuh lapis, yakni sebagai berikut:

Tabel 2.2 OSI Layer

Lapisan ke-	Nama lapisan	Keterangan
7	<b><i>Application layer</i></b>	Berfungsi sebagai antarmuka dengan aplikasi dengan fungsionalitas jaringan, mengatur bagaimana aplikasi dapat mengakses jaringan, dan kemudian membuat pesan-pesan kesalahan. Protokol yang berada dalam lapisan ini adalah HTTP, FTP, SMTP, dan NFS.
6	<b><i>Presentation layer</i></b>	Pada lapisan ini data yang diterima oleh lapisan application ditentukan format datanya sebelum dikirim pada jaringan. Format ini berupa format data untuk dipresentasikan. Layer ini pada dasarnya adalah sebagai penerjemah dan melakukan fungsi pengkodean dan konversi. Layer Presentation memastikan agar data yang berasal dari layer Application di satu komputer dapat dibaca oleh layer Application di komputer yang lain. Contoh: JPEG, MIDI, MPEG, RTF.
5	<b><i>Session Layer</i></b>	Lapisan ini bertugas mengatur sesi komunikasi data diantara station (komputer-komputer) yang saling berkomunikasi, yaitu

		<p>mengatur kapan sebuah sesi komunikasi di buka atau di putus. layer ini juga bertanggungjawab membangun hubungan antara komputer pengirim data dengan komputer penerima data. Layer Session melakukan koordinasi komunikasi antar sistem dan menawarkan tiga mode berikut: <i>simplex</i>, <i>half-duplex</i> dan <i>full-duplex</i>. Layer Session pada dasarnya menjaga terpisahnya data dari aplikasi yang satu dengan data dari aplikasi yang lain.</p> <p>Contoh: Network File System (NFS), Remote Procedure Call (RPC), X Window, AppleTalk Session Protocol (ASP), NetBEUI.</p>
4	<b>Transport</b>	<p>Layer ini bertugas mengatur urutan pengiriman data dalam jaringan. Layer ini juga berfungsi untuk mengontrol kemungkinan terdapatnya duplikat data yang dikirim sehingga tidak terjadi pengiriman data yang kembar. Selain itu layer ini juga bertanggungjawab memotong data yang besar menjadi bentuk paket – paket data yang lebih kecil.</p> <p>Contoh: TCP, UDP,</p>

		SPX(Protokol ciptaan Novell).
3	<i>Network</i>	<p>Pada layer ini data yang diterima dari lapisan di atasnya dipecah menjadi paket-paket data untuk dikirim melalui lapisan bawahnya. Layer Network juga mengelola pengalamatan peralatan, melacak lokasi peralatan di jaringan dan menentukan cara terbaik untuk memindahkan data, artinya layer Network harus mengangkut lalu lintas antar peralatan yang tidak terhubung secara lokal. Layer ini juga berfungsi mengurutkan paket – paket data yang terpisah bila data sudah sampai di komputer tujuan.</p> <p>Contoh: IP, IPX, ARP, RARP.</p>
2	<i>Datalink</i>	<p>Pada layer ini terjadi proses pengkodean data yang dikirim atau diterima. Data link layer terbagi menjadi 2 yaitu MAC Layer yang berfungsi memberikan identitas ke 25thernet dengan memanfaatkan MAC Address yang ada di Ethernet Card dan LLC (Logical Link Control) Layer yang berfungsi memotong paket – paket data menjadi bentuk yang lebih kecil lagi yang disebut frame.</p>

		Contoh device yang bekerja pada layer ini adalah bridge dan Ethernet card.
1	<b><i>Physical</i></b>	Merupakan lapisan paling bawah dalam struktur lapisan jaringan. Layer Physical melakukan dua hal yaitu mengirim bit dan menerima bit. Layer ini berhubungan dengan masalah listrik, prosedural, mengaktifkan, menjaga, dan menonaktifkan hubungan fisik. Layer ini berupa perangkat keras koneksi jaringan. Perangkat Hub berada pada lapisan ini. Contoh: 10BaseT, 100BaseTX.

### 2.2.6 Hubungan Komunikasi Data & Voice

Komunikasi data dan *voice* selalu mengembangkan perpindahan informasi dari suatu tempat ke tempat lain, tetapi bentuk perpindahan informasi data dan *voice* berbeda. Untuk *voice* memakai konversi suara (*audible tone conversion*) dan untuk *digital* menggunakan pensinyalan *digital* yang hanya dapat dimasukkan melalui terminal data (tanpa modem). Jaringan telepon disediakan oleh telepon *office* melalui pensinyalan *analog*, sedangkan informasi data dilewatkan melalui jalur komunikasi yang berupa pulsa *digital* (bit) yang akan diterima oleh modem yang kemudian diolah oleh terminal data (komputer).

Dapat dilihat bahwa antara komunikasi data dan *voice* mempunyai hubungan yang erat, terlebih lagi keduanya digunakan pada jalur komunikasi yang sama. Hubungan keduanya lebih erat lagi ketika



*voice* dan sinyal data dapat ditransmisikan secara bersama dalam satu jalur atau kanal digital.

## 2.3 VoIP (*Voice Over Internet Protocol*)

### 2.3.1 Konsep Dasar VoIP

Teknik dasar Voice over Internet Protocol atau yang biasa dikenal dengan sebutan VoIP adalah teknologi yang memungkinkan kemampuan melakukan percakapan telepon dengan menggunakan jalur komunikasi data pada suatu jaringan (*networking*). Sehingga teknologi ini memungkinkan komunikasi suara menggunakan jaringan berbasis IP (*internet protocol*) untuk dijalankan diatas infrastruktur jaringan packet network. Jaringan yang digunakan bisa berupa internet atau intranet. Teknologi ini bekerja dengan jalan merubah suara menjadi format digital tertentu yang dapat dikirimkan melalui jaringan IP.

Tujuan pengimplementasian VoIP adalah untuk menekan biaya operasional perusahaan maupun individu dalam melakukan komunikasi jarak jauh (interlokal/SLI). Penekanan biaya itu dapat dilakukan dengan cara memanfaatkan jaringan data yang sudah ada. Sehingga apabila kita ingin membuat jaringan telekomunikasi VoIP tidak perlu membangun infrastruktur baru yang mengeluarkan biaya yang sangat besar. Dengan menggunakan jaringan data yang ada, maka kita melakukan percakapan interlokal maupun internasional hanya dikenakan biaya lokal melalui PSTN.

*Internet telephony* lebih mengacu pada layanan komunikasi suara (*voice*), *faksimili*, dan *voice messaging applications*. Teknologi ini pada dasarnya mengkonversi sinyal *analog* (suara) ke format *digital* dan kemudian dikompres atau ditranslasikan ke dalam paket-paket IP yang kemudian ditransmisikan melalui jaringan internet.



Gambar 2.11 Gambaran VoIP

VoIP dalam penerapannya menggunakan sistem jaringan LAN dan *protocol-protocol* VoIP. Standarisasi *protocol* komunikasi pada teknologi VoIP seperti SIP (*session initiation protocol*) dan H.323.

### 2.3.2 Latar Belakang Teknologi VoIP

Latar belakang dibangunnya teknologi *Voice over Internet Protocol* (VoIP), antara lain:

1. Latar belakang perkembangan teknologi
  - ✓ Perkembangan teknologi komunikasi data. Semakin handalnya kualitas media transmisi – transmisi sinyal elektrik membutuhkan sebuah media transmisi yang normalnya membentuk jalur transmisi. Tipe media transmisi sangat penting untuk ditentukan pada awal pembentukan jalur komunikasi karena mempengaruhi jumlah maksimum bit (*binary digit*) yang dapat ditransmisikan.
  - ✓ Perkembangan teknologi sistem kompresi.
  - ✓ Perkembangan teknologi pemrosesan data.
2. Latar belakang perkembangan bisnis.
  - ✓ Persaingan di bidang bisnis telekomunikasi.
  - ✓ Tuntutan konsumen akan biaya komunikasi yang murah.
3. Efisiensi penggunaan media transmisi.

### 2.3.3 Kelebihan VoIP

Adapun kelebihan *Voice over Internet Protocol* (VoIP), adalah sebagai berikut:

- 1) Biaya lebih rendah untuk sambungan langsung jarak jauh. Penekanan utama dari VoIP adalah biaya. Dengan dua lokasi yang terhubung dengan internet maka biaya percakapan menjadi sangat rendah.
- 2) Memanfaatkan infrastruktur jaringan data yang sudah ada untuk suara. Berguna jika perusahaan sudah mempunyai jaringan. Jika memungkinkan jaringan yang ada bisa dibangun jaringan VoIP

dengan mudah. Tidak diperlukan tambahan biaya bulanan untuk penambahan komunikasi suara.

- 3) Penggunaan bandwidth yang lebih kecil daripada telepon biasa. Dengan majunya teknologi, penggunaan bandwidth untuk voice sekarang ini menjadi sangat kecil. Teknik pemampatan data memungkinkan suara hanya membutuhkan sekitar 8kbps bandwidth.
- 4) Memungkinkan digabung dengan jaringan telepon lokal yang sudah ada. Dengan adanya gateway bentuk jaringan VoIP bisa disambungkan dengan PABX (*Privat Automated Branch exchange*) yang ada dikantor. Komunikasi antar kantor bisa menggunakan pesawat telepon biasa.
- 5) Berbagai bentuk jaringan VoIP bisa digabungkan menjadi jaringan yang besar. Contoh di Indonesia adalah VoIP Rakyat.
- 6) Variasi penggunaan peralatan yang ada, misal dari PC sambung ke telepon biasa, *IP phone handset*.

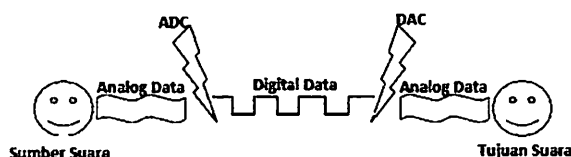
#### 2.3.4 Kekurangan VoIP

Adapun kekurangan *Voice over Internet Protocol* (VoIP), adalah sebagai berikut:

- 1) Kualitas suara tidak sejernih Telkom. Merupakan efek dari kompresi suara dengan bandwidth kecil maka akan ada penurunan kualitas suara dibandingkan jaringan PSTN konvensional.
- 2) Ada jeda dalam berkomunikasi. Proses perubahan data menjadi suara, jeda jaringan, membuat adanya jeda dalam komunikasi dengan menggunakan VoIP.
- 3) Peralatan relatif mahal. Peralatan VoIP yang menghubungkan antara VoIP dengan PABX (*IP telephony gateway*) relatif berharga mahal. Diharapkan dengan makin populernya VoIP ini maka harga peralatan tersebut juga mulai turun harganya.
- 4) Jika pemakaian VoIP semakin banyak, maka jaringan data yang ada menjadi penuh jika tidak diatur dengan baik

### 2.3.5 Cara Kerja VoIP

Pengiriman sebuah sinyal ke *remote destination* dapat dilakukan secara digital, yaitu sebelum dikirim data yang berupa sinyal analog, diubah dulu ke bentuk data digital dengan ADC (*analog to digital converter*), kemudian ditransmisikan, dan dipenerima dipulihkan kembali menjadi data analog dengan DAC (*digital to analog converter*). Begitu juga dengan VoIP, digitalisasi voice dalam bentuk packets data, dikirimkan dan dipulihkan kembali dalam bentuk voice dipenerima. Voice diubah dulu kedalam format digital karena lebih mudah dikendaikan dalam hal ini dapat dikompresi, dan dapat diubah ke format yang lebih baik. dan data digital lebih tahan terhadap noise dari pada analog.



Gambar 2.12 Cara kerja VoIP

### 2.3.6 Protokol Penunjang VoIP

Protokol-protokol yang menunjang terjadinya komunikasi VoIP adalah :

#### 1. TCP (*Transmission Control Protocol*)

Dalam mentransmisikan data pada layer transport ada dua protokol yang berperan yaitu TCP dan UDP. TCP merupakan protokol yang connection-oriented yang artinya menjaga reliabilitas hubungan komunikasi *end-to-end*. Konsep dasar cara kerja TCP adalah mengirim dan menerima segment – segment informasi dengan panjang data bervariasi pada suatu datagram internet. TCP menjamin realibilitas hubungan komunikasi karena melakukan perbaikan terhadap data yang rusak, hilang atau kesalahan kirim.

Hal ini dilakukan dengan memberikan nomor urut pada setiap oktet yang dikirimkan dan membutuhkan sinyal jawaban positif dari penerima berupa sinyal ACK (*acknowledgment*). Jika sinyal ACK ini tidak diterima pada interval pada waktu tertentu, maka data akan dikirimkan kembali. Pada sisi penerima, nomor urut tadi berguna

untuk mencegah kesalahan urutan data dan duplikasi data. TCP juga memiliki mekanisme *flow control* dengan cara mencantumkan informasi dalam sinyal ACK mengenai batas jumlah oktet data yang masih boleh ditransmisikan pada setiap segment yang diterima dengan sukses.

Dalam hubungan VoIP, TCP digunakan pada saat *signaling*, TCP digunakan untuk menjamin setup suatu call pada sesi *signaling*. TCP tidak digunakan dalam pengiriman data suara pada VoIP karena pada suatu komunikasi data VoIP penanganan data yang mengalami keterlambatan lebih penting dari pada penanganan paket yang hilang.

## **2. UDP (*User Datagram Protocol*)**

UDP yang merupakan salah satu protokol utama diatas IP merupakan transport protokol yang lebih sederhana dibandingkan dengan TCP. UDP digunakan untuk situasi yang tidak mementingkan mekanisme reliabilitas. *Header* UDP hanya berisi empat *field* yaitu *source port*, *destination port*, *length* dan *UDP checksum* dimana fungsinya hampir sama dengan TCP, namun fasilitas *checksum* pada UDP bersifat opsional.

UDP pada VoIP digunakan untuk mengirimkan audio stream yang dikirimkan secara terus menerus. UDP digunakan pada VoIP karena pada pengiriman audio streaming yang berlangsung terus menerus lebih mementingkan kecepatan pengiriman data agar tiba di tujuan tanpa memperhatikan adanya paket yang hilang walaupun mencapai 50% dari jumlah paket yang dikirimkan.

Karena UDP mampu mengirimkan data streaming dengan cepat, maka dalam teknologi VoIP UDP merupakan salah satu protokol penting yang digunakan sebagai header pada pengiriman data selain RTP dan IP. Untuk mengurangi jumlah paket yang hilang saat pengiriman data (karena tidak terdapat mekanisme pengiriman ulang) maka pada teknologi VoIP pengiriman data banyak dilakukan pada *private*

*network.*

### 3. IP (*Internet Protocol*)

*Internet Protocol* didesain untuk interkoneksi sistem komunikasi komputer pada jaringan *packet-switched*. Pada jaringan TCP/IP, sebuah komputer diidentifikasi dengan alamat IP. Tiap-tiap komputer memiliki alamat IP yang unik, masing-masing berbeda satu sama lainnya. Hal ini dilakukan untuk mencegah kesalahan pada transfer data. Terakhir, protokol data akses berhubungan langsung dengan media fisik. Secara umum protokol ini bertugas untuk menangani pendeteksian kesalahan pada saat transfer data. Untuk komunikasi datanya, Internet Protokol mengimplementasikan dua fungsi dasar yaitu *addressing* dan fragmentasi.

Salah satu hal penting dalam IP dalam pengiriman informasi adalah metode pengalamatan pengirim dan penerima. Saat ini terdapat standar pengalamatan yang sudah digunakan yaitu IPv4 dengan alamat terdiri dari 32 bit. Jumlah alamat yang diciptakan dengan IPv4 diperkirakan tidak dapat mencukupi kebutuhan pengalamatan IP sehingga dalam beberapa tahun mendatang akan diimplementasikan sistem pengalamatan yang baru yaitu IPv6 yang menggunakan sistem pengalamatan 128 bit.

#### 2.3.7 Format Paket VoIP

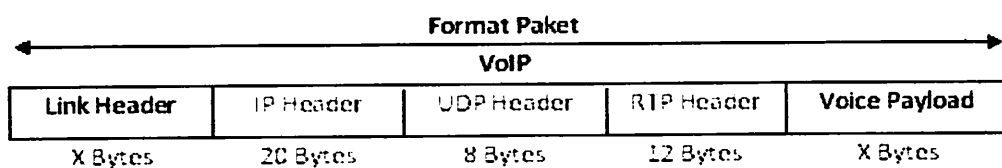
Tiap paket VoIP terdiri atas dua bagian, yakni header dan *payload* (beban). Header terdiri atas IP header, *Real-time Transport Protocol* (RTP) header, *User Datagram Protocol* header.

IP header bertugas menyimpan informasi routing untuk mengirimkan paket-paket ke tujuan. Pada tiap header IP disertakan tipe layanan atau *type of service* (ToS) yang memungkinkan paket tertentu seperti paket suara diperlakukan berbeda dengan paket yang *non real time*.

UDP header memiliki ciri tertentu yaitu tidak menjamin paket akan mencapai tujuan sehingga UDP cocok digunakan pada aplikasi *real*

*time* yang sangat peka terhadap delay.

RTP header adalah header yang dapat dimanfaatkan untuk melakukan *framing* dan segmentasi data *real time*. Seperti UDP, RTP juga mendukung realibilitas paket untuk sampai di tujuan. RTP menggunakan protocol kendali yang mengendalikan RTCP (*real-time transport control protocol*) yang mengendalikan QoS dan sinkronisasi media stream yang berbeda.



Gambar 2.13 Format Paket Voip

Untuk link header, besarnya sangat bergantung pada media yang digunakan. Tabel berikut menunjukkan ukuran header untuk media yang berbeda dengan metode kompersi G.729.

Tabel 2.3 Ukuran header pada kompersi G.729

Media	Link Layer Header Size	Bit Rate
Ethernet	14 Byte	29.6 kbps
PPP	6 Byte	26.4 kbps
Frame Relay	4 Byte	25.6 kbps
ATM	5 Byte	42.4 kbps

### 2.3.8 Komponen VoIP

Komponen-komponen VoIP terdiri dari user agent, proxy, protokol VoIP, codec dan lain-lain. Komponen-komponen tersebut adalah komponen yang dibutuhkan untuk komunikasi VoIP.

#### 1. User Agent

*User agent* merupakan sistem akhir (*end system*) yang digunakan untuk berkomunikasi. User agent terdiri atas dua bagian, yaitu:

- *User agent* berbasis software (*softphone*)

Contoh-contoh *user agent* berbasis *software* :

1. *Softphone SIP*

- X-Lite
- SJphone

2. *Softphone IAX*

- Idefisk
- IaxLite

3. *Softphone H.323*

- Netmeeting

- User Agent berbasis hardware

1. *IP Phone*

Berbentuk seperti telepon biasa, terhubung langsung ke jaringan IP (tidak melalui perangkat lain)

2. *USB Phone*

Berbentuk seperti telepon genggam, menggunakan kabel dan terhubung ke PC melalui port USB

3. *Internet Telephony Gateway (ITG)*

ITG mempunyai beberapa port, port-port itu terdiri dari FXS dan FXO. *Port* FXS terhubung ke telepon biasa dan FXO terhubung ke PSTN langsung atau melalui PABX.

4. *Analog Telephone Adaptor (ATA)*

ATA adalah telepon biasa yang dihubungkan ke ITG melalui port FXS.

2. **Proxy**

Proxy merupakan komponen penengah antar *user agent*, bertindak sebagai server yang menerima *request message* dari *user agent* dan menyampaikan pada *user agent* lainnya. Contoh-contoh aplikasi *proxy VoIP server* :



- *Open Source*
  - Asterisk (<http://www.asterisk.org>)
  - OpenSER (<http://www.openser.org>)
  - SER (<http://www.iptel.org/ser/>)
  - Yate (<http://yate.null.ro>)
- *Non Open Source*
  - Axon (<http://www.nch.com.au/pbx/>)
  - OnDO SIP Server (<http://www.brekeke.com/>)

### 3. Protokol VoIP

Protokol-protokol pada teknologi VoIP adalah sebagai berikut:

- H.323
- SIP (*session initiation protocol*)
- MGCP (*media gateway control protocol*)
- IAX

#### 2.3.9 Codec

Codec adalah kependekan dari *compression/decompression*, mengubah signal audio dan dimampatkan ke bentuk data digital untuk ditransmisikan kemudian dikembalikan lagi ke bentuk signal audio seperti data yang dikirim. Codec berfungsi untuk penghematan *bandwidth* di jaringan. Codec melakukan perubahan dengan cara *sampling signal audio* sebanyak 1000 kali per detik. Sebagai gambaran G.711 codec men-sample signal audio 64.000 kali per detik. Kemudian merubahnya ke bentuk data digital dan di mampatkan kemudian ditransmisikan.

Beberapa jenis rata-rata waktu men-sampling VoIP untuk codec yang sering digunakan:

- *64,000 times per second*
- *32,000 times per second*
- *8,000 times per second*

Contoh-contoh codec :

- *Open Source dan Free*
  - GSM (*codec bit rate 13,2Kbps*)

- iLBC (*codec bit rate 15,2Kbps*)
- G711 (*codec bit rate 64Kbps*)
- *Licensed*
  - G729 (*codec bit rate 8Kbps*)
  - G723 (*codec bit rate 5,3Kbps*)

## 2.4 SIP (*Session Initiation Protocol*)

*Session Initiation Protocol* atau disingkat SIP adalah suatu protokol yang dikeluarkan oleh group yang tergabung dalam *Multiparty Multimedia Session Control* (MMUSIC) yang berada dalam organisasi *Internet Engineering Task Force* (IETF) yang di dokumentasikan ke dalam dokumen *request for command* (RFC) 2543 pada bulan Maret 1999. SIP merupakan protocol yang berada pada layer aplikasi yang mendefinisikan proses awal, pengubahan, dan pengakhiran (pemutusan) suatu sesi komunikasi multimedia. Sesi komunikasi ini termasuk hubungan multimedia, *distance learning*, dan aplikasi lainnya.

SIP berkarakteristik *client-server*, karena *request* diberikan oleh client dan *request* ini dikirimkan ke server. Kemudian server mengolah *request* dan memberikan tanggapan (*response*) terhadap *request* tersebut ke client. *Request* dan tanggapan (*response*) terhadap *request* disebut transaksi SIP.

SIP saat ini dipandang sebagai protokol signaling yang baik untuk *Voice over IP* (VoIP). SIP juga dipandang sebagai pengganti dari protokol H.323 yang sebelumnya digunakan juga dalam teknologi VoIP.

### 2.4.1 Fungsi *Session Initiation Protocol* (SIP)

Adapun fungsi SIP terdiri dari tiga fungsi, yaitu:

#### 1. *Call Initiation*

- Membangun sebuah sesi komunikasi
- Negosiasi media *transfer protocol*
- Mengundang user agent lain untuk bergabung di dalam sesi komunikasi.

## 2. *Call Modification*

- Bila perlu, SIP dapat memodifikasi sesi komunikasi

## 3. *Call Termination*

- Menutup sesi komunikasi

### 2.4.2 Susunan Protokol SIP

SIP adalah *signalling protocol*, bukan *media transfer protocol*, sehingga SIP tidak membawa paket data voice atau video. Sehingga protokol SIP didukung oleh beberapa protokol lain, diantaranya RSVP (*Resource Reservation Protocol*) yang bertugas untuk melakukan pemesanan pada jaringan, RTP (*real-time transport protocol*) dan RTCP (*real-time transport control protocol*) yang bertugas mentransmisikan media dan mengetahui kualitas layanan, serta SDP (*session description protocol*) bertugas untuk mendeskripsikan sesi media. Secara *default* SIP menggunakan protokol UDP tetapi pada beberapa kasus dapat juga menggunakan TCP sebagai *protocol transport*.

- **RSVP (*Resource Reservation Protocol*)**

RSVP bekerja pada *layer transport*. Digunakan untuk menyediakan bandwidth agar data suara yang dikirimkan tidak mengalami *delay* ataupun kerusakan saat mencapai alamat tujuan *unicast* maupun *multicast*.

- **RTP (*Real-time Transport Protocol*)**

Protokol RTP menyediakan transfer media secara *real-time* pada jaringan paket. Protokol RTP menggunakan protokol UDP dan header RTP mengandung informasi kode bit yang spesifik pada tiap paket yang dikirimkan, hal ini membantu si penerima untuk melakukan antisipasi jika terjadi paket yang hilang.

- **RTCP (*Real-time Transport Control Protocol*)**

Protokol RTCP merupakan protokol yang mengendalikan transfer media. Protokol ini bekerja sama dengan protokol RTP. Dalam satu sesi komunikasi, protokol RTP mengirimkan paket

RTCP secara periodik untuk memperoleh informasi transfer media dalam memperbaiki kualitas layanan.

- **SDP (*Session Description Protocol*)**

Protokol SDP merupakan protokol yang mendeskripsikan media dalam suatu komunikasi. Tujuan protokol SDP adalah untuk memberikan informasi aliran media dalam satu sesi komunikasi agar penerima yang menerima informasi tersebut dapat berkomunikasi. Hal-hal yang dicakup dalam protokol SDP, antara lain:

1. Nama sesi komunikasi dan tujuannya
2. Waktu sesi (jika aktif)
3. Media dalam sesi komunikasi
4. Informasi bagaimana cara menerima media (misal: *port*, *format*, dan sebagainya)
5. *Bandwidth* yang di gunakan dalam komunikasi

### 2.4.3 Komponen SIP

Dalam hubungannya dengan IP Telephony, ada dua komponen yang ada dalam sistem SIP, yaitu:

1. ***User agent***

User agent terdiri dari dua komponen utama yaitu:

- *User Agent Client (UAC)*  
User agent ini bertugas memulai sesi komunikasi
- *User Agent Server (UAS)*  
User agent ini bertugas menerima atau menanggapi sesi komunikasi.

2. ***Network Server***

Agar user pada jaringan SIP dapat memulai suatu panggilan dan dapat pula di panggil, maka user terlebih dahulu harus melakukan registrasi agar lokasinya dapat di ketahui. Registrasi dapat dilakukan dengan mengirimkan pesan *REGISTER* ke server SIP. Lokasi user dapat berbeda-beda sehingga untuk mendapatkan lokasi user

yang aktual diperlukan location server. Pada jaringan SIP, ada tiga tipe network server, yaitu:

- ***Proxy Server***

*Proxy Server* adalah komponen penengah antar user agent, bertindak sebagai server dan client yang menerima *request message* dari user agent dan menyampaikan pada user agent lainnya. Request dapat dilayani sendiri atau disampaikan (*forward*) pada *proxy server* lain. *Proxy server* bertugas menerjemahkan dan/atau menulis ulang *request message* sebelum menyampaikan pada user agent tujuan atau proxy lain. Selain itu, proxy server bertugas menyimpan seluruh state sesi komunikasi antara UAC dan UAS.

- ***Redirect server***

Komponen ini adalah komponen yang menerima *request message* dari *user agent*, memetakan alamat SIP *user agent* atau proxy server tujuan kemudian menyampaikan hasil pemetaan kembali pada user agent pengirim (UAC). Redirect Server tidak menyimpan state sesi komunikasi antara UAC dan UAS setelah pemetaan disampaikan pada UAC. Redirect server berbeda dengan proxy server, yaitu tidak dapat memulai inisiasi *request message*. Selain itu, redirect server tidak dapat menerima dan menutup sesi komunikasi.

- ***Registar Server***

Registar server adalah komponen yang menerima *request message* REGISTER. Registar server menyimpan database user untuk otentikasi dan lokasi sebenarnya (berupa IP dan port) agar user yang terdapat dapat dihubungi oleh komponen SIP lainnya (berfungsi sebagai *Location Server* juga). Registar server dapat

dipasangkan dengan Proxy Server.

#### 2.4.4 SIP Request

Telah di bahas diatas, bahwa pesan *request* pada SIP dikirimkan dari client ke server. Ada enam tipe pesan *request*, antara lain:

##### 1. **INVITE**

Pesan ini digunakan untuk memulai suatu komunikasi atau mengundang user agent lain untuk bergabung dalam sesi komunikasi. *Message body* pesan INVITE berisikan deskripsi media yang dapat digunakan dalam komunikasi.

##### 2. **ACK**

Pesan ini berfungsi memberitahukan bahwa client telah menerima tanggapan terakhir terhadap INVITE. *Message body* pada pesan ACK dapat membaca deskripsi media yang akan digunakan oleh user yang dipanggil.

##### 3. **BYE**

Pesan ini dikirim oleh client untuk mengakhiri komunikasi.

##### 4. **CANCEL**

Pesan CANCEL dikirim untuk membatalkan pesan *request* (INVITE).

##### 5. **OPTIONS**

Pesan ini dikirimkan oleh client ke server untuk mengetahui tentang informasi kemampuan server.

##### 6. **REGISTER**

*Client* dapat melakukan registrasi lokasinya dengan mengirimkan pesan REGISTER ke server SIP dimana server

yang menerima pesan REGISTER disebut SIP register.

### 2.4.5 SIP Response

Pesan respon di kirimkan setelah menerima pesan *request* yang menunjukkan status keberhasilan server. Pesan respon didefinisikan dengan tiga angka, angka pertama merupakan kelas respon (seluruhnya ada 6 kelas respon). Angka kedua dan ketiga menunjukkan arti dari respon tersebut. Table berikut menunjukkan arti kelas respon SIP

Tabel 2.4 Response SIP

Kelas Respon	Jenis Respon	Kategori Respon
1xx	Informational Message	Provisional
2xx	Successful Response	Final
3xx	Redirection Response	Final
4xx	Request Failure Response	Final
5xx	Server Failure Response	Final
6xx	Global Failures Response	Final

Pesan respon terbagi atas dua kategori, yakni:

#### 1. *Provisional response.*

Respon ini merupakan respon yang dikirim oleh server untuk menunjukkan proses sedang berlangsung, tapi tidak mengakhiri transaksi SIP.

#### 2. *Final reponse*

Respon ini merupakan respon yang mengakhiri transaksi SIP.

### 2.4.6 SIP Header

Header SIP ditulis dalam format teks, mirip dengan HTTP dan SMTP. Berikut adalah contoh SIP Header :

```
INVITE sip:102@192.168.0.1 SIP/2.0
```

```

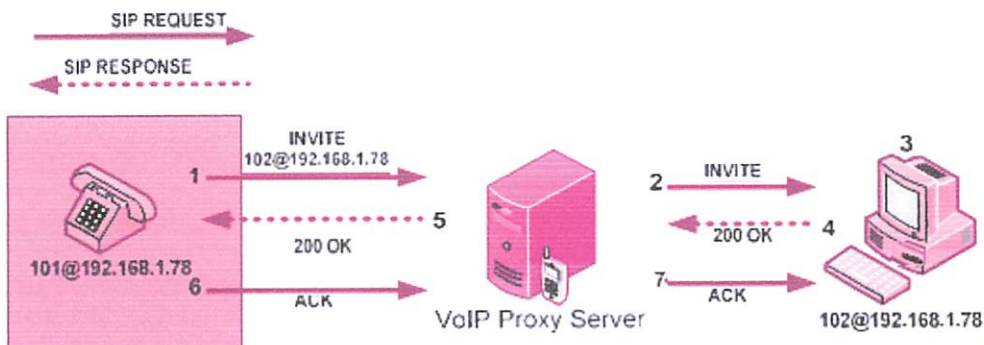
Via: SIP/2.0/UDP 192.168.1.78:5060;branch=z9hG4bK1385dfa1
From: "line1" <sip:101@192.168.1.78>;tag=as1da29517
To: <sip:102@192.168.0.1>
Contact: <sip:101@192.168.1.78;transport=UDP>
Call-ID: 362711e7794362431acdcd3a36f02774@192.168.1.78

CSeq: 102 INVITE
User-Agent:
Asterisk PBX
Max-Forwards: 70

Date: Mon, 05 Feb 2007 11:39:27 GMT
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER,
SUBSCRIBE, NOTIFY Content-Type: application/sdp...
...

```

## 2.4.7 Cara Kerja Protokol SIP



Gambar 2.14 Cara Kerja Protokol SIP

Secara umum langkah demi langkah standar/prosedur interaksi *internet telephony* yang normal, adalah:

1. Pemanggil akan mengirimkan sinyal INVITE ke proxy server.
2. Proxy server akan meneruskan message INVITE ke tujuan.
3. Bell akan berbunyi di komputer tujuan.
4. Jika tujuan ternyata bersedia menerima, maka tujuan akan



- mengirimkan message OK ke proxy server.
5. Proxy server akan meneruskan *message* OK ke pemanggil.
  6. Telepon pemanggil akan memberikan *message acknowledge* (ACK) ke proxy server.
  7. Proxy server akan meneruskannya ke mesin tujuan yang benar.
  8. Setelah proses pembentukan sambungan ini terbentuk maka hubungan komunikasi suara akan terjadi.

## 2.5 SNMP

Sejak dipublikasikan pertama kali pada tahun 1988, SNMP sudah menjadi alat manajemen jaringan yang banyak digunakan pada jaringan berbasis TCP/IP. SNMP mendefinisikan protokol untuk pertukaran informasi manajemen, SNMP juga mendefinisikan format untuk merepresentasikan informasi manajemen dan kerangka kerja untuk mengorganisasikan sistem distribusi kedalam sistem manajemen dan pengaturan agent. Selain itu, beberapa dari struktur basis data yang spesifik yang dinamakan MIB sudah didefinisikan sebagai bagian dari SNMP. MIB ini menspesifikasikan obyek-obyek yang telah diatur untuk subyek manajemen jaringan yang paling umum termasuk bridge, router dan LAN.

Pertumbuhan popularitas yang cepat dari SNMP pada akhir tahun 1980-an dan awal tahun 1990-an membuat kesadaran akan kekurangan-kekurangan yang ada pada SNMP, hal ini termasuk kedalam kategori deefisiensi fungsional, seperti ketidakmampuan menspesifikasikan *bulk-data transfer* dengan mudah, defisiensi keamanan, misalnya tidak adanya mekanisme autentifikasi dan privasi.

Untuk memperbaikinya beberapa group independen memulai mengerjakan peningkatan keamanan pada SNMPv2. Terdapat dua pendekatan yang muncul yang saling bersaing satu dengan lainnya yaitu SNMPv2u dan SNMPv2\* yang mana keduanya menjadi masukan bagi IETF SNMPv3 yang baru.

Hal yang sangat penting untuk diketahui, bahwa SNMPv3 bukan pengganti dari SNMPv1 dan atau SNMPv2. SNMPv3 mendefinisikan kemampuan keamanan yang dapat digunakan bersama dengan SNMPv2 atau SNMPv1.

### 2.5.1 Konsep Dasar SNMP

Ide dasar dari setiap manajemen jaringan adalah bahwa terdapat dua tipe sistem pada setiap jaringan terkonfigurasi yaitu : agen dan manajer. Yang ditempatkan pada setiap titik pada jaringan yang akan diatur, termasuk PC, Workstation, sever, bridge , router dan lainnya termasuk modul agen. Agen bertanggung jawab untuk:

- Mengumpulkan dan memelihara tentang lingkungan lokalnya.
- Menyediakan informasi tersebut kepada manajer, baik itu sebagai response ataupun tidak.
- Merespon terhadap perintah manajer untuk mengubah konfigurasi lokal atau parameter-parameter operasi.

Suatu konfigurasi juga termasuk satu atau lebih manajemen stasiun atau manajer. Manajer stasiun secara umum menyediakan antarmuka bagi pengguna sehingga manajer jaringan (manusia) dapat mengontrol dan melakukan observasi terhadap proses manajemen jaringan. Antarmuka ini mengijinkan pengguna untuk melakukan perintah (seperti mematikan link, mengumpulkan statistik dsb) dan menyediakan logika untuk proses rangkuman pemformatan informasi yang dikumpulkan oleh sistem.

Inti dari sistem manajemen jaringan adalah suatu set aplikasi yang memberikan pelayanan terhadap kebutuhan untuk manajemen jaringan. Minimal sistem akan memasukan aplikasi dasar untuk memonitor unjuk kerja, kontrol konfigurasi dan akunting. Sistem yang lebih canggih akan memasukan aplikasi pada kategori tersebut diatas ditambah fasilitas untuk isolasi kesalahan dan perbaikan dan untuk pengaturan fitur keamanan dari jaringan.

Semua aplikasi jaringan pada umumnya berbagi protokol manajemen jaringan yang umum. Protokol ini menyediakan fungsi- fungsi fundamental untuk mengambil informasi manajemen dari agent dan mengirimkan perintah kepada agent. Protokol ini kemudian menggunakan fasilitas komunikasi seperti TCP/IP atau OSI.

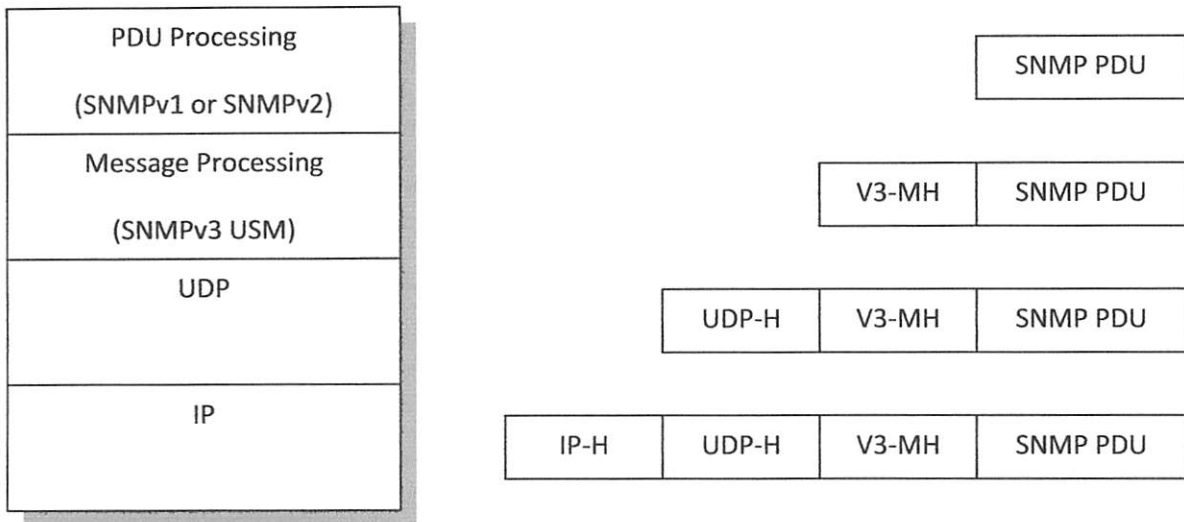
Akhirnya setiap agent memelihara basis informasi manajemen yang berisi informasi terbaru dan yang sebelumnya tentang konfigurasi dan

lalulintas lokalnya. Manajemen station akan memelihara basis informasi manajemen global dengan informasi berisi rangkuman dari semua agent.

Spesifikasi SNMPv1 dan SNMPv2 terdiri dari dokumen- dokumen yang mendefinisikan protokol manajemen jaringan, struktur umum dari basis informasi manajemen dan beberapa struktur data yang spesifik untuk tujuan manajemen yang spesifik pula. Spesifikasinya termasuk, minimal, aplikasi manajemen jaringan dan tanpa fasilitas presentasi pengguna. Banyak juga vendor- vendor yang sudah menyediakan aplikasi manajemen jaringan untuk berjalan diatas SNMP.

Perintah-perintah operasi SNMP sangat mudah. SNMP dirancang untuk mudah diimplementasikan dan membutuhkan sedikit prosesor dan sumberdaya jaringan. Oleh karena itu dapat digunakan sebagai alat untuk membangun 'bare-bone' fasilitas manajemen jaringan. Pada dasarnya protokol ini menyediakan fungsi sebagai berikut:

- **Get:** digunakan oleh manajer untuk mengambil suatu item dari agen MIB.
- **Set:** digunakan oleh manajer untuk men-set atau mengisikan harga suatu variabel pada agen MIB.
- **Trap :** digunakan oleh agent untuk mengirim peringatan kepada manajer.
- **Inform:** digunakan oleh manajer untuk mengirimkan peringatan kepada manejer yang lain.



Gambar 2.15 Alur Paket SNMP

Apa yang diberikan SNMP adalah struktur MIB standar sangat '*extensive*'. MIB pada agen memberikan petunjuk informasi apa yang dapat dikumpulkan dan disimpan oleh agent tersebut. Sebagai contoh, banyak sekali variabel pada MIB dasar yang berhubungan erat pada operasi TCP/IP yang ada dibawahnya, termasuk jumlah paket yang dikirim dan diterima, paket error dan seterusnya. Semua agen memelihara pasangan data- data variabel ini dan dapat dibuat suatu aplikasi pada manajemen station untuk mengeksploitasi informasi ini.

### 2.5.2 Arsitektur SNMP

Arsitektur SNMP terdiri dari kumpulan interaksi entitas dari SNMP yang terdistribusi. Setiap entitas mengimplementasikan bagian dari kemampuan SNMP dan bisa berperan sebagai node agent, node manajer, atau kombinasi dari keduanya. Setiap entitas SNMP berisi kumpulan dari modul- modul yang saling berinteraksi untuk menyediakan suatu layanan. Interaksi ini dapat dimodelkan sebagai primitif abstrak dan parameter.

Arsitektur dari RFC 2271 mencerminkan persyaratan rancangan kunci bagi SNMPv3: suatu rancangan arsitektur yang modular yang akan:

- Mengijinkan implementasi pada operasi lingkungan yang luas, dengan kebutuhan minimal, fungsi yang tidak mahal dan beberapa dapat mendukung fitur- fitur tambahan untuk mengatur jaringan yang besar.

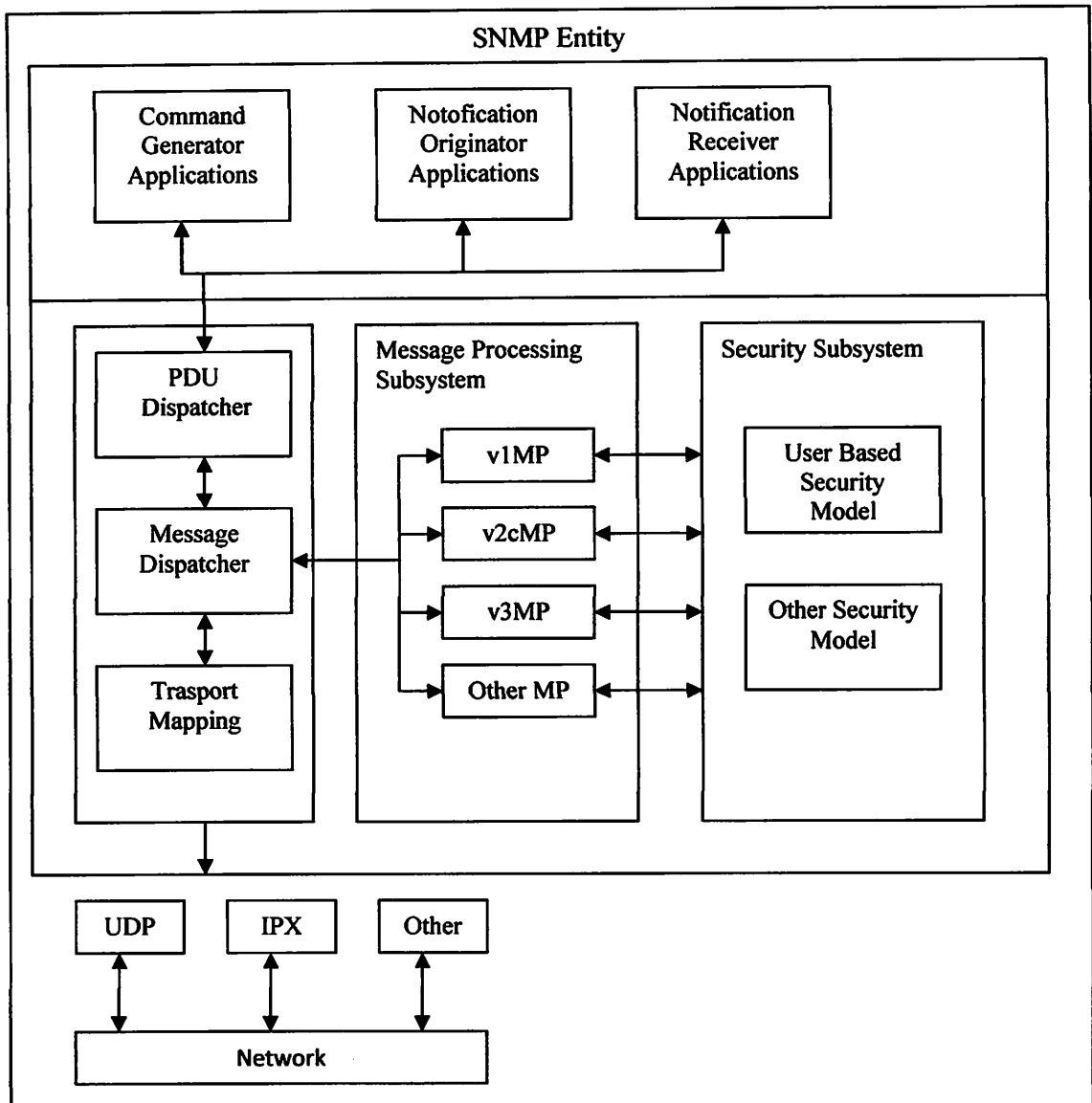
- Memungkinkan untuk memindahkan sebagian dari arsitektur pada 'standard track' walaupun jika konsensus tidak dapat diterima oleh semua bagian.
- Mengakomodasi model keamanan alternatif.

### 2.5.3 Entitas SNMP

Setiap entitas dari SNMP mengikutsertakan satu mesin SNMP. Suatu SNMP mesin mengimplementasikan fungsi untuk mengirimkan dan menerima pesan, melakukan autentifikasi dan melakukan enkripsi dan dekripsi suatu pesan dan juga mengontrol akses terhadap obyek-obyek yang ditangani. Fungsi- fungsi ini disediakan sebagai pelayanan terhadap satu atau lebih aplikasi yang dikonfigurasi dengan mesin SNMP untuk membentuk entitas SNMP.

Baik mesin SNMP dan aplikasi yang didukungnya keduanya didefinisikan sebagai kumpulan dari modul diskret. Arsitektur ini memberikan sejumlah keuntungan. Pertama, seperti yang dapat kita lihat, aturan dari entitas SNMP ditentukan oleh modul mana yang diimplementasikan pada entitas tersebut. Sebagai contoh, beberapa set modul dibutuhkan oleh *agent*, sementara beberapa set modul yang berbeda dibutuhkan oleh manajer SNMP. Kedua, struktur yang modular dari spesifikasi memungkinkan mendefinisikan setiap modul dalam versi yang berbeda-beda. Ini memungkinkan :

- Mendefinisikan alternatif atau meningkatkan kemampuan pada beberapa aspek dari SNMP tanpa harus membuat versi yang baru untuk keseluruhan SNMP
- Secara jelas menspesifikasikan keberadaan dan strategi transisi.



Gambar 2.16 Manajer SNMP Tradisional

Untuk mendapatkan pemahaman yang lebih baik terhadap aturan dari setiap modul dan hubungannya pada modul yang lain, sangat baik untuk melihat pada kegunaannya pada manajer dan agen SNMP tradisional. Istilah tradisional diartikan sebagai asli digunakan untuk menekankan fakta bahwa implementasi yang diberikan tidak harus manajer atau agent yang asli, tetapi mungkin sebuah modul yang memungkinkan entitas melakukan tugas baik manajemen ataupun agent.

## 2.6 Sistem Operasi Linux

Linux adalah sistem operasi yang pertama kali diciptakan oleh Linus Torvalds pada tahun 1991. Linux bersifat *open source* yang *free* dibawah lisensi GPL (GNU Public License), yang merupakan sistem operasi 32-64bit, dan dapat dijalankan pada berbagai platform baru. Linux itu sendiri adalah sistem operasi *multi-user* yang efisien, selain itu juga linux mampu mengerjakan lebih dari satu tugas pada waktu yang bersamaan, atau lebih sering kita sebut dengan sebutan *multitasking*.

Sistem operasi linux terdiri dari kernel, program sistem, dan beberapa program aplikasi. Kernel merupakan inti dari sistem operasi yang mengatur pengerjaan memori, piranti masukan keluaran, proses-proses, pemakaian file pada file sistem dan lain-lain.

Kernel linux terdiri dari beberapa bagian penting, seperti: manajemen proses, manajemen memori, *hardware device drivers*, *filesystem drivers*, manajemen jaringan dan lain-lain. Namun bagian yang terpenting dari kernel ialah manajemen proses dan manajemen memori. Manajemen memori menangani daerah pemakaian memori, daerah swap, bagian-bagian kernel dan untuk *buffer cache*. Manajemen proses menangani pembuatan proses-proses dan penjadwalan proses. Pada bagian dasar kernel berisi hardware device drivers untuk setiap jenis hardware yang didukung.

Program sistem dan semua program-program lainnya yang berjalan di atas kernel disebut *user mode*. Perbedaan mendasar antara program sistem dan program aplikasi adalah program sistem dibutuhkan agar suatu sistem operasi dapat berjalan sedangkan program aplikasi adalah program yang dibutuhkan untuk menjalankan suatu aplikasi tertentu. Contoh dari program sistem adalah daemon, sedangkan contoh dari program aplikasi adalah pengolah kata (*word processor*).

### 2.6.1 Linux Trixbox

Linux Trixbox (sebelumnya asterisk@home) adalah sebuah VoIP *phone system* yang mudah diinstalasi, memanfaatkan

software Asterisk PBX sebagai basis. Didesain baik untuk umum, maupun pengguna pribadi menyertakan CentOS Linux, SugarCRM, MySQL, HUDlite, freePBX, dan banyak paket software lainnya.

Trixbox dapat dikonfigurasi untuk menangani mulai dari satu sambungan telepon pribadi, puluhan atau ratusan sambungan telepon untuk perkantoran, dapat disambungkan ke beberapa saluran T1 di sebuah *call center* yang menghandle jutaan menit percakapan per bulan. Sebuah web GUI (*web based*) memungkinkan konfigurasi dan pengoperasian menjadi mudah.

### 2.6.2 Fitur Linux Trixbox

Fitur dari linux Trixbox adalah sebagai berikut:

1. AMP (*Asterisk Management Portal*) fitur ini adalah sebuah fitur yang sangat luar biasa, karena konfigurasi bisa dilakukan lewat *interface* web tanpa harus mengedit file konfigurasi.
2. ARI (*Asterisk Recording Interface*) fitur ini berfungsi untuk menyimpan percakapan, baik percakapan keluar (*outgoing*) maupun kedalam (*incoming*).
3. Flash Operator Panel adalah sebuah fitur yang berguna untuk monitor semua ekstensi secara *real time* berbasis web.
4. Cisco XML Services
5. Music On Hold (mpg123) Trixbox menggunakan mpg123 untuk *music on hold*.
6. Fax support (SpanDSP) adalah suatu fitur yang memperbolehkan Asterisk untuk menerima fax.

### 2.7 Pemrograman Borland Delphi 7

Delphi adalah kompiler / penterjemah bahasa Delphi (awalnya dari Pascal) yang merupakan bahasa tingkat tinggi sekelas dengan Basic atau C yang merupakan produk dari Borland corp. Bahasa Pemrograman di Delphi disebut



bahasa procedural artinya bahasa/sintaknya mengikuti urutan tertentu / prosedur. Delphi termasuk Keluarga Visual sekelas Visual Basic, Visual C, artinya perintah-perintah untuk membuat objek dapat dilakukan secara visual. Pemrogram tinggal memilih objek apa yang ingin dimasukkan kedalam Form/Window, lalu tingkah laku objek tersebut saat menerima event/aksi tinggal dibuat programnya. Ada jenis pemrograman non-prosedural seperti pemrograman untuk kecerdasan buatan seperti bahasa Prolog. Delphi merupakan bahasa berorientasi objek, artinya nama objek, properti dan *methode/procedure* dikemas menjadi satu kemasan (*encapsulate*).

Pada pemrograman berbasis windows, kita akan diperhadapkan pada satu atau beberapa jendela yang nampak dihadapan kita. Jendela ini dalam Delphi disebut juga dengan form. Delphi adalah sebuah perangkat lunak (bahasa pemrograman) untuk membuat program / aplikasi komputer berbasis windows. Delphi merupakan bahasa pemrograman berbasis objek, artinya semua komponen yang ada merupakan objek-objek.

Ciri sebuah objek adalah memiliki nama, properti dan method/procedure. Delphi disebut juga visual programming artinya komponen-komponen yang ada tidak hanya berupa teks (yang sebenarnya program kecil) tetapi muncul berupa gambar-gambar.

## BAB III

### ANALISA DAN DESAIN SISTEM

#### 3.1 Analisa Sistem

Pada bab ini dijelaskan mengenai analisa dan desain sistem aplikasi. Analisis ditujukan untuk memberikan gambaran secara umum terhadap aplikasi. Hal ini berguna untuk menunjang desain sistem aplikasi yang akan dikerjakan sehingga kebutuhan akan aplikasi tersebut dapat diketahui sebelumnya. Kemudian hasil analisa akan menjadi dasar untuk melakukan desain sistem atau perancangan aplikasi sesuai kebutuhan sistem.

Untuk dapat berkomunikasi antar *user*, maka memerlukan IP PBX. Sehingga setiap terjadi komunikasi antar *user*, *user* yang sudah terregistrasi tentunya, Data komunikasi yang terjadi akan terekam pada aplikasi yang telah dibuat pada aplikasi berbasis windows, lebih khususnya dikembangkan dengan menggunakan pemrograman Borland Delphi dan menggunakan protokol SNMP yang dipasang dan dikonfigurasi pada Server IP PBX.

Pada protokol SNMP agar dapat saling berkomunikasi antar manager dan agent untuk mendapatkan informasi *traffic*, maka SNMP memerlukan protokol. Cara yang biasa dipakai SNMP adalah manager mengirim permintaan (*request*) ke *agent* tentang informasi *traffic* berdasarkan *Object Identifier* (OID) atau memberikan perintah untuk melakukan pembaharuan keadaan *agent* dengan cara – cara tertentu. Idealnya *agent* cukup menjawab pertanyaan yang di minta atau mengkonfirmasi bahwa *agent* telah melakukan pembaharuan keadaan sesuai dengan permintaan manager. Jenis pesan – pesan SNMP dapat dilihat pada tabel berikut ini.

Tabel 3.1 Jenis pesan SNMP

Pesan	Deskripsi
Get – Request	Meminta nilai sebuah variabel atau lebih
Get - Next - Request	Meminta variabel setelah variabel saat itu
Get - Walk - Request	Meminta semua nilai variabel yang didefinisikan MIB
Get - Bulk - Request	Mengambil sebuah tabel berukuran besar
Set - Request	Memperbarui sebuah variabel atau lebih

Inform - Request	Pesan manajer ke manajer yang menjelaskan MIB lokal
SNMPV2 - Trap	Laporan tiap agen ke manager

### 3.1.1 Perintah yang digunakan untuk mengakses Agent

Perintah – perintah ini dijalankan oleh manager. *Software* yang dijalankan perintah ini biasanya sudah ada waktu kita menginstall *agent*. Selain itu juga terdapat *software* khusus yang menyediakan perintah-perintah untuk mengakses *agent*. Perintah dasar-dasar tersebut antara lain *Snmget*, *Snmgetnext*, *Snmwalk* dan *Snmtrap*. Dalam implmentasi aplikasi *monitoring traffic* ini, kita menggunakan dua dari empat jenis pesan SNMP tersebut untuk mengakses informasi dari agent, yaitu *Snmget* dan *Snmgetnext*.

Perintah *Snmget* digunakan untuk mengambil nilai sebuah variabel MIB dari sebuah *agent*. Kita harus tahu identifikasi kejadian secara tepat. Aturan penulisan perintahnya adalah sebagai berikut :

***Snmget hostname community\_name MIB\_object\_instance***

Perintah *snmpgetnext* digunakan untuk mengambil nilai sebuah *object* kejadian yang disebutkan dalam perintah (***MIB\_object\_instance***), Aturan penulisannya sebagai berikut :

***Snmgetnext hostname community\_name MIB\_object\_instance***

Hasilnya adalah nilai *object* kejadian yang urutannya setelah *object* kejadian yang disebutkan dalam *MIB\_object\_instance* tersebut.

### 3.1.2 Mekanisme Pendefinsian MIB

MIB standar yang digunakan dalam implementasi adalah MIB II yang didefinisikan pada RFC 1213 dan berisi 171 buah objek. Untuk memudahkan objek-objek dikelompokan dalam sepuluh kategori yang

berkaitan dengan sepuluh simpul yang berada dibawah MIB-II. Kesepuluh kategori tersebut dimaksudkan untuk menyediakan basis tentang apa saja yang harus dimengerti stasiun manajemen. Kategori tersebut dapat kita lihat pada table dibawah ini.

Tabel 3.2 Kelompok Object Internet MIB

Kategori	Objek	Deskripsi
Sistem	7	Nama,Lokasi dan penjelasan peralatan.
Interface	23	Interface jaringan dan lalu-lintas yang terukur
At	3	Penerjemah alamat
IP	42	Statistik Paket IP
ICMP	26	Statistik tentang pesan-pesan ICMP yang diterima
TCP	19	Algoritma,parameter dan statistic TCP
UDP	6	Statistik lalu-lintas UDP
EGP	20	Statistik lalu-lintas protocol gateway luar
Transmission	0	Dicadangkan untuk media khusus MIB-MIB
SNMP	29	Statistik lalu-lintas SNMP

Penjelasan yang terkait dengan kelompok *object* internet MIB-II :

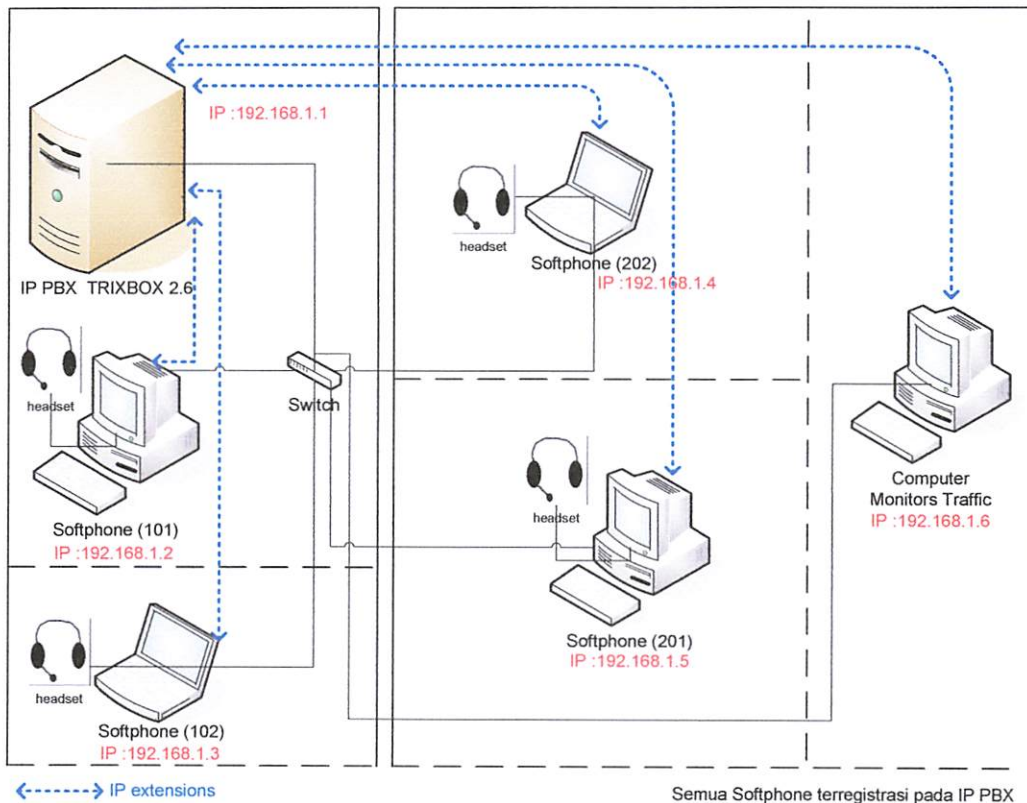
1. Kelompok *system* mengizinkan manajer untuk mengetahui nama peralatan, *hardware* dan *software* yang dikandung, lokasinya,dan tugas yang perlu dikerjakan,nama dan alamat administrator dari agent juga dapat diketahui pada kelompok ini sehingga dapat dihubungi jika ada masalah.
2. Kelompok *interface* berkaitan dengan adapter-adapter jaringan. Kelompok *interface* dapat mengetahui jumlah paket dan byte yang dikirimkan dan menerima dari jaringan, jumlah paket yang dibuang, jumlah *broadcast* dan ukuran antrian *output* saat itu.
3. Kelompok AT dapat ditemukan dalam MIB-I dan menyediakan informasi tentang pemetaan alamat, misalnya alamat Ethernet ke alamat IP. Informasi ini dipindahkan ke *protocol-specific* MIB pada SNMPv2.

4. Kelompok IP berkaitan dengan lalu lintas IP yang keluar dan masuk dalam simpul. Kelompok IP sangat baik dalam perhitungan jumlah paket yang dibuang dengan masing-masing alasan, misalnya karena route tujuannya tidak dikenal. Statistik tentang fragmentasi dan penyusunan kembali datagram juga dapat diperoleh. Seluruh item ini sangat penting khususnya untuk pengelolaan router.
5. Kelompok ICMP berkaitan dengan pesan-pesan kesalahan IP. Pada dasarnya kelompok ICMP mempunyai *counter* bagi setiap pesan ICMP yang merekam jumlah jenis tertentu yang diketahui.
6. Kelompok TCP memonitor jumlah koneksi terbuka saat ini kumulatifnya, *segment-segment* yang dikirim dan diterima dan statistik berbagai kegagalan.
7. Kelompok UDP melakukan log jumlah datagram UDP yang dikirim dan diterima, dan jumlah datagram UDP yang tidak dapat diantarkan sehubungan port yang tidak diketahui atau karena alasan lain.
8. Kelompok EGP digunakan oleh router untuk mendukung *protocol gateway eksterior*. Kelompok ini menghitung jumlah paket yang keluar, paket yang masuk dan diteruskan secara benar, paket yang masuk dan dibuang.
9. Kelompok transmisi merupakan tempat kedudukan media spesifik MIB, Misalnya *statistic* spesifikasi Ethernet yang dapat ditampung. Tujuan memasukkan kelompok yang kosong ke dalam MIB-II adalah untuk mencadangkan identifikasi.
10. Kelompok SNMP ditujukan untuk mengumpulkan statistik tentang operasi SNMP itu sendiri, jumlah pesan yang dikirim atau jenis pesan.

### 3.2 Desain Sistem

Sistem yang akan dibuat pada penelitian ini adalah sebuah aplikasi yang dapat memberikan informasi yang diperuntukkan sebagai *monitoring traffic* dengan memanfaatkan protokol SNMP yang sudah terpasang dan terkonfigurasi pada jaringan VoIP (IP PBX), sehingga memudahkan administrator dan *user* untuk memonitoring adanya komunikasi antar *Users*, Dengan Tanpa Membuka

atau melihat keseluruhan fitur Aplikasi Trixbox melalui Webmin yang tersedia terlebih dahulu.



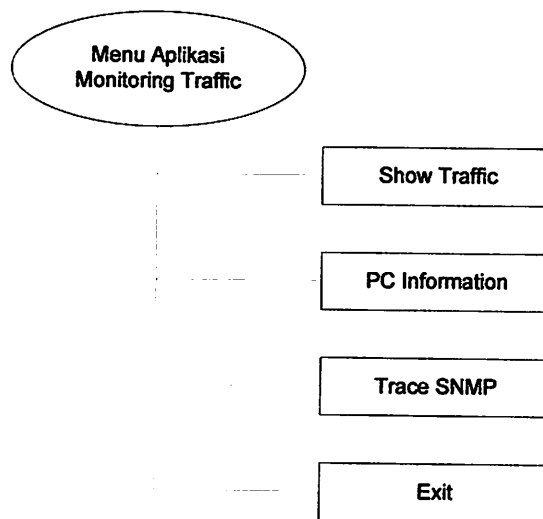
Gambar 3.1 Desain Sistem

Keterangan gambar dari desain sistem diatas:

1. VoIP Server merupakan IP PBX yang adalah sebuah sistem yang mempunyai fungsi utama menyediakan layanan Voip (*Voice Over Internet Protocol*) mulai dari registrasi *user*, *call routing*, *call conference*, *interactive voice response*, *call forwarding*, *caller id voice mail* dan sebagainya.
2. Dalam sebuah jaringan VoIP, selain terdapat IP PBX, juga terdapat beberapa *client* yang dapat saling berkomunikasi dengan baik melalui perantara IP PBX ini.
3. Untuk *computer monitor traffic* disini pengambilan data traffiknya memanfaatkan protokol SNMP yang sudah terpasang di IP PBX.

### Desain Aplikasi *Monitoring Traffic*

Pada menu utama ini terdapat beberapa submenu yang mewakili dari setiap fitur yang ada dalam aplikasi ini yaitu *Show Traffic*, *PC information* dan *Trace SNMP*. Menu-menu ini ketika dipilih langsung akan menghubungkan kita halaman dari fitur-fitur tersebut. *Monitoring traffic (Show Traffic)* jaringan VoIP merupakan fitur utama dari aplikasi ini, dimana pada menu *monitoring traffic* ini kita akan melihat *traffic* data dari jaringan VoIP yang akan di monitor. Mekanisme yang dijalankan pada proses *monitoring* ini terdiri dari tiga tahapan yaitu *request* nilai *traffic*, konversi nilai dan tampilan grafik.



Gambar 3.2 Desain Menu Aplikasi *Monitoring Traffic* Jaringan VoIP

Pada aplikasi ini juga terdapat fitur tambahan seperti *PC information* yang nantinya dapat digunakan untuk melihat informasi *software* atau aplikasi yang terinstall. Selain itu juga terdapat Fitur tambahan *Trace SNMP* yang dapat digunakan untuk melihat jalur OID pada protokol SNMP.

#### 3.2.1.1 Desain Proses *Monitoring Traffic*

*Monitoring traffic* jaringan VoIP merupakan fitur utama dari aplikasi ini, dimana pada menu *monitoring traffic* ini kita akan melihat *traffic* data dari jaringan VoIP yang akan di monitor. Mekanisme yang dijalankan pada proses *monitoring* ini terdiri dari tiga tahapan yaitu *request* nilai *traffic*, konversi nilai dan tampilan grafik.

### 3.2.1.2 Desain Proses *Request* Nilai *Traffic*

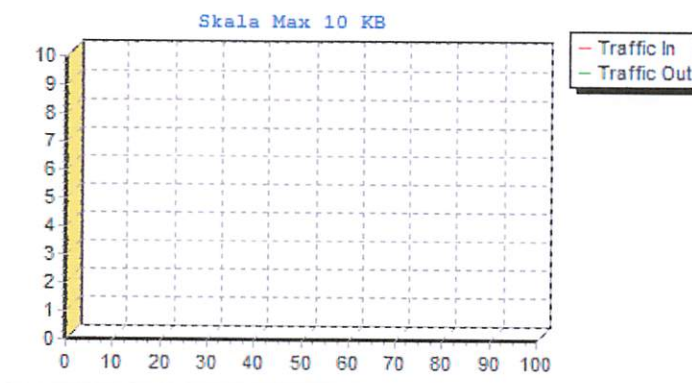
Proses ini adalah proses yang pertama dilakukan oleh aplikasi ini, yaitu meminta data *traffic* dari *agent* yang dalam hal ini berupa IP PBX Trixbox 2.6. Protokol yang digunakan dalam proses ini adalah protokol SNMP, kemudian komponen yang digunakan adalah komponen idSNMP yang terletak pada palette indy client di Borland Delphi 7. Untuk *community* yaitu '*community public*'. Kemudian perintah yang digunakan untuk meminta *traffic* adalah perintah SNMP yaitu `PDUGetRequest` dan `PDUGetNextRequest`.

### 3.2.1.3 Desain Proses Konversi Nilai

Pada tahapan ini nilai yang didapat dari *agent* akan dikonversi karena nilai yang didapat berupa Bytes sehingga sesuai dengan kebutuhan aplikasi ini kita akan mengubah kedalam tiga ukuran yaitu kilobytes (KB), megabytes(MB) dan gigabytes(GB). Untuk proses pengkonversian data yaitu dengan cara dibagi dengan nilai 1000,1000000 dan 1000000000 untuk mendapat hasil konversi yang diinginkan.

### 3.2.1.4 Desain Tampilan Grafik

Setelah melakukan konversi data, maka proses selanjutnya ialah menampilkan data *traffic* jaringan VoIP kedalam bentuk grafik agar mudah dilihat oleh pengguna aplikasi ini nantinya.

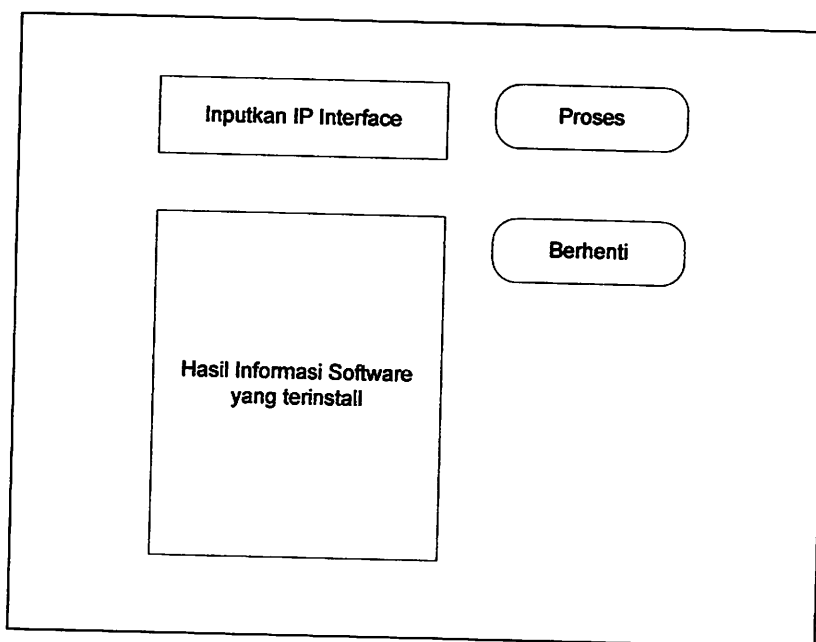


Gambar 3.3 Desain Tampilan Grafik



### 3.2.2 Desain Menu PC Information

Pada menu *PC Information* ini merupakan fitur pendukung dalam aplikasi *monitoring traffic* jaringan VoIP. Aplikasi ini berfungsi untuk menampilkan informasi mengenai software apa saja yang terinstall di dalam PC yang terhubung ke *server IP PBX Trixbox*. Mekanisme dari aplikasi ini terdiri dari dua tahapan yaitu meminta data informasi *software* dengan menggunakan protokol *SNMP*, kemudian menampilkannya dalam *Memo software* apa saja yang telah terinstall.



Gambar 3.4 Desain Tampilan Menu PC Information

#### 3.2.2.1 Request Informasi Software yang terinstal

Tahapan ini adalah tahapan pertama yang dilakukan dalam menu *PC Information*, yaitu meminta data informasi *software* dengan memanfaatkan *SNMP* sebagai protokol, kemudian menggunakan perintah *GetRequest* dan *GetNextRequest* untuk mendapatkan informasi *software* atau aplikasi.

#### 3.2.2.2 Menampilkan Informasi Software yang terinstall

Pada tahap ini, informasi yang telah di *request* akan ditampilkan dalam bentuk memo agar *user* dengan mudah dapat melihat informasi tersebut.

### 3.2.3 Desain Menu Trace SNMP

Pada menu *Trace SNMP* ini merupakan fitur pendukung dalam aplikasi *monitoring traffic* jaringan VoIP. Aplikasi ini berfungsi untuk menampilkan informasi mengenai variabel oid pada protokol SNMP yang nantinya dapat berguna untuk kebutuhan tambahan pada aplikasi *monitoring traffic* ini. Mekanisme dari aplikasi ini terdiri dari dua tahapan yaitu meminta data informasi *software* dengan menggunakan protokol SNMP, kemudian menampilkannya dalam Memo variabel-variabel oid dari server IP PBX.

#### 3.2.3.1 Request nilai variabel OID

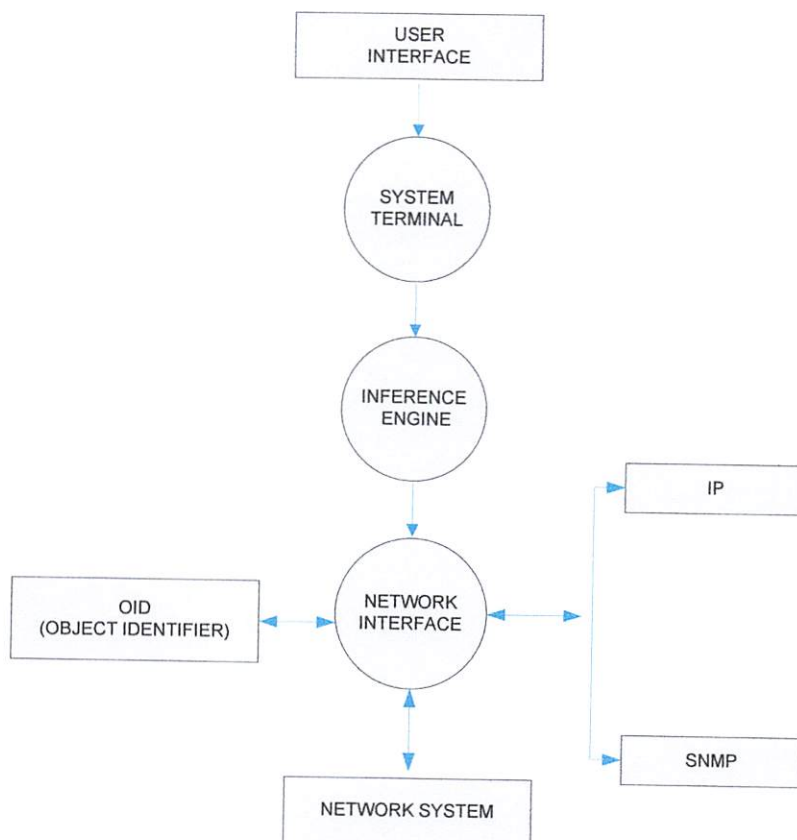
Tahapan ini adalah tahapan pertama yang dilakukan dalam menu *Trace SNMP*, yaitu meminta data informasi *nilai variabel* dengan memanfaatkan SNMP sebagai protokol, kemudian menggunakan perintah *GetRequest* dan *GetNextRequest* untuk mendapatkan informasi nilai *OID*.

#### 3.2.3.2 Menampilkan Informasi OID

Pada tahap ini, informasi nilai oid yang telah di *request* akan ditampilkan dalam bentuk memo agar *user* dengan mudah dapat melihat informasi tersebut.

### 3.2.4 Digram Blok

Untuk membuat sebuah aplikasi yang dapat memberikan informasi mengenai kondisi *traffic* suatu jaringan VoIP, dibutuhkan beberapa komponen yang membantu baik dalam proses mendapatkan informasi dari *agent* maupun proses pengiriman paket data dari manager ke *agent* SNMP. Berikut ini adalah diagram blok dari sistem yang akan dibuat, diagram blok ini menggambarkan interaksi antar komponen-komponen yang terlibat dalam pembuatan *system*.



Gambar 3.5 Diagram Blok Sistem

Keterangan dari masing-masing komponen diatas adalah sebagai berikut:

1. *User Interface*

*User interface* merupakan komponen yang menghubungkan *user* dengan mesin untuk menjalankan sistem, dalam hal ini dilakukan oleh tampilan yang dihasilkan dari *coding* aplikasi Borland Delphi 7.

2. *System Terminal*

*Sistem terminal* adalah sebuah mesin yang menjalankan *system*, pada mesin inilah aplikasi ini akan dibuat dan dijalankan, mesin ini diwakili oleh sebuah *computer*.

3. *Informasi Traffic*

Merupakan informasi yang dihasilkan dari eksekusi perintah dari manager ke agent SNMP melalui jenis pesan tertentu, sehingga di dapatkan suatu informasi mengenai keadaan lalulintas data suatu *device* jaringan dalam suatu jaringan VoIP.

4. *Inference Engine*

*Inference Engine* adalah komponen mengontrol kerja *system* darimana dan kemana paket akan dilewatkan sesuai dengan informasi yang diterima, *control* pada *inference engine* ini dilakukan dalam prosedur-prosedur *coding*.

#### 5. *Network Interface*

*Network Interface* merupakan komponen-komponen fisik pada jaringan komputer, komponen ini dibutuhkan untuk membentuk jaringan komputer yang akan diambil informasinya.

#### 6. *Network Sistem*

*Network System* merupakan komponen-komponen *logic* yang mendukung kerja jaringan komputer, komponen inilah yang akan menangani bagaimana dan kemana informasi pada jaringan itu disampaikan.

#### 7. *OID (Object Identifier)*

Merupakan sebuah ID *numeric* yang digunakan untuk membedakan masing – masing *variable* beserta posisinya dalam MIB dan didalam pesan SNMP.

#### 8. *Internet Portocol*

*Internet Protocol* (IP) berfungsi untuk menyampaikan paket-paket yang dikirimkan melalui jaringan dari satu titik ke titik lainnya.

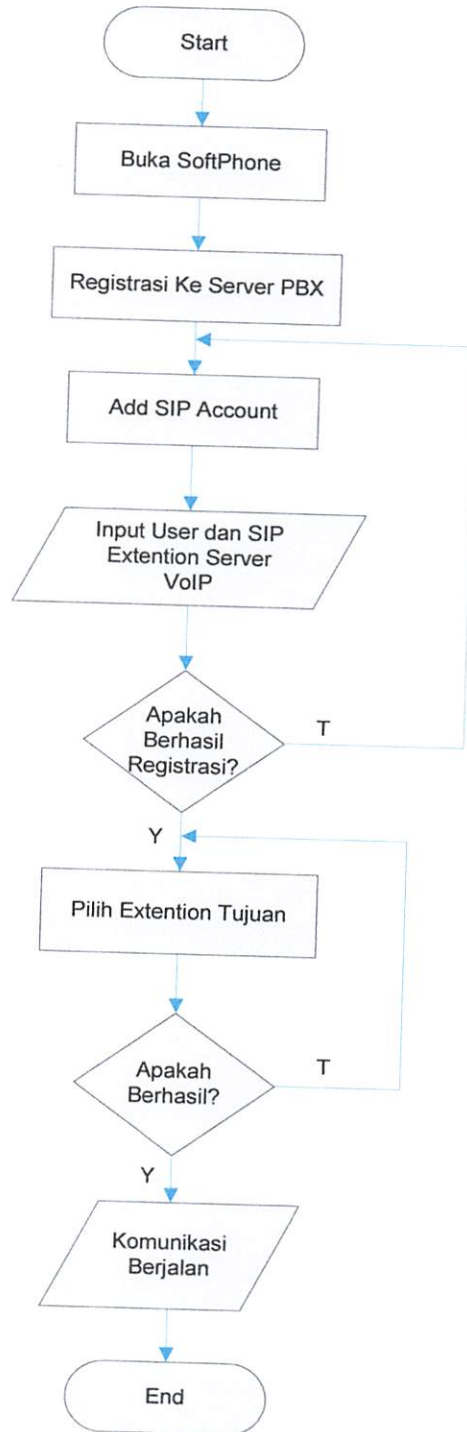
#### 9. *SNMP (Simple Network Management Protocol)*

*Simple Network Management Protocol* (SNMP) merupakan sebuah protocol yang didesain untuk memberikan kemampuan kepada pengguna untuk memonitor dan mengatur *device* jaringan komputer (*personal computer, switch,router* dll) dari management station.

### 3.2.5 **Flowchart**

Berdasarkan desain sistem yang dibuat , maka terdapat dua diagram alir dari aplikasi *monitoring traffic* jaringan VoIP adalah sebagai berikut :

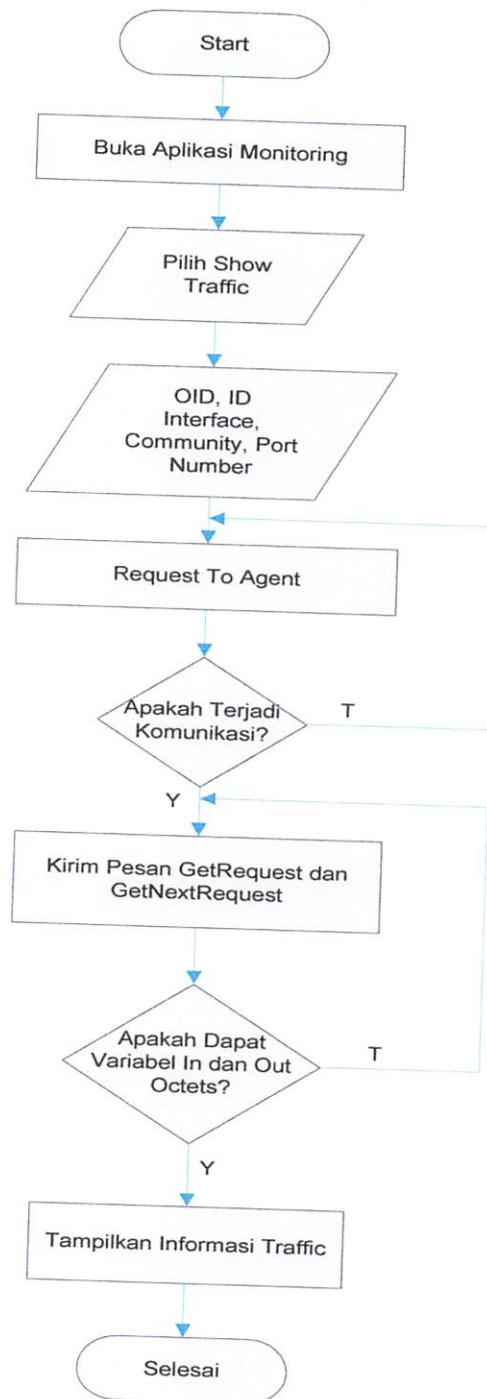
1. Flowchart Komunikasi antar User



Gambar 3.6 Flowchart Komunikasi Antar *User*

*Flowchart* ini menggambarkan bagaimana proses komunikasi antar *user* melalui IP PBX, mulai dari registrasi, penambahan *SIP Account* dan tujuan *Extention*. Sehingga komunikasi dapat berjalan dengan baik.

## 2. Flowchart *Monitoring Traffic* Jaringan VoIP



Gambar 3.6 Flowchart Aplikasi *Monitoring Traffic*

*Flowchart* ini menggambarkan proses utama dari mekanisme permintaan informasi atau variabel dari suatu *agent*, dimana manager SNMP harus membangun terlebih dahulu koneksi dengan suatu *agent* untuk selanjutnya dapat

berkomunikasi dan mendapatkan informasi atau nilai variabel dari suatu *device*, dalam hal ini mengenai lalulintas data.

**BAB IV**  
**IMPLEMENTASI DAN PENGUJIAN SISTEM**

**4.1 Implementasi sistem**

Implementasi dilakukan dengan menerapkan hasil desain yang telah dibuat kedalam bahasa pemrograman (*coding*) Borland Delphi 7, sehingga prosedur-prosedur yang telah dibuat dapat dimengerti oleh mesin dan menghasilkan keluaran seperti apa yang diharapkan. Berikut ini adalah perlengkapan dan kebutuhan yang digunakan dalam implementasi sistem :

Tabel 4.1 Spesifikasi Perlengkapan Implementasi

No	Perlengkapan	Spesifikasi	Keterangan
1	Software PC	Sistem Operasi	Windows XP Service Pack 2
		Bahasa Pemrograman	Borland Delphi 7
2	Personal Komputer	Processor	P4. 3.0 Ghz
		Memory	512 Mb DDR1
		Harddisk	80 GB
3	Headset	Frequency Responce	20 Hz-20KHz
		Jack Plug	OD 3.5mm Stereo



4	Switch AT- FSW708	Fast Ethernet	8 Port

## 4.2 Aplikasi Monitoring Traffic Jaringan VoIP

### 4.2.1 Instalasi Server VoIP

Dalam Instalasi VoIP server, diperlukan persiapan yang cukup, utamanya peralatan yang digunakan serta *software* yang hendak di *install*. Dibawah ini adalah beberapa alat dan bahan yang harus disiapkan dalam menginstalasi dan konfigurasi VoIP server.

#### 4.2.1.1 Alat dan Bahan

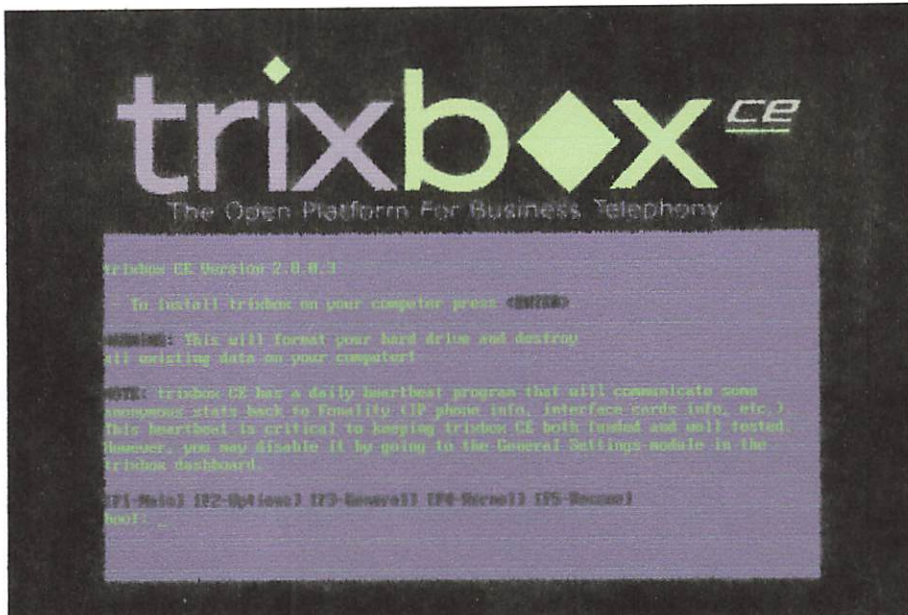
Alat dan bahan yang diperlukan untuk membuat server VoIP adalah sebagai berikut :

1. Satu buah PC untuk server dengan spesifikasi minimum :
  - Pentium II 250Mhz
  - 128 MB RAM
  - 4 GB Harddisk Space
  - 10/100 NIC
  - CD-ROM Drive
  - CD Linux Trixbox

#### 4.2.1.2 Instalasi Linux

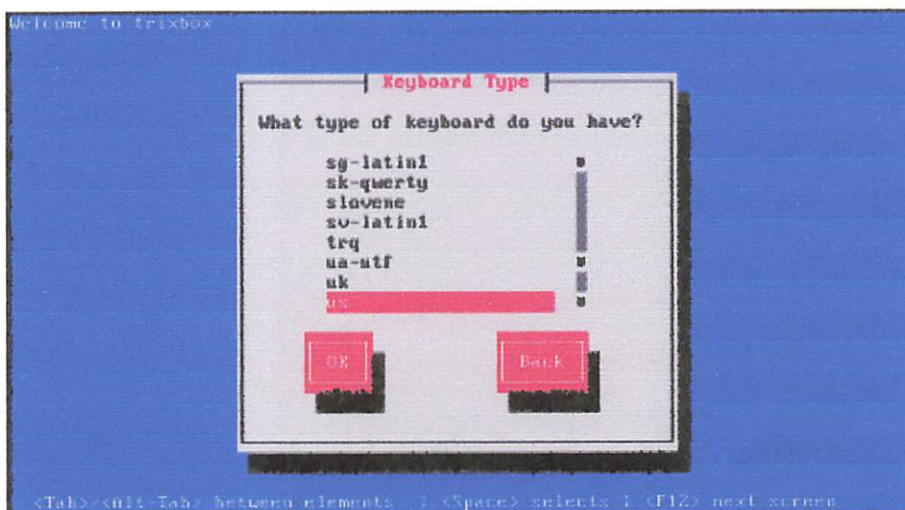
Server VoIP yang dibuat berbasiskan sistem operasi linux. Oleh karena itu berikut akan dijelaskan langkah-langkah menginstall trixbox :

- Siapkan CD instalasi Trixbox.
- Pastikan bahwa *setting* pada BIOS akan mem-boot CD-ROM pertama kali, rubah terlebih dahulu jika *setting* BIOS belum benar.
- Masukkan CD Trixbox dan *restart* komputer. Tunggu hingga keluar tampilan seperti berikut ini:



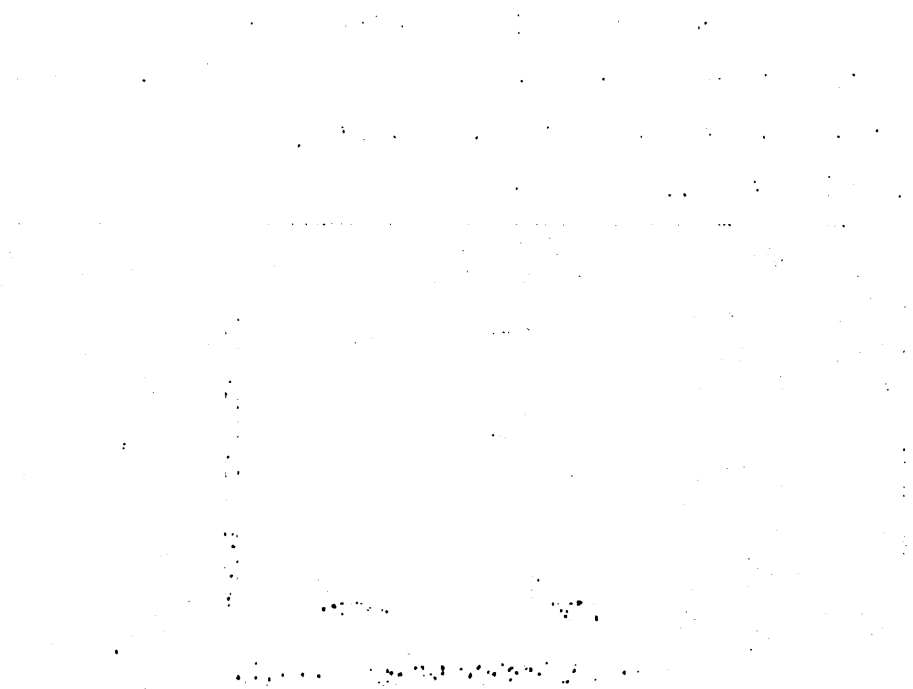
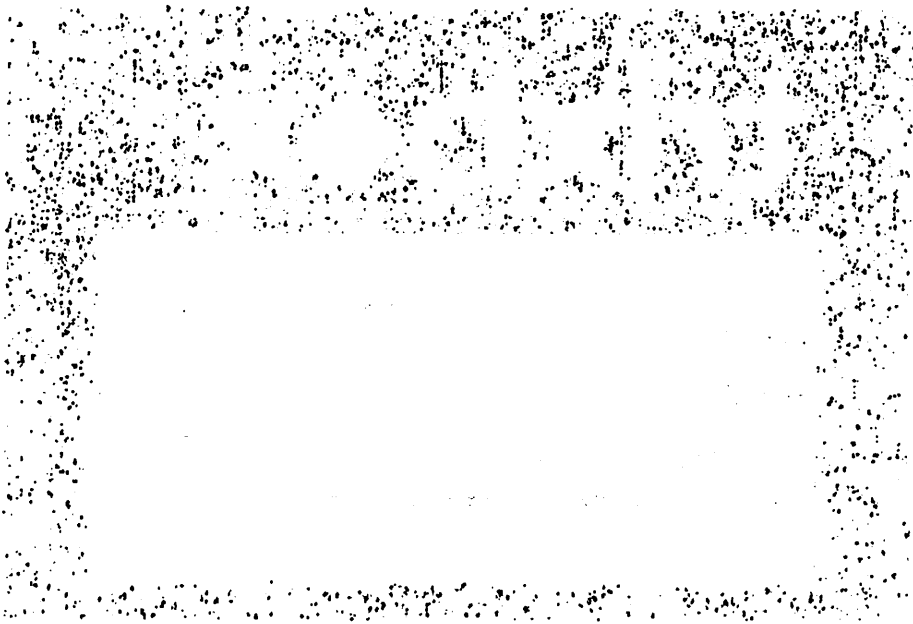
Gambar 4.1 Tampilan awal *booting* Trixbox

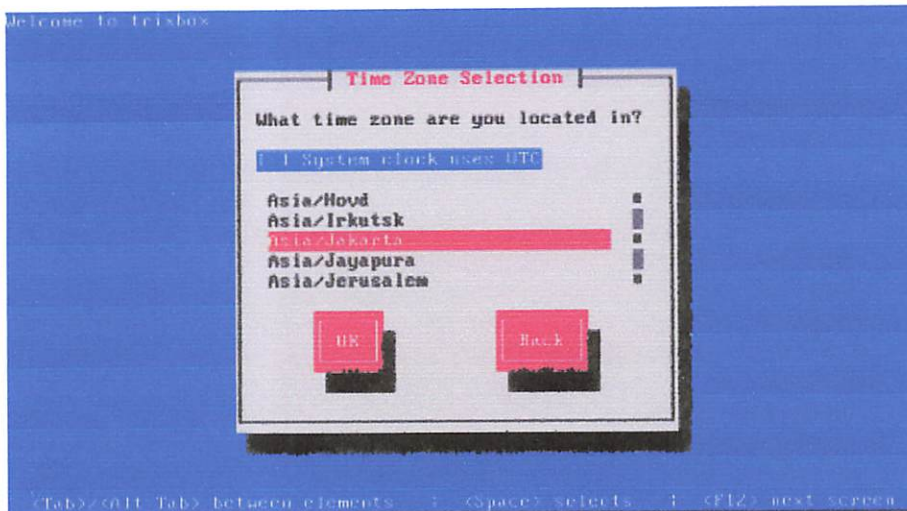
- Selanjutnya ikuti perintah yang muncul. Kemudian keluar kotak dialog tentang *type keyboard* yang kita gunakan, pilih *type keyboard "us"* kemudian "OK" dengan menekan tombol Enter.



Gambar 4.2 Menu *Selecting Keyboard*

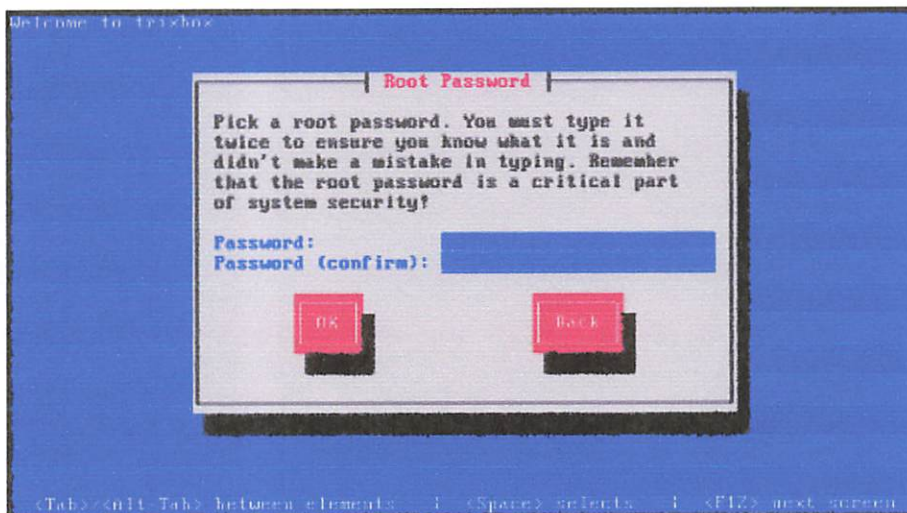
- Setelah itu akan keluar kotak dialog *Time Zone*, pilih lokasi sesuai daerah kita, pilih Asia/Jakarta, Pilih "OK" lalu tekan Enter.





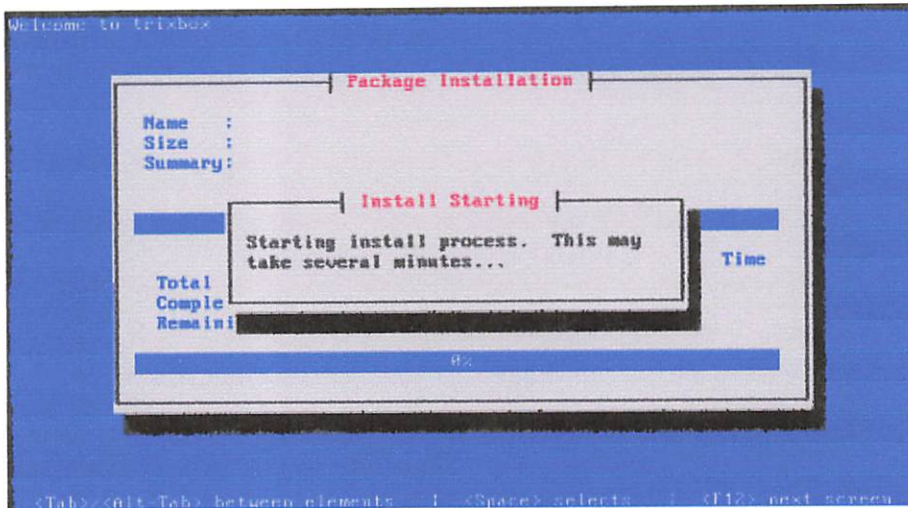
Gambar 4.3 Menu *Selecting Time Zone*

- Langkah berikutnya adalah masukan/ketikan *password* root untuk sistem linux. Misal *password* untuk root adalah 12345.

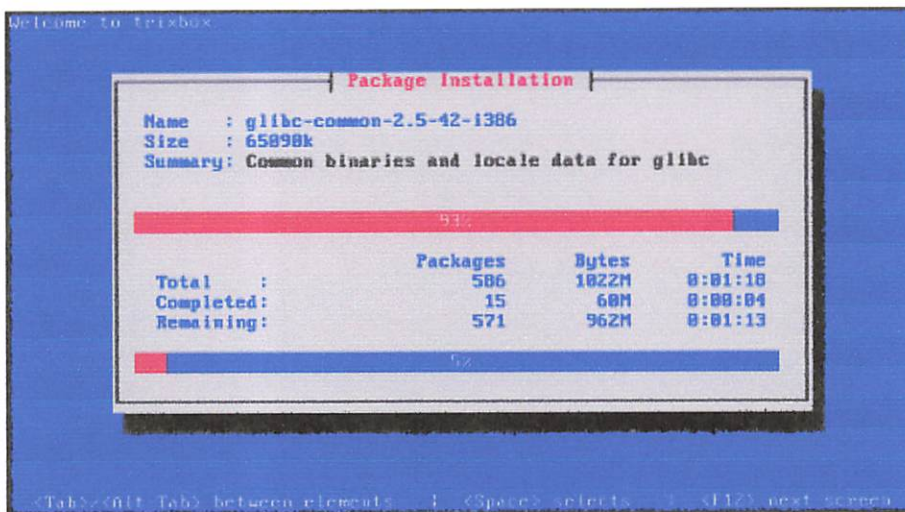


Gambar 4.4 Menu *Setting password root*

- Setelah kita memasukan password untuk root, maka proses instalasi *base system (operating system)* dimulai, tunggu sampai proses instalasi selesai maka setelah itu computer akan reboot.

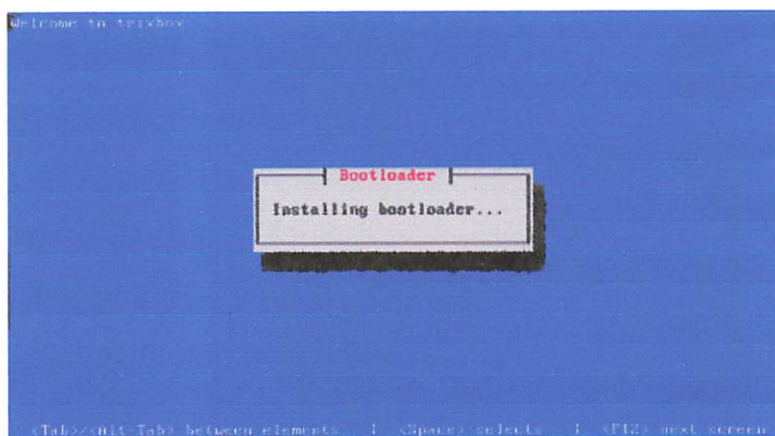
Gambar 4.5 Tampilan *install starting*

- Proses instalasi .

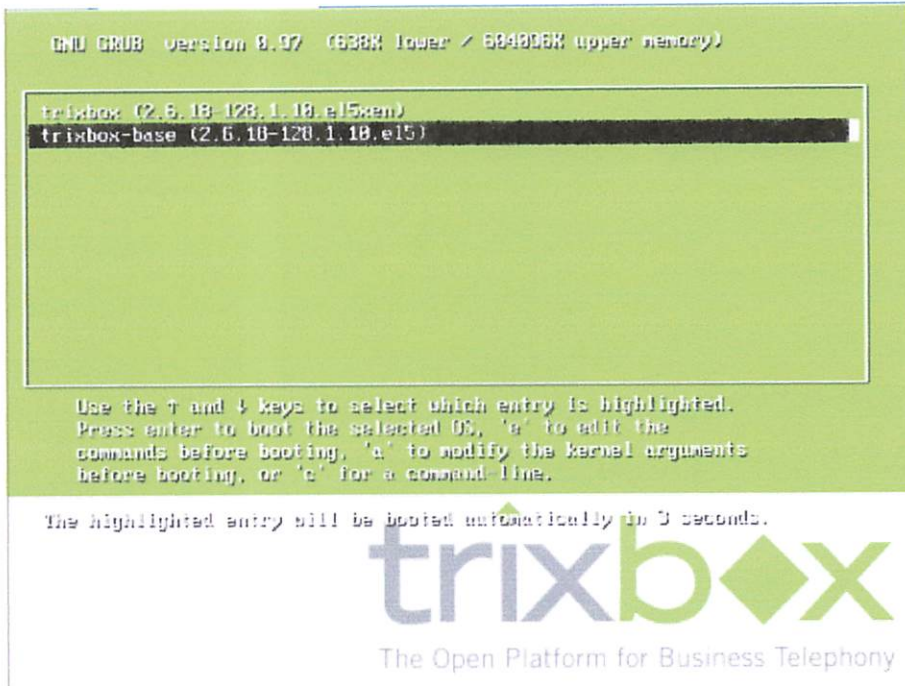


Gambar 4.6 Proses Instalasi

- Proses instalasi Boot loader.

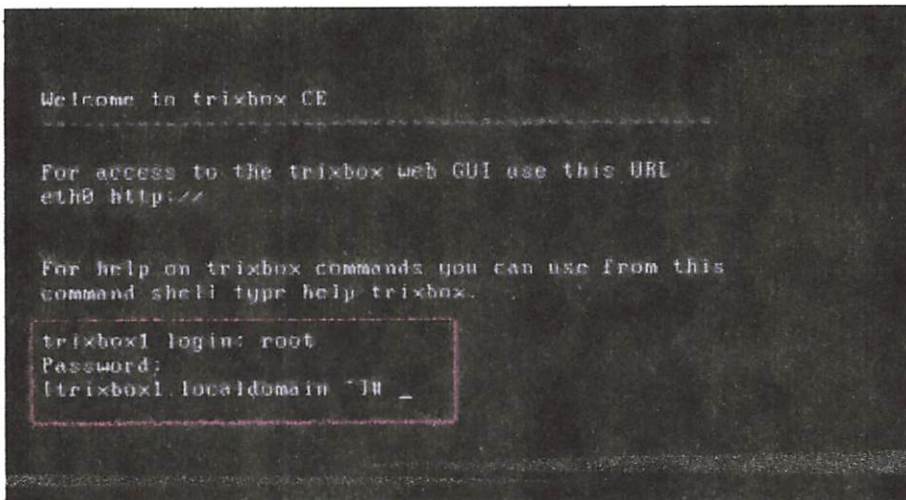
Gambar 4.7 Proses *installing bootloader*

- Setelah proses instalasi selesai maka akan reboot sampai sistem operasi tersebut *booting* untuk pertama kali seperti dibawah ini.



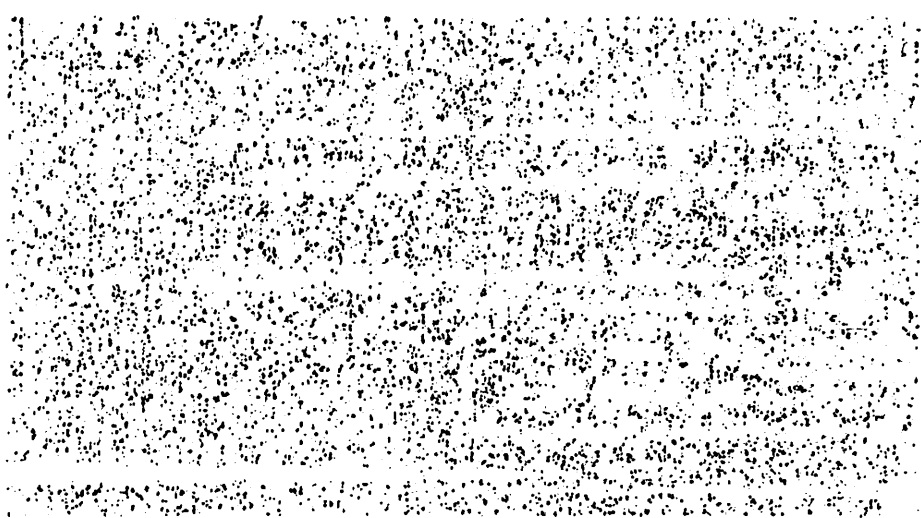
Gambar 4.8 Tampilan *Bootloader* Linux Trixbox

- Setelah itu pengisian *user* dan *password* untuk bisa login ke mode CLI (*Command Line Interface*).jika *login* berhasil maka akan keluar tampilan seperti dibawah ini.



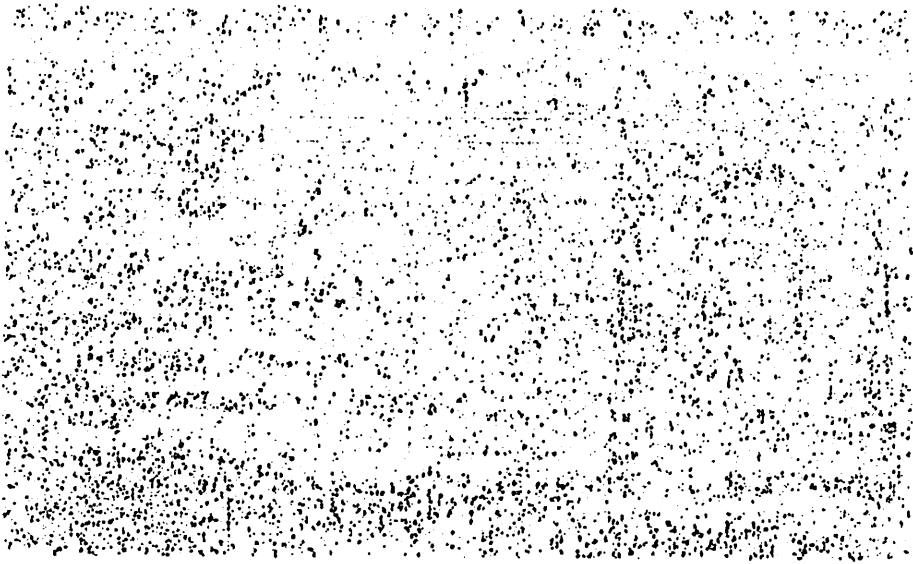
Gambar 4.9 Tampilan *Start Up* Linux

- Untuk meminta bantuan mengenai *command* pada trixbox ketikkan "*help-trixbox*".

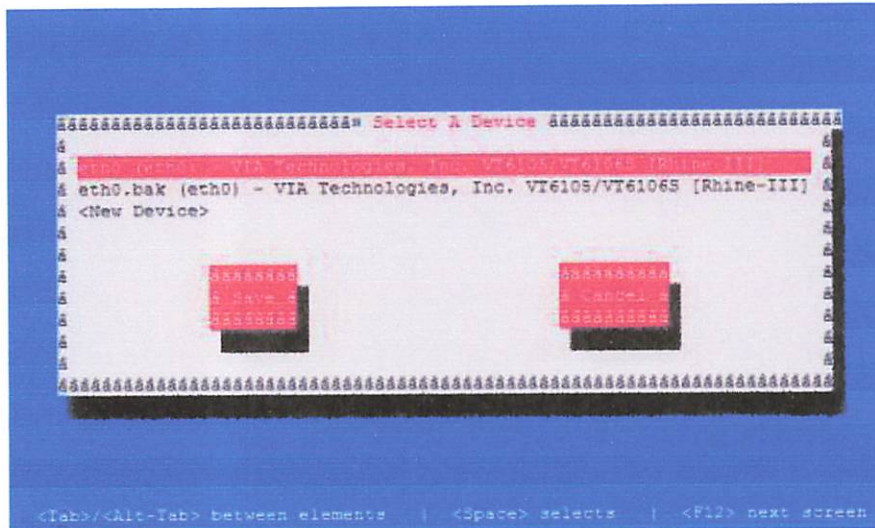






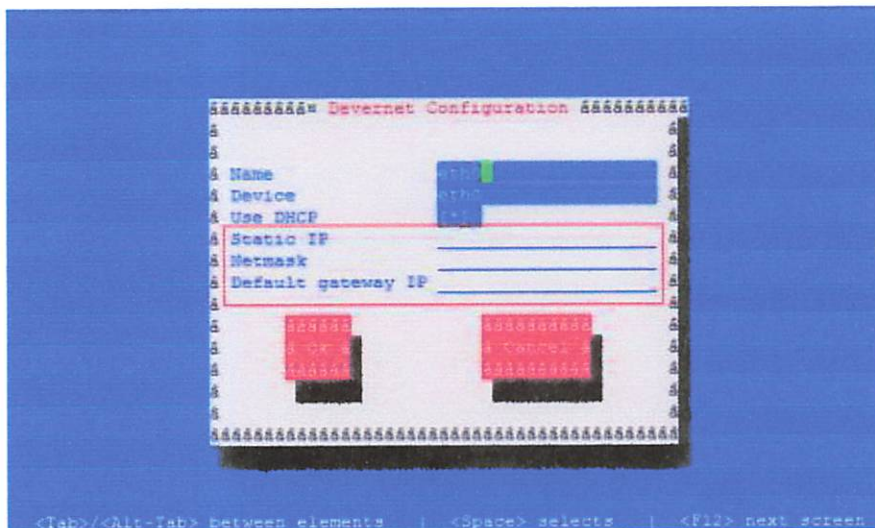


1990



Gambar 4.12 Pemilihan Ethernet

- Kemudian pilih konfigurasi *static* IP atau DHCP. Pada kasus kali ini konfigurasi akan dilakukan secara *static* yaitu dengan memberi *IP address* 192.168.1.1 dan netmask-nya dengan 255.255.255.0. Pilih OK, lalu tekan Enter.



Gambar 4.13 Pemberian alamat IP

- Setelah selesai mengkonfigurasi *network*, lalu *restart* konfigurasi network dengan cara mengetikkan perintah `/etc/init.d/network restart` lalu tekan enter.

#### 4.2.2 Konfigurasi Server

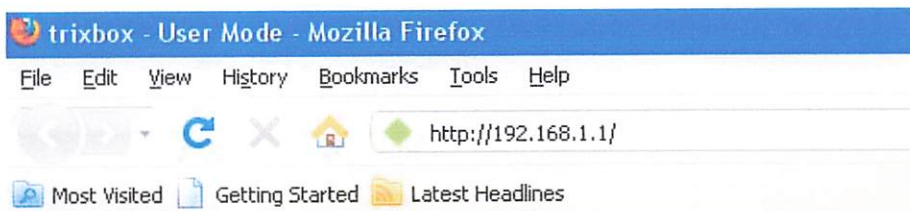
Dalam konfigurasi VoIP server, diperlukan satu buah komputer *client* yang digunakan untuk melakukan konfigurasi VoIP. Konfigurasi VoIP server

dilakukan dengan berbasis web (*web base*) , maka dari itu komputer *client* harus mempunyai aplikasi *web browser*.

#### 4.2.2.1 Instalasi Modul

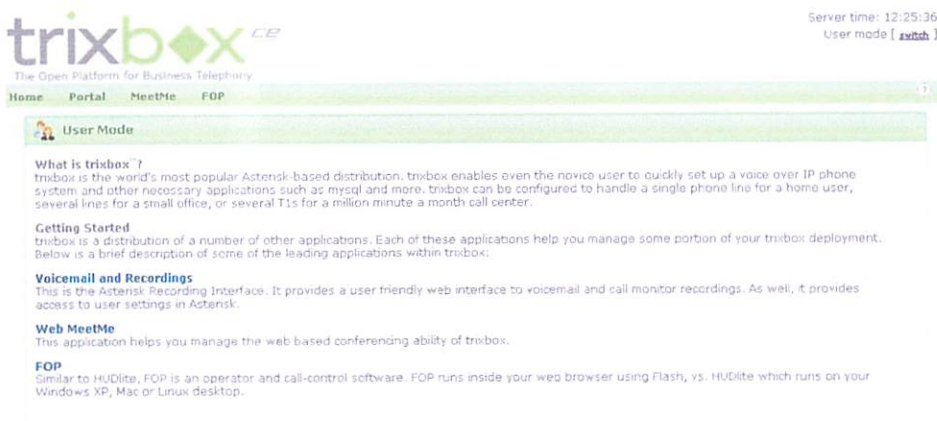
Secara *default* modul-modul yang digunakan untuk aplikasi VoIP pada Free PBX belum terinstal. Berikut ini adalah langkah-langkah instalasi modul yang diperlukan untuk membuat VoIP server.

- Buka browser (mozilla firefox) pada komputer *client*, lalu pada *address bar* isikan `http://192.168.1.1` (192.168.1.1 merupakan alamat IP server VoIP).



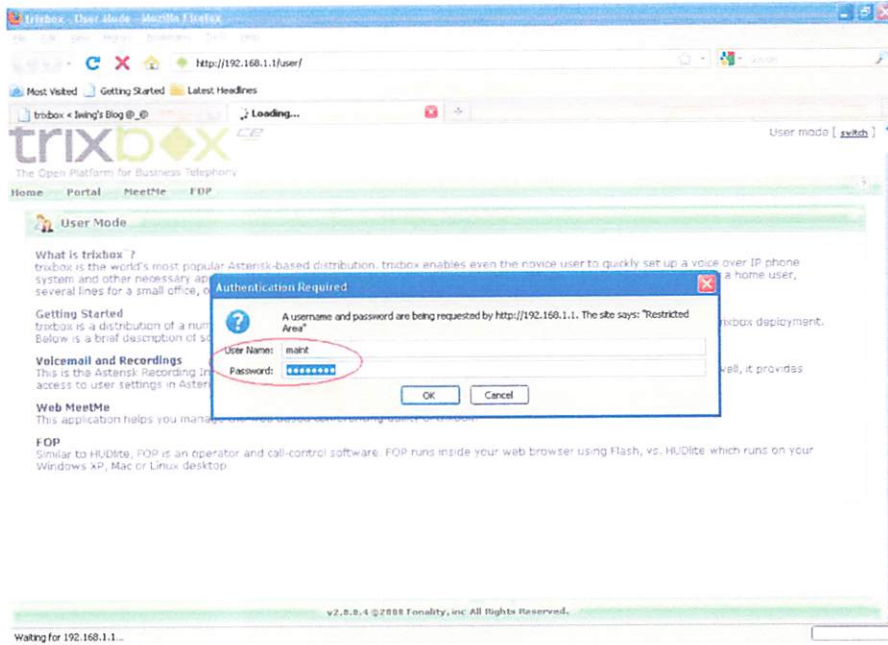
Gambar 4.14 Tampilan *address bar* pada browser

- Setelah kita memasukan alamat server VoIP pada *address bar*. Maka tampilan pertama dari trixbox akan keluar, Pilih menu *System Administration* untuk masuk kehalaman *Configuration and Administration*.



Gambar 4.15 Tampilan awal web konfigurasi trixbox

- Setelah memilih menu *system administration* maka akan keluar kotak dialog login, isikan *username* = maint ; *password* = password kemudian pilih “OK”.



Gambar 4.16 Kotak dialog login system administration

- Jika login berhasil maka akan masuk kehalaman Configuration and Administration, pilih menu PBX Setting untuk masuk kehalaman PBX Setting.



Gambar 4.17 Halaman Configuration and Administration

#### 4.2.2.2 Konfigurasi SIP Extensions

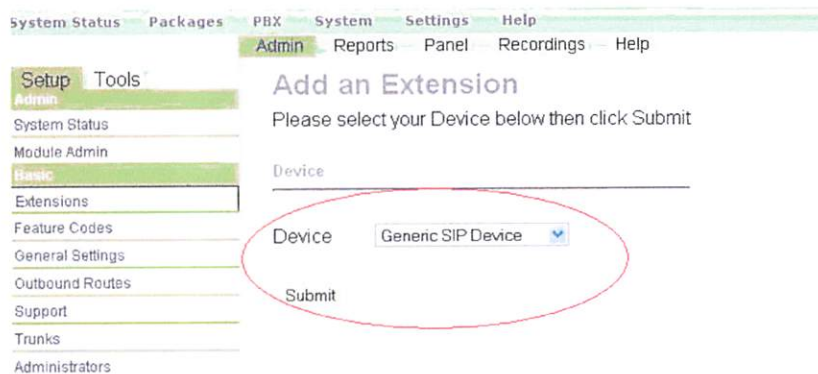
SIP *Extensions* adalah nomor atau user yang diperuntukan untuk client VoIP, yang digunakan untuk *register* ke server. Berikut akan dijelaskan langkah-langkah membuat SIP Extension.

- Pilih menu PBX Setting maka akan masuk ke halaman PBX Setting yang nantinya akan digunakan untuk penambahan *extension*



Gambar 4.18 Tampilan Halaman PBX Setting

- Langkah selanjutnya, Pilih *Extensions - Add extension - Device: Generic SIP device - Submit*



Gambar 4.19 Halaman Add Extension

- Selanjutnya, isikan *User Extension* dan *Display Name* pada halaman *Add Extension* ini.

Admin Reports Panel Recordings Help

## Add SIP Extension

Add Extension

---

User Extension 204  
 Display Name line6  
 CID Num Alias  
 SIP Alias

Extension Options

Gambar 4.20 Penambahan *User Extension*

- Kemudian isi *Secret* untuk keamanan PBX ini.

Conferences  
 DISA  
 Languages  
 Music on Hold  
 PIN Sets  
 Paging and Intercom  
 Parking Lot  
 System Recordings  
 VoiceMail Blasting

Device Options

This device uses sip technology

secret 12345  
 dtmfmode rfc2833

Gambar 4.21 Pengisian *Secret*

- Jika sudah, *Submit* dan *Apply Configuration Changes*.

PBX System Settings Help  
 Admin Reports Panel Recordings Help

Apply Configuration Changes

## Add an Extension

Please select your Device below then click Submit

Device

Device Generic SIP Device

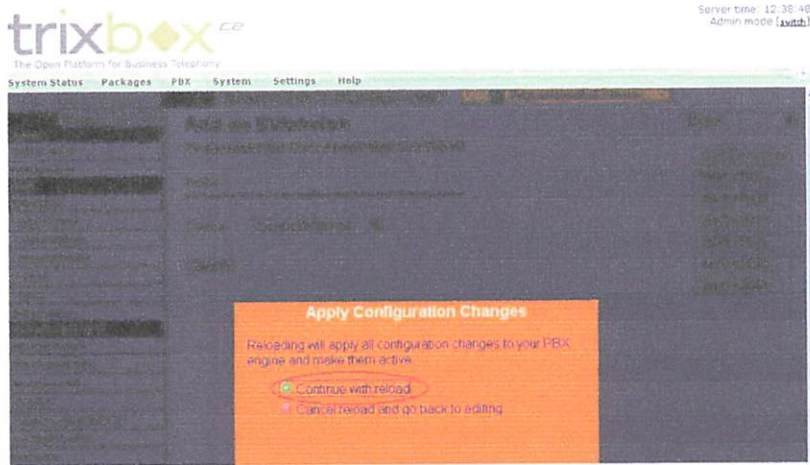
Submit

line1 <101>  
 line2 <102>  
 line3 <201>  
 line4 <202>  
 line5 <203>  
 line6 <204>

You have made changes to the configuration that have not yet been applied. When you are finished making all changes, click on apply Configuration Changes to get them into effect.

Gambar 4.22 *Submit* dan *Apply Configuration Changes*.

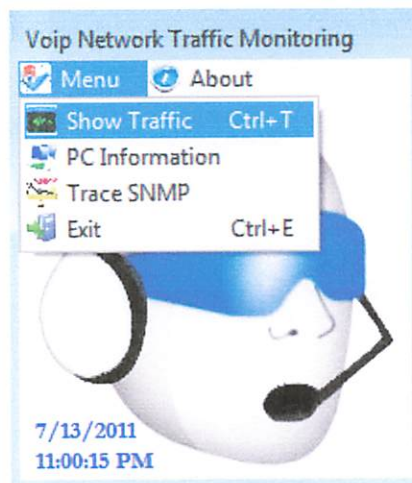
- Kemudian *Continue with Reload*, sampai posisi ini konfigurasi server trixbox selesai.



Gambar 4.23 *Continue with reload*

#### 4.2.2.3 Tampilan Menu Aplikasi *Monitoring Traffic*

Aplikasi *Monitoring Traffic* pada jaringan VoIP merupakan aplikasi utama sebagai proses permintaan informasi *traffic* yang terjadi pada jaringan VoIP. Pada aplikasi ini *administrator* harus memasukkan alamat IP *interface* yang merupakan pusat yang akan di *monitoring*.



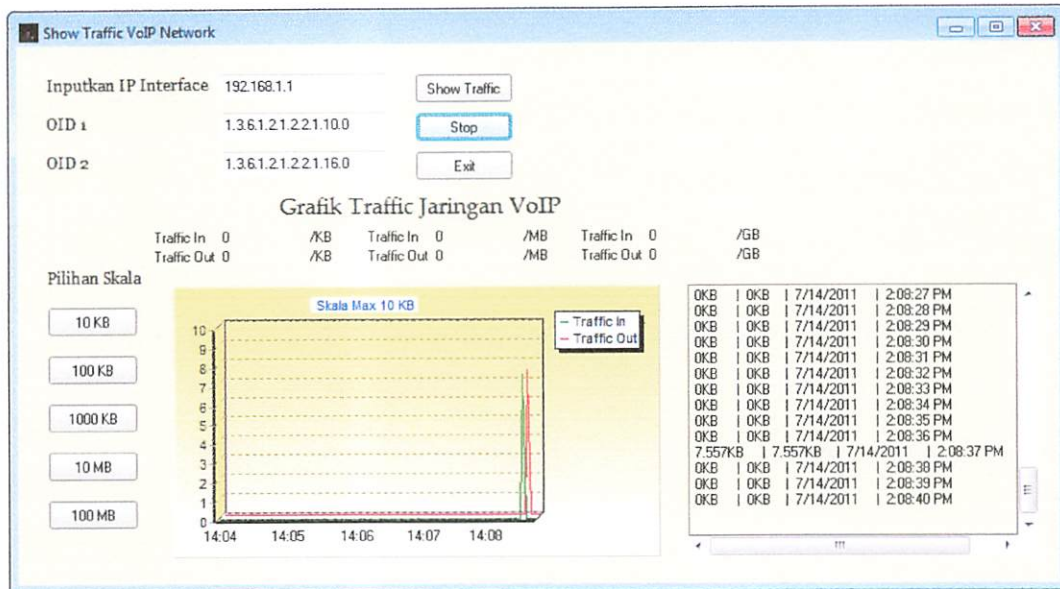
Gambar 4.24 Menu Aplikasi *Monitoring Traffic*

Pada menu di atas untuk melihat *traffic* jaringan VoIP, pilih menu *Show Traffic*



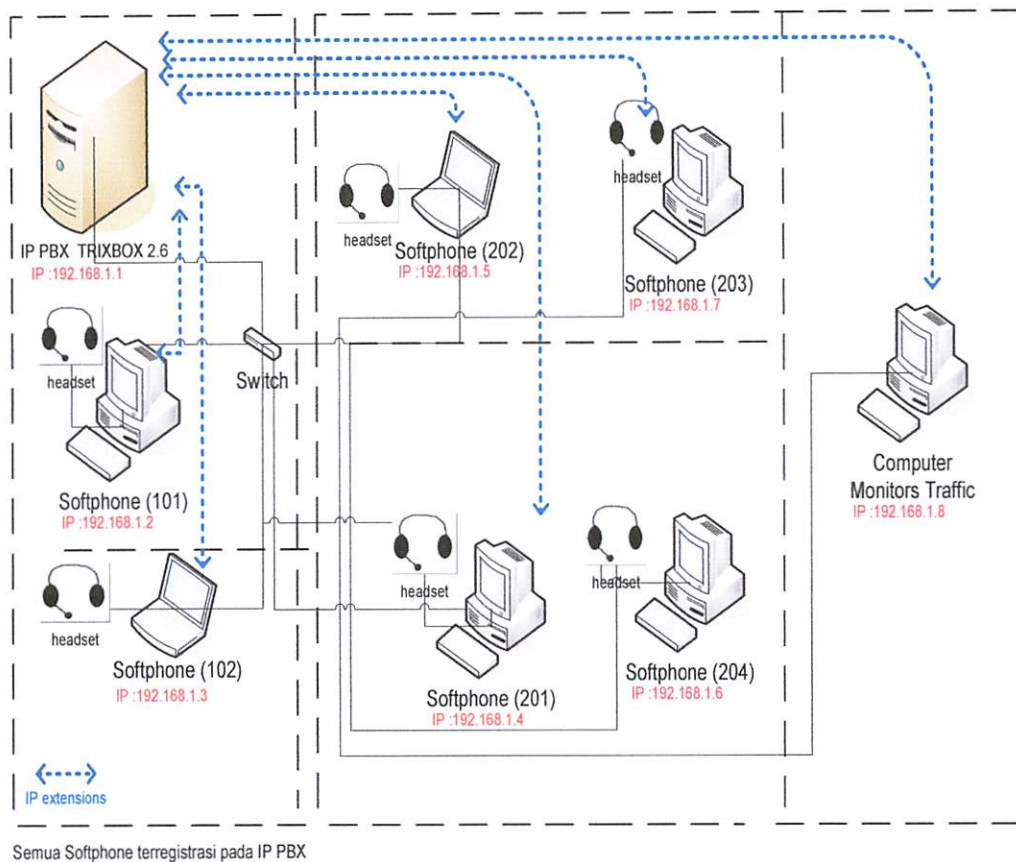


Dalam implementasi tersebut penulis mendesain sebuah jaringan VoIP, dimana jaringan tersebut terdiri dari beberapa *device* jaringan diantaranya, *Switch (8 port)* serta satu komputer sebagai IP PBX dan *Softphone* serta *Headset*. Berikut ini adalah hasil implementasi tersebut :



Gambar 4.27 Hasil *Monitoring Traffic*

#### 4.4 Pengujian Berdasarkan Topologi Jaringan VoIP yang Sederhana



Gambar 4.28 Desain Pengujian Jaringan VoIP

Pada pengujian kali ini akan dilakukan komunikasi antar *Extension User*. Disini semua *softphone (client)* sudah teregistrasi pada IP PBX.

##### 4.4.1 Pengujian Pada Saat Terjadi komunikasi antara *Extension user 101* dengan *Extension User 102*

Setelah implementasi sistem dilakukan, pengujian dilakukan secara keseluruhan. Pengujian dengan cara yang sama dilakukan pada komunikasi *extension user* dan setelah ter-registrasi yang melakukan komunikasi tentunya. Untuk pengujian Pada Jaringan VoIP ini ada dua lingkungan (*environment*) yang bertindak sebagai sisem penguji yaitu manusia dan *hardware*.

Untuk lingkungan manusia, sistem yang diuji cukup dengan membutuhkan beberapa orang yang bertindak sebagai users. Sedangkan untuk lingkungan

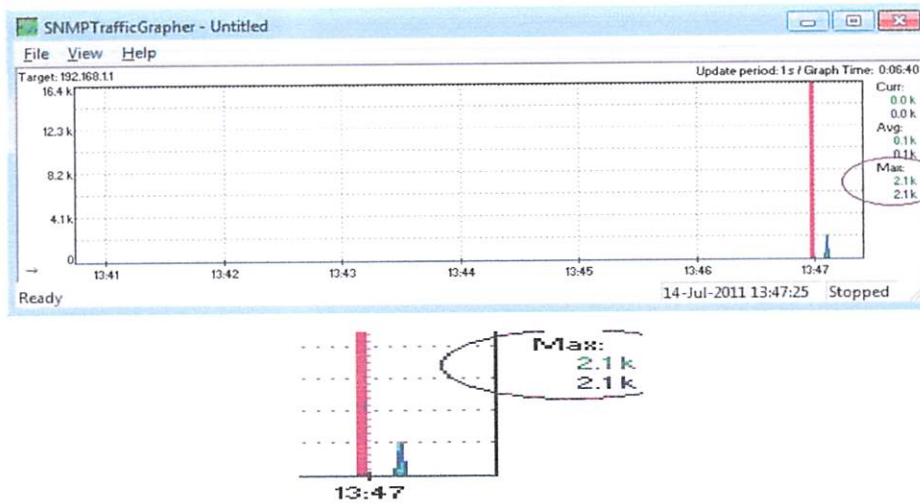
*hardware* pengujian dilakukan pada beberapa komputer, dimana komputer tersebut bertindak sebagai server dan *computer monitor* serta *softphone*.

Dalam pengujian ini setelah dilakukan komunikasi antara *Extension user* 101 dengan *Extension user* 102 pada aplikasi *monitoring traffic* jaringan VoIP didapatkan *traffic* data yang terpantau oleh aplikasi *monitoring traffic* ini sebesar 2,167 KB, Nilai data ini merupakan nilai dari *traffic* masuk dan *traffic* keluar. Pada pengujian di jaringan VoIP antara *Traffic in* dan *Traffic Out* memiliki nilai yang sama yaitu 2,167 KB.

Pada pengujian ini juga digunakan software SNMP Traffic Grapher (STG) sebagai software pembanding. Alasan menggunakan software ini karena disini dalam pengambilan data *traffic* sama-sama memanfaatkan *protocol* SNMP. Mekanisme dari pengujian ini yaitu ketika terjadi suatu komunikasi antar *extension user* 101 dan *extension user* 102, kemudian secara bersama-sama aplikasi *monitoring traffic* jaringan VoIP dan STG akan menampilkan *traffic* yang terjadi pada saat proses komunikasi yang terjadi di jaringan VoIP.



Gambar 4.29 Pengujian komunikasi antara *extension user* 101 dengan *extension user* 102

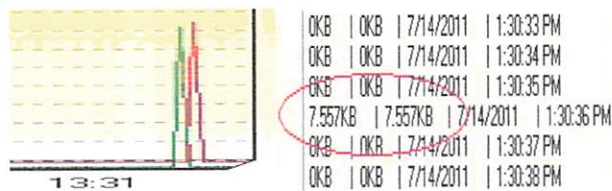
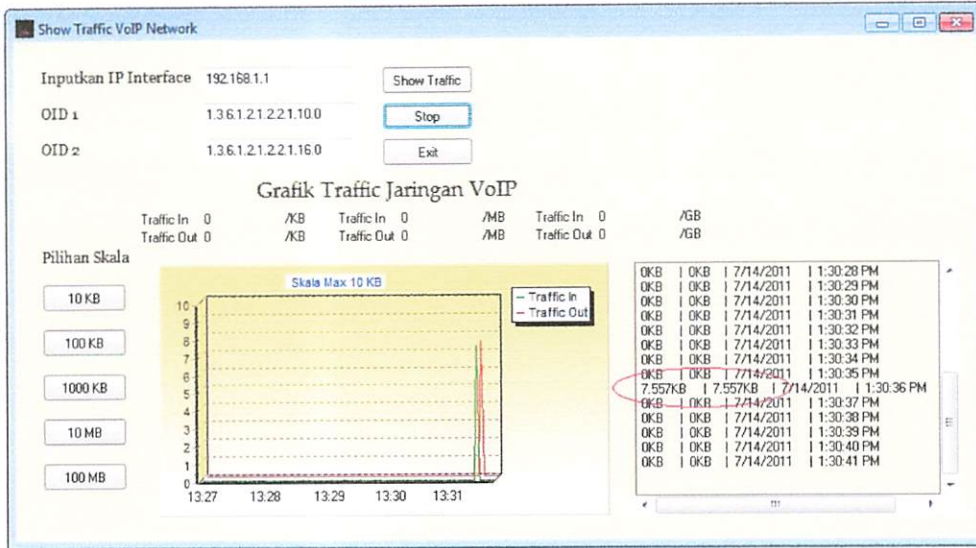


Gambar 4.30 Pengujian komunikasi antara *extension user 101* dengan *extension user 102* dengan menggunakan STG.

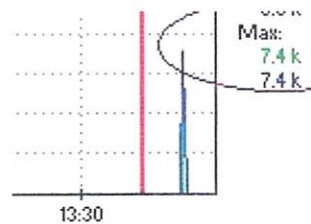
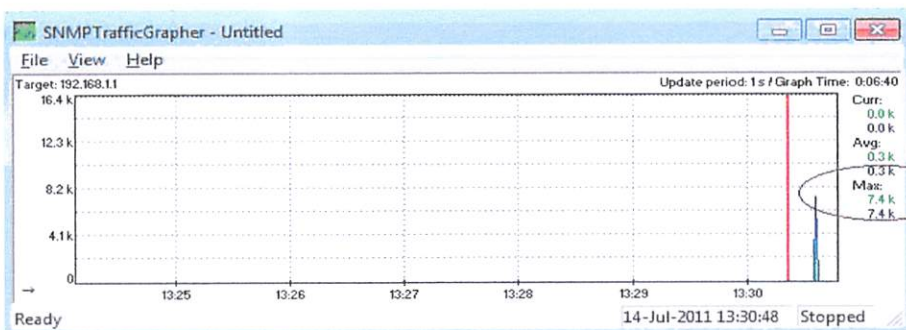
Dari gambar diatas bisa kita amati bahwa hasil *traffic* yang terpantau dari aplikasi *monitoring traffic* jaringan VoIP didapatkan nilai *traffic* sebesar 2,167 KB dan pada STG terlihat *traffic* sebesar 2,1 KB. Disini sudah bisa di lihat dan di analisa bahwa *traffic* yang terlihat pada aplikasi *monitoring traffic* jaringan VoIP melebihi nilai *traffic* data dari *software* STG hanya selisih nilai sebesar 0.067.

#### 4.4.2 Pengujian Pada Saat Terjadi komunikasi antara *Extension user 101* dengan *Extension User 102* dan *Extension user 201* dengan *Extension User 204*

Dalam pengujian ini setelah dilakukan komunikasi *antara Extension user 101* dengan *Extension user 102* dan *Extension user 201* dengan *Extension user 204* pada aplikasi *monitoring traffic* jaringan VoIP didapatkan *traffic* data yang terpantau oleh aplikasi *monitoring traffic* jaringan VoIP sebesar 7,557 KB, Nilai data ini merupakan nilai dari *traffic* masuk dan *traffic* keluar.



Gambar 4.31 Pengujian komunikasi antara *extension user 101* dengan *extension user 102* dan *extension user 201* dengan *extension user 204*

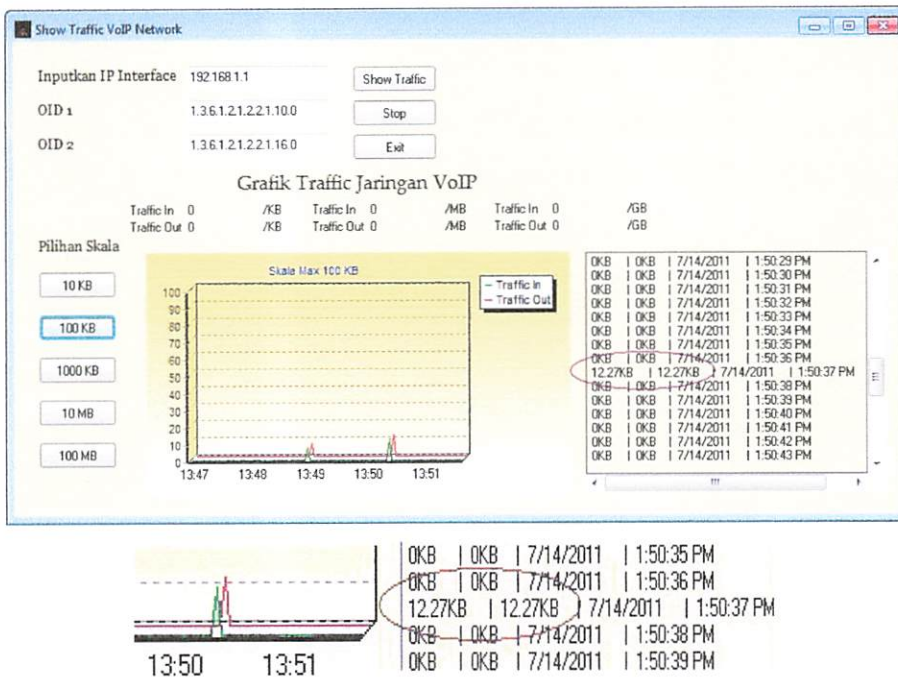


Gambar 4.32 Pengujian komunikasi antara *extension user 101* dengan *extension user 102* dan *extension user 201* dengan *extension user 204* dengan menggunakan STG

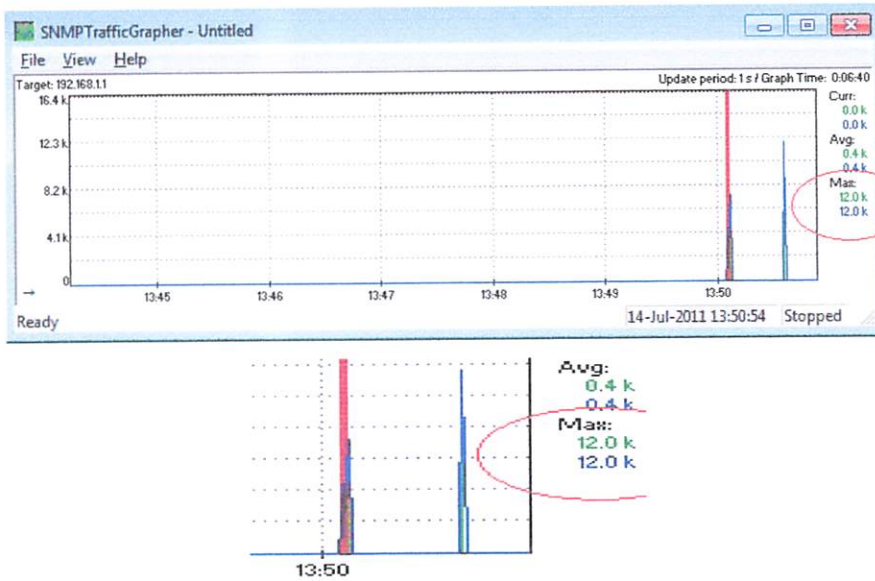
Dari gambar diatas bisa diamati bahwa hasil *traffic* yang terpantau dari aplikasi *monitoring traffic* jaringan VoIP didapatkan nilai *traffic* sebesar 7,557 KB dan pada STG terlihat *traffic* sebesar 7,4 KB. Disini sudah bisa di lihat dan di analisa bahwa *traffic* yang terlihat pada aplikasi *monitoring traffic* jaringan VoIP melebihi nilai *traffic* data dari *software* STG hanya selisih nilai sebesar 0,157.

#### 4.4.3 Pengujian Pada Saat Terjadi komunikasi antara *Extension user 101* dengan *Extension User 102*, *Extension user 201* dengan *Extension User 204* dan *Extension user 202* dengan *Extension User 203*

Dalam pengujian ini setelah dilakukan komunikasi antara *Extension user 101* dengan *Extension user 102*, *Extension user 201* dengan *Extension user 204* dan *Extension user 202* dengan *Extension user 203* pada aplikasi *monitoring traffic* jaringan VoIP didapatkan *traffic* data yang terpantau oleh aplikasi *monitoring traffic* jaringan VoIP sebesar 12,27 KB, Nilai data ini merupakan nilai dari *traffic* In dan *traffic* Out. Pada gambar dibawah ini skala yang digunakanpun berubah menjadi skala 100KB.



Gambar 4.33 Pengujian komunikasi antara *extension user 101* dengan *extension user 102*, *extension user 201* dengan *extension user 204* dan *extension user 202* dengan *extension user 203*.



Gambar 4.34 Pengujian komunikasi antara *extension user* 101 dengan *extension user* 102 ,*extension user* 201 dengan *extension user* 204 dan ,*extension user* 202 dengan *extension user* 203 dengan menggunakan STG.

Dari gambar diatas bisa diamati bahwa hasil *traffic* yang terpantau dari aplikasi *monitoring traffic* jaringan VoIP didapatkan nilai *traffic* sebesar 12,27 KB dan pada STG terlihat *traffic* sebesar 12.0 KB. Disini sudah bisa di lihat dan di analisa bahwa *traffic* yang terlihat pada aplikasi *monitoring traffic* jaringan VoIP melebihi nilai *traffic* data dari *software* STG hanya selisih nilai sebesar 0,27.

Berikut ini adalah tabel perbandingan hasil pengujian antara aplikasi *monitoring traffic* jaringan VoIP dan STG.

Tabel 4.2 Perbandingan hasil pengujian Traffic

No	Komunikasi <i>Extension User</i>	Aplikasi <i>Monitoring Traffic Jaringan VoIP</i>	<i>SNMP Traffic Grapher (STG)</i>
		<i>Traffic (KB)</i>	
1	101 - 102	2,167	2,1
2	101 - 102	7,557	7,4
	201 - 204		
3	101 - 102	12,27	12,0
	201 - 204		
	202 - 203		

#### 4.4.4 Penyimpanan *Log Traffic* Ke Database

##### 4.4.4.1 Koneksi Database Dari Delphi 7 ke Ms Access

Membuat koneksi database \*.mdb dari Delphi ke Ms Access bisa dilakukan melalui ODBC Data *Source Administrator* yang ada pada *Control Panel* -> *Administrative Tools* (Windows XP). Namun bagi anda yang suka sekali bermain script pada registry bisa mencoba cara yang kedua.

##### 4.4.4.2 Koneksi Melalui ODBC Data *Source Administrator*

Berikut ini adalah langkah-langkah mengkoneksikan *database log traffic* dari Delphi ke Ms Acces :

- Membuat Database Traffic

Pada sub bab ini tidak di bahas mengenai langkah-langkah pembuatan database. Pada intinya terdapat lima *field* yaitu ID, Traffic In, Traffic Out, Date dan Time.

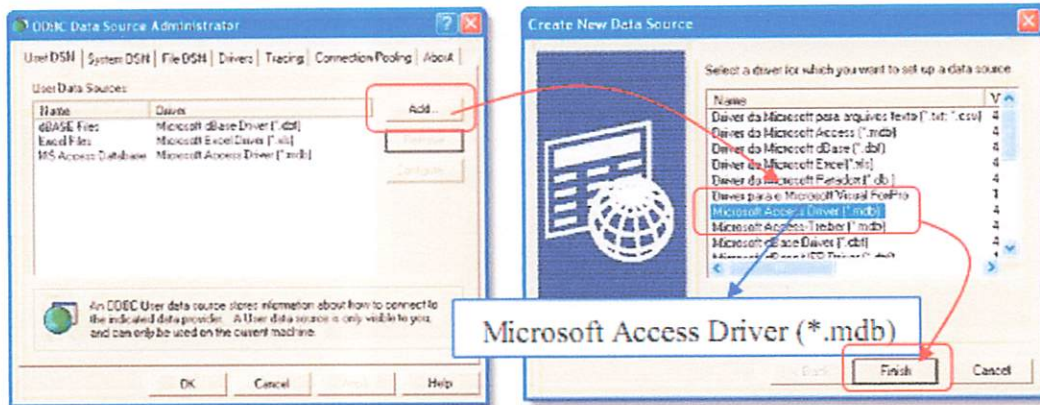
	ID	Trafik_In	Trafik_Out	Date	Time
	1				
	2				
	3				
	4				
▶	5				
*	(number)				

Gambar 4.35 Tabel Traffic

- Mengatur Konfigurasi ODBC DAN BDE

Buka ODBC Data *Source Administrator*, (untuk Windows XP) caranya klik tombol *Start* -> *All Programs* -> *Control Panel* -> *Administrative Tools* -> *Data Sources* (ODBC). Atau melalui *Start* menu -> *Run* kemudian ketikkan *odbcad32* dan klik *Ok*, Pada tab *User DSN*, klik tombol *Add*, kemudian pilih *Microsoft Access Driver (\*.mdb)*. dan klik *finish*.





Gambar 4.36 Setting ODBC

Kemudian muncul form ODBC Microsoft Access Setup. Disini user akan mengatur nama alias untuk database (Data Source Name), database file serta password. Isikan Data Source Name dengan nama dari database. Untuk database filenya klik select pada group database. Apabila anda ingin menambahkan password untuk keamanan data, klik tombol *Advanced*, lalu isikan *Login Name* dan *Password* sesuai keinginan anda. Apabila tidak menghendaki adanya password, maka opsi *advanced* bisa diabaikan. Kemudian klik *Ok*. Database tersebut sudah terdaftar.

Jika sudah terdaftar, langkah berikutnya yaitu dengan cara *coding* pada *form show traffic*. Pada kasus ini penulis memsukan *sourcode* agar *log traffic* tersebut dapat tersimpan ke dalam database. Berikut ini *sourcode* agar *log traffic* dapat tersimpan kedalam database.

```

procedure TForm4.log;
var
  InB,OutB : String;
begin
  InB := lbl_inbound.Caption;
  OutB := lbl_outbound.Caption;
  memo1.Lines.Add(' '+InB + 'KB   | ' + OutB+'KB   | ' + DateToStr(Now) + '
| '+ TimeToStr(Now) );
  ADOTable1.Open;
  ADOTable1.Append;

```

```

ADOTable1.FieldByName('Traffic In').AsString:=InB;
ADOTable1.FieldByName('Traffic Out').AsString:=OutB;
ADOTable1.FieldByName('Date').AsString:=DateToStr(now);
ADOTable1.FieldByName('Time').AsString:=TimeToStr(Now);
ADOTable1.Post;
end:

```

Sehingga hasilnya ketika di runing akan terlihat seperti berikut :

ID	Traffic In	Traffic Out	Date	Time
91	-358124.698	624382.67	8/17/2011	9:44:50 PM
92	0.148	0.203	8/17/2011	9:44:50 PM
93	0.148	0.203	8/17/2011	9:44:51 PM
94	0.148	0.203	8/17/2011	9:44:52 PM
95	0.148	0.203	8/17/2011	9:44:53 PM
96	0.148	0.203	8/17/2011	9:44:54 PM
97	0.148	0.203	8/17/2011	9:44:55 PM
98	0.148	0.203	8/17/2011	9:44:56 PM
99	0.148	0.203	8/17/2011	9:44:57 PM
100	0.148	0.203	8/17/2011	9:44:58 PM
101	0.148	0.203	8/17/2011	9:44:59 PM
102	0.148	0.203	8/17/2011	9:45:00 PM
103	0	0.203	8/17/2011	9:45:06 PM
104	0.296	0.203	8/17/2011	9:45:06 PM
105	0.148	0.203	8/17/2011	9:45:07 PM
106	0.148	0.203	8/17/2011	9:45:08 PM
107	0.148	0.203	8/17/2011	9:45:09 PM
108	0.148	0.203	8/17/2011	9:45:10 PM
109	0	0.203	8/17/2011	9:45:16 PM
110	0.296	0.203	8/17/2011	9:45:16 PM
111	0.148	0.203	8/17/2011	9:45:17 PM
112	0.148	0.203	8/17/2011	9:45:18 PM
113	0.148	0.203	8/17/2011	9:45:19 PM
114	0.148	0.203	8/17/2011	9:45:20 PM
115	0.148	0.203	8/17/2011	9:45:21 PM
116	0.148	0.203	8/17/2011	9:45:22 PM
117	0.148	0.203	8/17/2011	9:45:23 PM
118	0.148	0.203	8/17/2011	9:45:24 PM
119	0.148	0.203	8/17/2011	9:45:25 PM
120	0	0.203	8/17/2011	9:45:31 PM
<Auto				

Gambar 4.37 Hasil Log Traffic Pada Database

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Dari pengujian yang dilakukan, diperoleh kesimpulan sebagai berikut :

1. Dari grafik yang ditampilkan pada aplikasi *monitoring traffic* jaringan VoIP memperlihatkan grafik untuk paket data bukan tingkat komunikasi suara.
2. Dari hasil pengujian komunikasi antar *extension user* didapatkan *traffic* data sebagai berikut:
  - Untuk komunikasi *extension user* 101 – 102 didapatkan *traffic* sebesar 2,167 KB.
  - Untuk komunikasi *extension user* 101 – 102 dan 201 – 204 didapatkan *traffic* sebesar 7,557 KB
  - Untuk komunikasi *extension user* 101 – 102, 201 – 204 dan 202 - 203 didapatkan *traffic* sebesar 12,27 KB
3. Untuk nilai *traffic in* (*traffic* yang masuk ke IP PBX) dan *traffic Out* (*traffic* yang keluar ke IP PBX) pada saat pengujian memiliki nilai *traffic* yang sama.
4. Pada pengujian aplikasi *monitoring traffic* ini, Dalam waktu satu menit *traffic* data dapat terbaca tetapi tidak dapat ditentukan pada detik berapa *traffic* tersebut melintas.
5. Aplikasi ini dapat memonitor komunikasi *multiple* sebanyak *user extension* yang telah disediakan oleh administrator.

## 5.2 Saran-saran

Berikut ini adalah saran yang diberikan untuk pengembangan selanjutnya.

1. Untuk pengembangan selanjutnya dapat ditambahkan beberapa jenis monitoring seperti kualitas suara pada VoIP.
2. Pengujian aplikasi monitoring traffic pada jaringan VoIP ini tidak hanya diimplementasikan pada Jaringan VoIP sebatas Jaringan LAN tetapi dapat juga dikembangkan untuk Jaringan WAN, Internet maupun Jaringan VPN.
3. Untuk pengembangan aplikasi monitoring *traffic* selanjutnya bisa ditambahkan fasilitas deteksi alamat IP *Interface* secara otomatis tanpa harus menginputkan terlebih dahulu.

## DAFTAR PUSTAKA

- [1].Dharma Oetomo, Budi Sutedjo. *Konsep dan Perancangan Jaringan Komputer*. Yogyakarta ; Andi offset. 2003.
- [2].Kristanto Andri. *Jaringan Komputer*. Yogyakarta : Graha Ilmu. 2003.
- [3].Dharma Oetomo, Budi Sutedjo. *Konsep dan Perancangan Jaringan Komputer*. yogyakarta ; Andi offset. 2003.
- [4].Sugeng, Winarno. *Membangun Telepon Berbasis VoIP*, Bandung, 2005.
- [5].Schaum's. *Computer Networking*. ERLANGGA. 2004.
- [7].Drew, Heywood. *Konsep dan Penerapan TCP/IP*. Andi Yogyakarta. 2008.
- [8]. Todd, Lammle. *Cisco Certified Network Associate. Elexmedia Komputindo. Jakarta Administration.2008.*
- [9].Taufiq, Mochammad. *Membuat SIP Extension Pada Linux TrixBos Untuk Server VoIP*.
- [10].Efendi, Yusuf. *Merancang Jaringan Komunikasi VoIP Sederhana Dengan Server VoIP TrixBos Yang Dilengkapi VOManager dan Open VPN*.
- [11].About Delphi Programming From Zarko Gajic, your Guide to Delphi Programming.
- [12].[http://opensource.telkomspeedy.com/wiki/index.php/Internet\\_Telepon](http://opensource.telkomspeedy.com/wiki/index.php/Internet_Telepon)



# LAMPIRAN



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG  
**INSTITUT TEKNOLOGI NASIONAL MALANG**

FAKULTAS TEKNOLOGI INDUSTRI  
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN  
PROGRAM PASCASARJANA MAGISTER TEKNIK

ERSERO) MALANG  
NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145  
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

**BERITA ACARA UJIAN SKRIPSI  
FAKULTAS TEKNOLOGI INDUSTRI**


NAMA : ARIF NARENDRA  
NIM : 07.12.519  
JURUSAN : Teknik Komputer dan Informatika S-1  
JUDUL : **DESAIN DAN IMPLEMENTASI APLIKASI MONITORING  
TRAFFIC PADA JARINGAN VOIP**

Dipertahankan dihadapan Tim Penguji Skripsi Jenjang Program Strata Satu (S-1)

Pada Hari : Selasa  
Tanggal : 9 Agustus 2011  
Dengan Nilai : 84,25 ( A )<sup>2</sup>


**PANITIA UJIAN SKRIPSI**

**KETUA,**



**Ir. Yusuf Ismail Nakhoda, MT**  
NIP. X.1018800189

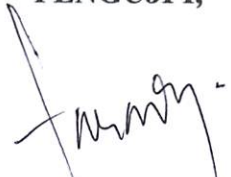
**SEKRETARIS,**



**Dr. Aryuanto S, ST, MT**  
NIP. Y.1030800417


**ANGGOTA PENGUJI**

**PENGUJI I,**



**Irmalia S. Faradisa, ST, MT**  
NIP. P.1030000365

**PENGUJI II,**



**Ahmad Faisol, ST**  
NIP. P.1031000431



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG  
**INSTITUT TEKNOLOGI NASIONAL MALANG**

FAKULTAS TEKNOLOGI INDUSTRI  
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN  
PROGRAM PASCASARJANA MAGISTER TEKNIK

ERSERO) MALANG  
NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145  
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

**BERITA ACARA UJIAN SKRIPSI  
FAKULTAS TEKNOLOGI INDUSTRI**


NAMA : ARIF NARENDRA  
NIM : 07.12.519  
JURUSAN : Teknik Komputer dan Informatika S-1  
JUDUL : **DESAIN DAN IMPLEMENTASI APLIKASI MONITORING  
TRAFFIC PADA JARINGAN VOIP**

Dipertahankan dihadapan Tim Penguji Skripsi Jenjang Program Strata Satu (S-1)


Pada Hari : Selasa  
Tanggal : 9 Agustus 2011  
Dengan Nilai : 84,25 ( A ) *rs*

**PANITIA UJIAN SKRIPSI**

**KETUA,**


  
**Ir. Yusuf Ismail Nakhoda, MT**  
NIP.Y.1018800189

**SEKRETARIS,**


  
**Dr. Aryuanto S, ST. MT**  
NIP.Y.1030800417

**ANGGOTA PENGUJI**

**PENGUJI I,**

  
**Irmalia S. Faradisa, ST, MT**  
NIP.P.1030000365

**PENGUJI II,**

  
**Ahmad Falsol, ST**  
NIP. P.1031000431





INSTITUT TEKNOLOGI NASIONAL MALANG  
FAKULTAS TEKNOLOGI INDUSTRI  
JURUSAN TEKNIK ELEKTRO

### Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika / T. Infokom, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : ARIF NARENDRA  
NIM : 0712515  
Perbaikan melalui :

Kesimpulan no 3 & 4 Alasannya

Malang,

5-8-11

(Wana)



INSTITUT TEKNOLOGI NASIONAL MALANG  
FAKULTAS TEKNOLOGI INDUSTRI  
JURUSAN TEKNIK ELEKTRO

### Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika / T. Infokom, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : Arip Norendana  
NIM : 0712519  
Perbaikan melalui :

- Tambahkan fasilitas penyimpanan log  
on-line ke database (Access / MySQL)
- Tampilkan laporan hasil penyimpanan log  
baik berupa data angka maupun grafik

Malang, 9-08-2011



## PERMOHONAN PERSETUJUAN SKRIPSI

Yang betanda tangan dibawah ini :

Nama : Arif Narendra  
NIM : 0712519  
Semester : VIII  
Fakultas : Teknologi Industri  
Jurusan : Teknik Elektro S-1  
Konsentrasi : ~~TEKNIK ELEKTRONIKA~~  
~~TEKNIK ENERGI LISTRIK~~  
~~TEKNIK KOMPUTER DAN INFORMATIKA~~  
~~TEKNIK KOMPUTER~~  
~~TEKNIK TELEKOMUNIKASI~~  
Alamat : Jln. hoki no 8 Malang

Dengan ini kami mengajukan permohonan untuk mendapatkan persetujuan untuk membuat *SKRIPSI Tingkat Sarjana*. Untuk melengkapi permohonan tersebut, bersama kami lampirkan persyaratan-persyaratan yang harus dipenuhi.

Adapun persyaratan-persyaratan pengambilan *SKRIPSI* adalah sebagai berikut :

1. Telah melaksanakan semua praktikum sesuai dengan konsentrasinya (.....)
2. Telah lulus dan menyerahkan Laporan Praktek Kerja (.....)
3. Telah lulus seluruh mata kuliah keahlian (MKB) sesuai konsentrasinya (.....)
4. Telah menenpuh mata kuliah  $\geq 134$  sks dengan IPK  $\geq 2$  dan tidak ada nilai E (.....)
5. Telah mengikuti secara aktif kegiatan seminar skripsi yang diadakan Jurusan (.....)
6. Memenuhi persyaratan administrasi (.....)

Demikian permohonan ini untuk mendapatkan penyelesaian lebih lanjut dan atas perhatiannya kami ucapkan terima kasih.

Telah diteliti kebenaran data tersebut diatas  
Recording Teknik Elektro

(*[Signature]*)  
(.....)

Malang, .....201

Penohon

(*[Signature]*)

(Arif Narendra.....)

Disetujui  
Ketua Jurusan Teknik Elektro

(*[Signature]*)  
Ir. Yusuf Ismail Nakhoda, MT  
NIP. Y. 1018800189

Mengetahui  
Dosen Wali

(*[Signature]*)  
(Ahmad Faizal, ST.....)

Catatan :

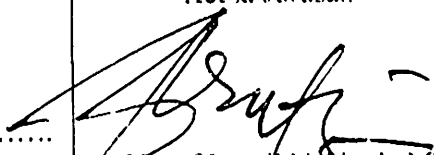
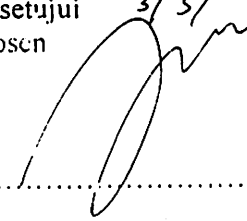
Bagi mahasiswa yang telah memenuhi persyaratan mengambil SKRIPSI agar membuat proposal dan mendapa persetujuan dari Ketua Jurusan/Sekretaris Jurusan T. Elektro S-1

1. IPK 4.94 / 3.58
2. 138
3. praktikum lengkap



## LEMBAR PENGAJUAN JUDUL SKRIPSI JURUSAN TEKNIK ELEKTRO S-1

Konsentrasi : Teknik Energi Listrik / Teknik Elektronika / Teknik Komputer &  
Informatika / Teknik Komputer / Teknik Telekomunikasi\*)

1.	Nama Mahasiswa: <u>ARIF NARENDRA</u>	Nim: <u>0712519</u>							
2.	Waktu Pengajuan	Tanggal: _____ Bulan: _____ Tahun: _____							
3.	Spesifikasi Judul (berilah tanda silang)**)								
	<table style="width: 100%; border: none;"> <tr> <td style="width: 50%; border: none;">a. Sistem Tenaga Elektrik</td> <td style="width: 50%; border: none;">e. Elektronika &amp; Komponen</td> </tr> <tr> <td style="border: none;">b. Energi &amp; Konversi Energi</td> <td style="border: none;">f. Elektronika Digital &amp; Komputer</td> </tr> <tr> <td style="border: none;">c. Tegangan Tinggi &amp; Pengukuran</td> <td style="border: none;">g. Elektronika Komunikasi</td> </tr> <tr> <td style="border: none;">d. Sistem Kendali Industri</td> <td style="border: none;">h. lainnya .....</td> </tr> </table>		a. Sistem Tenaga Elektrik	e. Elektronika & Komponen	b. Energi & Konversi Energi	f. Elektronika Digital & Komputer	c. Tegangan Tinggi & Pengukuran	g. Elektronika Komunikasi	d. Sistem Kendali Industri
a. Sistem Tenaga Elektrik	e. Elektronika & Komponen								
b. Energi & Konversi Energi	f. Elektronika Digital & Komputer								
c. Tegangan Tinggi & Pengukuran	g. Elektronika Komunikasi								
d. Sistem Kendali Industri	h. lainnya .....								
4.	Konsultasikan judul sesuai materi bidang ilmu kepada Dosen*)  <u>Dr. Arjuanto, ST, MT</u>	Ketua Jurusan  <u>Dr. Yusuf Ismail N. Noda, MT</u> NIP. Y. 1018800189							
5.	Judul yang diajukan mahasiswa:	<u>DESAIN DAN IMPLEMENTASI APLIKASI MONITORING TRAFFIC PADA SARINGAN VoIP</u>							
6.	Perubahan judul yang disetujui Dosen sesuai materi bidang ilmu	..... ..... .....							
7.	Catatan: ..... ..... .....								
	Persetujuan Judul skripsi yang dikonsultasikan kepada Dosen materi bidang ilmu	Disetujui <u>3/3/</u> 201 <u>1</u> Dosen 							

**Perhatian:**

1. Formulir pengajuan ini harap dikembalikan kepada jurusan paling lambat satu minggu setelah disetujui kelompok dosen keahlian dengan dilampirkan proposal skripsi beserta persyaratan skripsi sesuai form S-1
2. Keterangan: \*) Coret yang tidak perlu  
\*\*) dilingkari a, b, c, ..... atau g sesuai bidang keahlian

**INSTITUT TEKNOLOGI NASIONAL**

**Jln. Raya Karanglo, Km. 2 Malang**

---

Lampiran : 1 (satu) berkas

**Pembimbing Skripsi**

Kepada : Yth. Bapak Dr. Eng. Aryunto S,ST,MT

Dosen Institut Teknologi Nasional Malang

Yang bertanda tangan di bawah ini:

Nama : Arif Narendra

NIM : 07.12.519

Jurusan : Teknik Elektro S-1

Konsentrasi : Teknik Komputer dan Informatika

Dengan ini mengajukan permohonan, kiranya Bapak bersedia menjadi Dosen Pembimbing Pendamping untuk peyusunan Skripsi dengan judul (proposal terlampi.):

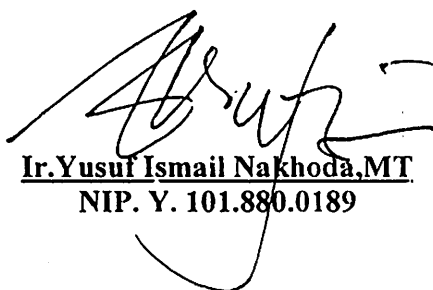
**“ Desain dan Implementasi Aplikasi Monitoring *Traffic* Pada Jaringan VoIP “**

Adapun tugas tersebut sebagai salah satu syarat untuk menempuh Ujian Akhir Sarjana Teknik.

Demikian permohonan kami buat dan atas kesediaan Bapak kami ucapkan terima kasih.

Mengetahui

Ketua Jurusan Teknik Elektro



**Ir. Yusuf Ismail Nakhoda, MT**  
NIP. Y. 101.880.0189

Melampirkan,

Format kami



**Arif Narendra**

Form S-3a

**PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI**

Sesuai permohonan dari mahasiswa/i:

Nama : Arif Narendra

NIM : 07.12.519

Semester : VIII (Delapan)

Jurusan : Teknik Elektro S-1

Konsentrasi : Teknik Komputer dan Informatika

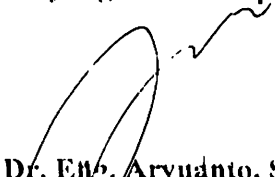
Dengan ini menyatakan bersedia/ tidak bersedia\*) Membimbing skripsi dari mahasiswa tersebut, dengan judul:

**“ Desain dan Implementasi Aplikasi Monitoring *Traffic* Pada Jaringan VoIP “**

Demikian surat pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Malang, 02 - 03 - 2011

Kami yang membuat pernyataan

  
**Dr. Eng. Aryanto, ST, MT**  
NIP.Y.1030800417

**Catatan :**

Setelah disetujui agar formulir ini

Diserahkan mahasiswa/i yang bersangkutan

Kepada Jurusan untuk diproses lebih lanjut.

\*) Coret yang tidak perlu

INSTITUT TEKNOLOGI NASIONAL

Jln. Raya Karanglo, Km. 2 Malang

---

**PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI**

Sesuai permohonan dari mahasiswa/i:

Nama : Arif Narendra

NIM : 07.12.519

Semester : VIII (Delapan)

Jurusan : Teknik Elektro S-1

Konsentrasi : Teknik Komputer dan Informatika


Dengan ini menyatakan bersedia/ tidak bersedia\*) Membimbing skripsi dari mahasiswa tersebut, dengan judul:

**“ Desain dan Implementasi Aplikasi Monitoring *Traffic* Pada Jaringan VoIP “**

Demikian surat pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Malang, 08 Maret 2011

Kami yang membuat pernyataan



Soehar'i, ST  
NIP.Y.103' 700309

**Catatan :**

Setelah disetujui agar formulir ini

Diserahkan mahasiswa/i yang bersangkutan

Kepada Jurusan untuk diproses lebih lanjut.

\*) Coret yang tidak perlu

Form S-3b



## BERITA ACARA SEMINAR PROPOSAL SKRIPSI JURUSAN TEKNIK ELEKTRO S-1

Konsentrasi : Teknik Energi-Listrik/Teknik Elektronika/Teknik Komputer & Informatika\*)

1.	Nama Mahasiswa: <b>ARIF NABENDRA</b>	Nim: <b>0712519</b>		
2.	Keterangan	Tanggal	Waktu	Tempat
	Pelaksanaan	<b>2 - APRIL - 2011</b>		Ruang:
Spesifikasi Judul (berilah tanda silang)**)				
3.	a. Sistem Tenaga Elektrik	e. Elektronika & Komponen		
	b. Energi & Konversi Energi	f. Elektronika Digital & Komputer		
	c. Tegangan Tinggi & Pengukuran	g. Elektronika Komunikasi		
	d. Sistem Kendali Industri	<input checked="" type="checkbox"/> h. lainnya <b>komputer dan informatika</b>		
4.	Judul Proposal yang diseminarkan Mahasiswa	<b>DESAIN DAN IMPLEMENTASI APLIKASI MONITORING TRAFFIC PADA JARINGAN VOIP</b>		
5.	Perubahan Judul yang diusulkan oleh Kelompok Dosen Keahlian			
6.	Catatan:			
Catatan:				
Persetujuan Judul Skripsi				
7.	Disetujui, Dosen Keahlian I	Disetujui, Dosen Keahlian II		
		 <b>M. Ibrahim Ashari</b>		
	Mengetahui, Ketua Jurusan.	Disetujui, Calon Dosen Pembimbing ybs		
 <b>Ir. Yusuf Ismail Nakhoda, MT</b> NIP. Y. 1018800189		Pembimbing I	Pembimbing II	

Perhatian

1. Keterangan. \*) Coret yang tidak perlu

\*\*) dilingkari. a, b, c, ..... atau g sesuai bidang keahlian.





PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG  
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI  
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN  
PROGRAM PASCASARJANA MAGISTER TEKNIK

INSTITUT TEKNOLOGI NASIONAL MALANG  
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigurgura No. 2 Telp. (0341) 551431 (Hunting) Fax. (0341) 553015 Malang 65145  
Kampus II : Jl. Raya Keranglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 04 April 2011

Nomor : ITN-212/I.TA/2/11  
Lampiran : -  
Perihal : BIMBINGAN SKRIPSI

Kepada : Yth. Sdr./i. **DR. ENG. ARYUANTO SOETEDJO, ST, MT**  
Dosen Institut Teknologi Nasional Malang

Dosen Pembimbing  
Jurusan Teknik Elektro S-1  
di  
Malang

Dengan hormat  
Seperti yang dimohonkan dan persetujuan dalam Proposal Skripsi  
Untuk Mahasiswa :

Nama : ARIF NARENDRA  
Nim : 0712519  
Fakultas : Teknologi Industri  
Jurusan : Teknik Elektro S-1  
Konsentrasi : Teknik **Komputer & Informatika**

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya  
kepada Saudara/i selama masa waktu (enam) 6 bulan, terhitung mulai  
tanggal :

02 April 2011 s/d 02 September 2011

Sebagai satu syarat untuk menempuh ujian Sarjana Teknik,  
Jurusan Teknik Elektro S-1  
Demikian agar perhatian dan atas perhatiannya kami sampaikan terima  
kasih



Ketua Jurusan  
Teknik Elektro S-1

*(Signature)*  
Ir Yusuf Ismail Nakhoda, MT  
Nip. Y.1018800189

Tembusan Kepada Yth :

1. Mahasiswa Yang bersangkutan
2. Arsip
3. Coret yang tidak perlu

Form. S 4a



## FORMULIR BIMBINGAN SKRIPSI

Nama : Arif Narendra  
Nim : 07.12.519  
Masa Bimbingan : 2 April 2011 s/d 2 September 2011 *Arif*  
Judul Skripsi : Desain dan Implementasi Aplikasi Monitoring Traffic Pada Jaringan VoIP

No	Tanggal	Uraian	Paraf Pembimbing
1	20-4-2011	Konsultasi protokol jaringan VoIP	<i>Arif</i>
2	1-6-2011	Konsultasi Pengambilan data traffic	<i>Arif</i>
3	22-6-2011	Bimbingan Laporan semua Bab (Revisi)	<i>Arif</i>
4	28-6-2011	Acc Bab I dan Bab II	<i>Arif</i>
5	4-7-2011	Acc Bab III	<i>Arif</i>
6	13-7-2011	Konsultasi pembuatan waktu seperti 570	<i>Arif</i>
7	14-7-2011	Bimbingan makalah seminar hasil (Acc)	<i>Arif</i>
8	3-8-2011	Acc bab IV dan Bab V	<i>Arif</i>
9			
10			

Malang,  
Dosen Pembimbing I

**Dr. Eng. Aryuanto, ST, MT**  
NIP.Y.1030800417



## FORMULIR BIMBINGAN SKRIPSI

Nama : Arif Narendra  
Nim : 07.12.519  
Masa Bimbingan : 2 April 2011 s/d 2 September 2011 *Benj*  
Judul Skripsi : Desain dan Implementasi Aplikasi Monitoring *Traffic* Pada Jaringan VoIP

No	Tanggal	Uraian	Paraf Pembimbing
1	4-5-2011	Konsultasi protokol VoIP dan request traffic	<i>Judi</i>
2	1-6-2011	Konsultasi program (kekurangan)	<i>Judi</i>
3	27-6-2011	Bimbingan Laporan semua Bab.	<i>Judi</i>
4	13-7-2011	Konsultasi pengujian program	<i>Judi</i>
5	14-7-2011	Acc Makalah seminar Hasil!	<i>Judi</i>
6	4-8-2011	Acc Laporan semua Bab	<i>Judi</i>
7			
8			
9			
10			

Malang, . . .  
Dosen Pembimbing II

Sotyohadi, ST  
NIP.Y.1039700309

## ❖ Listing Program Menu Utama

```
unit menu;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Menus, XPMAN, jpeg, ExtCtrls, StdCtrls;

type
  TForm4 = class(TForm);
  TForm1 = class(TForm)
  MainMenu1: TMainMenu;
  Menu1: TMenuItem;
  ShowTraffic1: TMenuItem;
  raceSNMP1: TMenuItem;
  Exit1: TMenuItem;
  About1: TMenuItem;
  PCInformation1: TMenuItem;
  Image1: TImage;
  Label1: TLabel;
  Timer1: TTimer;
  Label2: TLabel;
  procedure Exit1Click(Sender: TObject);
  procedure ShowTraffic1Click(Sender: TObject);
  procedure PCInformation1Click(Sender: TObject);
  procedure FormCreate(Sender: TObject);
  procedure raceSNMP1Click(Sender: TObject);
  procedure ArifNarendra1Click(Sender: TObject);
  procedure Timer1Timer(Sender: TObject);
  procedure About1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

uses Show_Trafic, Trc_snmp, Pc_Info, About;
```

```
{ $R *.dfm }
```

```
procedure TForm1.Exit1Click(Sender: TObject);  
begin  
    Application.Terminate;  
end;
```

```
procedure TForm1.ShowTraffic1Click(Sender: TObject);  
begin  
    Form4.Show;  
    Form2.Hide;  
    Form3.Hide;  
    Form5.Hide;  
end;
```

```
procedure TForm1.PCInformation1Click(Sender: TObject);  
begin  
    Form3.Show;  
    Form2.Hide;  
    Form4.Hide;  
    Form5.Hide;  
end;
```

```
procedure TForm1.FormCreate(Sender: TObject);  
begin  
    Form1.Top:=0;  
    Form1.Left:=0;  
end;
```

```
procedure TForm1.raceSNMP1Click(Sender: TObject);  
begin  
    Form2.Show;  
    Form4.Hide;  
    Form3.Hide;  
    Form5.Hide;  
end;
```

```
procedure TForm1.ArifNarendra1Click(Sender: TObject);  
begin  
    Form5.Show;  
    Form4.Hide;  
    Form2.Hide;  
    Form3.Hide;  
end;
```

```
procedure TForm1.Timer1Timer(Sender: TObject);  
begin  
    Label1.Caption:=TimeToStr(time);
```

```

Label2.Caption:=DateToStr(Date);
end;

procedure TForm1.About1Click(Sender: TObject);
begin
Form5.Show;
Form4.Hide;
Form2.Hide;
Form3.Hide;
end;

end.

```

### ❖ Sourcecode Show Traffic

```

unit Show_Trafik;

interface

uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, TeEngine, Series, ExtCtrls, TeeProcs, Chart,
IdBaseComponent, IdComponent, IdUDPBase, IdUDPClient, IdSNMP, XPMan,
ComCtrls, IdDNSResolver;

type
TForm4 = class(TForm)
XPManifest1: TXPManifest;
Memo1: TMemo;
lbl_outboundmb: TLabel;
lbl_outboundgb: TLabel;
lbl_outbound: TLabel;
lbl_out: TLabel;
lbl_inboundmb: TLabel;
lbl_inboundgb: TLabel;
lbl_inbound: TLabel;
lbl_in: TLabel;
Label4: TLabel;
Label3: TLabel;
Label2: TLabel;
Label1: TLabel;
Edit3: TEdit;
Edit2: TEdit;
Edit1: TEdit;
Chart5: TChart;
FastLineSeries7: TFastLineSeries;
FastLineSeries8: TFastLineSeries;

```

Chart4: TChart;  
FastLineSeries5: TFastLineSeries;  
FastLineSeries6: TFastLineSeries;  
Chart3: TChart;  
FastLineSeries3: TFastLineSeries;  
FastLineSeries4: TFastLineSeries;  
Chart2: TChart;  
FastLineSeries1: TFastLineSeries;  
FastLineSeries2: TFastLineSeries;  
Chart1: TChart;  
Button4: TButton;  
Button3: TButton;  
Button2: TButton;  
Button1: TButton;  
btn\_mb: TButton;  
btn\_kb: TButton;  
btn\_gb: TButton;  
IdSNMP1: TIdSNMP;  
Timer1: TTimer;  
Label5: TLabel;  
Label6: TLabel;  
Label7: TLabel;  
Label8: TLabel;  
Label9: TLabel;  
Label10: TLabel;  
Label11: TLabel;  
Label12: TLabel;  
Label13: TLabel;  
Label14: TLabel;  
Label15: TLabel;  
Label16: TLabel;  
Label17: TLabel;  
Button5: TButton;  
Label18: TLabel;  
Timer2: TTimer;  
Label19: TLabel;  
Label20: TLabel;  
Label21: TLabel;  
Label22: TLabel;  
Label23: TLabel;  
Label24: TLabel;  
Label25: TLabel;  
Label26: TLabel;  
Label27: TLabel;  
Label28: TLabel;  
Label29: TLabel;  
Label30: TLabel;  
Label31: TLabel;

```
Label32: TLabel;  
Label33: TLabel;  
Label34: TLabel;  
Label35: TLabel;  
Label36: TLabel;  
Label37: TLabel;  
Label38: TLabel;  
Label39: TLabel;  
Label40: TLabel;  
Label41: TLabel;  
Label42: TLabel;  
Series1: TFastLineSeries;  
Series2: TFastLineSeries;  
procedure Button1Click(Sender: TObject);  
procedure Button2Click(Sender: TObject);  
procedure btn_gbClick(Sender: TObject);  
procedure btn_kbClick(Sender: TObject);  
procedure btn_mbClick(Sender: TObject);  
procedure Button3Click(Sender: TObject);  
procedure Button4Click(Sender: TObject);  
procedure Timer1Timer(Sender: TObject);  
procedure FormCreate(Sender: TObject);  
procedure Button5Click(Sender: TObject);  
procedure Timer2Timer(Sender: TObject);
```

```
private  
  { Private declarations }  
public  
  { Public declarations }  
  procedure log;  
  procedure ReadtoLog;  
  procedure WritetoLog;  
end;
```

```
var  
  Form4: TForm4;  
  nil_1:real=0;  
  nil_2:real=0;  
  konversi :real;  
  konversi1,konversi2,konversi3:real;  
  konout:real;  
  konout1,konout2,konout3:real;  
  A:Array[1..301] Of real;  
  B:Array[1..301] Of real;  
  C:Array[1..301] Of real;  
  D:Array[1..301] Of real;  
  E:Array[1..301] Of real;  
  F:Array[1..301] Of real;
```



```
mfile:TextFile;  
textStr:String;  
x:integer;  
t:string;
```

implementation

uses DateUtils;

```
{ $R *.dfm }
```

```
procedure TForm4.Button1Click(Sender: TObject);
```

```
var
```

```
  i:Integer;
```

```
begin
```

```
  Timer1.Enabled:=True;
```

```
  IdSNMP1.Community:='public';
```

```
  IdSNMP1.Host:=Edit1.Text;
```

```
  IdSNMP1.Active:=True;
```

```
  IdSNMP1.Query.Port:=161;
```

```
  IdSNMP1.Query.Clear;
```

```
  IdSNMP1.Query.MIBAdd(Edit2.Text,"");
```

```
  IdSNMP1.Query.PDUType:=PDUGetRequest;
```

```
  IdSNMP1.Query.PDUType:=PDUGetNextRequest;
```

```
  if IdSNMP1.SendQuery then
```

```
    begin
```

```
      for i:=0 to IdSNMP1.Reply.ValueCount -1 do
```

```
        konversi:=StrToFloat(Format('%s',[IdSNMP1.Reply.Value[i]]));
```

```
      end
```

```
    else
```

```
      Memo1.Lines.Add('Inputkan IP Interface');
```

```
      IdSNMP1.Query.Clear;
```

```
      IdSNMP1.Query.MIBAdd(Edit3.Text,"");
```

```
      IdSNMP1.Query.PDUType:=PDUGetRequest;
```

```
      IdSNMP1.Query.PDUType:=PDUGetNextRequest;
```

```
      if IdSNMP1.SendQuery then
```

```
        begin
```

```
          for i:=0 to IdSNMP1.Reply.ValueCount -1 do
```

```
            konout:=StrToFloat(Format('%s',[IdSNMP1.Reply.Value[i]]));
```

```
            lbl_in.Caption:=FloatToStr(konversi-nil_1);
```

```
            nil_1:=konversi;
```

```
            konversi1:=StrToFloat(lbl_in.Caption)/1000;
```

```
            konversi2:=StrToFloat(lbl_in.Caption)/1000000;
```

```
            konversi2:=StrToFloat(lbl_in.Caption)/1000000000;
```

```
            lbl_inbound.Caption:=FloatToStr(konversi1);
```

```
            lbl_inboundmb.Caption:=FloatToStr(konversi2);
```

```
            lbl_inboundgb.Caption:=FloatToStr(konversi3);
```

```
            lbl_out.Caption:=FloatToStr(konout-nil_2);
```

```

        nil_2:=konout;
        konout1:=StrToFloat(lbl_out.Caption)/1000;
        konout2:=StrToFloat(lbl_out.Caption)/1000000;
        konout3:=StrToFloat(lbl_out.Caption)/1000000000;
        lbl_outbound.Caption:=FloatToStr(konout1);
        lbl_outboundmb.Caption:=FloatToStr(konout2);
        lbl_outboundgb.Caption:=FloatToStr(konout3);
        log;// menyimpan ke memo
    end
    else;
    IdSNMP1.Active:=False;
end;

```

```

procedure TForm4.Button2Click(Sender: TObject);
begin
    Timer1.Enabled:=False;
    WritetoLog;
end;

```

```

procedure TForm4.btn_gbClick(Sender: TObject);
begin
    Chart1.Visible:= False;
    Chart2.Visible:= False;
    Chart3.Visible:= True;
    Chart4.Visible:= false;
    Chart5.Visible:= false;
    Chart3.BottomAxis.DateTimeFormat:='HH:MM';
end;

```

```

procedure TForm4.btn_kbClick(Sender: TObject);
begin
    Chart1.Visible:= True;
    Chart2.Visible:= False;
    Chart3.Visible:= False;
    Chart4.Visible:= False;
    Chart5.Visible:= False;
    Chart1.BottomAxis.DateTimeFormat:='HH:MM';
end;

```

```

procedure TForm4.btn_mbClick(Sender: TObject);
begin
    Chart1.Visible:= False;
    Chart2.Visible:= True;
    Chart3.Visible:= False;
    Chart4.Visible:= False;
    Chart5.Visible:= False;
    Chart2.BottomAxis.DateTimeFormat:='HH:MM';
end;

```



```

procedure TForm4.Button3Click(Sender: TObject);
begin
  Chart1.Visible:= False;
  Chart2.Visible:= False;
  Chart3.Visible:= False;
  Chart4.Visible:= False;
  Chart5.Visible:= True;
  Chart5.BottomAxis.DateTimeFormat:='HH:MM';
  Chart5.BottomAxis.ExactDateTime:=true;
end;

```

```

procedure TForm4.Button4Click(Sender: TObject);
begin
  Chart1.Visible:= False;
  Chart2.Visible:= False;
  Chart3.Visible:= false;
  Chart4.Visible:= True;
  Chart5.visible:= False;
  Chart4.BottomAxis.DateTimeFormat:='HH:MM';
end;

```

```

procedure TForm4.Timer1Timer(Sender: TObject);
Var
  AA,BB,CC,DD,EE,FF:real;
  I:integer;
begin
  Button1 Click(Sender);
  AA:=StrToFloat(lbl_inbound.Caption);
  BB:=StrToFloat(lbl_outbound.Caption);
  CC:=StrToFloat(lbl_inboundmb.Caption);
  DD:=StrToFloat(lbl_outboundmb.Caption);
  EE:=StrToFloat(lbl_inboundgb.Caption);
  FF:=StrToFloat(lbl_outboundgb.Caption);
  A[300]:=AA;
  B[300]:=BB;
  C[300]:=CC;
  D[300]:=DD;
  E[300]:=EE;
  F[300]:=FF;
  i:=1;
  Repeat
    A[i]:=A[i+1];
    B[i]:=B[i+1];
    C[i]:=C[i+1];
    D[i]:=D[i+1];
    E[i]:=E[i+1];
    F[i]:=F[i+1];

```

```

I:=I+1;
Until I>299;
  Chart1.Series[0].Clear;
  Chart1.Series[1].Clear;
  Chart2.Series[0].Clear;
  Chart2.Series[1].Clear;
  Chart3.Series[0].Clear;
  Chart3.Series[1].Clear;
  Chart4.Series[0].Clear;
  Chart4.Series[1].Clear;
  Chart5.Series[0].Clear;
  Chart5.Series[1].Clear;
  for I:=1 To 300 DO
  Begin
    Chart1.Series[0].AddXY(i,A[i],"clRed);
    Chart1.Series[1].AddXY(i,B[i],"clGreen);
    Chart4.Series[0].AddXY(i,A[i],"clRed);
    Chart4.Series[1].AddXY(i,B[i],"clGreen);
    Chart5.Series[0].AddXY(i,A[i],"clRed);
    Chart5.Series[1].AddXY(i,B[i],"clGreen);
    Chart2.Series[0].AddXY(i,C[i],"clRed);
    Chart2.Series[1].AddXY(i,D[i],"clGreen);
    Chart3.Series[0].AddXY(i,C[i],"clRed);
    Chart3.Series[1].AddXY(i,D[i],"clGreen);
  end;
end;

```

```

procedure TForm4.log;
var
  InB,OutB : String;
begin
  InB := lbl_inbound.Caption;
  OutB := lbl_outbound.Caption;
  memo1.Lines.Add(' '+InB + 'KB | ' + OutB+'KB | ' + DateToStr(Now) + '
| '+ TimeToStr(Now) );
end;

```

```

procedure TForm4.ReadtoLog;
begin
  AssignFile(mFile, 'log.ini');
  Reset(mFile);
  while not Eof(mFile) do
  begin
    Readln(mFile, textStr );
    Memo1.Lines.Add(textStr);
  end;
  CloseFile(mFile);
end;

```

```

procedure TForm4.WritetoLog;
begin
  AssignFile(mFile, 'loghistory.ini');
  Rewrite(mFile);
  Writeln(mFile, Memo1.text);
  CloseFile(mFile);
end;
procedure TForm4.FormCreate(Sender: TObject);
begin
  Timer1.Enabled:=False;
  Edit1.Text:='192.168.1.1';
  Edit2.Text:='1.3.6.1.2.1.2.2.1.10.0';
  Edit3.Text:='1.3.6.1.2.1.2.2.1.16.0';
  Chart1.Visible:= False;
  Chart2.Visible:= False;
  Chart3.Visible:= False;
  Chart4.Visible:= False;
  lbl_in.Visible:=false;
  lbl_out.Visible:=false;
end;

procedure TForm4.Button5Click(Sender: TObject);
begin
  Close;
end;

procedure TForm4.Timer2Timer(Sender: TObject);
var
  waktu,dt:TDateTime;
begin
  dt:=1/24/60; // 1 hari/24 jam/60 menit
  waktu:=now;

  //Chart1--> untuk menit 1 (label18)
  label18.Caption:=FormatDateTime('hh:nn',waktu);
  label19.caption:=FormatDateTime('hh:nn',waktu-dt);
  label20.caption:=FormatDateTime('hh:nn',waktu-(2*dt));
  label21.caption:=FormatDateTime('hh:nn',waktu-(3*dt));
  label22.caption:=FormatDateTime('hh:nn',waktu-(4*dt));

  //chart 2
  label23.Caption:=FormatDateTime('hh:nn',waktu);
  label24.caption:=FormatDateTime('hh:nn',waktu-dt);
  label25.caption:=FormatDateTime('hh:nn',waktu-(2*dt));
  label26.caption:=FormatDateTime('hh:nn',waktu-(3*dt));
  label27.caption:=FormatDateTime('hh:nn',waktu-(4*dt));

```

```

//chart 3
label28.Caption:=FormatDateTime('hh:nn',waktu);
label29.caption:=FormatDateTime('hh:nn',waktu-dt);
label30.caption:=FormatDateTime('hh:nn',waktu-(2*dt));
label31.caption:=FormatDateTime('hh:nn',waktu-(3*dt));
label32.caption:=FormatDateTime('hh:nn',waktu-(4*dt));

//chart4
label33.Caption:=FormatDateTime('hh:nn',waktu);
label34.caption:=FormatDateTime('hh:nn',waktu-dt);
label35.caption:=FormatDateTime('hh:nn',waktu-(2*dt));
label36.caption:=FormatDateTime('hh:nn',waktu-(3*dt));
label37.caption:=FormatDateTime('hh:nn',waktu-(4*dt));

//chart5
label38.Caption:=FormatDateTime('hh:nn',waktu);
label39.caption:=FormatDateTime('hh:nn',waktu-dt);
label40.caption:=FormatDateTime('hh:nn',waktu-(2*dt));
label41.caption:=FormatDateTime('hh:nn',waktu-(3*dt));
label42.caption:=FormatDateTime('hh:nn',waktu-(4*dt))
end;

end.

```

#### ❖ Sourcecode PC Information

```

unit Pc_Info;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, IdBaseComponent, IdComponent, IdUDPBase, IdUDPClient,
  IdSNMP, ExtCtrls, XPMan;

type
  TForm3 = class(TForm)
    XPManifest1: TXPManifest;
    Timer1: TTimer;
    Memo2: TMemo;
    Memo1: TMemo;
    Label3: TLabel;
    Label2: TLabel;
    Label1: TLabel;
    IdSNMP1: TIdSNMP;
    Edit2: TEdit;
    Edit1: TEdit;

```

```

Button3: TButton;
Button2: TButton;
Button1: TButton;
Button4: TButton;
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure FormActivate(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure FormCreate(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

```

```

var
  Form3: TForm3;

```

implementation

```
{ $R *.dfm }
```

```

procedure TForm3.Button1Click(Sender: TObject);
var
  i:integer;
  banding:string;
begin
  Memo1.Lines.Clear;
  Timer1.Enabled:=True;
  IdSNMP1.Community:= 'public';
  IdSNMP1.Host:= Edit2.Text;
  IdSNMP1.Active := true;
  IdSNMP1.Query.Clear;
  IdSNMP1.Query.Port:=161;
  Idsnmp1.Query.MIBAdd(Edit1.Text,""); //enter your oid here
  IdSNMP1.Query.PDUType:= PDUGetRequest;
  IdSNMP1.Query.PDUType := PDUGetNextRequest; //PDUGetNextRequest for
WALK
  if idSNMP1.SendQuery then
    begin
      for i :=0 to IdSNMP1.Reply.ValueCount -1 do
        Memo1.lines.Add(Format('%s',[IdSNMP1.Reply.Value[i]]));
        Memo1.Lines.Add(Format('%s',[IdSNMP1.Reply.ValueOID[0]]));
      end
    else
      Memo1.Lines.Add('inputkan lagi oid');

```



```

Edit1.Text := Memo1.Lines[1];
banding:= Memo1.Lines[0];
//delete(banding,1,1);
if banding <> '0.0' THEN
  begin
    Memo2.Lines.Add(Memo1.Lines[0]);
  end
  else
    Timer1.Enabled:=false;
    IdSNMP1.Active := false;
end;

procedure TForm3.Button2Click(Sender: TObject);
begin
  Timer1.Enabled := False;
end;

procedure TForm3.Button3Click(Sender: TObject);
begin
  Memo1.Clear;
  Memo2.Clear;
end;

procedure TForm3.Button4Click(Sender: TObject);
begin
  Close;
end;

procedure TForm3.FormActivate(Sender: TObject);
begin
  Edit1.text:= '1.3.6.1.2.1.25.6.3.1.2.1 ';
end;

procedure TForm3.Timer1Timer(Sender: TObject);
begin
  Button1Click(sender);
end;

procedure TForm3.FormCreate(Sender: TObject);
begin
  Timer1.Enabled := False;
end;

end.

```

## ❖ Sourcecode Trace SNMP

```
unit Trace_SNMP;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, IdBaseComponent, IdComponent, IdUDPBase, IdUDPClient,  
IdSNMP, ScktComp, ExtCtrls, XPMan;
```

```
type
```

```
TForm2 = class(TForm)  
Label1: TLabel;  
Label2: TLabel;  
Button1: TButton;  
Memo1: TMemo;  
Edit1: TEdit;  
Edit2: TEdit;  
Button2: TButton;  
Memo2: TMemo;  
Button4: TButton;  
Button5: TButton;  
IdSNMP1: TIdSNMP;  
Timer1: TTimer;  
XPManifest1: TXPManifest;  
procedure Button1Click(Sender: TObject);  
procedure FormCreate(Sender: TObject);  
procedure Timer1Timer(Sender: TObject);  
procedure Button2Click(Sender: TObject);  
procedure Button4Click(Sender: TObject);  
procedure Button5Click(Sender: TObject);
```

```
private
```

```
{ Private declarations }
```

```
public
```

```
{ Public declarations }
```

```
end;
```

```
var
```

```
Form2: TForm;
```

```
implementation
```

```
{ $R *.dfm }
```

```
procedure TForm2.Button1Click(Sender: TObject);
```

```
var
```

```

i:integer;
oid:string;
begin
Memo1.Lines.Clear;
Timer1.Enabled:=True;
IdSNMP1.Community:= 'public';
IdSNMP1.Host:= Edit2.Text;
IdSNMP1.Active := true;
IdSNMP1.Query.Clear;
IdSNMP1.Query.Port:=161;
idsnmp1.Query.MIBAdd(Edit1.Text,""); //enter your oid here
IdSNMP1.Query.PDUType:= PDUGetRequest;
IdSNMP1.Query.PDUType := PDUGetNextRequest; //PDUGetNextRequest for
WALK
  if idSNMP1.SendQuery then
    begin
      for i :=0 to IdSNMP1.Reply.ValueCount - 1 do
        memo1.lines.Add(Format(' Value: %s',[IdSNMP1.Reply.Value[i]]));
        Memo1.Lines.Add(Format(' %s',[IdSNMP1.Reply.ValueOID[0]]));
      end
    else
      Memo1.Lines.Add('inputkan lagi oid');
      Edit1.Text := Memo1.Lines.ValueFromIndex[1];
      Memo2.Lines.Add(Memo1.Lines.ValueFromIndex[0]);
      Memo2.Lines.Add('oid:'+Memo1.Lines.ValueFromIndex[1]);
      IdSNMP1.Active := false;
    end;

procedure TForm2.FormCreate(Sender: TObject);
begin
  Edit2.Text:='192.168.1.1';
end;

procedure TForm2.Timer1Timer(Sender: TObject);
begin
  Button1.Click(Sender);
end;

procedure TForm2.Button2Click(Sender: TObject);
begin
  Timer1.Enabled := False;
end;

procedure TForm2.Button4Click(Sender: TObject);
begin
  Close;
end;

```

```
procedure TForm2.Button5Click(Sender: TObject);
begin
  Memo1.Clear;
  Memo2.Clear ;
end;

end.
```

#### ❖ Sourcecode About

```
unit About;
```

```
interface
```

```
uses
```

```
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls, jpeg;
```

```
type
```

```
  TForm5 = class(TForm)
```

```
    Label1: TLabel;
```

```
    Label2: TLabel;
```

```
    Label3: TLabel;
```

```
    Label4: TLabel;
```

```
    Label5: TLabel;
```

```
    Label6: TLabel;
```

```
    Label7: TLabel;
```

```
    Label8: TLabel;
```

```
    procedure FormClick(Sender: TObject);
```

```
  private
```

```
    { Private declarations }
```

```
  public
```

```
    { Public declarations }
```

```
end;
```

```
var
```

```
  Form5: TForm5;
```

```
implementation
```

```
{ $R *.dfm }
```

```
procedure TForm5.FormClick(Sender: TObject);
begin
  CLOSE;
end;
```

end.

## ❖ Set SNMP On Trixbox 2.8

```
#vi /etc/asterisk/res_snmp.conf
```

```
[general]
```

```
subagent = yes
```

```
enabled = yes
```

```
#chown asterisk:asterisk /etc/asterisk/res_snmp.conf
```

```
#cd /etc/snmp
```

```
#mv snmpd.conf snmpd.conf.orig
```

```
#vi snmpd.conf
```

```
rocommunity public
```

```
master agentx
```

```
#agentXSocket tcp:YOUR.ASTERISK.IP.ADDY:705
```

```
agentXperms 0660 0550 nobody asterisk
```

```
com2sec local localhost public
```

```
com2sec mynetwork0 192.168.1.0/8 public
```

```
#com2sec mynetwork1 YOUR.NETWORK/24 YOUR.COMMUNITY.NAME
```

```
#com2sec mynetwork2 YOUR.NETWORK/24 YOUR.COMMUNITY.NAME
```

```
#com2sec mynetwork3 YOUR.NETWORK/24 YOUR.COMMUNITY.NAME
```

```
group MyROGroup v1 local
```

```
group MyROGroup v2c local
```

```
group MyROGroup v1 mynetwork0
```

```
group MyROGroup v2c mynetwork0
```

```
#group MyROGroup any mynetwork1
```

```
#group MyROGroup any mynetwork2
```

```
#group MyROGroup any mynetwork3
```

```
view all included .1
```

```
access MyROGroup "" any noauth prefix all none none
```

```
#chmod 0755 /var/agentx
```

```
#service snmpd restart
```

```
#service snmptrapd restart
```

```
#amportal stop
```

```
#amportal start
```

```
#yum install net-snmp-utils
```

```
#export MIBS="+ASTERISK-MIB
```

```
#snmpwalk -v 1 -c public localhost asterisk
```

```
ASTERISK-MIB::astVersionString.0 = STRING: 1.4.22-4 RPM by vc-  
rpms@voipconsulting.nl
```

ASTERISK-MIB::astVersionTag.0 = Gauge32: 10422  
ASTERISK-MIB::astConfigUpTime.0 = Timeticks: (209041) 0:34:50.41  
ASTERISK-MIB::astConfigReloadTime.0 = Timeticks: (209042) 0:34:50.42  
ASTERISK-MIB::astConfigPid.0 = INTEGER: 2986  
ASTERISK-MIB::astConfigSocket.0 = STRING: /var/run/asterisk/asterisk.ctl  
ASTERISK-MIB::astNumModules.0 = INTEGER: 161  
ASTERISK-MIB::astNumIndications.0 = INTEGER: 40  
ASTERISK-MIB::astCurrentIndication.0 = STRING: us  
ASTERISK-MIB::astIndIndex.1 = INTEGER: 1  
ASTERISK-MIB::astIndIndex.2 = INTEGER: 2  
ASTERISK-MIB::astIndIndex.3 = INTEGER: 3  
ASTERISK-MIB::astIndIndex.4 = INTEGER: 4  
ASTERISK-MIB::astIndIndex.5 = INTEGER: 5  
ASTERISK-MIB::astIndIndex.6 = INTEGER: 6  
ASTERISK-MIB::astIndIndex.7 = INTEGER: 7  
ASTERISK-MIB::astIndIndex.8 = INTEGER: 8  
ASTERISK-MIB::astIndIndex.9 = INTEGER: 9