

BAB II DASAR TEORI

2.1 Literature Review

Pada tabel 2.1 merupakan *review* terhadap *literature - literature* penelitian sebelumnya yang telah mempelajari aspek yang terkait dalam penelitian penulis. Pengembangan yang terkait dengan penelitian penulis dapat dilihat beberapa *literature review* pada tabel 2.1.

Tabel 2.1 *Literature Review*

No.	Author & Judul Penelitian	Kelebihan & Kekurangan	Parameter Perbandingan Skripsi Penulis
1.	Muhammad Saefulloh, "Rancang Bangun VoIP Menggunakan <i>Software Open Source AsteriskNOW</i> "	(+)Perbandingan tiga buah <i>sofphone</i> pada <i>client</i> Laptop (+)Membahas MOS (-)Instalasi dengan <i>VirtualBox</i> pada Laptop. <i>server</i> (-)Tidak ada perbandingan waktu pengambilan	<ul style="list-style-type: none"> • Analisa kualitas layanan VoIP berdasarkan parameter QoS berdasarkan 3 skenario dan jumlah aktif call, dan waktu pengambilan • Server berbasis AsteriskNOW yang langsung di install di PC server yang didalamnya berisi Asterisk 11 dan FreePBX 6.12.65 dengan tampilan GUI server VoIP dengan fitur yang lebih lengkap • Softphone yang digunakan dalam penelitian yaitu Zoiper • Menggunakan dua AP dengan Wi-Fi 802.11n
2.	Irfan Irmansyah, "Analisis Perbandingan Performasi Protokol SIP dan H323 pada Aplikasi VoIP dalam jaringan LAN".	(+)Membahas perbandingan protokol SIP dan H323 (-)Perlu digunakan <i>server</i> lainnya dalam pengembangan Asterisk (-)Perlu digunakan <i>sofphone</i> lain selain X-Lite	
3.	Risky Agri Syafindra, "Analisa Performansi dan Kualitas Kanal VoIP Pada Sistem <i>Enmbedded Wireless Berbasis 802.11g</i> "	(+)Menggunakan <i>OpenWRT</i> untuk perangkat AP. (-)Hanya memakai satu AP (-)Perlu digunakan Wi-Fi selain 802.11g	

2.2 Voice Over Internet Protocol (VoIP)

Teknologi *Voice Over Internet Protocol (VoIP)* merupakan suatu teknik dalam dunia telekomunikasi yang dapat mengirimkan paket suara melalui jaringan *Internet Protocol (IP)* [1]. Konsep kerja VOIP adalah dengan merubah paket suara yang disalurkan ke dalam paket-paket data secara *real time* melalui jaringan IP [4]. Jaringan melalui IP sendiri adalah jaringan komunikasi data yang berbasis *packet switch*, jadi dalam bertelepon menggunakan Internet [5]. Pada saat melalui jaringan tersebut, percakapan suara berupa data analog dilewatkan pada jalur *packet switch* dan dikonversikan terlebih dahulu ke bentuk *digital* melalui proses *coder-decoder* sehingga dapat di *transport* melalui jaringan *internet*. Bentuk data yang dikirimkan akan sama ketika diterima sebagaimana

data tersebut sebelum dikirim baik berupa suara (*voice*) ataupun video [6]. Jenis layanan VoIP dapat diimplementasikan di kehidupan nyata yaitu sebagai berikut [7]:

- a. Dari Laptop ke Laptop melewati jaringan *internet*.

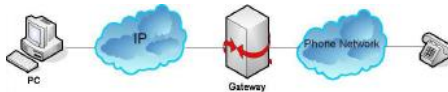
Pada hubungan ini kedua Laptop memiliki *protocol* dan *link* yang sama untuk digunakan antar *interface* pada masing-masing terminal.



Gambar 2.1 Hubungan antar PC ke PC [7]

- b. Dari Laptop ke *Phone/mobile* dan sebaliknya.

Pada Laptop ke *Phone/mobile* melewati jaringan, sehingga hubungan ini memerlukan sebuah *gateway* yang berfungsi untuk melakukan penyesuaian standar antar media termasuk penyesuaian kanal kontrol pensinyalan antar media. *Gateway* ini bisa berupa PC atau *router*.



Gambar 2.2 Hubungan antar PC ke *Phone* [7]

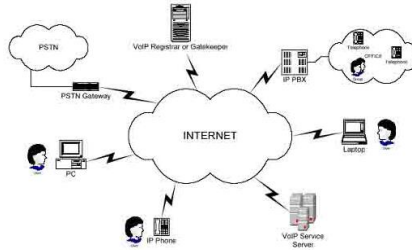
- c. Dari *Phone* ke *Phone* melewati jaringan *internet*.

Pada hubungan ini, protokol yang sama digunakan antar *interface* masing-masing terminal, namun pada *link* digunakan protokol yang berbeda, sehingga keberadaan *gateway* tetap dibutuhkan.



Gambar 2.3 Hubungan antar *phone* yang sama dengan menggunakan jaringan internet [7]

Penerapan teknologi VoIP dapat diimplementasikan pada suatu jaringan *wireless LAN* (WLAN), sehingga akan memudahkan setiap *client* dalam berkomunikasi. Hal tersebut mempunyai keuntungan dalam bidang komunikasi, yaitu biaya operasional akan lebih murah dibandingkan tarif telepon biasa. Topologi jaringan VoIP dapat dilihat pada gambar 2.4 [8].

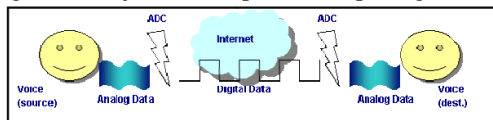


Gambar 2.4 Topologi Jaringan VoIP [8]

2.2.1 Cara Kerja VoIP [6]

Konsep cara kerja VoIP itu sendiri adalah melakukan pengiriman sebuah sinyal secara digital. Sebelum proses transmisi (pengiriman) dilakukan, data yang dikirim berupa sinyal analog akan dikonverikan terlebih dahulu dengan *Analog to Digital Converter* (ADC) menjadi paket data digital. Setelah itu akan dikonversi kembali menjadi data sinyal analog kembali dengan *Digital to Analog Converter* (DAC), sehingga data yang diterima oleh sumber tujuan sesuai dengan data yang di transmisikan.

Format digital lebih mudah dikendalikan dalam hal ini dapat dikompresi, dan dapat diubah ke format yang lebih baik. dan data digital lebih tahan terhadap noise daripada analog. Cara kerja VoIP dapat dilihat pada gambar 2.5.



Gambar 2.5 Cara Kerja VoIP [6]

Bentuk paling sederhana dalam sistem VoIP yaitu dua buah komputer atau laptop terhubung dengan *internet*. Dengan dukungan *software* khusus, kedua pemakai komputer atau laptop dapat saling terhubung dalam koneksi VoIP. Bentuk hubungan tersebut bisa dalam bentuk pertukaran *file*, suara, gambar. Penekanan utama dalam VoIP adalah hubungan keduanya dalam bentuk suara.

2.2.2 Komponen Penunjang VoIP [8]

Dalam melakukan komunikasi menggunakan VoIP dibutuhkan beberapa komponen pendukung. Beberapa komponen yang harus ada dalam VoIP, yaitu :

1. *User Agent*, merupakan komponen pada bagian *client* atau pengguna. Komponen tersebut berupa laptop ataupun *mobile*. Pada saat ingin berkomunikasi, *client* memerlukan *software* berupa telepon virtual yang

dapat di akses pada laptop ataupun pada *mobile*. *User Agent* itu sendiri terbagi menjadi dua bagian, yaitu:

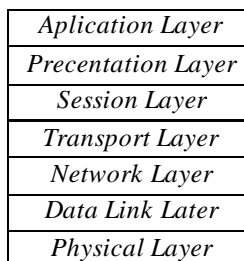
- a. *User Agent Client*, yaitu bagian dari komponen VoIP yang bertugas untuk memulai sesi inisiasi pada saat komunikasi akan berjalan.
- b. *User Agent Server*, yaitu bagian dari komponen VoIP yang bertugas untuk menerima, mengirim, dan menutup sesi komunikasi yang masuk dari *client*.

2. VoIP *Server*, yaitu *server* yang di dalamnya terdiri dari beberapa komponen. Komponen yang terdapat pada VoIP *server* adalah sebagai berikut:

- a. *Proxy Server*, sebagai media perantara *user agent client* dengan *user agent server*. *Proxy* berfungsi menyimpan sesi komunikasi.
- b. *Redirect Server*, yaitu berfungsi hanya untuk menerima *request message* dari *user agent*, memberikan alamat IP tujuan dan menyampaikan ke *user agent client*.
- c. *Registrar Server*, yaitu media penyimpanan dari data - data autentikasi *user*, data yang disimpan dalam bentuk alamat IP dan *port* dari setiap *client* yang terdaftar pada jaringan VoIP.

2.3 Konsep Dasar Model OSI Layer [9]

Model referensi *Open System Interconnection* (OSI) merupakan sebuah model yang ditujukan untuk jaringan komputer yang dikembangkan oleh *International Organization for Standardization* (ISO). Pada awalnya model OSI dimaksudkan untuk keperluan standarisasi protokol, namun protokol TCP/IP yang sekarang ini menjadi arsitektur model lapisan jaringan dari *Internet Protocol* (IP) yang akan terus dikembangkan diperluas standarnya. Model referensi OSI memiliki standar tujuh lapis, yaitu seperti gambar 2.6.



Gambar 2.6 Lapisan OSI Layer [9]

Model OSI disusun atas 7 lapisan, yaitu :

a. *Physical Layer* (Lapisan Fisik)

Physical layer atau lapisan fisik adalah lapisan yang berhubungan dengan media fisik sebagai prosedur standar dalam jaringan komunikasi data, sebagai media transmisi jaringan, sebagai metode pensinyalan, sinkronisasi bit, dan arsitektur jaringan (*Ethernet* atau *Token Ring*), topologi jaringan dan pengabelan.

b. *Data Link Layer*

Lapisan data *link* merupakan salah satu layer yang memiliki peran untuk menentukan bagaimana bit-bit data dikelompokkan menjadi format yang disebut sebagai frame. dan menentukan bagaimana perangkat jaringan seperti hub, repeater, dan switch layer 2 dijalankan. Spesifikasi IEEE 802, membagi layer ini menjadi dua, yaitu lapisan *Logical Link Control* (LLC) dan lapisan *Media Access Control* (MAC).

c. *Network Layer*

Lapisan *network* mendefinisikan alamat-alamat IP untuk membuat *header* untuk paket-paket, dan melakukan *routing* melalui *internetworking* sehingga paket dapat dikirim keluar dari *segment network* lokal ke suatu tujuan. Protokol pada lapisan *network* seperti *Internet Protocol* (IP).

d. *Transport Layer*

Lapisan *transport* digunakan untuk memecah data ke dalam paket-paket data serta memberikan nomor paket tersebut. Paket yang berhasil diterima akan diberi tanda (*acknowledgement*), sedangkan paket yang hilang atau rusak di tengah jalan akan dikirim ulang. Protokol lapisan *transport* seperti UDP, TCP.

e. *Session Layer*

Lapisan Sesi akan menyediakan fasilitas pada *client* untuk melakukan percakapan suatu komunikasi, dan bagaimana dapat terjadinya koneksi dapat dibuat serta diakhiri.

f. *Prezentation Layer*

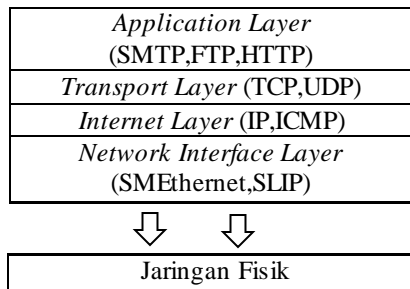
Pada lapisan presentasi terjadi perubahan data, dimana akan mentranslasikan data yang akan di transmisikan melalui jaringan. Kompresi dan enkripsi data yang terjadi seperti contohnya layanan presentasi adalah *encoding* data, dimana seperti nama atau tanggal akan dinyatakan dalam bentuk karakter atau struktur data.

g. *Application Layer*

Layer aplikasi sesungguhnya *client* “berinteraksi dengan jaringan, dimana terjadi antarmuka (penghubung) dengan aplikasi fungsionalitas jaringan berupa program atau aplikasi tingkatan layanan informasi. *Layer* aplikasi juga mengatur bagaimana aplikasi dapat mengakses jaringan. Protokol pada lapisan aplikasi seperti FTP, HTTP.

2.4 Konsep Dasar Protokol TCP/IP [9]

Model jaringan TCP/IP (*Transmission Control Protocol/Internet Protocol*) merupakan model jaringan komputer dan rangkaian protokol komunikasi yang digunakan di *internet*. Berbeda seperti model OSI yang memiliki tujuh lapis, model TCP/IP memiliki standar empat lapis dapat dilihat pada gambar 2.7.



Gambar 2.7 Pemodelan *Layer* TCP/IP [7]

Model TCP/IP terdiri atas protokol yang masing-masing bertanggung jawab atas bagian tertentu dari komunikasi data.

1. *Application Layer*

Proses teratas pada lapisan aplikasi terjadi proses antarmuka dengan layanan *transport* untuk menggunakan jaringan.

2. *Transport Layer*

Pada lapisan *transport* data yang masuk akan terlihat status koneksi, sehingga lapisan ini akan mengizinkan tujuan untuk berkomunikasi.

3. *Network Layer*

Lapisan akses *network* merupakan lapisan pertama yang mengetahui data secara fisik yang dikirim melalui jaringan.

4. *Physical Layer*

Pada layer fisik yaitu berhubungan langsung dengan media fisik, dan mendefinisikan protokol TCP/IP mendukung semua standar protokol.

2.5 TCP dan UDP [9]

Transmission Control Protocol (TCP) dan *User Datagram Protocol* (UDP) merupakan protokol terpenting dalam *layer Transport*, dan dua protokol ini dapat digunakan pada aplikasi TCP/IP.

2.5.1 *Transmission Control Protocol* (TCP) [9]

Protokol TCP berada di *transport layer* dan bertanggung jawab untuk pengiriman data secara berurutan dari sumber ke tujuan dengan benar. Karakteristik protokol TCP yaitu *connection oriented* yang berarti pada saat proses transmisi data terjadi, dua aplikasi TCP harus melakukan pertukaran kontrol informasi (*handshaking*), dan bersifat *reliable* karena menerapkan fitur deteksi kesalahan dan retransmisi apabila ada data yang rusak, sedangkan *byte stream service* artinya paket akan dikirimkan ke tujuan secara berurutan (*sequencing*). TCP digunakan untuk koneksi yang membutuhkan kehandalan yang tinggi, seperti aplikasi FTP, HTTP, dan beberapa layanan lainnya.

Bit offset	0-3	4-7	8-15								16-31	
0	Source Port (16)								Destination Port (16)			
32	Sequence Number (32)											
64	Acknowledgement Number (32)											
96	Data offset	Reserved	C W R	E C E	U R E	A C K	P S H	R S T	S S T	F I N	F I N	Window Size
128	Checksum (16)										Urgent (16)	
160	Options (0 or 32 if any)											
160/192++	Data (varians)											

Gambar 2. 8 *Header* Protokol TCP [9]

2.5.2 *User Data Protocol* (UDP) [9]

Protokol UDP menyediakan layanan pengiriman atau menyediakan fasilitas *multiplexing* aplikasi (via nomor *port*) dan verifikasi atau mendeteksi kesalahan (via *checksum*) yang disediakan pada *header* dan *payload*. Pada saat melakukan pengiriman data tidak dilakukan proses *handshaking*, tidak ada *sequencing* datagram, dan tidak memberikan jaminan penyampaian pesan tertentu atau datagram ke soket (*end node*) pada sistem penerima, dimana jumlah pesan yang dikirim lebih sedikit dibanding TCP. Overhead yang dibutuhkan untuk mengirimkan pesan dalam bentuk datagram atau paket UDP sangat kecil. Sehingga UDP digunakan pada aplikasi yang membutuhkan *query* dan *response* cepat yaitu pada transmisi audio/video, seperti VoIP, audio/video streaming. UDP kurang baik digunakan untuk mengirim paket dengan ukuran yang besar, karena akan memperbesar jumlah paket yang *loss* atau hilang.

Bit	0-15	15-31
0	Source Port (16)	Destination Port (16)
32	Header Length (16)	Checksum (16)
64	Data (varies)	

Gambar 2.9 Header Protokol UDP [9]

2.6 Internet Protokol (IP) [2]

Konsep jaringan komputer adalah dimana komputer satu dan komputer lainnya saling terhubung dengan mekanisme pengalamatan, sehingga antar komputer dapat saling bertukar data maka dibutuhkan peras dari protokol. *Internet protocol* berada di layer *network*, dan merupakan protokol yang mengatur bagaimana suatu data dapat dikenal dan dikirim dari satu komputer yang satu dan lainnya, dan mendefinisikan skema pengalamatan *IP address*. *IP address* merupakan pemberian identitas setiap *host* bagi setiap *interface* komputer pada jaringan *internet*, dan berfungsi sebagai petunjuk alamat *interface* untuk menentukan suatu *route* jaringan yang di lalui oleh sebuah pengiriman data.

Pada jaringan yang menggunakan protokol TCP/IP setiap host akan memiliki alamat IP versi 4 yang memiliki nilai berukuran 32 bit, yang dibagi atas 4 segmen dan setiap segmen terdiri atas 8 bit. Nilai ini digunakan untuk mengenali nomor unik *host* yang bersangkutan di jaringan tertentu. Konsep pengalamatan berguna untuk memudahkan pengaturan *IP address* seluruh komputer pengguna jaringan *Internet*. Pembagian kelas-kelas *IP address* didasarkan pada dua komponen, yaitu *Network ID* sebagai bagian *IP address* untuk menunjukkan jaringan tempat komputer berada, dan *Host ID* yang digunakan untuk menunjukkan *workstation*, *server router*, serta semua *host* jaringan tersebut. Perlu diperhatikan, *Host ID* tidak boleh sama dalam satu jaringan. Pengalamatan IP dapat dilihat dari *octet* pertama, yaitu sebagai berikut:

	← 8 bit →	← 8 bit →	← 8 bit →	← 8 bit →
Kelas A	0 Prefik		Suffix	
Kelas B	10 Prefix		Suffix	
Kelas C	110 Prefix		Suffix	
Kelas D	1110 Alamat Multicast			
Kelas E	1111 Alamat Cadangan			

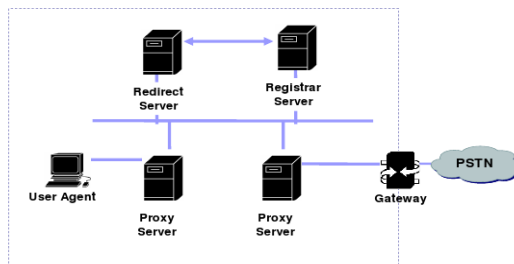
Gambar 2.10 Alokasi alamat IP dengan skema *classfull* [2]

Tabel 2.2 Kelas IP Address [2]

No	Kelas IP	Range IP Address	Jumlah Host	Jumlah Network
1.	Kelas A	0.0.0.0-127.225.225.225	16.777.216	127
2.	Kelas B	128.0.0.0-191.225.225.225	1.048.576	16.384
3.	Kelas C	192.0.0.0-233.255.255.255	65.356	2.097.152
4.	Kelas D	224.0.0.0-239.255.255.255	Tidak Didefinisikan	Tidak Didefinisikan
5.	Kelas E	240.0.0.0-255.255.255.255	Tidak Didefinisikan	Tidak Didefinisikan

2.7 Protokol Control/Signaling VoIP

Protokol *control/signaling* pada VoIP yaitu protokol *Session Initiation Protocol* (SIP). Protokol SIP merupakan *control/signalling* protocol yang membuat *user/client* VoIP dapat saling berkomunikasi. SIP memanfaatkan *Real Time Protocol* (RTP) untuk media transfer. Komponen perancangan VoIP sehingga berbeda dengan komponen *server* pada VoIP. [8]



Gambar 2.11 Arsitektur SIP [6]

Komponen : [7]

1. *User Agent*

- a. *User agent client* (UAC) : Komponen yang memulai sesi komunikasi.
- b. *User agent server* (UAS) : Komponen yang menerima dan menanggapi sesi komunikasi.

Format *messages* pada SIP didefinisikan menjadi dua:

a. *SIP Request*

Request message pesan pada SIP yang dikirim dari *user agent* ke *user agent server*. Fungsi pesan-pesan tersebut dapat dilihat pada tabel berikut:

Tabel 2.3 Pesan *Request Message* [7]

Pesan	Keterangan
INVITE	Menginisialisasi sesi komunikasi, dimana mengajak <i>user agent</i> lain untuk bergabung dalam sesi komunikasi.
ACK	Mengkonfirmasi bahwa <i>user agent</i> telah menerima pesan terakhir dari serangkaian pada pesan <i>INVITE</i> .
BYE	Perintah yang digunakan untuk mengakhiri sesi oleh <i>user agent client</i>
CANCEL	Perintah untuk membatalkan <i>INVITE</i> .
REGISTER	Permintaan untuk daftar di <i>registrar server</i> .
OPTIONS	Digunakan untuk meminta <i>server</i> memberikan informasi alamat dengan <i>server SIP</i> .
INFO	Membawa informasi ketika suatu panggilan di proses, berisi pesan informasi.

b. SIP Responen

Tabel 2.4 Pesan *Response Message* [7]

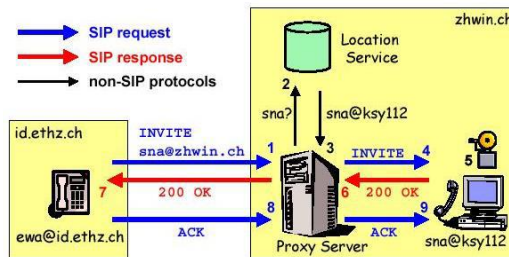
No. Serial	Status Code	Keterangan
1xx	<i>Informasional message Request</i>	berarti telah diterima dan sedang melanjutkan proses. Contoh: 100(<i>trying</i>), 180 (<i>ringing</i>), 183 (<i>progress</i>)
2xx	<i>successful response</i>	sukses diterima dipahami dan disetujui. Contoh : 200 (OK).
3xx	<i>Redirectional response</i>	<i>Redirectional response</i> , yaitu memproses permintaan. Contoh: 302(<i>temporarily moved</i>), 305(<i>use proxy</i>).
4xx	<i>Request failure response, Client error</i>	Berisi <i>syntax</i> yang salah sehingga tidak bisa diproses. Contoh : 403 (<i>forbidden</i>).
5xx	<i>Server failure response, server error</i>	Berarti <i>server</i> gagal untuk memproses dan mengenali permintaan. Contoh : 500 (<i>server internal error</i>), dan 501 (<i>not implemented</i>).
6xx	<i>Global failure response,</i>	Berarti permintaan tidak dapat dipenuhi oleh server manapun. Contoh : 606 (<i>not acceptable</i>).

Pada sistem SIP pada saat ingin berkounikasi, maka *client* harus terlebih dulu melakukan registrasi ke *server* agar lokasinya dapat diketahui. *Network server* terdiri [7]:

1. *Proxy Server*

- a. Komponen penengah antar *user agent*, sebagai *server* dan *user* yang menerima *request* dan menyampaikan pada *user agent* lainnya, serta menyimpan sesi komunikasi antara UAC dan UAS.
- b. *Request* dilayani sendiri atau disampaikan (*forward*) pada *proxy* lain atau *server* lain.

2. *Redirect Server*
 - a. Menyediakan informasi mengenai tujuan berikutnya dari users
 - b. *Redirect server* tidak menyimpan *state* sesi komunikasi antara UAC dan UAS setelah pemetaan disampaikan pada UAC.
 - c. Tidak seperti *proxy server*, *redirect server* tidak dapat memulai inisiasi *request message*.
3. *Registrar Server*
 - a. Komponen yang menerima *request message REGISTER*.
 - b. *Registrar* dapat menambahkan fungsi otentikasi *user* untuk validasi dan menyimpan *database* user untuk otentikasi dan lokasi sebenarnya (berupa IP dan *port*).

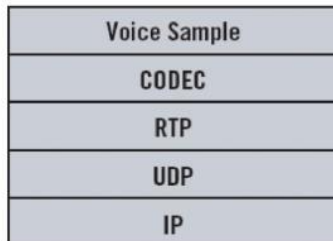


Gambar 2.12 Komunikasi Dalam SIP [7]

2.8 Protocol Data VoIP [10]

a. Real Time Protocol (RTP)

RTP merupakan protokol yang digunakan *client* untuk *voice*. Tiap paket RTP berisi potongan paket dari *voice conversation*. Besarnya ukuran paket *voice* bergantung pada CODEC yang digunakan. Informasi RTP dienkapsulasi dalam paket UDP. Pada susunan protokol RTP terlihat pada gambar berikut :



Gambar 2.13 Susunan Protokol RTP [10]

b. Compressed RTP

Compressed Real Time Protocol (cRTP) adalah varian dari RTP. Sistem dengan cRTP juga dapat melakukan panggilan dua kali lebih banyak dibandingkan standar RTP.

c. Real Time Control Protocol (RTCP)

Real Time Control Protocol (RTCP) merupakan protocol data VoIP yang jarang digunakan, karena membutuhkan *extra bandwidth* untuk membawa RTCP stream ke tujuan.

2.9 Codec G.711

Percakapan yang terjadi pada VoIP perlu adanya *Codec* (*Coder Decoder*) untuk mengubah sinyal atau mengkonversi data dalam bentuk format *digital* dan dikonversi kembali menjadi bentuk yang dikirim [1]. Proses perancangan sistem ini menggunakan *codec* G.711. Codec ini menggunakan teknik *Pulse Code Modulation* (PCM) dalam pengiriman suara. Proses di dalam PCM yaitu terdapat proses *sampling*, *quantizing*, *coding*. Pada sistem teleponi, proses *sampling* terhadap data analog yang di dalamnya dapat mengkonversi sinyal *analog* ke bentuk *digital* dengan melakukan *sampling* sinyal *analog* 8000 kali/detik dalam bentuk angka, berfrekuensi 300Hz - 3400Hz. Pada pros. Sinyal tersebut diubah dalam bentuk diskrit. Format PCM 8bit dengan menghasilkan 8000 sampel/detik dengan 8bit/sampel, sehingga menghasilkan 64.000bit/detik. Bit rate 64kbps merupakan standar transmisi untuk satu kanal telepon digital [11].

Standar G.711 mempunyai algoritma kompresi, yaitu algoritma A-Law dan U-Law. Kedua algoritma ini menghasilkan dua *codec* yang berbeda, yaitu G.711a atau PCMA (A-Law) dan G.711u atau PCMU (U-Law). Algoritma pengodean yang direkomendasikan yaitu dapat dilihat di tabel 2.4 [12].

Tabel. 2.5 Perbandingan Jenis *Codec* [10]

<i>Codec</i>	Bit rate (Kbps)	NEB (Kbps)
G.711	64	87.2
G.729	8	31.2
G.723.1	6.3	21.9
G.726	32	55.2
G.728	16	31.5

2.10 Software Penunjang VoIP

2.10.1 Software Open Source AsteriskNOW

AsteriskNOW merupakan *server* aplikasi yang berbasis *open source*, dimana distribusi *Linux* disesuaikan dengan mencakup Asterisk dan *FreePBX*, dan berbagai aplikasi yang mendukung sistem Asterisk. AsteriskNOW menyediakan layanan yang dapat memudahkan *client* untuk penginstalan. Komponen AsteriskNOW yaitu Asterisk, *drivers* DAHDI, dan *Libpri*, *FreePBX*. AsteriskNOW didasarkan pada CentOS *Linux* AsteriskNOW yang menangani sebagian besar dari kompleksitas Asterisk dengan *Linux*, manajemen pemaketan *yum* dan mudah untuk melakukan konfigurasi Asterisk karena telah berbasis *web interface* yang memiliki GUI [3].

Asterisk adalah implementasi perangkat lunak dari “*telephone private branch exchange* (PBX)”. Seperti PBX lainnya, dimungkinkan memasang pesawat telepon dan melakukan panggilan ke satu dengan lainnya, termasuk ke layanan telepon pribadi dan publik, termasuk layanan jaringan telepon umum (PSTN) dan *Voice over Internet Protocol* (VoIP). Nama Asterisk berasal dari *(tanda bintang). Asterisk dirilis menganut model lisensi ganda, menggunakan GNU/GPL sebagai lisensi perangkat lunak bebas [13]. Pada penggunaan AsteriskNOW, pengembang pada aplikasi dan integrator dalam sistem menyiapkan “*dial plans*” dan menghubungkan *hardware* yang diperlukan, tanpa perlu mendalami mengenai konfigurasi *server* dan instalasi dasar Asterisk.

Pada versi baru AsteriskNOW ini, datang dengan sejumlah perbaikan dan peningkatan termasuk peningkatan pada Asterisk ke versi 11 menggantikan v1.8. Selanjutnya basis distro ditingkatkan ke CentOS 6.4 dari sebelumnya menggunakan CentOS 5.x. Pada versi baru pada paket *FreePBX* digunakan 2.11 beta menggantikan 2.10, pada modul terbaru dituliskan “*Digium-phones addon*” dengan menambahkan banyak opsi, dan diperbaiki sejumlah *issues*, termasuk masalah perekaman “*call detail records*” yang telah ditemukan di versi 2.x [14].



Gambar 2.14 Tampilan AsteriskNOW [14]

Fitur yang paling penting dari AsteriskNOW adalah *setup wizard*, yang hanya membawa pengguna melalui pengaturan waktu dan tanggal, terhubung ke

penyedia layanan dan dari mereka pergi. Pada implementasinya, solusi telepon berbasis Asterisk menawarkan serangkaian fitur yang kaya dan fleksibel, fungsi PBX klasik dan interoperasi dengan sistem telepon berbasis standar dan VoIP sistem. Fitur selanjutnya pada *AsteriskNow* [15]:

- a. *Web* berbasis antarmuka yang dapat mengkonfigurasi dan membuat manajemen *point-and-klik*.
- b. Dapat membangun *data-driven* aplikasi dengan dukungan terintegrasi untuk ODBC dan HTTPS
- c. *Wizard-driven* yaitu konfigurasi koneksi VoIP memudahkan untuk terhubung.
- d. Dapat menginstall *pre-built*, yaitu aplikasi dikemas menggunakan *Application Manager*
- e. Dapat mendeteksi dan konfigurasi *hardware Digium analog* dan telepon *digital*.
- f. *Dialplan* dan AEL skrip editor dengan sintaks dan validasi menyederhanakan pengembangan.
- g. *Log* dan *Call Record Detail* (CDR) penampil yang memberikan akses cepat ke aktivitas sistem.
- h. Adanya bantuan untuk aplikasi, fungsi, CLI, AGI dan perintah AMI.
- i. Pengembang aplikasi yang dapat dibuat dengan AsteriskNOW seperti membuat *VoIP Gateway*, *IP PBX*, *Call Center ACD*, *Conference Bridge*, *IVR Server*, *Voicemail Sistem*, *Fax Server*, dan *pidato server*.

2.10.2 FreePBX [16]

Aplikasi *FreePBX* merupakan sebuah program aplikasi berbasis *Graphical User Interface* (GUI) yang dapat bekerja secara bersama-sama dengan Asterisk maupun program lainnya untuk memudahkan *admin* dalam melakukan konfigurasi *server* VoIP secara *remote* melalui *web server* dengan memasukkan alamat IP *address server* pada kolom *address bar*.

Aplikasi *FreePBX* mempunyai tampilan sangat grafis, sehingga lebih mudah dalam melakukan konfigurasi untuk *server* VoIP pada saat dijalankan. Tampilan berbasis *Graphic User Interface* (GUI) akan memudahkan *admin* dalam melakukan konfigurasi seperti *add extention* maupun dalam melakukan *control* secara menyeluruh terhadap jaringan VoIP secara *remote* melalui *web browser* dengan memasukkan alamat IP *address server*.

2.11 Jaringan Komputer [2]

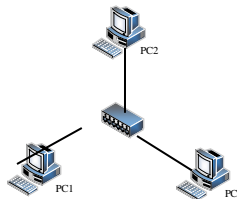
Jaringan merupakan konsep dalam hubungan atau interkoneksi antar perangkat yang satu sama lain harus saling terhubung. Perangkat dalam konsep jaringan secara umum terdapat dua jenis perangkat, yaitu perangkat akhir (*end device*) yang digunakan oleh pengguna jaringan untuk menerima atau mengirimkan data, sedangkan perangkat yang kedua yaitu perangkat tengah (*intermediary device*) hanya digunakan untuk menghubungkan antar perangkat dalam sebuah jaringan.

2.11.1 Topologi Fisik

Topologi fisik berkaitan dengan pola atau bentuk jaringan jika dilihat secara fisik. Ada beberapa topologi jaringan yang digunakan, yaitu topologi *star*, *mesh*, *partial - mesh*.

a. Topologi *Star*

Topologi *star* merupakan topologi yang setiap perangkatnya dihubungkan pada satu perangkat penghubung (sentral) ke perangkat lainnya. Penggunaan topologi *star* cukup sederhana, yaitu menghubungkan ketiga komputer pada perangkat penghubung sentral. Pada sisi penggunaan kabel tidak banyak, sehingga biaya pembangunan jaringan relatif lebih murah. Sisi kelemahan topologi *star* yaitu pada perangkat sentral, karena jika perangkat penghubung mati (*down*) dapat mengakibatkan seluruh sistem akan mati, dan komputer tidak dapat bertukar data.

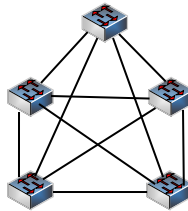


Gambar 2.15 Topologi *Star* dengan Hub sebagai penghubung (sentral) [2]

b. Topologi *Mesh*

Pada setiap perangkat setiap topologi *mesh* mempunyai hubungan secara dedicated (*point-to-point*) dengan perangkat lainnya. *Dedicated* disini berarti trafik data yang mengalir pada sebuah jalur hanya berasal dari dua perangkat yang dihubungkan saja. Jika

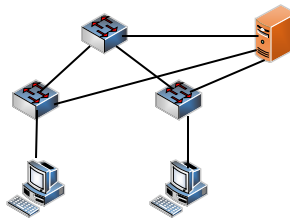
dibandingkan antara penggunaan topologi *star* an *mesh*. Jumlah jalur (*link*) yang digunakan pada topologi *mesh* lebih banyak, sehingga biaya pembangunan jaringan juga besar. Keuntungan menggunakan topologi *mesh* yaitu *bandwidth* saluran tidak terbagi karena sifatnya *dedicated*, kemudian mempunyai jaringan yang kuat, dan dipastikan hanya penerima yang menerima data karena hanya dua perangkat yang terhubung.



Gambar 2.16 Topologi *Mesh* menggunakan lima perangkat penghubung (*switch*) [2]

c. Topologi *Partial - Mesh*

Topologi *partial - mesh* atau dapat disebut juga dengan topologi *mesh* tidak penuh merupakan topologi yang pada mulanya dilihat dari beberapa perangkat yang tidak perlu diberikan jalur cadangan (*backup*), namun hanya jalur yang menuju ke arah perangkat utama saja yang diberikan jalur cadangan.



Gambar 2.17 Topologi *Partial - Mesh* [2]

2.11.2 Tipe Jaringan [2]

Pada penamaan tipe dalam sebuah jaringan dapat dilihat dari letak atau penempatannya. Berdasarkan penempatan jaringan, ada tiga macam jaringan, yaitu *Local Area Network* (LAN), *Wide Area Network* (WAN), dan *Internet*.

a. Local Area Network (LAN)

Local Area Network (LAN) merupakan jaringan yang terdapat dalam satu gedung, kantor, atau kampus. Untuk LAN tidak terpaku pada jumlah perangkat yang terhubung, namun pada penempatan jaringan pada tempat yang sama.

b. Wide Area Network (WAN)

Jaringan WAN merupakan hubungan antar jaringan LAN, namun mempunyai letak wilayah geografis yang berbeda. WAN akan menghubungkan jaringan LAN yang terletak pada jarak yang relatif jauh, seperti antar provinsi.

c. Internet

Jaringan *Internet* lebih dikenal dengan jaringan umum (*public*). *Internet* mempunyai konsep menghubungkan beberapa wilayah jaringan yang berbeda. Pada *Interconnection – networking* (*Internet*) menggunakan standar TCP/IP sebagai protokol pertukaran paket untuk melayani pengguna seluruh dunia.

2.12 Jaringan Wireless Local Area Network (WLAN)

Pada teknik jaringan berbasis *wireless* LAN (WLAN) yaitu dimana memanfaatkan jaringan lokal yang memanfaatkan gelombang radio sebagai media transmisi. Untuk dapat terhubung ke dalam jaringan menggunakan perangkat yang dihubungkan dengan *switch* atau *router*. Jaringan *wireless* LAN membutuhkan perangkat *Access Point* (AP) sebagai perangkat penghubung [1].

Komponen dalam membangun WLAN yaitu dibutuhkan beberapa komponen, yaitu sebagai berikut [17]:

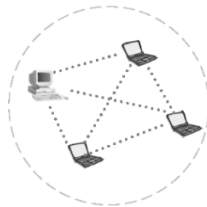
1. *Access Point* (AP), dimana AP merupakan alat untuk mentransmisikan data dan biasanya terhubung pada jaringan LAN melalui kabel. Fungsi AP adalah mengirim, dan menerima data, sebagai *buffer* data antara WLAN dengan *Wired* LAN, dan AP sudah mencapai spesifikasi 802.11n yang mampu mentransmisikan data dengan lebar *bandwidth* mencapai 300 Mbps.
2. *Dekstop PC/Mobile*, yaitu perangkat *end-user* sebagai media akses oleh pengguna.
3. *Extention Point*, seperti *repeater* untuk *user* di tempat yang jauh.
4. Antena, dimana untuk mentranformasikan sinyal radio menjadi gelombang elektromagnetik untuk memperkuat daya pancar. Antena yang digunakan seperti *omni* maupun *directional*.

2.12.1 Arsitektur Jaringan *Wireless LAN*

Pada implementasi jaringan yang digunakan pada skripsi ini yaitu menggunakan arsitektur jaringan pada WLAN dengan konsep jaringan *Extended Service Set (ESS)*.

a. *Independent Basic Service Set (IBSS)*

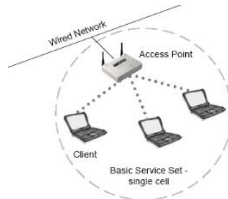
Pada implementasi jaringan ini merupakan konfigurasi jaringan yang setara dengan ‘*peer-to-peer*’ atau mode ‘*ad-hoc*’ (khusus). Implementasi mencakup wilayah terbatas dan umumnya tidak dihubungkan ke jaringan yang lebih besar [18]. Arsitektur jaringan IBSS dapat dilihat pada gambar 2.19.



Gambar 2.18 Topologi Jaringan IBSS [19]

b. *Basic Service Sets (BSS)*

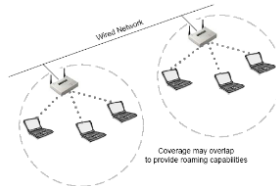
Pada implementasi jaringan ini merupakan konsep jaringan *wireless LAN* yang hanya membutuhkan satu buah *Access Point (AP)* dalam menghubungkan antar user. Suatu BSS mempunyai satu *Service Set Identifier (SSID)* yang berbeda [19].



Gambar 2.19 Topologi jaringan BSS [19]

c. *Extended Service Set (ESS)*

Pada implementasi jaringan ini merupakan topologi *Extended Service Set (ESS)*, dimana pada topologi ini terdiri dari beberapa BSS dalam suatu jaringan. Pada ESS penggunaan *Access Point (AP)* sebagai penghubung antar *user* lebih dari satu [18].

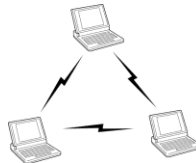


Gambar 2.21 Topologi jaringan ESS [19]

2.12.2 Mode Jaringan Wireless LAN

a. Ad-Hoc

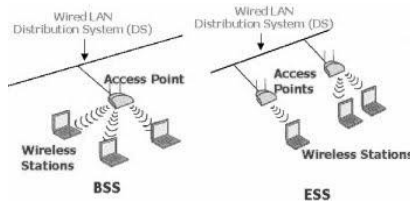
Konsep jaringan *wireless LAN* yang sederhana, karena tidak memerlukan *Access Point (AP)* untuk berkomunikasi secara langsung satu sama lain. Jaringan *ad-hoc* menggunakan konsep topologi jaringan *mesh*.



Gambar 2.22 Mode jaringan Ad-Hoc [20]

b. Mode Infrastruktur

Pada mode infrastruktur *access point* berfungsi melayani komunikasi utama pada jaringan wireless. *Access Point* mengirimkan data pada PC/Laptop pada jangkauan tertentu pada suatu area. Penambahan dan pengaturan letak AP dapat memperluas jangkauan dari jaringan WLAN [20].



Gambar 2.23 Mode Jaringan Infrastruktur [20]

2.12.3 Teknologi Jaringan *Wireless Fidelity* (Wi-Fi)

Wireless Fidelity atau lebih dikenal dengan sebutan Wi-Fi merupakan sebuah konsep pada pembangunan jaringan *Local Area Network* (LAN) yang menggunakan media *wireless* sebagai transmisinya [21]. Standar yang digunakan pada jaringan Wi-Fi yaitu standar yang dikeluarkan oleh IEEE yaitu 802.11 adalah jenis spesifikasi dari Wi-Fi tersebut. Teknologi Wi-Fi saat ini mampu menyediakan akses dengan *bandwidth* mencapai 300 Mbps untuk standar pada spesifikasi 802.11n. Beberapa hal yang harus dipenuhi supaya komunikasi dapat berjalan yaitu adanya penggunaan parameter yang sama antara perangkat yang akan dihubungkan [22].

Pada jaringan *wireless*, frekuensi menjadi bagian yang penting. Rentang frekuensi yang digunakan perangkat *wireless* LAN terletak pada frekuensi 2.4GHz-2.5GHz untuk implementasi jaringan *indoor* (dalam ruangan) dan 5.7GHz-5.8 GHz untuk implementasi *outdoor* (luar ruangan) [2]. Berikut adalah spesifikasi dari standar 802.11. Ada beberapa jenis spesifikasi dari 802.11 berdasarkan tingkat kecepatan yaitu 802.11a, 802.11b, 802.11g, dan 802.11n. Berikut adalah spesifikasi 802.11 dapat dilihat pada tabel 2.5 [23].

Tabel 2.6 Spesifikasi 802.11 [24]

802.11 Network Standards							
802.11 Protocol	Frequency (GHz)	Data Rate (Mbps)	Approximate Indoor Range		Approximate Outdoor Range		Ket
		Max	Meter	Feet	Meter	Feet	
802.11a	5Ghz	54 Mbps	35	115	120	390	a
802.11b	2,4Ghz	11 Mbps	38	125	140	460	b
802.11g	2,4Ghz	54 Mbps	38	125	140	460	b,g
802.11n	2,4Ghz	300 Mbps	70	230	250	820	b,g,n

2.12.3.1 IEEE 802.11n [23]

IEEE 802.11n merupakan standar *wireless* untuk meningkatkan *throughput* dari standar sebelumnya yaitu 802.11b dan 802.11g. Beberapa fitur dari IEEE 802.1n yaitu adanya MIMO (*Multiple-Input Multiple-Output*), dimana menggunakan beberapa antena untuk mentransmisikan informasi. Fitur selanjutnya pada IEEE 802.11 yaitu adanya saluran 40 MHz dari yang sebelumnya yaitu 20MHz, maka data rate 802.11 akan meningkat dua kali lipat dari 150 Mbps

menjadi 300Mbps. Selain peningkatan *bandwidth*, terdapat juga peningkatan performa pada jarak dan reliabilitas.

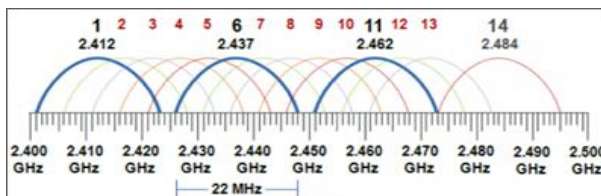
2.12.3.2 Channel Frequency Wi-Fi 2,4 GHz [25]

Pada konsep jaringan *wireless* sama dengan stasiun radio, yakni menggunakan dua alokasi frekuensi yaitu 2,4 GHz dan 5 GHz. Frekuensi 2,4 GHz yang digunakan oleh 802.11b/g/n dibagi menjadi *channel - channel* seperti pembagian frekuensi untuk stasiun radio.

Organisasi *International Telecommunication Union* (ITU) membagi frekuensi 2,4GHz menjadi 14 *channel*, tetapi setiap Negara mempunyai kebijakan tertentu terhadap *channel* tersebut. Pada kebijakan Negara Indonesia belum ada kebijakan, tetapi *Access Point* di Indonesia jumlah *channel* yang dimiliki kebanyakan yaitu 11-13. Frekuensi yang digunakan pada setiap *channel* dapat dilihat pada tabel 2.7.

Tabel 2.7 Channel Frequency [25]

Channel	Frequency (GHz)
1	2.412
2	2.327
3	2.422
4	2.427
5	2.432
6	2.437
7	2.447
8	2.452
9	2.457
10	2.462
11	2.462
12	2.467
13	2.472
14	2.484



Gambar 2.24 Alokasi Frekuensi 14 Channel [26]

Pada komunikasi *wireless*, penggunaan *channel* 1 dan *channel* 2 secara bersamaan akan menimbulkan interferensi yang akan menimbulkan rusaknya data-data yang dikirim. Untuk menanggulangi hal tersebut, maka diperlukan strategi penggunaan *channel* yang baik [27].

2.13 Parameter Pengukuran Quality of Service (QoS)

Pada saat komunikasi berlangsung, *client* membutuhkan adanya layanan yang lebih baik pada trafik tertentu. Hal tersebut menjadikan *Quality of Service* (QoS) menjamin data yang penting yang bersifat *real time* seperti suara akan mendapatkan prioritas utama untuk diteruskan ke perangkat tujuan. QoS tidak diperoleh langsung dari infrastruktur yang ada, melainkan diperoleh langsung dengan mengimplementasikannya pada jaringan bersangkutan [1]. Kualitas paket data pada jaringan VoIP ditunjukkan dengan parameter-parameter *Quality of Service* (QoS) berikut ini :

a. Standar Perhitungan *Throughput*

Throughput dalam jaringan telekomunikasi merupakan rata-rata pengiriman sukses dalam satu pengiriman (satuan bps). Sedangkan sistem *Throughput* atau jumlah *Throughput* merupakan jumlah rata-rata paket data yang sukses dikirimkan oleh semua terminal pada sebuah jaringan. *Throughput* diukur digunakan untuk mentransfer file dari ukuran tertentu [28]. Berdasarkan standarisasi TIPHON TR 101 329 untuk menghitung *throughput* dapat dihitung dengan menggunakan persamaan (2-1):

$$Throughput = \frac{\text{jumlah bit data diterima benar}}{\text{waktu pengiriman bit}} \text{ Bps} \dots \dots (2-1) \quad [29]$$

b. *Delay (Latency)*

Secara teknis, *delay* merupakan banyaknya waktu yang diperlukan sebuah paket untuk melakukan perjalanan dari sumber ke tujuan. *Delay* dapat dipengaruhi oleh jarak, media fisik, dan proses waktu yang lama [28]. Berdasarkan standarisasi TIPHON TR 101 329 parameter *delay* untuk layanan *audio* dapat dilihat pada tabel 2.7.

Tabel 2.8 Klasifikasi Standar *Delay* [30]

Kategori Latensi	Besar <i>Delay</i> (ms)	Indeks
Sangat Bagus	<150 ms	4
Bagus	150 s/d 300ms	3
Sedang	300 s/d 450 ms	2
Jelek	>450 ms	1

Untuk menghitung *delay* dapat dihitung dengan menggunakan persamaan (2-2):

$$Delay(n) = T_{out}(n) = \text{serving time} + T_{in}(n) \dots \dots \dots (2-2) \quad [29]$$

Dimana :

$T_{out}(n)$ = waktu data ke-n keluar antrian dan siap ditransmisi

$T_{in}(n)$ = waktu data ke-n keluar antrian

c. Standar Perhitungan Jitter

Besarnya nilai *jitter* akan sangat dipengaruhi oleh variasi beban trafik dan besarnya tumbukan antar paket (*congestion*) yang ada dalam jaringan IP. Semakin besar beban trafik di dalam jaringan akan menyebabkan semakin besar pula peluang terjadinya *congestion* dengan demikian nilai *jitter*-nya akan semakin besar [28]. Menurut TIPHON TR 101 329, *jitter* dapat diketahui standar yang diberikan seperti pada tabel 2.9.

Tabel 2.9 Klasifikasi Standar *Jitte* [30]

Kategori Degradasi	Peak Jitter	Indeks
Sangat Bagus	0 ms	4
Bagus	0 s/d 75ms	3
Sedang	75 s/d 125 ms	2
Jelek	125 s/d 225 ms	1

Untuk menghitung *jitter* dapat dihitung dengan menggunakan persamaan seperti berikut :

$$Jitter = \frac{\text{Total variasi Delay}}{\text{Total Paket yang diterima}} \dots\dots\dots(2-3) \quad [29]$$

Total variasi *delay* diperoleh dari Persamaan (2-2):

variasi *Delay*= delay – (rata-rata delay)2-4)

d. Standar Perhitungan Packet loss

Packet loss merupakan banyaknya paket yang gagal mencapai tempat tujuan saat pengiriman paket. Jika paket yang dikirim gagal maka paket tersebut tidak akan dikirim kembali, atau dengan kata lain paket tersebut hilang akan menjadi masalah jika *packet loss* [28]. Menurut TIPHON TR 101 329, *packet loss* dapat diketahui standar yang diberikan seperti pada tabel 2.10.

Tabel 2.10 Klasifikasi Standar *Packet loss* [30]

Kategori Degredasi	Packet loss	Indeks
Sangat Bagus	0%	4
Bagus	3%	3
Sedang	15%	2
Jelek	25%	1

Untuk menghitung *packet loss* dapat dihitung dengan menggunakan persamaan (2-4):

$$Packetloss = \frac{\text{jml bit dikirim} - \text{jml bit diterima}}{\text{jml bit yang dikirim}} \times 100\% \dots\dots\dots(2-5) \quad [29]$$