

BAB II

TINJAUAN PUSTAKA

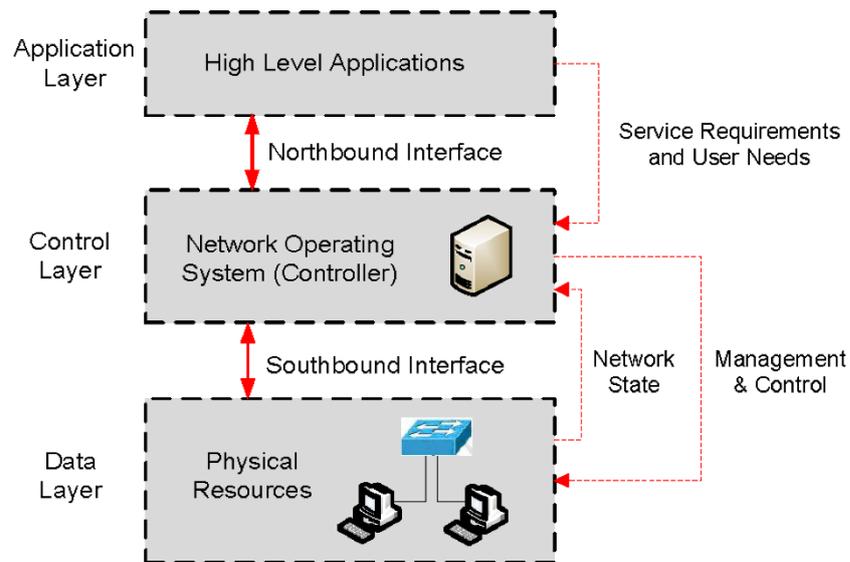
2.1 Software-Defined Networking

Software Defined Network (SDN) adalah sebuah konsep yang memungkinkan/memperbolehkan operator jaringan untuk mengelola *router* dan *switch* secara fleksibel menggunakan *software* yang berjalan di *server* eksternal [1]. Dalam pengertian yang lainnya juga menyebutkan bahwa *Software Defined Network* (SDN) adalah sebuah arsitektur jaringan yang baru dimana kontrol jaringan dipisahkan dari *forwarding* dan diprogram secara langsung [2].

Beberapa aspek penting dari *Software Defined Network* adalah:

1. Adanya pemisahan secara fisik/eksplisit antara *forwarding/data-plane* dan *control-plane*.
2. Antarmuka standard (*vendor-agnostic*) untuk memprogram perangkat jaringan.
3. *Control-plane* yang terpusat (secara logika) atau adanya sistem operasi jaringan yang mampu membentuk peta logika (*logical map*) dari seluruh jaringan dan kemudian memrepresentasikannya melalui *Application Programming Interface* (API).
4. Virtualisasi dimana beberapa sistem operasi jaringan dapat mengontrol bagian-bagian (*slices* atau *substrates*) dari perangkat yang sama.

Arsitektur SDN terdiri dari tiga *layer* yaitu sebagai berikut, seperti tampak pada Gambar 2.1:



Gambar 2.1 Arsitektur SDN [3]

1. *Application Plane*: berada pada lapisan teratas, berupa aplikasi yang dapat secara langsung dan eksplisit mendefinisikan *network requirement* dan *network behavior* yang diinginkan.
2. *Controller Plane*: yaitu entitas kontrol yang memiliki tugas yaitu mentranslasikan *network requirement* yang telah didefinisikan oleh *Application Layer* menjadi instruksi-instruksi yang sesuai untuk *Infrastructure Layer*, dan memberikan *abstract view* yang dibutuhkan bagi *Application Layer* (*abstract view* meliputi informasi statistik dan *event* yang terjadi di jaringan).
3. *Data Plane*: terdiri dari elemen jaringan yang dapat menerima instruksi dari *Control Layer*.
4. *Northbound API*: SDN menggunakan *northbound API* untuk berkomunikasi dengan *application layer*. *Northbound API* juga membantu operator dalam mengimplementasikan *software* baru.
5. *Southbound API*: SDN menggunakan *southbound API* untuk menyampaikan informasi ke *switch* dan *router*. OpenFlow merupakan salah satu protokol yang digunakan pada *southbound API* dan menjadi bagian umum dari SDN.

Control Plane merupakan otak dari jaringan SDN, dapat dijalankan secara terpisah dari *Data Plane*. Sedangkan *Data Plane* merupakan perangkat-perangkat keras jaringan yang terprogram secara khusus dan dikendalikan penuh oleh *Control Plane*. Pada SDN tersedia *open interface* yang memungkinkan sebuah entitas *software* atau aplikasi untuk mengendalikan konektivitas dari sumber daya jaringan, mengendalikan aliran lalu lintas paket data, serta melakukan inspeksi atau memodifikasi lalu lintas paket data tersebut.

2.2 OpenFlow

OpenFlow adalah sebuah protokol standar yang digunakan dalam komunikasi antar *control-plane* dan *data-plane* [4]. OpenFlow mendefinisikan infrastruktur *flow-based forwarding* dan Application Programming Interface (API) standar yang memungkinkan controller untuk mengarahkan fungsi dari switch melalui saluran yang aman (secure channel) [5].

Untuk bisa menggunakan OpenFlow diperlukan *controller* SDN yang mendukung jalannya protokol OpenFlow. POX *controller* adalah salah satu *controller* SDN yang mendukung protokol OpenFlow. POX adalah *controller* yang berbasis bahasa *Python*, dan dapat dijalankan pada sistem operasi Windows, MacOS dan Linux. POX masih dalam tahap pengembangan, dan merupakan perkembangan dari *controller* terdahulu yaitu NOX (berbasis bahasa C). Dalam jaringan SDN, *controller* harus diaktifkan dahulu sebelum menjalankan jaringan itu sendiri.

2.3 Mininet

Mininet adalah suatu *software emulator* yang memungkinkan untuk melakukan percobaan pada jaringan yang luas dengan hanya menggunakan satu mesin. Mininet digunakan untuk mengemulasi *data plan* atau *infrastructure layer* dalam perancangan jaringan SDN [6].

2.4 Jaringan Komputer

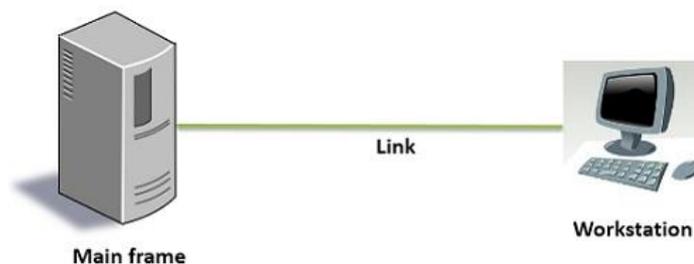
Jaringan adalah seperangkat alat (atau disebut *node*) yang dihubungkan oleh jalur komunikasi. Sebuah *node* dapat berupa komputer atau peralatan lain yang mampu mengirim atau menerima data dari *node* lain dalam suatu jaringan [7].

2.4.1 Struktur Fisik Jaringan Komputer

Perangkat-perangkat dalam suatu jaringan saling terhubung menggunakan link. Link adalah sebuah jalur komunikasi untuk mengirim data dari satu perangkat ke perangkat lain. Terdapat dua jenis koneksi, yaitu *point-to-point* dan *multipoint* [7].

1. Point-to-point

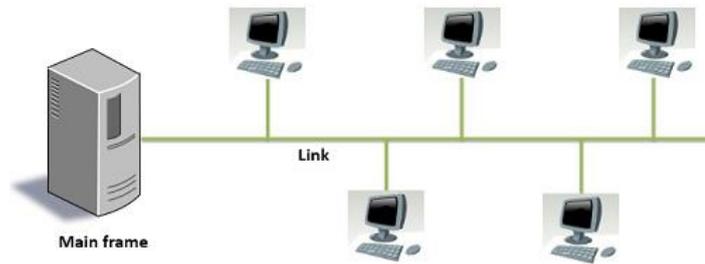
Sebuah koneksi *point-to-point* adalah dua buah perangkat yang saling berkomunikasi menggunakan sebuah saluran. Koneksi *point-to-point* sering menggunakan kabel sebagai medianya atau media nirkabel seperti *microwave*, *link* satelit. Berikut ini merupakan contoh gambar *point-to-point*:



Gambar 2.2 *Point-to-point*

2. Multipoint

Sebuah koneksi *multipoint* adalah koneksi antara satu atau beberapa perangkat dalam satu saluran yang digunakan bersama. Berikut ini merupakan contoh gambar *multipoint*:



Gambar 2.3 *Multipoint*

2.4.2 Topologi Jaringan Komputer

Topologi adalah suatu aturan/*rules* bagaimana menghubungkan komputer (*node*) satu sama lain secara fisik dan pola hubungan antara komponen-komponen yang berkomunikasi melalui media/peralatan jaringan, seperti: *server*, *workstation*, *hub/switch*, dan pengabelannya, sedangkan jaringan merupakan sebuah sistem yang terdiri atas komputer, perangkat komputer, tambahan dan perangkat jaringan lainnya yang saling berhubungan dengan menggunakan media tertentu dengan aturan yang sudah ditetapkan [8].

2.4.2.1 Topologi Fisik Jaringan

Topologi fisik jaringan adalah representasi geometris dari hubungan semua saluran dan perangkat yang saling berhubungan satu sama lain. Topologi fisik jaringan merupakan desain aktual yang dibangun untuk menghubungkan beberapa perangkat dalam sebuah jaringan [7]. Berikut adalah jenis-jenis topologi fisik:

1. Topologi Bus

Topologi Bus merupakan topologi yang menggunakan koneksi *multipoint*, karena memiliki satu saluran yang digunakan bersama untuk berbagi data. Dalam topologi Bus setiap *node* dihubungkan oleh kabel yang langsung dihubungkan dengan saluran utama. Setiap kabel yang berhubungan dengan saluran utama dihubungkan oleh tap. Sinyal akan semakin melemah jika jarak yang ditempuhnya semakin jauh. Oleh karena itu perlu adanya beberapa tap untuk menjaga agar sinyal tidak melemah. Topologi ini seringkali dijumpai pada sistem *client/server*, dimana salah satu mesin pada jaringan tersebut difungsikan sebagai *file server*, yang berarti

bahwa mesin tersebut dikhususkan hanya untuk pendistribusian data dan biasanya tidak digunakan untuk pemrosesan informasi.

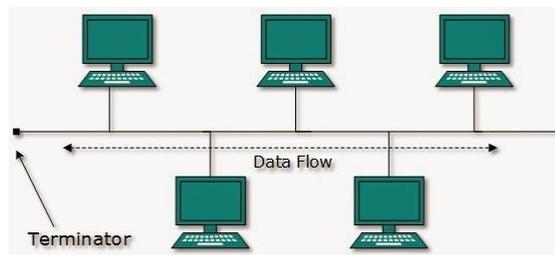
Kelebihan dari topologi Bus:

1. Hemat kabel.
2. Layout kabel sederhana.
3. Pengembangan jaringan atau penambahan workstation baru dapat dilakukan dengan mudah tanpa mengganggu workstation lain.

Kekurangan dari topologi Bus:

1. Deteksi dan isolasi kesalahan sangat kecil.
2. Kepadatan lalu lintas data.
3. Bila salah satu client rusak, maka jaringan tidak bisa berfungsi.
4. Diperlukan repeater untuk jarak jauh.

Berikut ini merupakan gambar dari topologi Bus:



Gambar 2.4 Topologi Bus [9]

2. Topologi Ring

Pada topologi Ring setiap perangkat berhubungan hanya dengan perangkat yang berada di sebelahnya menggunakan koneksi *point-to-point*, membentuk jalur melingkar seperti cincin. Sinyal dilewatkan dalam satu arah dari satu perangkat ke perangkat lainnya hingga mencapai tujuan. Setiap perangkat dihubungkan dengan saluran utama menggunakan *repeater*. Sinyal akan semakin melemah jika jarak yang ditempuhnya semakin jauh, sehingga diperlukan *repeater*. Dengan adanya fungsi *repeater* ini maka sinyal data tersebut akan di-*generate* kembali sebelum diteruskan ke komputer yang dituju. Pada topologi Ring, komunikasi data dapat

terganggu jika satu titik mengalami gangguan. Jaringan *Fiber Distributed Data Interface* (FDDI) mengantisipasi kelemahan ini dengan mengirim data searah jarum jam dan berlawanan dengan arah jarum jam secara bersamaan.

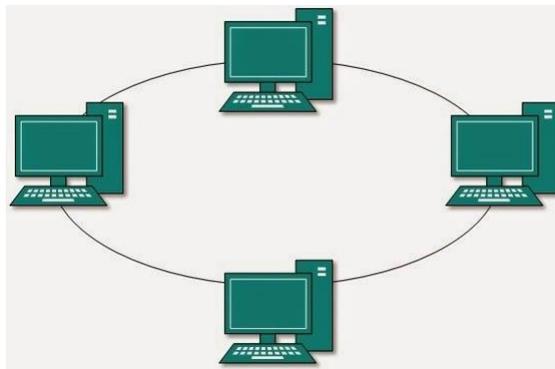
Kelebihan dari topologi Ring:

1. Hemat kabel.
2. Tidak terjadi *collision* atau tabrakan pengiriman data seperti pada topologi Bus, karena hanya satu nod dapat mengirim data pada satu waktu.

Kekurangan dari topologi Ring:

1. Setiap *node* dalam jaringan akan selalu ikut serta mengelola informasi yang dilewatkan dalam jaringan, sehingga jika terdapat gangguan di suatu *node* maka seluruh jaringan akan terganggu.
2. Pengembangan jaringan lebih kaku.

Berikut ini merupakan gambar dari topologi Ring:



Gambar 2.5 Topologi Ring [9]

3. Topologi Tree

Topologi Tree pada dasarnya merupakan bentuk yang lebih luas dari topologi Star. Seperti halnya topologi Star, perangkat (*node, device*) yang ada pada topologi Tree juga terhubung kepada sebuah pusat pengendali (sentral *hub* atau *switch*) yang berfungsi mengatur lalu lintas di dalam jaringan. Meskipun demikian, tidak semua perangkat pada topologi Tree terhubung secara langsung ke sentral *hub*. Pada topologi Tree terdapat dua atau lebih *hub* yang digunakan untuk menghubungkan

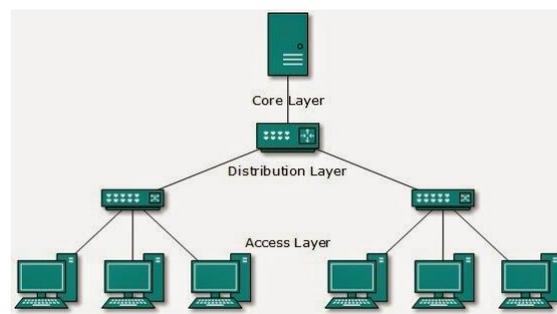
setiap perangkat ke dalam jaringan. Berdasarkan fungsinya, *hub* tersebut terbagi menjadi dua bagian, yaitu *hub* aktif dan *hub* pasif. *Hub* aktif berfungsi tidak hanya sekedar sebagai penerus sinyal data dari satu komputer ke komputer lainnya, tetapi juga memiliki fungsi sebagai *repeater*. Sedangkan *hub* pasif hanya berfungsi sebagai penerus sinyal data dari komputer ke komputer lainnya.

Kelebihan dari topologi Tree:

1. Pemasangannya relatif mudah, karena setiap alat atau komputer hanya membutuhkan satu *port I/O*.
2. Seteksi kesalahan cukup mudah.
3. Kerusakan pada salah satu saluran tidak mempengaruhi sambungan yang lain, asalkan *hub* masih berfungsi.
4. Adanya *hub* sekunder memungkinkan lebih banyak perangkat yang dapat tersambung ke sebuah *hub* sentral, sehingga menambah jarak jangkauan jaringan.

Kekurangan topologi Tree hampir sama dengan kekurangan pada topologi Star, yaitu jika *node* sentral mengalami kerusakan, maka seluruh jaringan akan terhenti.

Berikut ini merupakan gambar dari topologi Tree:



Gambar 2.6 Topologi Tree [9]

4. Topologi Star

Pada topologi Star, setiap perangkat memiliki saluran *point-to-point* hanya ke sebuah perangkat pusat, biasanya disebut *hub*. Perangkat tidak berhubungan langsung dengan perangkat lain. Pada topologi Star jika suatu perangkat ingin mengirimkan dat ke perangkat lain, perangkat pengirim harus mengirimkannya

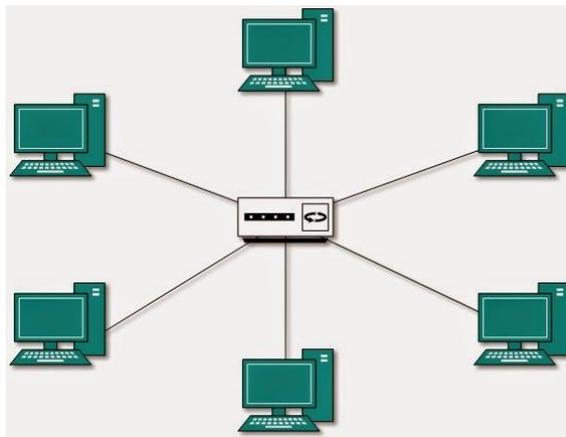
melalui *hub*, kemudian *hub* tersebut menyampaikan ke perangkat yang terhubung lainnya.

Kelebihan dari topologi Star:

1. Kerusakan pada satu saluran hanya akan mempengaruhi jaringan pada saluran tersebut dan perangkat yang terpaat.
2. Tahan terhadap lalu lintas jaringan yang sibuk.
3. Penambahan dan pengurangan perangkat dapat dilakukan dengan mudah.

Kekurangan dari topologi Star adalah jika *node* tengah mengalami kerusakan, maka seluruh jaringan akan terhenti.

Berikut ini merupakan gambar dari topologi Star:

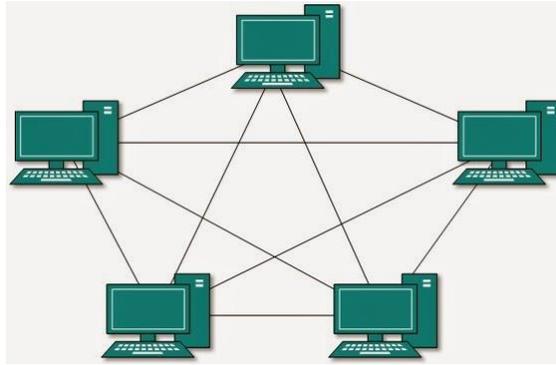


Gambar 2.7 Topologi Star [9]

5. Topologi Mesh

Topologi Mesh dibangun dengan memasang banyak *link* pada setiap komputer. Hal ini dimungkinkan karena pada setiap komputer terdapat lebih dari satu NIC. Topologi ini secara teori memungkinkan akan tetapi tidak praktis dan biayanya cukup tinggi. Topologi Mesh memiliki tingkat *redundancy* yang tinggi.

Berikut ini merupakan gambar dari Topologi Mesh:

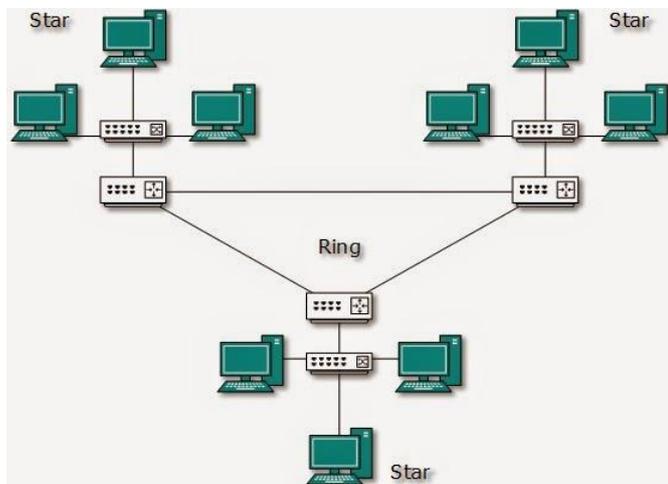


Gambar 2.8 Topologi Mesh [9]

6. Topologi Hybrid

Topologi Hybrid adalah jaringan yang dibentuk dari berbagai topologi dan teknologi. Sebuah topologi Hybrid memiliki semua karakteristik dari topologi dasar yang terdapat dalam jaringan tersebut.

Berikut ini merupakan gambar dari topologi Hybrid:



Gambar 2.9 Topologi Hybrid [9]

2.4.2.2 Topologi Logis Jaringan

Topologi jaringan secara logis menggambarkan bagaimana perangkat berkomunikasi, atau dengan kata lain bentuk komunikasi. Berbeda dengan desain topologi jaringan secara fisik yang dilihat bentuknya secara fisik seperti kabel atau perangkat jaringan fisik lainnya, desain topologi jaringan secara logis lebih terkait

dengan protokol jaringan. Topologi jaringan logis bisa sama, lebih sederhana, ataupun lebih rumit dari topologi jaringan fisiknya [10]. Berikut ini adalah jenis topologi logis jaringan [11]:

1. Ethernet

Protocol *Ethernet* sejauh ini adalah yang paling banyak digunakan, *Ethernet* menggunakan metode akses yang disebut *Carrier Sense Multiple Access/Collision Detection* (CSMA/CD). Sistem ini menjelaskan bahwa setiap komputer memperhatikan ke dalam kabel dari network sebelum mengirimkan sesuatu ke dalamnya. Jika di dalam jaringan tidak ada aktifitas atau bersih komputer akan mentransmisikan data, jika ada transmisi lain di dalam kabel, komputer akan menunggu dan akan mencoba kembali transmisi jika jaringan telah bersih. Kadangkala dua buah komputer melakukan transmisi pada saat yang sama, ketika hal ini terjadi, masing-masing komputer akan mundur dan akan menunggu kesempatan secara acak untuk mentransmisikan data kembali. Metode ini dikenal dengan koalisi, dan tidak akan berpengaruh pada kecepatan tranmisi dari *network*. Protocol *Ethernet* dapat digunakan untuk pada model jaringan Bus, Star, atau Tree. Data dapat ditransmisikan melewati kabel *twisted pair*, *coaxial*, ataupun kabel *fiber optic* pada kecepatan 10Mbps.

2. LocalTalk

LocalTalk adalah sebuah protokol yang di kembangkan oleh Apple Computer, Inc. untuk mesin-mesin komputer Macintosh. Metode yang digunakan oleh *LocalTalk* adalah *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA). Hampir sama dengan CSMA/CD. Adapter *LocalTalk* dan kabel *twisted pair* khusus dapat digunakan untuk menghubungkan beberapa komputer melewati *port* serial. Sistem Operasi Macintosh memungkinkan koneksi secara jaringan *peer-to-peer* tanpa membutuhkan tambahan aplikasi khusus. Protokol *LocalTalk* dapat digunakan untuk model jaringan Bus, Star, ataupun model Tree dengan menggunakan kabel *twisted pair*. Kekurangan yang paling mecolok yaitu kecepatan transmisinya. Kecepatan transmisinya hanya 230Kbps.

3. Token Ring

Protocol Token di kembangkan oleh IBM pada pertengahan tahun 1980. Metode Aksesnya melalui lewatnya sebuah token dalam sebuah lingkaran seperti Cincin. Dalam Token Ring, komputer-komputer dihubungkan satu dengan lainnya seperti sebuah cincin. Sebuah sinyal token bergerak berputar dalam sebuah lingkaran (cincin) dalam sebuah jaringan dan bergerak dari sebuah komputer menuju ke komputer berikutnya, jika pada persinggahan di salah satu computer ternyata ada data yang ingin ditransmisikan, token akan mengangkutnya ke tempat dimana data itu ingin ditunjukkan, token bergerak terus untuk saling mengkoneksikan diantara masing-masing komputer. Protokol Token Ring membutuhkan model jaringan Star dengan menggunakan *twisted pair* atau kabel *fiber optic*. Dan dapat melakukan kecepatan transmisi 4Mbps atau 16Mbps. Sejalan dengan perkembangan *Ethernet*, penggunaan Token Ring makin berkurang sampai sekarang.

4. FDDI

Fiber Distributed Data Interface (FDDI) adalah sebuah protokol jaringan yang menghubungkan antara dua atau lebih jaringan bahkan pada jarak yang jauh. Metode aksesnya yang digunakan oleh FDDI adalah metode token. FDDI menggunakan dua buah topologi Ring secara fisik. Proses transmisi biasanya menggunakan satu buah Ring, namun jika ada masalah ditemukan akan secara otomatis menggunakan Ring yang kedua. Sebuah keuntungan dari FDDI adalah kecepatan dengan menggunakan kabel *fiber optic* pada kecepatan 100Mbps.

5. ATM

ATM adalah singkatan dari *Asynchronous Transfer Mode* (ATM) yaitu sebuah protokol jaringan yang mentransmisikan pada kecepatan 155Mbps atau lebih. ATM mentransmisikan data ke dalam satu paket dimana pada protocol yang lain mentransfer pada besar-kecilnya paket. ATM mendukung variasi media seperti video, CD-audio, dan gambar. ATM bekerja pada model topologi Star, dengan menggunakan Kabel *fiber optic* ataupun kabel *twisted pair*. ATM pada umumnya

digunakan untuk menghubungkan dua atau lebih LAN. Dia juga banyak dipakai oleh *Internet Service Providers* (ISP) untuk meningkatkan kecepatan akses Internet untuk klien mereka.

2.4.3 IP Address Versi 4

IP Address terdiri dari dua versi, yaitu IPv4 dan IPv6. Namun dalam subbab ini hanya akan menjelaskan IPv4. IPv4 memiliki ukuran sebesar 32-bit. IPv4 dibagi ke dalam lima kelas, yaitu A, B, C, D, dan E. Metode ini disebut dengan *classful addressing*. Pada umumnya yang digunakan dalam pengalamatan IPv4 untuk setiap *host* hanya tiga kelas, yaitu A, B, dan C. Berikut ini pembagian setiap kelas-kelasnya [12]:

1. Kelas A

Format : 0nnnnnnn.hhhhhhhh.hhhhhhhh.hhhhhhhh (n = Net ID, h = Host ID)

Byte pertama : 0 – 127

Jumlah Network : 126

Range IP : 1.xxx.xxx.xxx.sampai 126.xxx.xxx.xxx (0 dan 127 dicadangkan)

Jumlah IP : 16.777.214

2. Kelas B

Format : 10nnnnnnn.nnnnnnnn.hhhhhhhh.hhhhhhhh (n = Net ID, h = Host ID)

Byte pertama : 128 – 191

Jumlah Network : 16.384

Range IP : 128.0.0.xxx sampai 191.255.xxx.xxx

Jumlah IP : 65.534

3. Kelas C

Format : 110nnnnn.nnnnnnnn.nnnnnnnn.hhhhhhhh (n = Net ID, h = Host ID)

Byte pertama : 192 – 223

Jumlah Network : 2.097.152

Range IP : 192.0.0.xxx sampai 255.255.255.xxx
Jumlah IP : 254

2.4.4 Quality of Service (QoS)

Quality of Service (QoS) merupakan metode pengukuran tentang seberapa baik jaringan dan merupakan suatu usaha untuk mendefinisikan karakteristik dan sifat dari satu servis. QoS digunakan untuk mengukur sekumpulan atribut kinerja yang telah dispesifikasikan dan diasosiasikan dengan suatu servis [13].

Ada beberapa parameter pengukuran menggunakan QoS, diantaranya [14]:

1. Delay

Delay atau waktu tunda merupakan selang waktu yang dibutuhkan oleh suatu paket data saat data mulai dikirim dan keluar dari proses antrian sampai mencapai titik tujuan. Waktu waktu tunda dinyatakan dalam satuan detik. Sebuah organisasi internasional yang membakukan dan meregulasi radio internasional dan telekomunikasi yaitu ITU-T, merekomendasikan batas nilai *delay* yang dianggap baik yaitu sebesar 0 – 150ms seperti yang terlampir pada Tabel 2.1 di bawah.

Tabel 2.1 Standar *Delay* Berdasarkan ITU-T

Delay (ms)	Kualitas
0 - 150	Baik
151 - 400	Cukup, masih dapat diterima
> 400	Buruk

2. Jitter

Jitter adalah perbedaan selang waktu kedatangan antar paket di terminal tujuan. *Jitter* dipengaruhi oleh variasi beban trafik dan besarnya tumbukan antar paket (*congestion*) yang ada dalam jaringan. Semakin besar beban trafik di dalam jaringan akan menyebabkan semakin besar pula peluang terjadinya *congestion* dengan demikian nilai *jitter*-nya akan semakin besar. ITU-T juga memberikan standar rekomendasi nilai *jitter* yang baik seperti yang tertera dalam tabel 2.2 di bawah.

Tabel 2.2 Standar *Jitter* Berdasarkan ITU-T

Jitter (ms)	Kualitas
0 – 20	Baik
21 - 50	Dapat diterima
> 50	Buruk

3. Throughput

Throughput adalah laju rata-rata dari paket data yang berhasil dikirim melalui kanal komunikasi atau dengan kata lain *throughput* adalah jumlah paket data yang diterima setiap detik. *Throughput* dinyatakan dalam satuan *bit per second* (bps) dan juga dalam satuan data paket per second.

2.4.5 Iperf

Iperf adalah *software* berbasis *console* yang digunakan untuk membangkitkan *traffic* UDP dan TCP, Iperf berjalan pada model *client-server*, *server* Iperf membangkitkan *traffic* UDP atau TCP pada sisi *client* [15].

2.5 Ubuntu

Ubuntu adalah salah satu distribusi Linux yang berbasiskan Debian GNU/Linux dan memiliki interface desktop. Ubuntu adalah sepenuhnya sistem operasi open source yang dibangun berdasarkan kernel Linux. Nama Ubuntu sendiri diambil dari bahasa kuno Afrika Zulu dan Xhosa (oo-boon-too) yang artinya “rasa perikemanusiaan terhadap orang lain”. Berdasarkan Ubuntu Documentation Project (2008) dijelaskan bahwa lingkungan desktop bawaan Ubuntu adalah GNOME, platform pengembangan dan keluarga desktop UNIX dan Linux terdepan. Skema penomoran versi Ubuntu didasarkan pada tanggal rilis sebuah versi dari distribusi. Nomor versi berasal dari tahun dan bulan rilis, bukan mencerminkan versi sebenarnya dari perangkat lunak [16].