

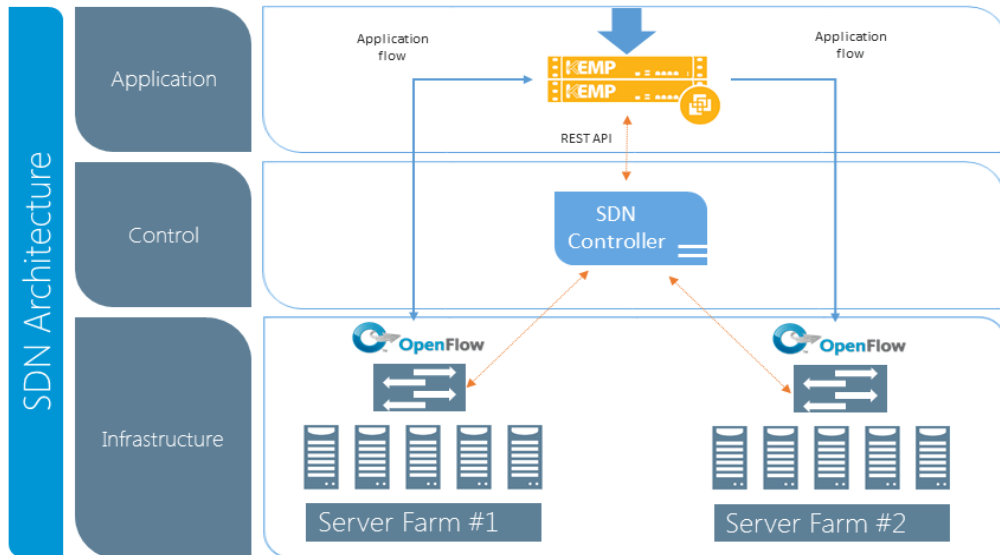
BAB II

DASAR TEORI

2.1 Software-Defined Network

Software-Defined Network (SDN) adalah sebuah arsitektur baru dalam bidang jaringan komputer, yang bersifat dinamis, *manageable*, *cost-effective*, dan *adaptable*, sehingga sangat ideal untuk kebutuhan jaringan saat ini yang bersifat dinamis dan bandwidth yang besar. Arsitektur ini memisahkan antara fungsi *control* dan fungsi *forwarding*, sehingga *network control* tersebut menjadi *directly programmable* (dapat diprogram secara langsung), sedangkan infrastruktur yang mendasarinya dapat diabstraksikan untuk layer aplikasi dan layanan jaringan [1].

Secara umum dalam perangkat jaringan terdapat tiga proses utama, yaitu *Application Plane*, *Control Plane* dan *Data Plane*. *Application Plane* merupakan bagian paling atas yang menyediakan interface kepada user dalam pembuatan program aplikasi yang kemudian akan mengatur dan mengoptimalkan penggunaan jaringan.



Gambar 2.1 Arsitektur SDN

Control Plane adalah bagian yang berfungsi untuk mengontrol jaringan, mengkonfigurasi sistem, manajemen jaringan, menentukan informasi *routing table* dan *forwarding table*. *Data Plane* adalah bagian yang bertanggung jawab

memforward/meneruskan paket, selain itu juga menguraikan header paket, mengatur QoS, dan enkapsulasi paket [2].

Tujuan utama dari SDN adalah untuk mencapai pengelolaan jaringan yang lebih baik dengan tingkatan dan kompleksitas yang besar serta memastikan bahwa semua keputusan dari sistem kontrol dibuat dari titik pusat (kontroler). SDN memperkenalkan suatu metode untuk meningkatkan tingkat abstraksi pada konfigurasi jaringan, menyediakan mekanisme yang secara otomatis bereaksi terhadap perubahan yang sering terjadi dan terus-menerus untuk jaringan [3].

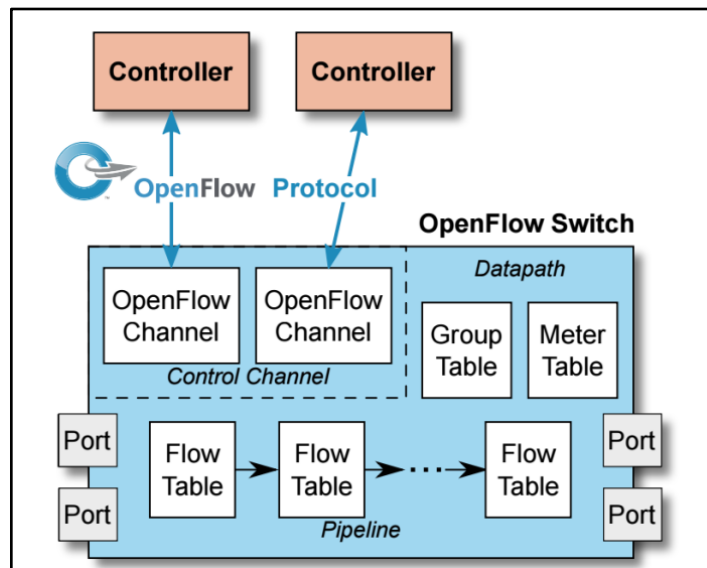
Berikut adalah karakteristik *Software-defined network* yang tertera pada website *Open Networking Foundation* [4] :

- a. *Directly programmable*, yaitu *network control* yang dapat di program secara langsung karena *control plane* yang terpisah dari *data plane*.
- b. *Agile*, yaitu abstraksi bidang control dari bidang data yang dapat memperbolehkan admin jaringan melakukan pengaturan jaringan yang dapat diubah sesuai kebutuhan.
- c. *Centrally managed*, yaitu sentralisasi kontrol jaringan dengan software berbasis SDN yang dapat menangani jaringan secara luas.
- d. *Programmatically configured*, yaitu dimana pada jaringan SDN admin jaringan dapat mengkonfigurasi, mengatur, melakukan pengamanan dan mengoptimalkan jaringan secara dinamis, cepat, dan otomatis karena tidak bergantung pada software khusus.
- e. *Open standards-based and vendor neutral*, yaitu dengan menyederhanakan desain dan operasi jaringan karena menyediakan controller dengan protokol jaringan yang dapat digunakan oleh berbagai vendor.

2.2 OpenFlow

OpenFlow adalah standar antarmuka komunikasi atau protokol yang menghubungkan antara *data plane* dengan *control plane* pada arsitektur SDN. *OpenFlow* mengizinkan akses langsung ke *data plane* dari sebuah perangkat jaringan seperti switch dan router baik fisik maupun virtual [2]. Protokol *OpenFlow* pada arsitektur jaringan SDN dapat mengatasi kebutuhan bandwidth yang tinggi, aplikasi dan layanan yang bersifat dinamis, dan secara signifikan mengurangi operasi kompleksitas manajemen pada jaringan yang bersifat adaptif

dan selalu berubah-ubah. Secara sederhana *OpenFlow* merupakan sebuah antarmuka antara SDN kontroler dengan perangkat switch [5].



Gambar 2.2 Struktur OpenFlow

2.3 OpenDaylight Controller

Kontroler SDN adalah aplikasi SDN yang mengelola *flow control* untuk mengaktifkan *intelligence networking*. Kontroler SDN bekerja berdasarkan protokol seperti *OpenFlow* yang memungkinkan server memberitahu kemana paket dikirimkan. Perangkat meneruskan paket data yang diterima berdasarkan aturan yang ditetapkan dari kontroler [6]. Kontroler menentukan apa yang harus dilakukan dengan paket dan, jika perlu, mengirimkan aturan baru untuk perangkat sehingga dapat menangani paket data di masa depan dengan cara yang sama. Terdapat banyak jenis kontroler yang tersedia secara open source salah satunya adalah *OpenDaylight*. *OpenDaylight* merupakan sebuah infrastruktur kontroler yang memiliki ketersediaan tinggi, modular, *extensible*, *scalable* dan multi-protokol, dibangun untuk penyebaran SDN di jaringan heterogen, multi-vendor, serta modern.

2.4 Mininet

Mininet adalah emulator jaringan SDN yang dapat mensimulasikan kinerja antara *end-host*, switch, router, kontroler, dan link dalam sebuah kernel Linux [6]. Mininet merupakan sebuah sistem virtualisasi yang dapat menggambarkan jaringan yang besar dengan hanya menggunakan sebuah laptop. Mininet bersifat

open source, sehingga proyek yang telah dilakukan berupa *source code*, *scripts*, dan dokumentasi yang dapat dikembangkan oleh siapapun.

Karakteristik yang dimiliki Mininet yaitu :

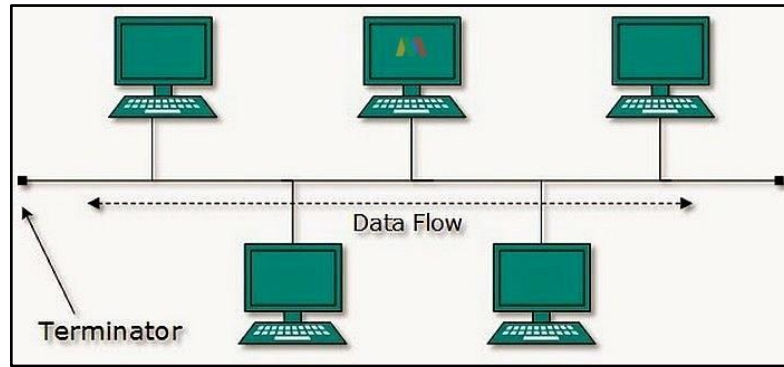
- a. *Flexible*, topologi dan fungsionalitas yang baru akan sistem yang dibuat harus didefinisikan dalam bentuk software menggunakan bahasa yang familiar dan operating system.
- b. *Deployable*, jika ingin membangun sebuah prototype dengan fungsionalitas yang benar untuk jaringan yang bersifat hardware-based, testbed yang dibuat tidak memerlukan perubahan terhadap kode atau konfigurasi yang dirubah sebelumnya.
- c. *Interactive*, mengatur dan menjalankan jaringan harus berdasarkan waktu yang riil, sebagaimana berhubungan dengan jaringan yang sebenarnya.
- d. *Scalable*, lingkup prototype ini dapat menggambarkan jaringan dengan jumlah switch dari ratusan hingga ribuan hanya dalam satu laptop.
- e. *Realistic*, perilaku dari prototipe dapat menunjukkan perilaku jaringan yang sebenarnya dengan tingkat kepercayaan yang tinggi.
- f. *Share-able*, prototipe yang telah dibentuk dapat saling dibagikan antar kolaborator untuk dijalankan atau dimodifikasi dalam eksperimen selanjutnya.

2.5 Topologi Jaringan

Topologi adalah suatu aturan bagaimana menghubungkan computer satu sama lain secara fisik dan pola hubungannya antara komponen-komponen yang berkomunikasi melalui media atau peralatan jaringan baik secara fisik (*physical topology*) maupun secara logik (*logic topology*).

2.5.1 Topologi Bus

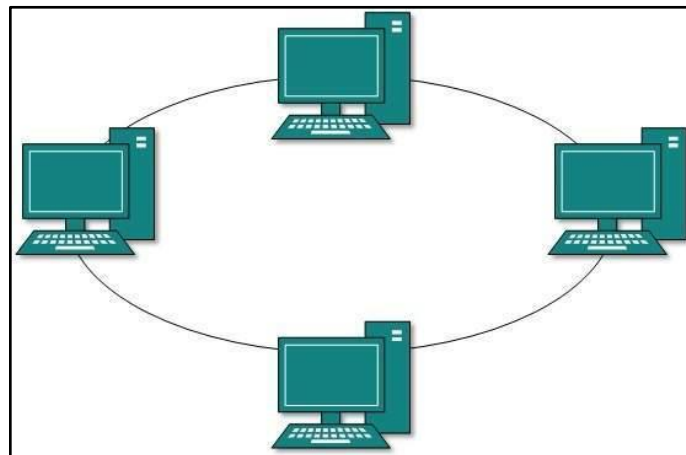
Topologi bus merupakan topologi yang banyak dipergunakan pada masa penggunaan kabel coaxial menjamur[6]. Karakteristik topologi ini yaitu satu kabel yang kedua ujungnya ditutup dimana sepanjang kabel terdapat node-node, paling prevevalent karena sederhana dalam instalasi, signal melewati kabel 2 arah dan mungkin terjadi collision.



Gambar 2.3 Topologi Bus

2.5.2 Topologi Ring

Topologi ring adalah topologi yang informasi dan data serta traffic disalurkan sedemikian rupa[6]. Umumnya fasilitas ini memanfaatkan fiber optic sebagai sarannya. Karakteristik topologi ini yaitu lingkaran tertutup yang berisi node-node, sederhana dalam layout, signal mengalir dalam satu arah sehingga menghindari terjadinya collision lihat Gambar 2.4.

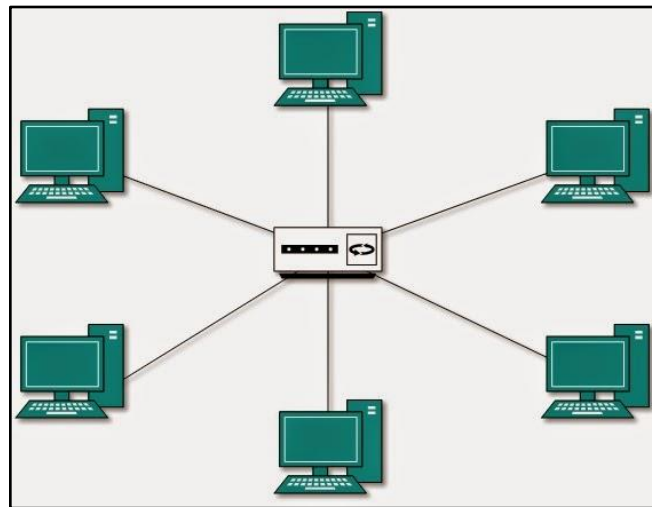


Gambar 2.4 Topologi Ring

2.5.3 Topologi Star

Topologi star merupakan topologi yang banyak digunakan diberbagai tempat, karena kemudahan untuk menambah, mengurangi, atau mendeteksi kerusakan jaringan yang ada[6]. Karakteristik topologi ini yaitu setiap node berkomunikasi langsung dengan central node, traffic data mengalir dari node ke central node dan kembali lagi, mudah dikembangkan karena setiap node hanya memiliki kabel yang langsung

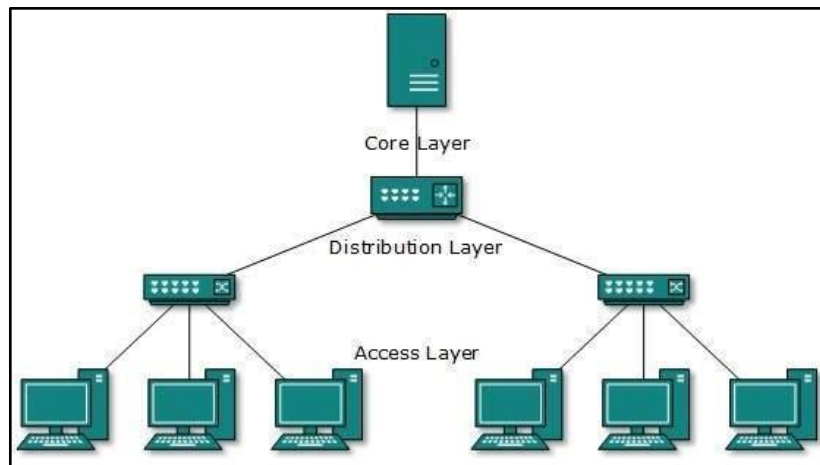
terhubung ke central node, keunggulan jika satu kabel node terputus maka yang lainnya tidak akan terganggu lihat Gambar 2.5.



Gambar 2.5 Topologi Star

2.5.4 Topologi Tree

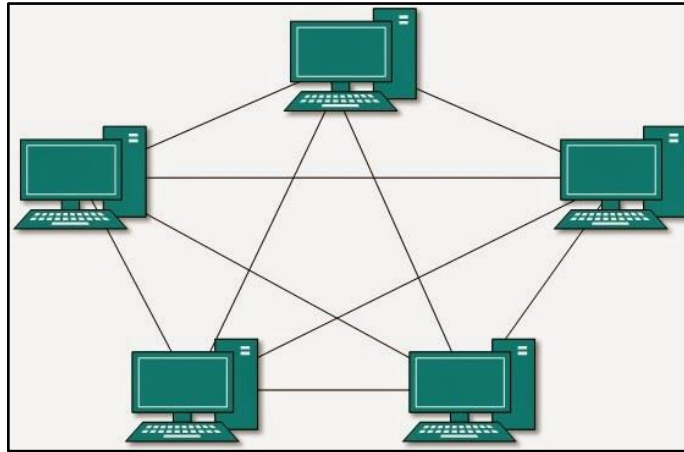
Topologi tree merupakan topologi jaringan dimana topologi ini merupakan gabungan atau kombinasi dari ketiga topologi yang ada yaitu topologi star, topologi ring, dan topologi bus[6] , lihat Gambar 2.6.



Gambar 2.6 Topologi Tree

2.5.5 Topologi Mesh

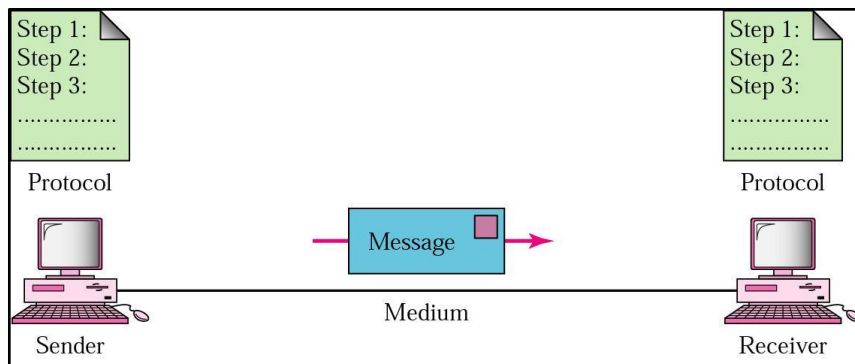
Topologi Mesh merupakan topologi dimana setiap perangkatnya saling terhubung secara dedicate point to point ke semua perangkat. Dan beban koneksi satu link tidak akan mengganggu link lainnya.



Gambar 2.7 Topologi Mesh

2.6 Komunikasi Data

Komunikasi data adalah hubungan atau interaksi (pengiriman dan penerimaan) antar device yang terhubung dalam sebuah jaringan, baik yang dengan jangkauan sempit maupun dengan jangkauan yang lebih luas. Efektivitas dari komunikasi data tergantung pada empat karakteristik mendasar yaitu pengiriman, akurasi, ketepatan waktu, dan jitter [7].



Gambar 2.8 Komunikasi Data

Komponen Komunikasi Data meliputi:

1. Pengirim, perangkat yang bertugas mengirimkan data.
2. Penerima, perangkat yang bertugas menerima data.
3. Data, informasi yang akan dikomunikasikan.
4. Media pengiriman, perantara yang digunakan untuk melakukan proses pengiriman data.
5. Protokol, aturan-aturan yang berfungsi sebagai penyelaras hubungan.

2.7 Quality of Service (QoS)

Quality of Service (QoS) adalah efek kolektif dari kinerja layanan yang menentukan derajat kepuasan seorang pengguna terhadap sebuah layanan. Ada banyak parameter yang menjadi acuan untuk QoS di antaranya yaitu *delay*, *throughput*, *jitter*, dan *packet loss*.

2.7.1 Delay

Delay adalah perbedaan waktu di antara waktu kirim dengan waktu terima sebuah paket. Dalam tugas akhir ini *delay* yang dimaksudkan adalah *one way delay*, yaitu waktu rata – rata pengiriman setiap paket dalam perjalanan dari satu titik kirim ke satu titik terima.

Tabel 2.1 Kategori Delay

Kategori	Delay (ms)
Sangat Baik	< 150
Baik	150 s/d 300
Sedang	300 s/d 450
Buruk	> 450

Sumber : TIPHON

Persamaan perhitungan *Delay* :

$$Delay = \frac{Packet\ length}{Link\ bandwidth} \quad (1)$$

2.7.2 Jitter

Jitter adalah variasi *delay* yang diakibatkan oleh panjang antrian dalam suatu pengolahan data dan *reassemble* paket data di akhir pengiriman akibat kegagalan sebelumnya. *Jitter* merupakan variasi selisih dari setiap *delay* dengan *delay* selanjutnya. Dalam tugas akhir ini *jitter* yang dimaksud adalah nilai dari *oneway delay variation*.

Tabel 2.2 Kategori Jitter

Kategori	Jitter (ms)
Sangat Baik	0
Baik	0 s/d 75
Sedang	75 s/d 125

Buruk	125 s/d 225
-------	-------------

Sumber : TIPHON

Persamaan perhitungan *Jitter* :

$$Jitter = \frac{\text{Delay} - (\text{rata-rata delay})}{\text{Paket data diterima}} \quad (2)$$

2.7.3 Throughput

Throughput adalah jumlah bit yang sukses diterima dari satu terminal tertentu di dalam sebuah jaringan, dari suatu titik jaringan ke titik jaringan lainnya dibandingkan dengan total waktu pengiriman.

Tabel 2.3 Kategori Throughput

Kategori	Throughput (bps)
Sangat Baik	100
Baik	75
Sedang	50
Buruk	< 25

Sumber : TIPHON

Persamaan perhitungan *Throughput* :

$$Throughput = \frac{\text{Paket data diterima}}{\text{Lama pengamatan}} \quad (3)$$

2.7.4 Packet Loss

Packet Loss merupakan suatu parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang dapat terjadi karena *collision* dan *congestion* pada jaringan.

Tabel 2.4 Kategori *Packet Loss*

Kategori	Packet Loss (%)
Sangat Baik	0
Baik	3
Sedang	15
Buruk	25

Sumber : TIPHON

Persamaan perhitungan *Packet Loss* :

$$Packet Loss = \frac{(Paket\ data\ dikirim - Paket\ data\ diterima)}{Paket\ data\ dikirim} \times 100\% \quad (4)$$