# **LAPORAN KULIAH KERJA PRAKTIK**

**PERANCANGAN SOFTWARE DEFINED NETWORK MENGGUNAKAN KONTROLER OPENDAYLIGHT PADA JARINGAN DISKOMINFO PEMDA KAB. BEKASI**

Diajukan Sebagai Salah Satu Syarat Untuk Menempuh

Tugas Akhir Pada Program Sarjana Ilmu Komputer



Disusun Oleh

MELVIN DICKY RUDOLF BENJAMIN

311610234

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS PELITA BANGSA**

**BEKASI**

**2022**

# **LEMBAR PERSETUJUAN**

**LAPORAN KULIAH KERJA PRAKTIK**

**PERANCANGAN SOFTWARE DEFINED NETWORK**

**MENGGUNAKAN KONTROLER OPENDAYLIGHT**

**PADA JARINGAN DISKOMINFO**

**PEMDA KAB. BEKASI**

Diajukan untuk memenuhi syarat kelulusan pada Program Studi Teknik Informatika

Universitas Pelita Bangsa

Disusun Oleh :

**MELVIN DICKY RUDOLF BENJAMIN**

311610234

Telah diperiksa dan disahkan

Pada tanggal : ...., ......, ......

|  |  |  |
| --- | --- | --- |
| Pembimbing Lapangan  **Sufajar Butsianto, S.Kom., M.Kom.**  **NIDN. 0424068106** |  | Dosen Pembimbing  **Aswan S. Sunge, S.E.,M.Kom.**  **NIDN. 0426018003** |

|  |
| --- |
| Mengetahui,  Ketua Program Studi Teknik Informatika  **Aswan Supriyadi Sange, S.E., M.Kom.**  **NIDN.0426018003** |

# **LEMBAR PENGESAHAN**

**LAPORAN KULIAH KERJA PRAKTIK**

**PERANCANGAN SOFTWARE DEFINED NETWORK**

**MENGGUNAKAN KONTROLER OPENDAYLIGHT**

**PADA JARINGAN DISKOMINFO**

**PEMDA KAB. BEKASI**

Disusun Oleh :

**MELVIN DICKY RUDOLF BENJAMIN**

311610234

Telah dipertahankan didepan Dewan Penguji

Pada tanggal : ...., .........., .......

|  |  |  |
| --- | --- | --- |
| Dosen Penguji I  **Sufajar Butsianto, S.Kom., M.Kom.**  **NIDN. 0424068106** |  | Dosen Penguji II  **Aswan S. Sunge, S.E.,M.Kom.**  **NIDN. 0426018003** |

|  |
| --- |
| Mengetahui,  Ketua Program Studi Teknik Informatika  **Aswan Supriyadi Sange, S.E., M.Kom.**  **NIDN.0426018003** |

# **UCAPAN TERIMAKASIH**

Dengan memanjatkan puji syukur kehadirat Tuhan Yang Maha Pengasih dan Maha Penyayang yang telah melimpahkan segala rahmat, hidayat dan inayah-Nya kepada Penulis, sehingga laporan Kerja Praktik dengan Judul:" Perancangan Software Defined Network Menggunakan Kontroler Opendaylight : Studi Kasus (Pada Jaringan Diskominfo Pemda Kab.Bekasi). dapat diselesaikan sesuai dengan rencana karena dukungan dari berbagai pihak yang tidak ternilai besarnya. Oleh karena itu penulis menyampaikan terima kasih kepada:

1. Bapak Hamzah Muhammad Mardiputra, S.K.M., MM sebagai Rektor Universitas Pelita Bangsa
2. Ibu Putri Anggun Sari, S.Pt., M.Si sebagai Dekan Teknik Universitas Pelita Bangsa
3. Bapak Aswan S. Sunge, SE, M.Kom sebagai Kaprodi Teknik Informatika - S1
4. Bapak Sufajar Butsianto, SKom, MKom sebagai dosen pembimbing KKP
5. Bapak Gustav M sebagai pembina kerja praktik dilokasi kerja praktik
6. Orang Tua saya yang selalu memberikan dukungan dan doa
7. Teman-teman di kelas TI.16.A.2 yang selalu memberikan motivasi dan semangat
8. Seluruh direksi Universitas Pelita Bangsa dan Kantor Diskominfo Pemda Kab.Bekasi yang tidak dapat saya sebutkan namanya satu persatu.

Semoga Tuhan Yang Maha Pengasih dan Maha Penyayang memberikan balasan yang lebih besar kepada beliau beliau, dan pada akhirnya penulis berharap agar laporan Kerja Praktik ini dapat bermanfaat dan berguna sebagaimana fungsinya.

|  |  |
| --- | --- |
|  | Bekasi, 24 April 2020 |
|  | MELVIN DICKY RUDOLF BENJAMIN  Nim 311610234 |

# **DAFTAR ISI**

[**LEMBAR PERSETUJUAN** i](#_Toc102742654)

[**LEMBAR PENGESAHAN** ii](#_Toc102742655)

[**UCAPAN TERIMAKASIH** iii](#_Toc102742656)

[**DAFTAR ISI** v](#_Toc102742657)

[**DAFTAR GAMBAR** vii](#_Toc102742658)

[**DAFTAR TABEL** viii](#_Toc102742659)

[**DAFTAR LAMPIRAN** ix](#_Toc102742660)

**BAB I :** [**PENDAHULUAN** 1](#_Toc102742662)

[**1.1.** **Latar Belakang** 1](#_Toc102742663)

[**1.2.** **Permasalahan dan Batasan** 2](#_Toc102742664)

[**1.3.** **Tujuan dan Manfaat Kerja Praktek** 2](#_Toc102742665)

**BAB II :** [**TINJAUAN PUSTAKA DAN LANDASAN TEORI** 4](#_Toc102742667)

[**2.1.** **Tinjauan Penelitian** 4](#_Toc102742668)

[**2.2.** **Tinjauan Pustaka** 11](#_Toc102742669)

[*2.2.1* *Software Defined Network (SDN)* 11](#_Toc102742670)

[*2.2.2.* *OpenDaylight* 13](#_Toc102742671)

[*2.2.3.* *Topologi Jaringan* 13](#_Toc102742672)

[*2.2.4.* *Topologi Pohon* 13](#_Toc102742673)

[*2.2.5.* *Topologi Cincin* 14](#_Toc102742674)

[*2.2.6.* *Topologi Bintang* 15](#_Toc102742675)

[*2.2.7.* *Topologi Bus* 15](#_Toc102742676)

[*2.2.8.* *Topologi Jala* 16](#_Toc102742677)

[*2.2.9.* *Jack Terdaftar (Registered Jack)* 17](#_Toc102742678)

[*2.2.10.* *Standar Penamaan* 18](#_Toc102742679)

[*2.2.11.* *Jenis Jack Terdaftar* 18](#_Toc102742680)

[*2.2.12.* *Kabel RJ11, RJ14, RJ25* 22](#_Toc102742681)

[*2.2.13.* *RJ21* 25](#_Toc102742682)

[*2.2.14.* *RJ45S* 28](#_Toc102742683)

[*2.2.15.* *RJ48* 29](#_Toc102742684)

[*2.2.16.* *RJ61* 30](#_Toc102742685)

[*2.2.17.* *Jack Serupa dan nama tidak resmi* 31](#_Toc102742686)

[**2.3.** **Bahasa Pemograman** 32](#_Toc102742687)

[*2.3.1.* *JAVA* 32](#_Toc102742688)

[**2.4.** **Perancangan Sistem** 32](#_Toc102742689)

[*2.4.1.* *Uml* 32](#_Toc102742690)

[*2.4.2.* *Use Case Diagram* 33](#_Toc102742691)

[*2.4.3.* *Activity Diagram* 35](#_Toc102742692)

[*2.4.4.* *Sequence Diagram* 36](#_Toc102742693)

[*2.4.5.* *Class Diagram* 38](#_Toc102742694)

**BAB III :** [**HASIL DAN PEMBAHASAN** 39](#_Toc102742696)

[**3.1.** **Sejarah Organisasi Tempat Kerja Praktik** 39](#_Toc102742697)

[**3.2.** **Visi dan Misi DISKOMINFOSANTIK** 39](#_Toc102742698)

[*3.2.1.* *Visi* 39](#_Toc102742699)

[*3.2.2.* *Misi* 40](#_Toc102742700)

[**3.3.** **Struktur Organisasi DISKOMINFOSANTIK** 40](#_Toc102742701)

[**3.4.** **Analisis Sistem yang Sedang Berjalan** 41](#_Toc102742702)

[*3.4.1.* *Permasalahan* 41](#_Toc102742703)

# **DAFTAR GAMBAR**

[**Gambar 2.1.1. Topologi Small/Home office** 5](#_Toc102745430)

[**Gambar 2.1.2. Mengaktifkan Controller pox** 5](#_Toc102745431)

[**Gambar 2.1.3. Mengaktifkan Mininet** 6](#_Toc102745432)

[**Gambar 2.1.4. Ping Host 1 ke Host 2** 7](#_Toc102745433)

[**Gambar 2.1.5. TCP DUMP** 7](#_Toc102745434)

[**Gambar 2.2.1. Contoh Topologi Pohon** 14](#_Toc102745435)

[**Gambar 2.2.2. Contoh Topologi Cincin** 14](#_Toc102745436)

[**Gambar 2.2.3. Contoh Topologi Bintang** 15](#_Toc102745437)

[**Gambar 2.2.4. Contoh Topologi Bus** 16](#_Toc102745438)

[**Gambar 2.2.5. Contoh Topologi Jala** 16](#_Toc102745439)

[**Gambar 2.2.6. Konektor RJ21** 17](#_Toc102745440)

[**Gambar 2.2.7. Konektor gaya crimp-on 6P4C yang biasa digunakan untuk RJ11 dan RJ14** 19](#_Toc102745441)

[**Gambar 2.2.8. konektor 6P6C menunjukkan lokasi pin 1** 23](#_Toc102745443)

[**Gambar 2.2.9. Konektor RJ21 perempuan** 26](#_Toc102745445)

[**Gambar 2.2.10. Konektor perempuan dengan kunci 8P8C (jack).** 28](#_Toc102745447)

[**Gambar 3.3.1 Struktur Organisasi** 40](#_Toc102745454)

# **DAFTAR TABEL**

[**Tabel 1.3.1. Tabel Jadwal Kegiatan Kerja Praktik** 3](#_Toc102745482)

[**Tabel 2.2.1. Jenis jack terdaftar yang diakui secara resmi** 19](#_Toc102745495)

[**Tabel 2.2.2. Tabel Pinout** 23](#_Toc102745497)

[**Tabel 2.2.3. Kode warna hitung genap** 26](#_Toc102745499)

[**Tabel 2.2.4. Kabel RJ48C dan RJ48X** 29](#_Toc102745501)

[**Tabel 2.2.5. Kabel RJ61 (USOC)** 30](#_Toc102745502)

[**Tabel 2.2.6. Use Case Diagram** 33](#_Toc102745503)

[**Tabel 2.2.7. Activity Diagram** 35](#_Toc102745504)

[**Tabel 2.2.8. Sequence Diagram** 37](#_Toc102745505)

[**Tabel 2.2.9. Multiplicity Class Diagram** 38](#_Toc102745506)

# **DAFTAR LAMPIRAN**

# **BAB I**

# **PENDAHULUAN**

## **Latar Belakang**

Layanan internet dan teknologi telah berkembang dengan berbagai kompleksitas, desain, manajemen dan operasional yang menyebabkan jumlah perangkat yang terhubung serta jumlah trafik dalam jaringan telah meningkat dengan sangat pesat dalam beberapa tahun terakhir. Selain itu, munculnya dan berkembangnya layanan baru dan penggunaan mobilitas yang tinggi, virtualisasi server, *cloud computing* dan *internet of thing* (IOT) yang sedang dikembang menyebabkan arsitektur jaringan yang ada sekarang sangatlah terbatas kemampuannya. Karena keterbatasan ini, maka industry jaringan mencapai satu titik tujuan untuk meresponnya dengan diciptakan sebuah arsitektur baru yang dikenal dengan arsitektur *Software-Defined Network*.

Dinas Komunikasi dan Informatika (DISKOMINFO) merupakan instansi yang bergerak di bidang komunikasi dan informatika yang meliputi telekomunikasi, sarana komunikasi dan diseminasi informasi, telematika, serta pengolahan data elektronik. Dinas Komunikasi dan Informatika (DISKOMINFO) juga mempunyai tugas mengkoordinasikan dan pembinaan Organisasi Perangkat Daerah (OPD) dan melaksanakan tugas lain dari Gubernur sesuai tugas dan fungsinya.

Dinas Komunikasi dan Informatika dalam pengolahan data elektronik telah menggunakan teknologi telekomunikasi berupa Jaringan Komputer dalam mengkomunikasikan antar Organisasi Perangkat Daerah (OPD) satu dengan lainnya yang terdapat di Propinsi Jawa Barat. Dengan menggunakan Jaringan Komputer, tugas-tugas komputasi dari berbagai daerah Jawa Barat dapat saling berkomunikasi dalam melaksanakan tugas-tugas tersebut.

Software Defined Network (SDN) merupakan sebuah konsep pendekatan baru untuk mendesain, mengelola dan mengimplementasikan arsitektur jaringan yang memisahkan antara sistem control (control plane) dan sistem forwarding (data plane) pada perangkat jaringan. Software Defined Network (SDN) memungkinkan suatu jaringan dapat secara dinamis menyesuaikan lingkungan untuk menyederhanakan manajemen dan meningkatkan skalabilitas jaringan yang diwujudkan melalui implementasi sederhana dari penambahan komponen dan layanan jaringan.

Berdasarkan uraian diatas penulisan dimana pada kesempatan kali ini penulis akan membahas topik tentang *“Perancangan Software Defined Network Menggunakan Kontroler Opendaylight Pada Jaringan Diskominfo Pemda Kab. Bekasi”*.

## **Permasalahan dan Batasan**

Berdasarkan latar belakang penelitian diatas, maka identifikasi masalah pada penelitian ini adalah sebagai berikut :

1. Banyaknya jumlah perangkat yang terhubung serta banyaknya jumlah trafik dalam jaringan yang meningkat
2. Arsitektur jaringan yang ada sekarang sangatlah terbatas kemampuannya

Dari identifikasi masalah yang dijabarkan diatas, maka penelitian ini fokus pada Perancangan Software Defined Network Menggunakan Kontroler Opendaylight Pada Jaringan Diskominfo Pemda Kab. Bekasi.

Berkaitan dengan latar belakang diatas, rumusan masalah yang bisa diambil adalah bagaimana Software Defined Network meningkatkan jaringan yang diwujudkan melalui implementasi penambahan komponen ?

## **Tujuan dan Manfaat Kerja Praktek**

Adapun tujuan dari penelitian ini adalah menganalisis skalabilitas jaringan Diskominfo Pemda Kab. Bekasi.

Sedangkan manfaatnya dari tujuan penelitian tersebut adalah dapat meningkatkan skalabilitas jaringan di DISKOMINFO

Adapun nama kegiatan yang penulis lakukan yaitu Kerja Praktek yang dilaksanakan pada :

Tanggal Kerja : 22 Juli 2019 s/d 23 Agustus 2019

Hari Kerja : Senin s/d Jumat

Waktu Kerja : 7 jam

Tempat Kerja :Dinas Komunikasi Informasi Persandian dan Statistik (Diskominfo) Kab. Bekasi

Jadwal kegiatan tersebut adalah kegiatan yang berlangsung selama kerja praktik di kantor Diskominfo Kab. Bekasi 1 bulan kebelakang.

#### **Tabel 1.3.1. Tabel Jadwal Kegiatan Kerja Praktik**

|  |  |  |
| --- | --- | --- |
| **Minggu ke** | **Tanggal** | **Jenis kegiatan** |
| **I** | 22 Juli | Mencari informasi dan menganalisa |
| Sampai Dengan | apa itu Network Monitoring |
| 26 Juli | Sistem |
| **II** | 29 Juli | Mencari Permasalahan |
| Sampai Dengan | Network Monitoring Sistem |
| 2 Agustus | di Diskominfo Kab. Bekasi |
| **III** | 5 Agustus | Mencari informasi dan |
| Sampai Dengan | menganalisa Software |
| 9 Agustus | Defined Network |
| **IV** | 12 Agustus | Menginstall aplikasi |
| Sampai Dengan | Opendaylight di Virtual |
| 16 Agustus | Box |
| **V** | 19 Agustus | Membuat topologi |
| Sampai Dengan | Menggunakan aplikasi |
| 23 Agustus | Opendaylight di Virtual Box |

# 

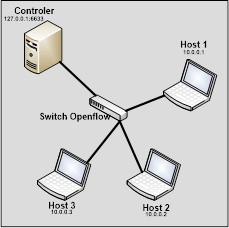
# **BAB II**

# **TINJAUAN PUSTAKA DAN LANDASAN TEORI**

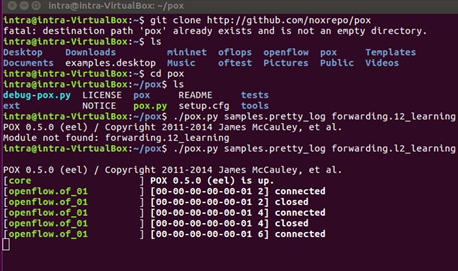
## **Tinjauan Penelitian**

*Arsitektur Software Defined Network: Implementasi Pada Small Network*, 2020, Software defined network masih terus dikembangkan dan dapat diadopsi dengan berbagai cara. Salah satu cara untuk memanfaatkannya adalah dengan menggunakan arsitektur SDN sebagai pengganti arsitektur jaringan tradisional. Dengan menggunakan arsitektur SDN, para pengguna jaringan Small office Home office dapat melakukan manajemen terhadap jaringan sehingga memiliki jaringan yang optimal karena disesuaikan dengan kebutuhan pengguna masing-masing. Penggunaan arsitektur SDN juga dapat meningkatkan keamanan dalam jaringan tersebut tanpa harus menggunakan berbagai perangkat keras jaringan lainnya sehingga dapat mengurangi biaya yang ditanggung oleh pengguna jaringan Small office Home office.

Simulasi jaringan SDN untuk Untuk Small/Home office pada system operasi linux Ubuntu versi 16.04.5 dengan menggunakan mininet dan pox sebagai controller. Berikut ini adalah topologi dari jaringan Small/home office yang dilakukan. Dari gambar 8 diatas dapat dilihat terdapat 3 host, 1 switch dan 1 buah controller, dimana controller tersebut menggunakan pox, ip range dari ketiga host tersebut adalah 10.0.0.1 sampai dengan 10.0.0.3 sedangkan ip yang digunakan oleh controller adalah 127.0.0.1:6633 karena controller tersebut berjalan pada local host. Langkah pertama dalam melakukan simulasi adalah dengan cara melakukan instalasi pox apabila dalam system operasi yang digunakan belum terinstall pox, setelah pox terinstal langkah selanjutnya adalah mengaktifkan controller pox tersebut.



#### **Gambar 2.1.1. Topologi Small/Home office**

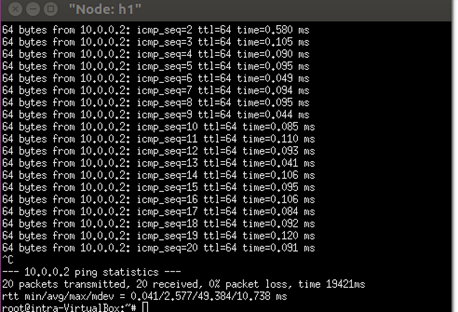


#### **Gambar 2.1.2. Mengaktifkan Controller pox**

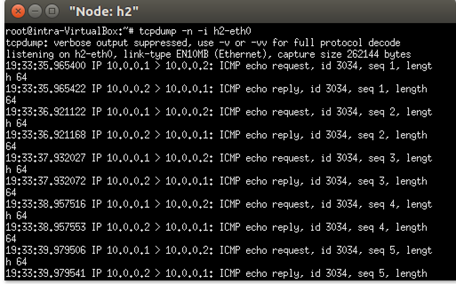
Setelah mengaktifkan controller pox langkah selanjutnya adalah melakukan installasi mininet, dan selanjutnya adalah membuat topologi jaringan yang akan digunakan seperti pada Gambar 2.1.2. Topologi ini dibuat dengan script python. Gambar dibawah ini menunjukan topologi yang sudah dibuat dan dijalankan dalam mininet. Pada gambar 2.1.3. dibawah dapat dilihat seluruh host switch dan controller telah terhubung, hal ini dapat dilihat pada saat dilakukan test ping keseluruh host. Setelah melakukan ping test ke 3 host tersebut yaitu host 1,2 dan 3 terdapat keterangan result:0% dropped (6/6 received). Pada Gambar 11 adalah gambar poses ping yang dilakukan pada Host 1(10.0.0.1) pada Host 2 (10.0.0.2), di gambar tersebut terlihat ada tulisan 0% packet loss yang berarti tidak ada data yang hilang atau terdapat koneksi yang terputus pada saat proses ping, hal ini berarti Host 1 dan Host 2 sudah terhubung. Gambar 2.1.4. dibawah adalah tampilan pada saat Host 2 Menjalankan TCP DUMP bersamaan pada saat Host 1 melakukan Ping ke Host 2, TCP DUMP sendiri adalah sebuah packet analyzer yang berfungsi untuk menangkap atau memantau traffic jaringan.

****

#### **Gambar 2.1.3. Mengaktifkan Mininet**

****

#### **Gambar 2.1.4. Ping Host 1 ke Host 2**

****

#### **Gambar 2.1.5. TCP DUMP**

*ANALISIS EFEK PENGGUNAAN KONTROLER FLOODLIGHT DAN OPENDAYLIGHT PADA PERFORMANSI JARINGAN SDN*, 2018,

1. Pada pengujian performansi kontroler Floodlight tanpa algoritma Johnson, Floodlight dengan algoritma Johnson dan OpenDaylight tanpa algoritma Johnson mempelihatkan perubahan jumlah switch dari 6 switch, 8 switch dan 10 switch berbanding lurus dengan perubahan nilai QoS terutama pada nilai delay dan jitter. Kontroler Floodlight tanpa algoritma Johnson lebih unggul dibandingkan kedua kontroler lainnya dari untuk layanan data dan VoIP. Ini terlihat dari jumlah delay pada layanan data dan VoIP. Adapun nilai delay maksimal pada model kontroler ini adalah 88.38ms. Namun untuk pengiriman layanan video, kontroler OpenDaylight lebih baik dari yang lainnya. Karena hanya kontroler OpenDaylight yang bisa mengirimkan paket video. Adapun nilai delay maksimal untuk pengiriman layanan video adalah 1195ms.
2. Untuk peningkatan background traffic yang berkisar dari 0-80Mbps, secara umum kontroler OpenDaylight memiliki perubahan nilai delay dan jitter yang berbanding lurus sesuai kenaikkan background traffic-nya untuk semua layanan. Perubahan nilai delay dan jitter untuk semua layanan sering terjadi pada background traffic yang bernilai di 60-80Mbps. Pada kontroler floodlight tidak terjadi perubahan seperti ini.
3. Untuk hasil pengujian resource utilization, konsumsi memori tertinggi dimiliki oleh kontroler OpenDaylight. OpenDaylight mengkonsumsi memori sebanyak 782MB (19.1% ) dari total memori virtual machine yaitu 4094MB. Lalu diikuti oleh Floodlight dengan algoritma Johnson sebanyak 262MB (6.4%) dan terakhir Floodlight tanpa algoritma johnson sebanyak 245MB (6%).
4. Berdasarkan standarisasi ITU-T G.1010 kontroler Floodlight tanpa algoritma Johnson, Floodlight menggunakan algoritma Johnson dan OpenDaylight hanya memenuhi standarisasi delay untuk layanan data saja. Untuk nilai jitter dan throughput-nya tidak memenuhi standar disemua layanan. Adapun nilai delay maksimal untuk layanan data pada tiga kontroler tersebut yaitu 88ms, 82ms dan 636ms dimana itu belum melebihi 60s.

Pengujian QoS (*Quality of Service*) pada layanan data secara umum kontroler floodlight tanpa algoritma johnson cenderung mengalami sedikit penurunan delay ketika jumlah background traffic ditambahkan . Walau terjadi sedikit fluktuasi dari bandwith 0Mbps ke 20Mbps, namun ini memperlihatkan karakteriskiknya. Hal yang sama tidak berlaku untuk kontroler floodlight dengan algoritma johnson. Pada hasil pengujian delay menggunakan layanan data diatas,peningkatan jumlah switch mempengaruhi nilai delay data. Untuk controller OpenDaylight kenaikan jumlah switch dan trafik berbanding lurus dengan delay datanya. Setiap terjadi penambahan switch, nilai delay meningkat dan ketika terjadi kenaikan nilai trafik delay-nya cendrung naik. Perubahan yang agak signifikan terlihat ketika kontroler floodlight berada pada trafik 60Mbps ke 80Mbps. Itu terjadi kaerna bandwith link sudah mendekati ambang. Jika kita menggunakan referensi standarisasi QoS ITU.T G.1010, tiga kontroler diatas memenuhi standarnya, karena nilai delay-nya tidak lebih dari 60 second. Nilai delay terkecil dimiliki oleh kontroler floodlight tanpa algoritma johnson.

Untuk layanan VoIP, kontroler floodlight tanpa algoritma johnson mengalami kenaikan delay ketika jumlah host nertambah. Semakin bertambah jumlah host, delay-nya akan naik. Untuk perubahan nilai background traffic tidak terlalu mempengaruhi delay-nya. Begitu juga yang terjadi pada kontroler floodlight ketika ditambahkan algoritma johnson dan OpenDaylight tanpa algoritma johnson. Terjadi kenaikan delay ketika jumlah host bertambah. Kontroler OpenDaylight mengalami cendrung mengalami peningkatan delay ketika background traffic-nya meningkat. Berdasarkan standarisasi QoS ITU.T G.1010. kontroler OpenDaylight tidak memenuhi standar ini. Karena delay-nya ada yang menyentuh nilai diatas 400ms.

Pengujian delay pada layanan video, hanya bisa dilakukan pada kontroler OpenDaylight. Karena floodlight tidak bisa mengirim packet dalam jumlah besar. Ini bug yang masih belum diperbaiki oleh developer kontroler floodlight sekarang.

Berdasarkan hasil pengujian, penambahan jumlah host berdampak pada kenaikan nilai delay. Jika jumlah host bertambah, maka nilai delay akan naik. Sorotan khusus pada kontroler OpenDaylight ketika memiliki background traffic 60Mbps dan 80Mbps. Topologi dengan model 10 switch tidak bisa mengirimkan data video karena mengalami hang pada PC yang digunakan. Ini terjadi karena background traffic yang sudah mendekati bandwith ambang pada link dan jumlah switch yang banyak dengan ukuran data yang besar. Jadi terjadilah keantrian dalam mengirim data, selama antrian juga akan memakan resource lebih banyak dan menyebabkan PC menjadi hang dan menyebabkan pengiriman data tidak bisa diteruskan.

Berdasarkan hasil pengujian diatas, nilai jitter pada layanan data meningkat seiring bertambahnya jumlah host. Perubahan jitter tiap perubahan background traffic-nya tidak terlalu signifikan. Namun nilai jitter terkecil dipegang oleh kontroler floodlight tanpa algoritma johnson. Berdasarkan standarisasi QoS ITU.T G.1010. ketiga kontroler ini tidak memenuhi standar. Karena jitter-nya melebihi angka 1ms.

Pada pengujian jitter untuk layanan VoIP, secara umum ketiga model kontroler mengalamami kenaikan nilai jitter saat terjadi penambahan jumlah host. Jika dilihat dari standarisasi QoS ITU.T G.1010, tidak ada satupun kontroler yang memenuhi standarisasi tersebut. Karena nilai jitternya lebih dari 1ms. Kontroler floodlight tanpa algoritma johnson lebih baik dalam nilai jitternya karena kontroler ini memiliki nilai jitter terkecil diantara dua kontroler lainnya.

Untuk pengujian jitter pada layanan video hanya bisa dilakukan pada kontroler opendaylight, karena kontroler floodlight tidak bisa meneruskan paket dengan ukuran besar. Jika diteruskan, nilai keterangan jitternya akan memiliki keluaran nan. Ini bug yang belum diperbaiki oleh developer floodlight. Untuk kontroler opendaylight perubahan jumlah host mempengaruhi kenaikan nilai jitter. Semakin bertambah jumlah host maka jitternya akan bertambah. Sama seperti pada layanan VoIP, terjadi hang pada komputer percobaan saat dilakukan pengujian pada model topologi 10 switch dengan bandwith 60Mbps dan 80Mbps. Ini terjadi karena background traffic yang sudah mendekati bandwith ambang pada link dan jumlah switch yang banyak dengan ukuran data yang besar. Jadi terjadilah keantrian dalam mengirim data, selama antrian juga akan memakan resource dan menyebabkan PC menjadi hang dan menyebabkan pengiriman data tidak bisa diteruskan.

Untuk perubahan nilai throughput pada layanan data, VoIP dan video di ketiga kontroler tersebut masih konstan. Secara umum, perubahan jumlah host dan kenaikan background traffic tidak mempengaruhi nilai thorughputnya. Hal ini dikarenakan besar background traffic masih dibawah besar maksimal yang disediakan link. Berdasarkan standarisasi QoS ITU.T G.1010 tidak ada kontroler yang memenuhi standar tersebut karena nilai throughput-nya lebih dari 64kbps. Semua data hasil pengujian QoS tertera pada lampiran.

Tujuan dilakukannya pengujian resource utilization adalah untuk mengetuhui seberapa besar konsumsi memori yang digunakan oleh kontroler ketika belum memiliki Algoritma Johson dan ketika sudah memiliki Algrotima Johnson didalamnya. Berdasarkan hasil pengujian, secara umum terjadi peningkatan konsumsi memori pada masing-masing kontroler pada setiap peningkatan jumlah switch. Kenaikan jumlah switch berbanding lurus dengan konsumsi memori oleh kontroler/ Kontroler OpenDaylight lebih terbilang paling banyak memakan memori sebesar 782.336MB pada jumlah switch 10. Semua data hasil pengujian resource utilization tertera pada lampiran.

## **Tinjauan Pustaka**

### *Software Defined Network (SDN)*

*Software Defined Network (SDN)* adalah sebuah paradigma arsitektur baru dalam bidang jaringan komputer, yang memiliki karakteristik dinamis, manageable, cost-effective, dan adaptable, sehingga sangat ideal untuk kebutuhan aplikasi saat ini yang bersifat dinamis. Arsitektur ini memisahkan antara network 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 network services, (Ummah, Izzatul, Desianto A, 2016).

Dari penelitian yang dilakukan oleh Muhammad Hikam Hidayat, Nur Rohman Rosyid (2017) dengan judul “Analisis Kinerja dan Karakteristik Arsitektur Software-Defined Network Berbasis OpenDaylight Controller” Layanan internet dan teknologi telah berkembang yang menyebabkan jumlah perangkat yang terhubung serta jumlah trafik dalam jaringan telah meningkat dengan sangat pesat. Karena peningkatan ini, maka industri jaringan mencapai satu titik tujuan untuk meresponnya dengan diciptakan sebuah arsitektur baru yang dikenal dengan arsitektur Sofware Defined Network. Software Defined Network (SDN) merupakan sebuah konsep pendekatan baru untuk mendesain, mengelola, dan mengimplementasikan arsitektur jaringan dimana aliran data (data flows) dari sistem kontrol dipisahkan dari hardware. Pada penelitian ini dilakukan analisis kinerja dan karakteristik arsitektur SDN. OpenDaylight digunakan sebagai controller untuk menangani jaringan yang dibuat. Arsitektur SDN diterapkan pada perangkat MikroTik untuk diamati kinerja dan karakteristik yang dimiliki oleh jaringan tersebut kemudian dibandingkan dengan arsitektur tradisional.

Sedangkan dari penelitian ini dilakukan oleh Kartadie Rikkie, Utami Ema, Pramono Eko dengan Berjudul “Prototipe Infrastruktur Software Defined Network Dengan Protokol Openflow menggunakan Ubuntu Sebagai Kontroller”. OpenFlow dapat diimplementasikan pada NetFTPGA, digunakan sebagai firewall, dan bisa dikendalikan oleh wirehark perbedaannya Dasar dalam penelitian ini adalah perangkat penelitian, penelitian ini menggunakan tombol langsung OpenWrt terkait dengan controller, jadi ketika itu terjadi Keberhasilan implementasi infrastruktur jaringan bukan biaya yang besar, karena tidak ada perubahan mendasar dalam peralatan infrastruktur yang sudah ada Selain itu, Chunk Yin memantau hasil penelitian mereka menggunakan protokol wirehark dan monitoring TCP, sedangkan dalam penelitian ini, menggunakan wirechark, tapi yang dikendalikan oleh protocol sebuah percobaan tentang prototyping SDN dengan OpenFlow, Tapi majalah ini hanya membahas mininet dan penggunaannya. Mininet simulator sistem untuk prototipe jaringan besar dalam keterbatasan sumber daya laptop. Dalam hal ini, dibandingkan dengan penelitian yang naik dari jenis prototip menjadi Studi ini menyoroti prototipe yang dibangun menggunakan switch (sebagai perangkat keras), jadi responnya diharapkan pertanyaan apakah OpenFlow bisa bekerja tanpa menggunakan switch khusus untuk OpenFlow, Sementara dalam penelitiannya. Menekankan simulator mininet dalam skalabilitas.

### *OpenDaylight*

OpenDaylight merupakan sebuah kontroler SDN yang berlisensi terbuka atau opensource dan menggunakan bahasa pemrograman Java dan dikelola Linux Foundation dan didukung oleh lebih dari 40 perusahaan seperti IBM, Cisco, Juniper, VMWare, dan sejumlah vendor jaringan besar (Asadollahi et al., 2017).

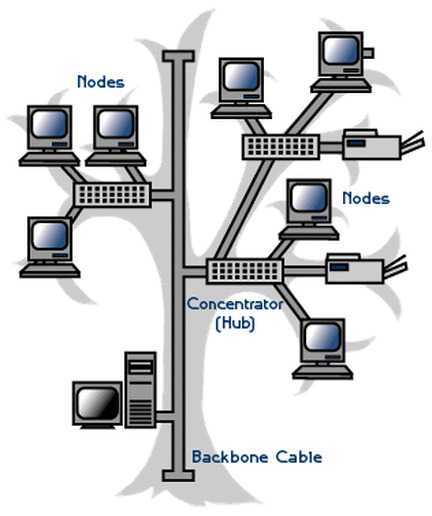
### *Topologi Jaringan*

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. (Iwan Sofana, 2008:7)

Topologi jaringan komputer adalah suatu cara menghubungkan komputer yang satu dengan komputer lainnya sehingga membentuk jaringan. Dalam suatu jaringan komputer jenis topologi yang dipilih akan mempengaruhi kecepatan komunikasi. Untuk itu maka perlu dicermati kelebihan/keuntungan dan kekurangan/kerugian dari masing-masing topologi berdasarkan karakteristiknya.

### *Topologi Pohon*

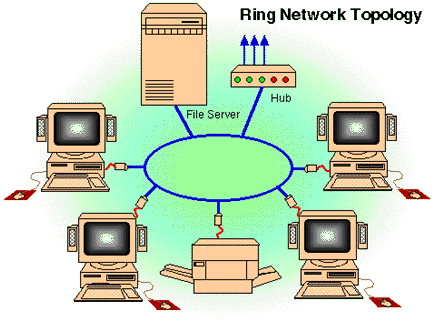
Topologi pohon merupakan topologi yang bisa digunakan pada jaringan didalam ruangan kantor yang bertingkat. Pada jaringan pohon, terdapat beberapa tingkatan simpul (node). Keunggulan jaringan model pohon adalah dapat membentuk suatu kelompok yang dibutuhkan pada setiap saat.



#### **Gambar 2.2.1. Contoh Topologi Pohon**

### *Topologi Cincin*

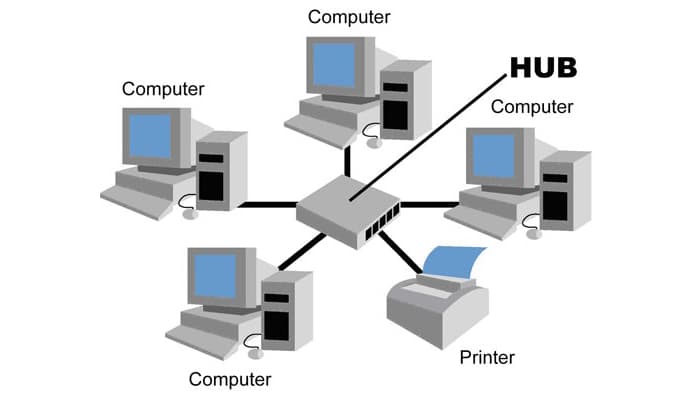
Topologi cincin adalah topologi jaringan disetiap komputer yang terhubung akan membuat lingkaran. Adapun kelebihan dari topologi ini adalah kabel yang digunakan bisa lebih dihemat.



#### **Gambar 2.2.2. Contoh Topologi Cincin**

### *Topologi Bintang*

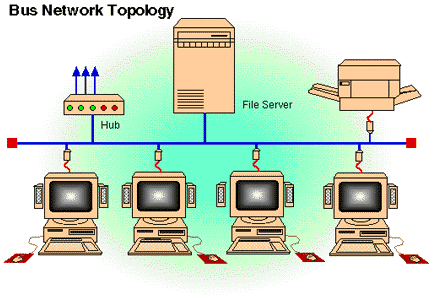
Topologi bintang atau yang lebih sering disebut dengan star topology sudah menggunakan bantuan alat lain untuk mengkoneksikan jaringan komputer. Alat yang dipakai disini adalah hub atau switch. Keuntungan dari topologi ini dapat memudahkan admin dalam mengelola jaringan, dapat memudahkan dalam penambahan komputer atau terminal serta memudahkan mendeteksi kerusakan dan kesalahan jaringan.



#### **Gambar 2.2.3. Contoh Topologi Bintang**

### *Topologi Bus*

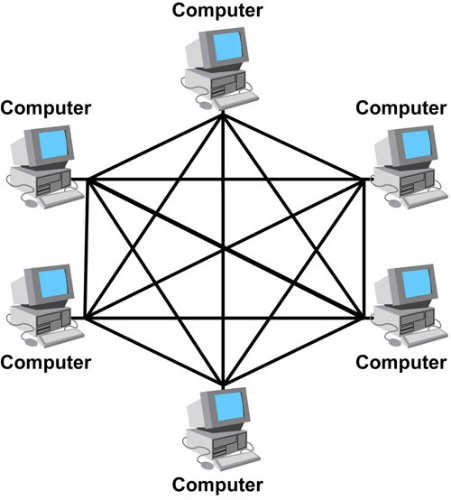
Topologi ini adalah topologi awal yang digunakan untuk menghubungkan komputer. Dalam topologi ini, masing-masing komputer akan terhubung ke satu kebel panjang dengan beberapa terminal dan pada akhir dari kabel harus diakhiri dengan satu terminator. Keuntungan dari topologi bus adalah pengembangan jaringan atau penambahan workstation baru dapat dilakukan dengan mudah tanpa mengganggu workstation lain.



#### **Gambar 2.2.4. Contoh Topologi Bus**

### *Topologi Jala*

Topologi jala atau Topologi mesh adalah suatu bentuk hubungan antar perangkat dimana setiap perangkat terhubung secara langsung ke perangkat lainnya yang ada di dalam jaringan. Akibatnya, dalam topologi mesh setiap perangkat dapat berkomunikasi langsung dengan perangkat yang dituju (dedicated links). Dengan demikian maksimal banyaknya koneksi antar perangkat pada jaringan bertopologi mesh ini dapat dihitung yaitu sebanyak n(n-1)/2.



#### **Gambar 2.2.5. Contoh Topologi Jala**

### *Jack Terdaftar (Registered Jack)*

Sebuah jack terdaftar (Registered Jack) adalah standar telekomunikasi antarmuka jaringan untuk menghubungkan suara dan data peralatan untuk layanan yang disediakan oleh operator pertukaran lokal atau operator jarak jauh . Antarmuka pendaftaran pertama kali ditentukan dalam sistem Kode Pemesanan Layanan Universal (Universal Service Ordering Code) Sistem Bel di Amerika Serikat untuk mematuhi program pendaftaran untuk peralatan telepon yang disediakan pelanggan yang diamanatkan oleh Federal Communications Commission (FCC) pada tahun 1970-an.

Konektor yang digunakan untuk pemasangan jack terdaftar terutama adalah konektor modular dan konektor pita miniatur 50-pin. Misalnya, RJ11 menggunakan konektor six-position two-conductor (6P2C), RJ14 menggunakan jack modular six-position four-conductor (6P4C), sedangkan RJ21 menggunakan konektor pita miniatur 25 pasang (50 pin).



#### **Gambar 2.2.6. Konektor RJ21**

Konektor modular kiri ke kanan:

* Colokan 8P8C delapan kontak yang digunakan untuk RJ45S, RJ49, RJ61 dan lainnya.
* Colokan 6P6C enam kontak yang digunakan untuk RJ25
* konektor 6P4C empat kontak yang digunakan untuk RJ14
* steker handset 4P4C empat kontak
* Jack 6P6C, dapat digunakan untuk RJ11, RJ14, dan RJ25.

### *Standar Penamaan*

Penunjukan jack terdaftar berasal dari proses standardisasi di Bell System di Amerika Serikat, dan menjelaskan rangkaian aplikasi dan bukan hanya geometri fisik konektor; pemeriksaan konektor tidak selalu menunjukkan kabel jack terdaftar mana yang digunakan. Jenis konektor modular yang sama dapat digunakan untuk aplikasi jack terdaftar yang berbeda.

Ada banyak kebingungan atas standar koneksi ini. Steker dan soket enam posisi yang sama yang biasa digunakan untuk sambungan saluran telepon dapat digunakan untuk RJ11, RJ14 atau bahkan RJ25, yang semuanya merupakan nama standar antarmuka yang menggunakan konektor fisik ini. Standar RJ11 menentukan koneksi pasangan kabel tunggal, sedangkan RJ14 adalah konfigurasi untuk dua saluran, dan RJ25 menggunakan keenam kabel untuk tiga saluran telepon. Namun, sebutan RJ hanya terkait dengan kabel jack, maka namanya Jack Terdaftar; itu biasa, tetapi tidak sepenuhnya benar, untuk merujuk ke steker yang tidak diinginkan dengan salah satu nama ini.

### *Jenis Jack Terdaftar*

Jack terdaftar yang paling banyak diterapkan dalam telekomunikasi adalah RJ11. Ini adalah konektor modular yang dikabelkan untuk satu saluran telepon, menggunakan dua kontak tengah dari enam posisi yang tersedia. Konfigurasi ini juga digunakan untuk telepon jalur tunggal di banyak negara selain Amerika Serikat. Ini juga dapat menggunakan konektor 6P4C, untuk menggunakan pasangan kabel tambahan untuk menyalakan lampu pada pesawat telepon. RJ14 mirip dengan RJ11, tetapi memiliki kabel untuk dua jalur dan RJ25 memiliki tiga jalur. RJ61 adalah soket terdaftar serupa untuk empat saluran, tetapi menggunakan konektor 8P8C.

Soket RJ45 (S) jarang digunakan dalam aplikasi telepon tetapi versi standar konektor modular 8P8C yang digunakan oleh RJ45 digunakan dalam jaringan Ethernet dan dalam konteks ini konektor sering disebut sebagai RJ45.



#### **Gambar 2.2.7. Konektor gaya crimp-on 6P4C yang biasa digunakan untuk RJ11 dan RJ14**

#### **Tabel 2.2.1. Jenis jack terdaftar yang diakui secara resmi**

|  |  |  |
| --- | --- | --- |
| **Kode** | **Penyambung** | **Pemakaian** |
| RJA1X | Adaptor 225A | Konektor untuk steker modular ke soket empat cabang |
| RJA2X | Adaptor 267A | Konektor untuk memisahkan satu jack modular menjadi dua jack modular |
| RJA3X | Adaptor 224A | Konektor untuk mengadaptasi steker modular ke soket 12-cabang |
| RJ2MB | 50-pin | 2–12 saluran telepon dengan pengaturan buat sibuk |
| RJ11 (C/W) | 6P2C | Membuat koneksi yang dijembatani untuk satu saluran telepon (6P4C jika daya pada pasangan kedua) |
| RJ12 (C/W) | 6P6C | Menetapkan koneksi yang dijembatani untuk satu saluran telepon dengan kontrol sistem telepon utama di depan rangkaian saluran |
| RJ13 (C/W) | 6P4C | Mirip dengan RJ12, tetapi di belakang rangkaian saluran |
| RJ14 (C/W) | 6P4C | Untuk dua saluran telepon (6P6C jika dihidupkan pada pasangan ketiga) |
| RJ15C | 3-pin tahan cuaca | Untuk satu saluran telepon untuk kapal di marina |
| RJ18 (C/W) | 6P6C | Untuk satu saluran telepon dengan pengaturan buat sibuk |
| RJ21X | 50-pin | Beberapa (hingga 25) jalur yang menjembatani konfigurasi T / R |
| RJ25 (C/W) | 6P6C | Untuk tiga saluran telepon |
| RJ26X | 50-pin | Untuk beberapa jalur data, universal |
| RJ27X | 50-pin | Untuk beberapa jalur data, diprogram |
| RJ31X | 8P8C | Mengizinkan sistem alarm menggunakan saluran telepon untuk melakukan panggilan keluar selama alarm. Jack ditempatkan lebih dekat ke antarmuka jaringan daripada semua peralatan lainnya. Hanya 4 konduktor yang digunakan. |
| RJ32X | 8P8C | Seperti RJ31X, kabel ini menyediakan ujung seri dan koneksi cincin melalui blok penghubung, tetapi digunakan ketika peralatan di tempat pelanggan dihubungkan secara seri dengan satu stasiun, seperti dialer otomatis. |
| RJ33X | 8P8C | Pengkabelan ini menyediakan ujung seri dan koneksi cincin dari jalur KTS di depan sirkuit jalur karena peralatan terdaftar memerlukan dering CO / PBX dan koneksi penghubung dari kabel A dan A1 dari belakang sirkuit jalur. Tip dan cincin adalah satu-satunya kabel yang dibuka saat steker CPE dimasukkan. Penggunaan umum adalah untuk pemanggil otomatis dan pembatas panggilan yang disediakan pelanggan. |
| RJ34X | 8P8C | Mirip dengan RJ33X, tetapi semua kabel dihubungkan di belakang rangkaian saluran. |
| RJ35X | 8P8C | Susunan ini menyediakan ujung seri dan koneksi dering ke saluran apa pun yang telah dipilih dalam perangkat telepon utama ditambah kabel penghubung A dan A1. |
| RJ38X | 8P4C | Mirip dengan RJ31X, dengan sirkuit kontinuitas. Jika steker dicabut dari jack, bar korsleting memungkinkan sirkuit telepon untuk melanjutkan ke telepon situs. Hanya 4 konduktor yang digunakan. |
| RJ41S | 8P8C, dikunci | Untuk satu jalur data, universal (kehilangan loop tetap dan terprogram) |
| RJ45S | 8P8C, dikunci | Untuk satu jalur data, dengan resistor pemrograman |
| RJ48C | 8P4C | Untuk jalur data empat kabel (DSX-1) |
| RJ48S | 8P4C, dikunci | Untuk jalur data empat kabel (DDS) |
| RJ48X | 8P4C dengan shorting bar | Untuk jalur data empat kabel (DS1) |
| RJ49C | 8P8C | Untuk ISDN BRI via NT1 |
| RJ61X | 8P8C | Untuk empat saluran telepon |
| RJ71C | 50-pin | Koneksi seri 12 baris menggunakan konektor 50-pin (dengan adaptor penghubung) di depan peralatan pelanggan. Kebanyakan digunakan untuk peralatan call sequencer. |

Banyak dari nama dasar memiliki sufiks yang menunjukkan subtipe:

* C: flush-mount atau surface mount
* F: pemasangan fleksibel
* W: dipasang di dinding
* L: dudukan lampu
* S: satu baris
* M: multi-garis
* X: jack kompleks

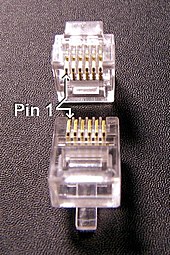
Misalnya, RJ11 hadir dalam dua bentuk: RJ11W adalah soket yang dapat digunakan untuk menggantung telepon dinding, sedangkan RJ11C adalah soket yang dirancang untuk ditancapkan kabel. Kabel juga dapat dipasang ke RJ11W.

### *Kabel RJ11, RJ14, RJ25*

Semua jack terdaftar ini dijelaskan sebagai mengandung sejumlah posisi kontak potensial dan jumlah kontak aktual yang dipasang dalam posisi ini. RJ11, RJ14, dan RJ25 semuanya menggunakan konektor modular enam posisi yang sama , sehingga secara fisik identik kecuali untuk jumlah kontak yang berbeda (masing-masing dua, empat dan enam) yang memungkinkan sambungan untuk satu, dua, atau tiga saluran telepon.

Kabel yang tersambung ke antarmuka RJ11 memerlukan konektor 6P2C. Namun demikian, kabel yang dijual sebagai RJ11 sering kali menggunakan konektor 6P4C (enam posisi, empat konduktor) dengan empat kabel. Dua dari enam kemungkinan posisi kontak menghubungkan ujung dan cincin, dan dua konduktor lainnya tidak digunakan.

Konduktor selain dua ujung pusat dan konduktor cincin dalam praktiknya digunakan secara beragam untuk saluran telepon kedua atau ketiga, arde untuk dering selektif , daya tegangan rendah untuk lampu panggil , atau untuk sirkuit anti-tinkle untuk mencegah telepon panggilan pulsa dari membunyikan bel di ekstensi lain.



#### **Gambar 2.2.8. konektor 6P6C menunjukkan lokasi pin 1**

Mengamati konektor jantan dari bukaan kabel, dengan cabang menghadap ke bawah, pin diberi nomor 1-6, kiri ke kanan:

#### **Tabel 2.2.2. Tabel Pinout**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Posisi** | **Pasangan** | **Tip atau Ring** | **±** | **RJ11** | **RJ14** | **RJ25** | **Warna US Bell System [a]** | **25-pasang kode warna [b]** | **Warna Jerman [c]** | **Warna Australia** | **Warna Belanda** |
| 1 | 3 | T | + |  |  | T3 | atau  putih atau oranye | putih / hijau | ungu | oranye | Tidak digunakan |
| 2 | 2 | T | + |  | T2 | T2 | hitam | putih / oranye | hijau | merah | oranye |
| 3 | 1 | R | - | R1 | R1 | R1 | merah | biru / putih | putih | biru | merah |
| 4 | 1 | T | + | T1 | T1 | T1 | hijau | putih biru | cokelat | putih | biru |
| 5 | 2 | R | - |  | R2 | R2 | kuning | oranye / putih | kuning | hitam | putih |
| 6 | 3 | R | - |  |  | R3 | atau  biru atau coklat | hijau / putih | batu tulis | hijau | Tidak digunakan |

1. Meskipun kode warna solid lama ditetapkan dengan baik untuk pasangan 1 dan biasanya pasangan 2, beberapa konvensi yang bertentangan ada untuk pasangan 3. Warnanya berasal dari vendor kabel konduktor datar 8-satin perak yang diklaim sebagai standar . Setidaknya satu vendor kabel datar 8 konduktor menggunakan urutan biru, oranye, hitam, merah, hijau, kuning, coklat dan putih / batu tulis.
2. 25-pasang kode warna yangditetapkan pada 1950-an untuk kabel konduktor berinsulasi polietilen (PIC).
3. Skema warna ini berasal dari standar nasional DIN 47100(ditarik).

Beberapa telepon seperti model telepon Western Electric Princess dan Trimline memerlukan daya tambahan (~ 6 V AC) untuk pengoperasian lampu panggil pijar. Daya ini dikirim ke perangkat telepon dari transformator oleh pasangan kabel kedua (pin 2 dan 5) dari konektor 6P4C.

### *RJ21*

RJ21 adalah standar jack terdaftar yang menggunakan konektor modular dengan kontak hingga lima puluh konduktor. Ini digunakan untuk mengimplementasikan koneksi hingga 25 jalur, atau sirkuit yang membutuhkan banyak pasangan kabel, seperti yang digunakan dalam sistem telepon kunci 1A2. The konektor pita miniatur dari interface ini juga dikenal sebagai konektor 50-pin telco, CHAMP (AMP), atau Amphenol konektor, yang terakhir menjadi merek dagang umum, sebagai Amphenol adalah terkemuka produsen ini pada satu waktu.

Skema warna kabel, yang dikenal sebagai kode warna hitung genap, ditentukan untuk 25 pasang konduktor sebagai berikut: Untuk setiap cincin, warna primer yang lebih menonjol dipilih dari rangkaian biru, oranye, hijau, coklat, dan batu tulis, dalam urutan itu, dan warna garis sekunder yang lebih tipis dari rangkaian putih, merah, hitam, kuning, dan warna ungu, dalam urutan itu. Skema warna konduktor ujung menggunakan warna yang sama dengan cincin yang cocok tetapi mengganti ketebalan garis warna primer dan sekunder. Karena himpunan sudah dipesan, oranye (warna 2 dalam himpunannya) dengan kuning (warna 4) adalah skema warna untuk pasangan kabel 4 · 5 + 2 - 5 = 17. Jika kuning lebih menonjol, garis lebih tebal, maka kawat adalah konduktor ujung yang menghubungkan ke pin bernomor 25 + pasangan #, yaitu pin 42 dalam hal ini. Konduktor cincin terhubung ke nomor pin yang sama dengan nomor pasangan.

Penghitungan konvensional dari pasangan warna kawat kemudian dimulai dengan biru (dan putih), oranye (dan putih), hijau (dan putih) dan coklat (dan putih), yang memasukkan konvensi kode warna yang digunakan pada kabel yang terdiri dari 4 pasang atau kurang (8 kabel atau kurang) dengan konektor 8P dan 6P.

Konektor pita 50-pin ganda sering digunakan pada punch block untuk membuat kotak breakout untuk private branch exchange (PBX) dan sistem telepon utama lainnya



#### **Gambar 2.2.9. Konektor RJ21 perempuan**

#### **Tabel 2.2.3. Kode warna hitung genap**

|  |  |  |  |
| --- | --- | --- | --- |
| **Warna** | **Pin (tip)** | **Pin (cincin)** | **Warna** |
| Putih biru | 26 | 1 | Biru/putih |
| Putih/oranye | 27 | 2 | Oranye/putih |
| Putih/hijau | 28 | 3 | Hijau/putih |
| Putih/coklat | 29 | 4 | Coklat/putih |
| Putih/batu tulis | 30 | 5 | Batu tulis/putih |
| Merah biru | 31 | 6 | Merah biru |
| Merah/oranye | 32 | 7 | Oranye/merah |
| Hijau merah | 33 | 8 | Hijau merah |
| Merah/coklat | 34 | 9 | Coklat/merah |
| Merah/batu tulis | 35 | 10 | Batu tulis/merah |
| Hitam biru | 36 | 11 | Biru hitam |
| Hitam/oranye | 37 | 12 | Oranye/hitam |
| Hitam/hijau | 38 | 13 | Hijau hitam |
| Hitam/coklat | 39 | 14 | Hitam kecoklatan |
| Hitam/batu tulis | 40 | 15 | Batu tulis/hitam |
| Kuning biru | 41 | 16 | Kuning biru |
| Kuning oranye | 42 | 17 | Oranye/kuning |
| Kuning hijau | 43 | 18 | Hijau kuning |
| Kuning coklat | 44 | 19 | Coklat kuning |
| Kuning/batu tulis | 45 | 20 | Batu tulis/kuning |
| Violet/biru | 46 | 21 | Biru/ungu |
| Violet/oranye | 47 | 22 | Oranye/ungu |
| Violet/hijau | 48 | 23 | Hijau/ungu |
| Violet/coklat | 49 | 24 | Coklat/ungu |
| Violet/batu tulis | 50 | 25 | Batu tulis/ungu |

### 

### *RJ45S*

RJ45S, soket standar yang pernah ditentukan untuk modem atau antarmuka data, menggunakan variasi bodi 8P8C yang dikunci secara mekanis dengan tab ekstra yang mencegahnya untuk kawin dengan konektor lain; perbedaan visual dari 8P8C yang lebih umum tidak kentara. Konektor modular 8P2C berkunci RJ45S asli memiliki pin 5 dan 4 kabel untuk ujung dan cincin dari saluran telepon tunggal, dan pin 7 dan 8 korslet resistor pemrograman, tetapi usang saat ini.

Konektor 8P8C sering salah disebut konektor RJ45 di beberapa bidang seperti telekomunikasi dan jaringan komputer tetapi tidak memiliki tab tambahan.



#### **Gambar 2.2.10. Konektor perempuan dengan kunci 8P8C (jack).**

### *RJ48*

RJ48 adalah jack terdaftar. Ini digunakan untuk penghentian T1 dan ISDN dan saluran data area lokal / layanan digital subrate. Ini menggunakan konektor modular delapan posisi (8P8C). RJ48C biasa digunakan untuk rangkaian T1 dan menggunakan nomor pin 1, 2, 4 dan 5.

RJ48X adalah variasi dari RJ48C yang berisi blok korslet di dalam jack sehingga dibuat loopback untuk mengatasi masalah saat dicabut dengan menghubungkan pin 1 dan 4, dan 2 dan 5. Terkadang ini disebut sebagai jack self-looping.

RJ48S biasanya digunakan untuk saluran data area lokal dan layanan digital subrate, dan memiliki satu atau dua saluran. Ini menggunakan berbagai kunci dari konektor modular 8P8C.

Konektor RJ48 diikat ke kabel berpelindung twisted pair (STP), bukan kabel standar unshielded twisted pair (UTP) CAT- (1–5).

#### **Tabel 2.2.4. Kabel RJ48C dan RJ48X**

|  |  |  |  |
| --- | --- | --- | --- |
| **Pin** | **Pasangan** | **Sinyal** | **Warna** |
| 1 | R | Cincin RX | Oranye/putih |
| 2 | T | Tip RX | Putih/oranye |
| 3 |  | Pendiam | Putih/hijau |
| 4 | R1 | Cincin TX | Biru/putih |
| 5 | T1 | Tip TX | Putih biru |
| 6 |  | Pendiam | Hijau/putih |
| 7 |  | Melindungi | Putih/coklat |
| 8 |  | Melindungi | Coklat/putih |

### *RJ61*

RJ61 adalah antarmuka fisik yang sering digunakan untuk mengakhiri kabel jenis twisted pair. Ini menggunakan konektor modular delapan posisi, delapan konduktor (8P8C).

Pinout ini hanya untuk penggunaan telepon multisaluran; RJ61 tidak cocok untuk digunakan dengan data berkecepatan tinggi, karena pin untuk pasangan 3 dan 4 ditempatkan terlalu lebar untuk frekuensi pensinyalan tinggi. Garis T1 menggunakan kabel lain untuk konektor yang sama, yang disebut RJ48. Ethernet over twisted pair (10BASE-T, 100BASE-TX dan 1000BASE-T) juga menggunakan kabel yang berbeda untuk konektor yang sama, baik T568A atau T568B . RJ48, T568A, dan T568B semuanya dirancang untuk menjaga pin tetap berdekatan untuk pasangan 3 dan 4.

Kabel datar satin perak delapan konduktor yang biasanya digunakan dengan telepon analog empat saluran dan soket RJ61 juga tidak cocok untuk digunakan dengan data berkecepatan tinggi. Kabel twisted pair diperlukan untuk aplikasi data. Kabel patch data twisted pair yang digunakan dengan ketiga standar data tersebut bukan merupakan pengganti langsung untuk kabel RJ61, karena RJ61 pair 3 dan 4 akan terpecah diantara kabel patch twisted pair yang berbeda, sehingga terjadi cross-talk antara jalur suara 3 dan 4 yang mungkin saja terlihat untuk kabel patch panjang.

Dengan munculnya sistem kabel terstruktur dan konvensi TIA / EIA-568-B, pinout RJ61 tidak lagi digunakan. Standar T568A dan T568B digunakan sebagai pengganti RJ61 sehingga standar kabel tunggal di fasilitas dapat digunakan untuk suara dan data.

#### **Tabel 2.2.5. Kabel RJ61 (USOC)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Pin** | **Pasangan** | **Sinyal** | **Warna** |
| 1 | 4 | Tip | Putih/coklat |
| 2 | 3 | Tip | Putih/hijau |
| 3 | 2 | Tip | Putih/oranye |
| 4 | 1 | Cincin | Biru/putih |
| 5 | 1 | Tip | Putih biru |
| 6 | 2 | Cincin | Oranye/putih |
| 7 | 3 | Cincin | Hijau/putih |
| 8 | 4 | Cincin | Coklat/putih |

### *Jack Serupa dan nama tidak resmi*

Nama gaya RJ berikut tidak mengacu pada jenis ACTA resmi. Label RJ9, RJ10, RJ22 yang beragam digunakan untuk 4P4C dan 4P2C konektor modular, paling biasanya diinstal pada handset telepon dan pengikatan mereka. Handset telepon tidak terhubung langsung ke jaringan publik, dan oleh karena itu tidak memiliki sebutan jack terdaftar.

RJ45 sering digunakan saat mengacu pada konektor 8P8C yang digunakan untuk T568A / T568B dan Ethernet. Ini berbeda dari antarmuka RJ45S resmi. RJ50 sering kali merupakan antarmuka 10P10C, sering digunakan untuk aplikasi data.

Konektor micro pita, pertama kali dibuat oleh Amphenol, yang digunakan dalam antarmuka RJ21, juga telah digunakan untuk menghubungkan Ethernet port dalam jumlah besar dari switch dengan port 50-pin ke CAT-5 dinilai patch panel, atau antara dua patch panel. Kabel dengan konektor 50-pin di salah satu ujungnya dapat mendukung 6 konektor 8P8C kabel penuh atau port Ethernet pada panel patch dengan satu pasangan cadangan. Sebagai alternatif, hanya pasangan yang diperlukan untuk 10/100 Ethernet yang dapat dihubungkan dengan kabel yang memungkinkan dua belas port Ethernet dengan satu pasangan cadangan.

Konektor ini juga digunakan dengan kunci jaminan pegas untuk koneksi SCSI-1. Beberapa printer komputer menggunakan versi 36-pin yang lebih pendek yang dikenal sebagai konektor Centronics. The 8P8C modular jack dipilih sebagai calon ISDN sistem. Untuk dipertimbangkan, sistem konektor harus ditentukan oleh standar internasional, yang mengarah pada pembuatan standar ISO 8877. Di bawah aturan proyek standar IEEE 802, standar internasional lebih disukai daripada standar nasional, jadi ketika versi kabel twisted-pair 10BASE-T asli dari Ethernet dikembangkan, konektor modular ini dipilih sebagai dasar untuk IEEE 802.3i- 1990.

## **Bahasa Pemograman**

### *JAVA*

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (bytecode) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (general purpose), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda, java dikenal pula dengan slogannya, "Tulis sekali, jalankan di mana pun". Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi.

### *PYTHON*

Python adalah bahasa pemrograman yang berdaya dan mudah dipelajari. Python memiliki struktur data tingkat tinggi yang efisien dan pendekatan yang sederhana namun efektif untuk pemrograman berorientasi objek. Sintaksis Python yang elegan dan tipe dinamis, bersama dengan sifatnya yang diinterpretasikan, menjadikannya bahasa yang ideal untuk skrip dan pengembangan aplikasi yang cepat di banyak area di sebagian besar platform.

Interpreter Python dan pustaka standar yang luas tersedia secara bebas dalam bentuk kode sumber atau biner untuk semua platform utama dari situs Web Python*,*[*https://www.python.org/*](https://www.python.org/)*,* dan dapat didistribusikan secara bebas. Situs yang sama juga berisi distribusi dan refernsi ke banyak modul Python gratis dari pihak ketiga, program dan alat, serta dokumentasi tambahan.

Interpreter Python mudah dikembangkan dengan fungsi dan tipe data baru diimplementasikan dalam C atau C ++ (atau bahasa lain yang bisa dipanggil dari C). Python juga cocok sebagai bahasa tambahan untuk aplikasi yang dapat disesuaikan.

## **Perancangan Sistem**

### *Uml*

*Unified Modeling Language* (UML) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat tool untuk mendukung pengembangan sistem tersebut. UML mulai diperkenalkan oleh Object Management Group, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP. UML merupakan dasar bagi perangkat (tool) desain berorientasi objek dari IBM.

UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem.

### *Use Case Diagram*

*Use case diagram* merupakan gambaran atau representasi dari interaksi yang terjadi antara sistem dan lingkungannya. Menurut Alistair Cockburn, use case mendeskripsikan tentang tingkah laku sistem di berbagai kondisi dan bagaimana sistem tersebut menanggapi permintaan pengguna. Use case didefinisikan dari sudut pandang aktor yang terlibat. Aktor merupakan orang atau alat yang menggunakan sistem tersebut atau bisa juga sebagai apapun yang berinteraksi dengan sistem.

#### **Tabel 2.2.6. Use Case Diagram**

|  |  |
| --- | --- |
| **Gambar** | **Keterangan** |
|  | *Use Case* menggambarkan fungsionalitas yang disediakan system sebagai unit – unit yang bertukar pesan antar unit dengan actor, yang dinyatakan dengan menggunakan kata kerja |
|  | *Actor* atau Aktor adalah Abstraction dari orang atau system yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasikan actor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan *Use Case*, tetapi tidak memiliki kontrol terhadap *use case* |
|  | Asosiasi antara aktor dan *use case*, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data |
|  | Asosiasi antara aktor dan *use case* yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem |
|  | *Include*, merupakan di dalam *use case* lain (*required*) atau pemanggilan *use case* oleh *use case* lain, contohnya adalah pemanggilan sebuah fungsi program |
|  | *Extend*, merupakan perluasan dari *use case* lain jika kondisi atau syarat terpenuhi |

### *Activity Diagram*

*Activity Diagram* adalah bentuk visual dari alir kerja yang berisi aktivitas dan tindakan, yang juga dapat berisi pilihan, pengulangan, dan *concurrency*. Dalam *Unified Modeling Language*, diagram aktivitas dibuat untuk menjelaskan aktivitas komputer maupun alur aktivitas dalam organisasi. Diagram aktivitas menggambarkan alur kontrol secara garis besar. Diagram aktivitas memiliki komponen dengan bentuk tertentu, dihubungan dengan tanda panah. Panah mengarahkan urutan aktivitas terjadi, dari awal sampai akhir. Diagram aktivitas dapat dianggap sebagai jenis alir kerja. Umumnya alir kerja tidak memiliki cara untuk menampilkan *concurrency*. Simbol penggabungan dan pemecahan pada diagram aktivitas dapat menjadi solusi untuk pemakaian yang sederhana.

#### **Tabel 2.2.7. Activity Diagram**

|  |  |
| --- | --- |
| **Gambar** | **Keterangan** |
|  | *Start Point*, diletakkan pada pojok kiri atas dan merupakan awal aktivitas |
|  | *End Point*, Akhir Aktivitas |
|  | *Activities*, menggambarkan suatu proses/kegiatan bisnis |
|  | *Fork*/percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu |
|  | *Join* (penggabungan) atau *rake*, digunakan untuk menunjukkan adanya dekomposisi |
|  | *Decision Points*, menggambarkan pilihan untuk pengambilan keputusan, *true* atau *false* |
|  | *Swimlane*, pembagian *activity diagram* untuk menunjukkan siapa melakukan apa |

### *Sequence Diagram*

*Sequence Diagram* menunjukkan interaksi objek yang diatur dalam urutan waktu. Ini menggambarkan objek yang terlibat dalam skenario dan urutan pesan yang dipertukarkan antara objek yang diperlukan untuk menjalankan fungsionalitas skenario. Diagram urutan biasanya dikaitkan dengan realisasi kasus penggunaan dalam Tampilan Logis dari sistem yang sedang dikembangkan. Diagram urutan terkadang disebut diagram peristiwa atau skenario peristiwa.

*Sequnce Diagram* menunjukkan, sebagai garis vertikal paralel (garis hidup), berbagai proses atau objek yang hidup secara bersamaan, dan, sebagai panah horizontal, pesan yang dipertukarkan di antara mereka, dalam urutan terjadinya. Ini memungkinkan spesifikasi skenario runtime sederhana dengan cara grafis.

#### **Tabel 2.2.8. Sequence Diagram**

|  |  |
| --- | --- |
| **Gambar** | **Keterangan** |
|  | *Entity Class*, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data |
|  | *Boundary Class*, berisi kumpulan kelas yang menjadi *interfaces* atau interaksi antara satu atau lebih actor dengan sistem, seperti tampilan form entry dan form cetak |
|  | *Control class*, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek |
|  | *Message*, simbol mengirim pesan antar *class* |
|  | *Recursive*, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri |
|  | *Activation*, mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi |
|  | *Lifeline*, garis titik-titik yang terhubung dengan objek, sepanjang *lifeline* terdapat *activation* |

### 

### *Class Diagram*

*Class Diagram* merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class Diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan.

*Class Diagram* secara khas meliputi : Kelas (*Class*), Relasi *Assosiations*, *Generalitation* dan *Aggregation*, attribut (*Attributes*), operasi (*operation*/*method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau attribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality*.

#### **Tabel 2.2.9. Multiplicity Class Diagram**

|  |  |
| --- | --- |
| **Multiplicity** | **Penjelasan** |
| 1 | Satu dan hanya satu |
| 0..\* | Boleh tidak ada atau 1 atau lebih |
| 1..\* | 1 atau lebih |
| 0..1 | Boleh tidak ada, maksimal 1 |
| n..n | Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimal 4 |

# **BAB III**

# **HASIL DAN PEMBAHASAN**

## **Sejarah Organisasi Tempat Kerja Praktik**

Dinas Komunikasi, Informatika, Persandian dan Statistik Kabupaten Bekasi yang selanjutnya disebut sebagai (Diskominfosantik) adalah perangkat daerah yang berfungsi sebagai sarana dalam menyebarkan informasi kepada masyarakat luas mengenai Informasi Publik, Infromasi Pelayanan dan Layanan digital dalam membangun Kabupaten Bekasi yang berbasis Smart City (Diskominfosantik) di dirikan pada tanggal 20 Desember 2016 berdasarkan pada Struktur Kelembagaan daerah terbaru yang ditetapkan dalam Peraturan Daerah Kabupaten Bekasi Nomor 07 tahun 2016 tentang Pembentukan dan Susunan Perangkat Daerah Kabupaten Bekasi.

Perkembangan yang terjadi kepada (Diskominfosantik) sangatlah pesat, berbagai macam prestasi ditorehkan oleh Diskominfosantik baik internal daerah maupun eksternal daerah. Seperti pencapaian Pelayanan terbaik dalam Layanan Informasi dan Dokumentasi tingkat Jawa Barat dan Media Sosial Terbaik tingkat Nasional. Diskominfosantik juga berperan dalam menjadikan Kabupaten Bekasi menjadi Kota Smart City yang menjadikan Kabupaten Bekasi menjadi Kota yang berbasis Informasi Teknologi.

Diskominfosantik Kabupaten Bekasi memiliki Struktur Kordinasi dan Organisasi yang sesuai dengan Peraturan Bupati Kabupaten Bekasi yang terbaru yaitu Peraturan Bupati Kabupaten Bekasi nomor 54 Tahun 2018 tentang Perubahan Kedua Atas Peraturan Bupati Kabupaten Bekasi nomor 79 Tahun 2016 tentang Kedudukan, Susunan Organisasi, Tugas Pokok dan Fungsi Serta Tata Kerja pada Dinas Komunikasi, Informatika, Persandian dan Statistik Kabupaten Bekasi.

## **Visi dan Misi DISKOMINFOSANTIK**

### *Visi*

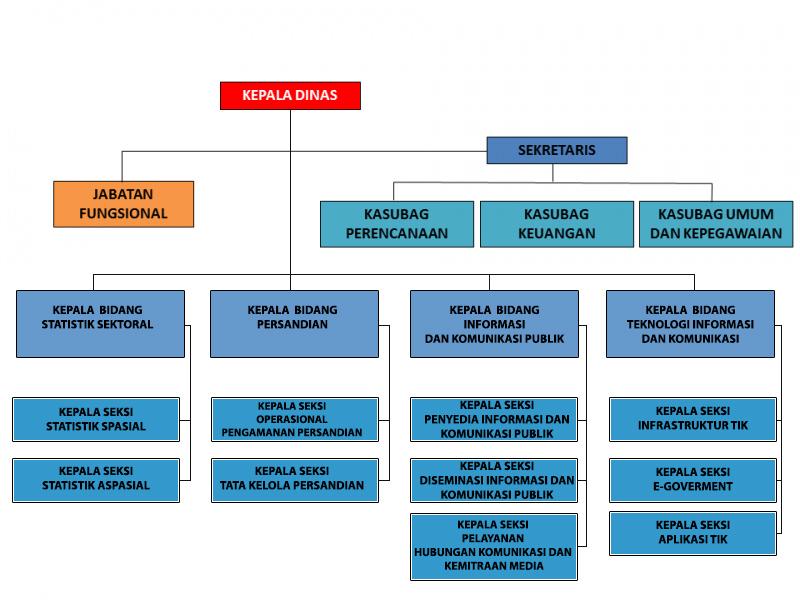
Sebagai Kabupaten Bekasi yang Informatif, Kreatif, Akuntabel Berbasis Teknologi Informasi sehingga dapat bersinergi dengan program Kabupaten Bekasi dalam program Kabupaten Bekasi yaitu melayani dengan cepat dan lebih dekat dengan masyarakat

### *Misi*

1. Meningkatkan Sarana dan Prasarana Teknologi Informasi
2. Mewujudkan Tata Kelola Kepemerintahan yang Baik Melalui Penerapan E - Government
3. Melaksanakan Pelayanan Informasi dan Komunikasi kepada Masyarakat serta Promosi Potensi.

## **Struktur Organisasi DISKOMINFOSANTIK**

Struktur organisasi DISKOMINFOSANTIK kabupaten Bogor dapat di lihat pada gambar 3.1. dibawah ini.



#### **Gambar 3.3.1 Struktur Organisasi**

## **Sistem yang Sedang Berjalan**

Analisis sistem merupakan gambaran tentang sistem yang saat ini sedang berjalan di DISKOMINFO PEMDA Bekasi pada bagian jaringan, sistem yang digunakan masih sederhana yaitu dengan menggunakan Mikrotik Router operating system (OS) dan aplikasi The Dude untuk membuat sebuah sistem monitoring jaringan. Analisis sistem ini bertujuan untuk membuat sistem yang baru agar tervirtualisasi sehingga dapat lebih efektif dan efisien.

## **Analisa Sistem**

### *Permasalahan*

Sistem yang digunakan di DISKOMINFO PEMDA Bekasi masih sangat sederhana, karena hanya menggunakan mikrotik router dan aplikasi The Dude, dimana aplikasi tersebut berfungsi sebagai monitoring saja. Sehingga ketika ada permasalahan di lokasi jaringan tertentu, teknisi diharuskan untuk meninjau kelokasi. Selain itu aplikasi The Dude juga masih memiliki kelemahan, yaitu aplikasi ini tidak bisa mendeteksi dan melakukan ping saat client firewall di tutup. Untuk menyelesaikan permasalahan tersebut, maka dibutuhkan sebuah sistem jaringan dinamis yang bersifat *programmable* dan *centralized*. *Software Defined Network* merupakan perkembangan jaringan yang dinamis dan memberikan kemudahan kepada penggunanya dalam mendesain, membangun, dan mengelola jaringan, sehingga lebih efektif dan efisien karna dikelola secara terpusat.

### *Kebutuhan Sistem (Requirement)*

Sistem yang diajukan yaitu Software Defined Network menggunakan controller openDaylight dengan kebutuhan sebagai berikut :

1. Minimum sistem
   1. CPU, 2 cores
   2. RAM, 2 Gb
   3. Storage, 16 Gb
2. Rekomendasi sistem
   1. CPU, 8 cores
   2. RAM, 8 Gb
   3. Storage, 64 Gb
3. *Operation System* (OS)
   1. CentOS 7.x
   2. Java Virtual Machine 11

## **Perancangan Sistem**

### *Use Case Diagram*