

SKRIPSI

**IMPLEMENTASI JARINGAN *SOFTWARE DEFINED NETWORK*
(SDN) MENGGUNAKAN RYU CONTROLLER PADA
RASPBERRY-PI**

Disusun dan diajukan oleh:

**DIRGA UTAMA KAMAL
D421 16 312**



**PROGRAM STUDI SARJANA TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS HASANUDDIN
GOWA
2023**

LEMBAR PENGESAHAN SKRIPSI
IMPLEMENTASI JARINGAN SOFTWARE DEFINED NETWORK (SDN)
MENGGUNAKAN RYU CONTROLLER PADA RASPBERRY-PI

Disusun dan diajukan oleh

DIRGA UTAMA KAMAL

D42116312

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka
Penyelesaian Studi Program Sarjana Program Studi Teknik Informatika Fakultas
Teknik Universitas Hasanuddin pada tanggal 08 Agustus 2023 dan dinyatakan
telah memenuhi syarat kelulusan.

Menyetujui,

Pembimbing Utama,

Pembimbing Pendamping,

Dr. Eng. Ir. Muhammad Niswar, S.T.,
M.InfoTech.
Nip. 197309221999031001

Dr. Eng. Adv. Wahyudi Paundu, ST., MT.
Nip. 197503132009121003

Seluruh Program Studi,



Prof. Dr. Ir. Indrabayati, S.T., M.T., M.Bus.Sys., IPM., ASEAN.Eng.
Nip. 19750716 200212 1 004

PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini ;
Nama : Dirga Utama Kamal
NIM : D42116312
Program Studi : Teknik Informatika
Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

IMPLEMENTASI JARINGAN SOFTWARE DEFINED NETWORK (SDN)
MENGUNAKAN RYU CONTROLLER PADA RASPBERRY-PI

Adalah karya tulisan saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain dan bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Semua informasi yang ditulis dalam skripsi yang berasal dari penulis lain telah diberi penghargaan, yakni dengan mengutip sumber dan tahun penerbitannya. Oleh karena itu semua tulisan dalam skripsi ini sepenuhnya menjadi tanggung jawab penulis. Apabila ada pihak manapun yang merasa ada kesamaan judul dan atau hasil temuan dalam skripsi ini, maka penulis siap untuk diklarifikasi dan mempertanggungjawabkan segala resiko.

Segala data dan informasi yang diperoleh selama proses pembuatan skripsi, yang akan dipublikasi oleh Penulis di masa depan harus mendapat persetujuan dari Dosen Pembimbing.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan isi skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Gowa, 08 Agustus 2023

Yang Menyatakan





Dirga Utama Kamal

ABSTRAK

DIRGA UTAMA KAMAL. *Implementasi jaringan Software Defined Network (SDN) menggunakan Ryu Controller pada Raspberry-Pi* (dibimbing oleh . Dr. Eng. Muhammad Niswar., ST, MIT. Dan Dr. Eng. Ady Wahyudi Paundu, S.T., M.T.)

Penelitian ini Software Defined Network (SDN) merupakan generasi baru dalam jaringan yang menggunakan teknologi standar yaitu Openflow. Openflow merupakan standar protokol pada jaringan berbasis Software Defined Network (SDN) yang memisahkan fungsi *control* dan *forwarding* sehingga dapat melakukan kendali jaringan terpusat dan memudahkan melakukan manajemen jaringan. Hubungan antara *controller* dan perangkat menjadi hal utama dalam teknologi ini. *Raspberry-Pi* sering disingkat dengan nama Raspi adalah mini PC yang menggunakan system operasi Linux seperti pada umumnya. Raspiberry-Pi dapat digunakan sebagai perangkat Software Defined Network. Software Defined Network juga memungkinkan network admisnistrasi unutm melakukan konfigurasi dan maintenance jaringan melalui sebuah controller tanpa fisik ke perangkat jaringan. Riset SDN dilakukan pada Raspi digunakan unutm sebagai perangkat Controller dan Forwarding namun memiliki sebuah kendala port pada perangkat raspi hanya bisa menggunakan satu Link dan kurang cocok digunakan sebagai witch .Pada proyek akhir ini,implementasi dilakukan implementasi jaringan yang berskala kecil menggunakan Open vSwitch sebagai fungsi Forwarding dimana dipasang ke Router yang telah diinstal menggunakan OpenWrt sebagai firmware sehingga mempunyai beberapa port dan Raspi dipasan Ryu Controller sebagai kendali terpusat dan fungsi control.Hasil dari implementasi jaringan berbasis Software Defined Network (SDN) pada proyek ini menunjukan performa dengan Rata-rata Throughput, Rata-rata Delay, Rata-rata Jitter, dan rata-rata Packet loss.

Kata kunci: Software Defined Network, Router, OpenSwitch, Raspberry-PI

ABSTRACT

DIRGA UTAMA KAMAL. *Implementation of Software Defined Network (SDN) network using Ryu Controller on Raspberry-Pi* (supervised by . Dr. Eng. Muhammad Niswar., ST, MIT. Dan Dr. Eng. Ady Wahyudi Paundu, S.T., M.T.)

In this study, Software Defined Network (SDN) is a new generation of networks that uses standard technology, namely Openflow. Openflow is a standard protokol on Software Defined Network (SDN) based networks that separates control and forwarding functions so that centralized network control can be performed and facilitates network management. The relationship between the controller and the device is the main thing in this technology. Raspberry-Pi, often abbreviated as Raspi, is a mini PC that uses the Linux operating system as usual. Raspiberry-Pi can be used as a Software Defined Network device. Software Defined Network also allows network administration to perform network configuration and maintenance through a controller without being physically connected to network devices. SDN research was carried out on Raspi used for Controller and Forwarding devices but has a port problem on Raspi devices that can only use one link and is not suitable for use as a witch. In this final project, implementation is carried out on a small scale network using Open vSwitch as the Forwarding function which is installed on a router that has been installed using OpenWrt as firmware so that it has several ports and Raspi installed Ryu Controller as centralized control and control functions. The results of the implementation of a Software Defined Network (SDN) based network on this project show performance with Average Throughput, Average -Average Delay, Average Jitter, and average Packet loss.

Keywords: Software Defined Network, Router, OpenSwitch, Raspberry-PI

DAFTAR ISI

LEMBAR PENGESAHAN SKRIPSI.....	i
PERNYATAAN KEASLIAN.....	ii
ABSTRAK	iii
ABSTRACT.....	iv
DAFTAR ISI.....	v
DAFTAR GAMBAR	vii
DAFTAR TABEL.....	ix
DAFTAR SINGKATAN DAN ARTI SIMBOL	x
DAFTAR LAMPIRAN.....	xi
KATA PENGANTAR	xii
BAB I PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah	3
1.3. Tujuan Penelitian.....	3
1.4. Manfaat Penelitian.....	4
1.5. Ruang Lingkup	4
BAB II TINJAUAN PUSTAKA.....	5
2.1. Software Defined Network (SDN)	5
2.2. Openflow	5
2.3. Openswitch.....	7
2.4. Ryu controller.....	7
2.5. OpenWrt	8
2.6. Raspberry-PI 3B+	8
2.7. Wireshark	9
2.8. Iperf	9

BAB III METODE PENELITIAN.....	10
3.1 Lokasi Penelitian	10
3.2 Tahapan Penelitian	10
3.3 Instrumen Penelitian.....	11
3.4 Teknik Pengambilan Data	12
3.5 Perancangan Sistem.....	12
BAB IV HASIL DAN PEMBAHASAN	32
4.1 Hasil Penelitian.....	32
4.2 Pembahasan	48
BAB V KESIMPULAN DAN SARAN.....	49
5.1 Kesimpulan.....	49
5.2 Saran	49
DAFTAR PUSTAKA	50
LAMPIRAN.....	51

DAFTAR GAMBAR

Gambar 1 Struktur OpenFlow	6
Gambar 2 OpenSwitch	7
Gambar 3 Lokasi Penelitian	10
Gambar 4 Tahapan Penelitan	10
Gambar 5 Flowchart Perancangan Sistem Switch	13
Gambar 6 Flowchart Perancangan Sistem Controller	14
Gambar 7 Kompilasi Update.....	15
Gambar 8 Package Instalasi	15
Gambar 9 Cloning OpenWrt	16
Gambar 10 Update OpenWrt.....	16
Gambar 11 Makefile	17
Gambar 12 Website Raspberry-PI.....	19
Gambar 13 FormatSD Card	19
Gambar 14 Raspberry-PI Imager.....	20
Gambar 15 Pengaturan Raspberry-PI Imager	21
Gambar 16 IP Local PC	22
Gambar 17 IP Adres dari Raspberry-PI	23
Gambar 18 Aplikasi Putty	23
Gambar 19 Login SSH	24
Gambar 20 Update Ubuntu	24
Gambar 21 Install Package Git	25
Gambar 22 Install Package Python3	25
Gambar 23 Upgrade pip	25
Gambar 24 Clone Ryu Controller	25
Gambar 25 Instalasi Ryu Controller	26
Gambar 26 Run Ryu Controller	26
Gambar 27 Instalasi iperf3.....	27
Gambar 28 Protokol TCP PC1	28

Gambar 29 Server PC2.....	29
Gambar 30 Protokol UDP PC1	29
Gambar 31 Server PC2.....	30
Gambar 32 Pengujian Pengiriman Data TCP dan UDP.....	31
Gambar 33 Pengujian Konvensional PC1	32
Gambar 34 Pengujian Konvensional PC2.....	32
Gambar 35 Pengujian Switch.....	33
Gambar 36 Raspberry-PI ke PC1	34
Gambar 37 Raspberry-PI ke PC2.....	34
Gambar 38 Raspberry-PI ke Switch.....	34
Gambar 39 Pengujian Ryu Controller.....	35
Gambar 40 Ryu Controller ke PC1	35
Gambar 41 Raspberry-PI ke PC2.....	36
Gambar 42 Raspberry-PI ke Switch.....	36
Gambar 43 Grafik TCP Throughput	37
Gambar 44 Grafik UDP Throughput.....	38
Gambar 45 Grafik TCP Delay	42
Gambar 46 Grafik UDP delay.....	43
Gambar 47 Grafik TCP Jitter	45
Gambar 48 Grafik UDP Jitter	46

DAFTAR TABEL

Tabel 1	Tabel TCP Throughput	37
Tabel 2	Tabel UDP Throughput.....	38
Tabel 3	Tabel TCP Packet Lose.....	40
Tabel 4	Tabel UDP Packet Lose	40
Tabel 5	Tabel TCP Delay.....	41
Tabel 6	Tabel UDP Delay	42
Tabel 7	Tabel TCP Jitter	44
Tabel 8	Tabel UDP Jitter.....	45

DAFTAR SINGKATAN DAN ARTI SIMBOL

Lambang / Singkatan	Arti dan Keterangan
-c	Client
-i	Interval
-p	Port
-t	Waktu
-l	Payload Size
-s	Server
-u	UDP

DAFTAR LAMPIRAN

Lampiran 1. Source Code.....	51
Lampiran 2. Dataset	51
Lampiran 3. Lembar Perbaikan Skripsi	52

KATA PENGANTAR

Assalamu'alaikum Warahmatullaahi Wabarakatuh.

Alhamdulillahirabbil'aalamiin, puji syukur senantiasa penulis panjatkan atas kehadiran Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya yang sehingga penulis dapat menyelesaikan laporan Tugas Akhir ini yang berjudul “Implementasi Jaringan Software Defined Network (SDN) Menggunakan RYU Controller Pada Raspberry-Pi” sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu (S1) pada Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin. Sholawat serta salam InsyaAllah selalu tercurahkan kepada junjungan kita Nabi Muhammad SAW yang telah menunjukkan dan mengajarkan akhlak mulia sehingga didapatkan kenyamanan.

Dalam proses pembuatan laporan akhir ini tidak lepas dari bantuan, bimbingan serta dukungan dari berbagai pihak, dari masa perkuliahan sampai dengan masa penyusunan tugas akhir ini. Oleh karena itu, penulis dengan senang mengucapkan rasa terima kasih kepada:

1. Allah Swt yang melalui berkat dan rahmat-Nya sebagai penolong disetiap Langkah penulis sehingga dapat menyelesaikan tugas akhir ini.
2. Kepada Orang Tua dan keluarga, Bapak Kamaluddin dan Ibu Nursiah yang selalu memberikan dukungan, motivasi, dan semangat serta selalu sabar menunggu ku hingga sarjana, tidak henti-hentinya mendoakan, merawat, menyayangi, terima kasih. ♥

3. Bapak Dr. Eng. Ir. Muhammad Niswar,S.T., M.InfoTech., selaku pembimbing I dan bapak Dr. Ady Wahyudi Paundu S.T., M.T., selaku pembimbing II yang selalu menyediakan waktu, tenaga, pikiran dan memberikan bimbingan dalam penyusunan tugas akhir ini.
4. Keluarga besar saya secara umum yang telah memberikan banyak sumbangsih baik secara materi maupun non-materi
5. Segenap Dosen dan Staff Departemen Teknik Informatika Fakultas Teknik Univesitas Hasanuddin yang telah banyak membantu dan memberikan banyak ilmu selama masa perkuliahan.
6. Teman-teman Anak Teknik baik Senior maupun Junior dan terkhusus teman Angkatan 2016 yang telah menemani suka duka dalam menjalani dunia kemahasiswaan dan perkuliahan dikampus kita Fakultas Teknik Universitas Hasanuddin.
7. Teman-teman I G N I T E R 1 6 atas dukungan dan semangat yang telah diberikan
8. Orang-orang berpengaruh lainnya yang tidak sempat penulis sebutkan satu persatu yang telah memberikan dukungan, semangat dan doanya selama penyusunan tugas akhir ini.

Akhirnya dengan segala kerendahan hati, penyusun menyadari bahwa masih terdapat kekurangan dalam penyusunan Tugas Akhir ini baik isi maupun cara penyajian. Oleh karena itu penyusun mengharapkan adanya saran dan kritik yang bersifat membangun demi kesempurnaan tugas akhir ini. Penulis berharap semoga tugas akhir ini dapat memberikan manfaat bagi pembaca pada umumnya dan manfaat bagi penulis khususnya.

Wassalamu'alaikum Warahmatullaahi Wabarakaatuh

Makassar, Juni 2023

Penulis

BAB I PENDAHULUAN

1.1. Latar Belakang

Software-Defined Network (SDN) merupakan pendekatan inovatif dalam menajemen arsitektur jaringan yang memisahkan kontrol jaringan (control plane) dari infrastruktur fisik (data plane) dan mengelola perangkat jaringan secara sentral melalui perangkat lunak. Dalam SDN, fungsi kontrol jaringan terpusat dan diatur melalui program-program perangkat lunak yang berjalan di lapisan kontrol, sedangkan perangkat keras jaringan berfungsi sebagai pengalihan lalu lintas tanpa memiliki keputusan pengalihan sendiri, Software-Defined Network (SDN) berkaitan erat dengan perkembangan dan kebutuhan infrastruktur jaringan yang semakin kompleks dan dinamis. Sebelum memahami SDN, penting untuk memahami bagaimana jaringan tradisional beroperasi. Jaringan tradisional terdiri dari perangkat keras (misalnya, router, switch) dan perangkat lunak (seperti firmware dan konfigurasi). Kebanyakan pengaturan jaringan, seperti routing dan switching, diimplementasikan secara langsung di perangkat keras tersebut. Pada akhir tahun 2000-an, dengan semakin berkembangnya teknologi virtualisasi dan cloud computing, kebutuhan akan jaringan yang lebih fleksibel, skaler, dan mudah dikelola semakin meningkat. Inilah latar belakang munculnya konsep Software-Defined Network (SDN). Keuntungan pada SDN meliputi Sentralisasi Pengelolaan, SDN memungkinkan manajemen jaringan terpusat yang dapat diotomatisasi, mempermudah konfigurasi dan pemantauan. Fleksibilitas dan Skalabilitas: Dengan abstraksi jaringan dan pemisahan kontrol dan data plane, jaringan dapat dengan mudah diperbesar dan disesuaikan sesuai dengan kebutuhan. Optimalisasi Lalu Lintas: SDN memungkinkan pengalihan lalu lintas yang cerdas dan dinamis, sehingga dapat mengoptimalkan kinerja jaringan secara real-time. Kemudahan Implementasi Kebijakan Keamanan: Dengan kontrol sentral, kebijakan keamanan dapat diterapkan dan dikelola dengan lebih efisien dan konsisten. Dengan semakin cepatnya perkembangan teknologi, SDN terus menjadi area penelitian dan inovasi yang menarik dalam dunia jaringan komputer. Meskipun telah banyak kemajuan, tantangan teknis dan keamanan tetap ada dan terus dipecahkan oleh para ahli dan praktisi.

SDN adalah pendekatan yang menarik untuk mengatur jaringan, karena menggantikan beberapa fungsi yang dulunya ditanamkan dalam perangkat keras khusus dengan perangkat lunak. Ini memungkinkan kita untuk memisahkan kontrol dari

perangkat keras, sehingga memberikan kemampuan baru untuk memanipulasi jaringan dengan cara yang sangat fleksibel (Nick McKeown., 2009). Terdapat beberapa variabel yang akan diuji dalam penelitian ini seperti throughput, delay, jitter dan packet loss. Data yang didapatkan akan diolah dan di analisis sehingga dapat diketahui sejauh mana arsitektur SDN berpengaruh dalam sebuah jaringan nyata dan bagaimana perbandingannya dengan arsitektur tradisional. Selain itu, dengan menggunakan beberapa skenario dapat diketahui bagaimana fleksibilitas dan skalabilitas yang dimiliki oleh arsitektur SDN. SDN mengusung sebuah paradigma baru dalam jaringan yang awalnya bersifat terdistribusi dan cenderung tertutup menjadi jaringan yang bersifat terpusat, programmable dan terbuka (open source). SDN memberikan konsep network topology virtualisation dan memungkinkan administrator untuk melakukan customize pada control plane.(Sudiyatmoko et al., 2016)

Penelitian arsitektur Software-Defined Network untuk diterapkan ke jaringan pada umumnya masih dilakukan dengan bantuan software mininet. Mininet merupakan emulator jaringan yang mensimulasikan koleksi dari host-end, switch, router, dan link pada single kernel Linux. Masing-masing elemen ini disebut sebagai "host" menggunakan virtualisasi ringan untuk membuat sistem tampilan tunggal sehingga terlihat seperti jaringan yang lengkap, menjalankan kernel, sistem, dan user code yang sama. Sebagai contoh,(Ummah, 2016), melakukan penelitian untuk merancang dan mensimulasikan jaringan virtual berbasis Software-Defined Network (SDN) yang bertujuan mengetahui representasi jaringan SDN terhadap jumlah node dalam jaringan. Pada penelitiannya dibuat beberapa skenario rancangan topologi menggunakan Visual Network Description untuk kemudian disimulasikan menggunakan simulator mininet. Skenario yang dibuat memiliki jumlah perangkat yang berbeda-beda untuk mengetahui ketangguhan arsitektur SDN. Ummah menggunakan tiga variabel untuk menguji jaringan yang dibuat yaitu delay, jitter, dan throughput untuk port TCP dan UDP.

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 (controller). 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.(Hidayat & Rosyid, 2017), SDN mengusung sebuah paradigma baru dalam jaringan yang awalnya bersifat terdistribusi dan cenderung tertutup menjadi jaringan yang bersifat terpusat, programmable dan terbuka (open source).

Berdasarkan pengertian pengimplementasikan jaringan Software Defined Network (SDN) yang merupakan sebuah paradigma atau konsep baru pada sebuah jaringan. Berbeda dengan jaringan konvensional yang fungsi forwarding dan fungsi Software Defined Network (selanjutnya disingkat SDN) merupakan sebuah paradigma/konsep baru fungsi forward dengan fungsi kontrol untuk memudahkan pengelolaan jaringan skala besar SDN. tersebut dalam perangkat yang berbeda. OpenFlow merupakan standar protokol yang memisahkan kontrol pada satu perangkat, konsep SDN melakukan pemisahan forwarding dan kontrol sehingga fungsi control.

Oleh karena itu, penulis mengusulkan judul “Implementasi Jaringan Software Defined Network (SDN) Menggunakan RYU Controller pada Raspberry-PI”.

1.2. Rumusan Masalah

Berdasarkan latar belakang masalah di atas, maka dapat dirumuskan permasalahan pada penelitian ini yaitu:

1. Bagaimana melakukan instalasi Ryu Controller pada Raspberry Pi dan Open vSwitch pada TP-Link agar dapat di implementasikan jaringan komputer berbasis SDN
2. Bagaimana membangun jaringan komputer berbasis SDN dengan menggunakan Ryu controller dan Open vSwitch
3. Bagaimana melakukan pengukuran pada jaringan berbasis SDN

1.3. Tujuan Penelitian

Berdasarkan rumusan masalah di atas, maka dapat dibuat tujuan antara lain:

1. Membuat sebuah topologi jaringan berbasis SDN dengan menggunakan Ryu Controller pada Raspberry Pi sebagai pengontrol dan OpenvSwitch pada router sebagai fungsi forwarding.
2. Membangun sebuah jaringan computer berbasis SDN yang dapat melakukan fungsi routing dan switching.
3. Mengukur parameter-parameter pengukuran Throughput, Packet Lose, Jitter dan delay pada jaringan berbasis SDN

1.4. Manfaat Penelitian

Dengan dilakukannya penelitian ini, maka manfaat yang diperoleh dari penelitian ini adalah:

1. Bagi peneliti, Sebagai implementasi jaringan SDN, Proyek akhir yang di buat diharapkan sebagai sebuah implementasi jaringan berbasis SDN yang dapat memisahkan antara fungsi forward dengan fungsi control,
2. Bagi Masyarakat, sebagai Media pembelajaran dalam konfigurasi jaringan berbasis SDN Sebagai media pembelajaran, sistem ini diharapkan dapat menjadi media pembelajaran dan pengembangan riset SDN kedepannya.
3. Bagi institusi pendidikan, penelitian ini dapat digunakan sebagai referensi ilmiah untuk penelitian-penelitian selanjutnya.

1.5. Ruang Lingkup

Yang menjadi batasan masalah dari penelitian ini adalah sebagai berikut:

1. Perangkat yang digunakan dalam implementasi jaringan komputer berbasis SDN ini adalah Raspberry Pi, TP-Link dan dua buah laptop sebagai client dan server
2. Implementasi jaringan komputer berbasis SDN sebagai fungsi routing dan switching

BAB II

TINJAUAN PUSTAKA

2.1. Software Defined Network (SDN)

Software-Defined Network (SDN) adalah pendekatan revolusioner dalam arsitektur jaringan yang memisahkan logika kontrol (control plane) dari infrastruktur fisik (data plane) dan mengelola seluruh jaringan melalui perangkat lunak sentral yang disebut SDN Controller. Dalam SDN, administrator jaringan dapat mengatur, mengoptimalkan, dan mengelola jaringan secara sentral menggunakan perangkat lunak, tanpa perlu melakukan konfigurasi manual pada setiap perangkat keras secara individual.

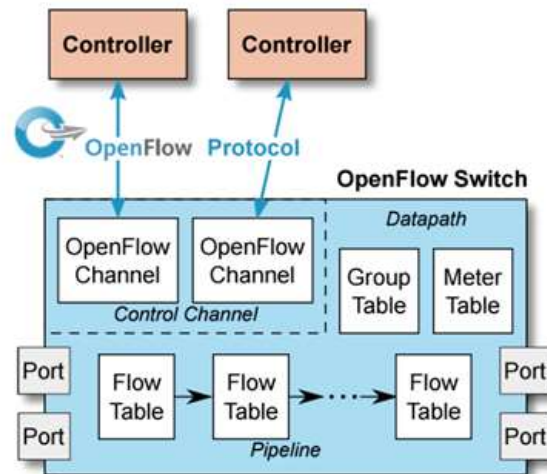
Dilansir dari Open Network Foundation (2022) Software-Defined Networking (SDN) adalah arsitektur jaringan baru yang dinamis, mudah dikelola, hemat biaya, dan mudah beradaptasi, menjadikannya ideal untuk bandwidth tinggi, sifat dinamis dari aplikasi saat ini. Arsitektur ini memisahkan kontrol jaringan dan fungsi penerusan yang memungkinkan kontrol jaringan menjadi dapat diprogram secara langsung dan infrastruktur dasar untuk diabstraksikan menjadi aplikasi dan layanan jaringan. Protokol OpenFlow adalah elemen dasar untuk membangun jaringan SDN.

Dalam arsitektur SDN, jaringan dibagi menjadi dua bagian: *control plane* dan *data plane*. *control plane* bertanggung jawab untuk mengelola dan mengonfigurasi jaringan, sedangkan *data plane* bertanggung jawab untuk meneruskan dan merutekan paket data. *control plane* biasanya diimplementasikan sebagai komponen perangkat lunak terpisah, yang dikenal sebagai *controller*, yang berkomunikasi dengan perangkat jaringan untuk mengonfigurasi dan mengelolanya. Hal ini memungkinkan pemisahan *controller* dan *data plane*, membuatnya lebih mudah untuk mengubah arsitektur jaringan dengan memodifikasi perangkat lunak controller, daripada harus mengonfigurasi ulang perangkat individual.

2.2. OpenFlow

OpenFlow adalah protokol yang memungkinkan SDN Controller untuk menjadi otak dari jaringan, menginstruksikan perangkat keras bagaimana harus mengalihkan lalu lintas. Dengan demikian, kami dapat mencapai efisiensi operasional yang lebih tinggi dan memberdayakan inovasi dalam pengelolaan jaringan (Dan Pitt, Mantan Direktur Eksekutif ONF). Openflow sebuah protokol yang memungkinkan pengaturan, pengiriman, penjaluran paket pada sebuah switch. Dalam arsitektur jaringan konvensional Openflow protokol yang dikembangkan open networking foundation untuk menangani proses komunikasi antara control plane dan data plane sehingga aliran paket

dijaringan dapat diatur dan dikonfigurasi secara independen. Hal ini dapat dilakukan dengan membuat algoritma dan forwarding rules nya dibagian controller yang kemudian didistribusikan ke switch yang berada di jaringan. Metodologi ini dikenal Software Defined Network (SDN) ini sudah diimplementasikan diberbagai perusahaan seperti Google, HP dan IBM.



Gambar 1 Struktur Openflow

Terdapat 2 jenis Openflow yang dibutuhkan untuk implementasi Openflow, Diantaranya adalah sebagai berikut:

2.2.1 OpenFlow controller

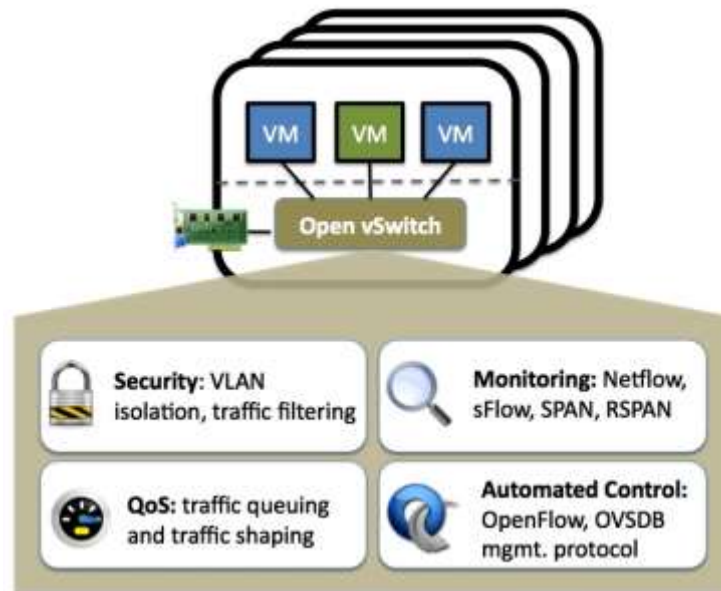
OpenFlow Controller bertugas memformulasikan flow, mengontrol path dan mengatur kerja dari OpenFlow switch.

2.2.2 OpenFlow Switch

Open Flow Switch merupakan sebuah perangkat yang berfungsi sebagai switch hanya bertugas untuk arus data forwarding yang nantinya akan difungsi control seperti routing dan switching dilakukan oleh OpenFlow Controller. Pada proyek ini menggunakan OpenvSwitch sebagai arus data forwarding yang diinstal pada OpenWrt

2.3. OpenSwitch

OpenVSwitch adalah sebuah switch virtual berbasis linux. OpenVSwitch sendiri merupakan perangkat lunak sumber terbuka yang berada dalam lisensi Apache 2.0. OpenVSwitch mendukung pengontrolan secara otomatis dengan protokol openflow



Gambar 2 OpenSwitch

Beberapa fitur yang didukung oleh Open vSwitch, diantaranya adalah:

- Automated Control
- Monitoring
- Quality of Service (QoS)
- Security

2.4. Ryu Controller

Ryu merupakan salah satu controller dalam software defined network yang dirancang untuk meningkatkan kemampuan dalam jaringan yang bermanfaat untuk mempermudah dan mengatur. Secara umum controller merupakan fungsi otak dari software defined network. RYU merupakan open source yang dikembangkan oleh NTT. Dalam RYU application program interface (API) sudah didefinisikan dengan sangat baik yang berarti dapat melakukan pengembangan dengan mudah untuk membuat suatu network management yang baru. Controller RYU ini mendukung beberapa protokol dalam software defined network diantaranya OpenFlow, Netconf, OF-config serta lainnya.

2.5. OpenWrt

OpenWrt adalah sebuah sistem operasi distribusi GNU / Linux yang berbasis firmware. Sistem operasi ini digunakan untuk perangkat yang tertanam terutama pada perangkat router. Selain itu, OpenWrt menggunakan antarmuka dari baris perintah, tetapi tetap memiliki antarmuka WEB yang selalu meningkat. Dukungan teknis juga di sediakan pada kebanyakan proyek perangkat lunak bebas melalui forum dan saluran IRC.

OpenWrt sudah dikonfigurasi menggunakan antarmuka baris perintah (ash shell) atau antarmuka web (Luci). Bahkan sudah ada sekitar 3500 paket perangkat lunak opsional yang tersedia untuk dapat dilakukan instalasi melalui sistem manajemen paket opkg.

2.6. Raspberry-PI

Raspberry PI adalah sebuah komputer papa tunggal (single-board computer) atau SBC seukuran kartu kredit yang dapat digunakan untuk menjalankan program perkantoran, permainan komputer. Raspberry PI diambil dari nama buah, Yaitu buah Raspberry, sedangkan PI diambil dari kata python sebuah nama dari Bahasa pemrograman, Python dijadikan sebagai Bahasa pemrograman utama dari Raspberry PI.

Raspberry Pi pertama kali dikembangkan di laboratorium Komputer Universitas Cambridge oleh Eben Upton, Rob Mullins, Jack Lang, dan Alan Mycroft. Mereka kemudian mendirikan yayasan Raspberry Pi bersama dengan Pete Lomas dan David Braben pada tahun 2009. Pada tahun 2012, Raspberry Pi Model B memasuki produksi massal. Raspberry Pi memiliki komponen yang hampir serupa dengan komputer pada umumnya. Seperti CPU, GPU, RAM, Port USB, Audio Jack, HDMI, Ethernet, dan GPIO. Untuk penyimpanan data menggunakan Micro SD. Raspberry PI memiliki 2 model utama, Yaitu model A dan model B. pada penelitian ini menggunakan Raspberry-PI Model 3 B+, model ini dirilis pada Februari 2016 dengan prosesor quad-core 64-bit dan 1GB RAM. Model B+ dirilis pada Maret 2018 dengan beberapa peningkatan pada konektivitas dan daya dibandingkan dengan Model 3 B, Penggunaan Raspberry-pi ini mendukung untuk melakukan penelitian yang sedang dilakukan sebab Raspberry-PI mudah ditemukan pada marketplace dan dapat dijangkau dengan mudah.

2.7. Wireshark

Wireshark adalah Perangkat lunak analisis jaringan yang memungkinkan pengguna untuk merekam menganalisis lalu lintas jaringan dalam waktu nyata. Wireshark dapat digunakan untuk memecahkan masalah jaringan, mengidentifikasi gangguan, mengoptimalkan kinerja jaringan, dan bahkan memecahkan masalah keamanan. Wireshark adalah tools analisa paket *open source* yang digunakan untuk pemecahan masalah jaringan, analisis, pengembangan perangkat lunak dan protokol komunikasi, dan penelitian. Wireshark mampu menganalisis berbagai protokol, termasuk Ethernet, DNS, HTTP. Wireshark tersedia untuk Windows, macOS, Linux, dan platform lainnya.

Wireshark juga dapat digunakan untuk menganalisa trafik pada protokol lain yang digunakan di SDN, seperti BGP-LS, PCEP, PCE dan lainnya. Wireshark dapat membantu mengidentifikasi masalah perutean dan rekayasa lalu lintas, serta membantu mengoptimalkan kinerja jaringan. Menurut E.V Josy *et al* dalam *Information and Communication Technology for Competitive Strategies (ICTCS 2020)* disini dapat dengan mudah mengintegrasikan wireshark dan SDN. Di ekosistem SDN, Wireshark dapat digunakan untuk memeriksa protokol OpenFlow, yang biasanya digunakan untuk berkomunikasi antara *control plane* dan *data plane* di SDN. Wireshark dapat berguna untuk mengidentifikasi dan memecahkan masalah terkait dengan tabel aliran pada switch, atau masalah komunikasi antara controller dan switch.

2.8. Iperf

Menurut “M. Abdullah, N. Al-awad, and F. W. Hussein, *Performance Comparison and Evaluation of Different Software Defined Networks Controllers,*” *International Journal of Computing and Network Technology*, vol. 6, no. 2, 2018” Iperf adalah alat diagnostik berbasis command line yang berfungsi untuk menganalisis dan mengukur kinerja jaringan. Dengan menggunakan Iperf, throughput jaringan maksimum yang dapat ditangani oleh suatu server dapat diukur, memungkinkan deteksi dan pemecahan masalah terkait performa jaringan. Iperf menciptakan aliran data antara klien dan server Iperf, dan mengevaluasi throughput yang dapat dicapai melalui aliran ini. Evaluasi ini memberikan pemahaman mendalam tentang kemampuan suatu tautan jaringan dalam menangani trafik, atau bagaimana suatu jaringan berperforma dalam skenario tertentu .