

IMPLEMENTASI JARINGAN KOMPUTER BERBASIS *SOFTWARE DEFINED NETWORK* MENGGUNAKAN RYU *CONTROLLER* DAN OPENVSWITCH

IMPLEMENTATION OF COMPUTER NETWORK BASED-ON SOFTWARE DEFINED NETWORK USING RYU *CONTROLLER* AND OPENVSWITCH

Mohammad Nuruzzamanirridha¹, Indrarini Dyah¹, Yuli Sun Hariyani¹

¹Prodi D3 Teknik Telekomunikasi, Fakultas Ilmu Terapan, Universitas Telkom

mohammadnuruz@gmail.com, indrarini@telkomuniversity.ac.id, yulisun@tass.telkomuniversity.ac.id

Abstrak

Software Defined Network (selanjutnya disingkat SDN) merupakan sebuah paradigma/konsep baru dalam sebuah jaringan. Berbeda dengan jaringan konvensional yang fungsi *forwarding* dan fungsi kontrol pada satu perangkat, konsep SDN melakukan pemisahan *forwarding* dan kontrol sehingga fungsi tersebut dalam perangkat yang berbeda. OpenFlow merupakan standar protokol yang memisahkan fungsi *forward* dengan fungsi kontrol untuk memudahkan pengelolaan jaringan skala besar SDN. Penelitian yang telah dilakukan sebelumnya menggunakan Raspberry Pi dengan Ryu sebagai fungsi kontrol dan TP-Link WR1043ND v2 dengan *firmware* Open vSwitch telah memberikan hasil berupa pengukuran *bandwidth* dan *link down*[3]. Pada jurnal kali ini adalah lanjutan dari penelitian sebelumnya dengan konfigurasi *Switch* yang lebih stabil, pengujian jaringan dengan *static routing* dan *Spanning Tree Protocol* (STP), serta parameter pengukuran berupa *bandwidth*, *jitter*, *packet Loss*, *delay*, dan *throughput*.

Kata Kunci : *Software Defined Network*, OpenFlow, Raspberry Pi, TP-Link WR1043ND, Open vSwitch, OpenWRT

Abstract

Software Defined Network (abbreviated as SDN) is a new paradigm/concept in a network. In contrast to conventional networks that forwarding functions and control functions in one, SDN concept of the separation of forwarding and control in different devices. OpenFlow is a standard protocol that separates the functions between forwarding and control to facilitate large-scale networks SDN. The research that has been done before using a Raspberry Pi with Ryu as a control function and TP-Link WR1043ND v2 with *firmware* Open vSwitch has resulted in a measurement *bandwidth* and *link down*[3]. At this time the journal is a continuation of previous studies with more stable *Switch* configuration, network testing using *Static Routing* and *Spanning Tree Protocol* (abbreviated as STP), and then measures its parameters such as *bandwidth*, *jitter*, *packet loss*, *delay*, and *throughput*.

Keywords: *Software Defined Network*, OpenFlow, Raspberry Pi, TP-Link WR1043ND, Open vSwitch, OpenWRT

1. Pendahuluan

Dewasa ini pertumbuhan penggunaan teknologi informasi dan komunikasi bertambah pesat seiring dengan bertambahnya kebutuhan layanan data. Kebutuhan dari pengguna itu sendiri menyebabkan kenaikan trafik dan menambah kompleksitas jaringan. Namun, infrastruktur jaringan saat ini masih dianggap belum mampu menangani kebutuhan tersebut sehingga muncul konsep jaringan baru yang disebut *Software Defined Network* (selanjutnya disingkat SDN). SDN merupakan sebuah metode dalam jaringan komputer yang sistem kontrolnya dipisahkan dari perangkat keras. Metode ini dikembangkan sekitar tahun 2008 di *University of California*, *Barkley* dan *Stanford University*.

Berdasarkan penelitian sebelumnya yang menggunakan Raspberry Pi sebagai *controller* dan TP-Link WR1043ND v2 dengan Open vSwitch sebagai *switch*. Namun, penelitian sebelumnya hanya mengimplementasikan jaringan SDN untuk mengecek kompatibilitas pada perangkat, pengujian menggunakan satu SDN *Applications*, dan parameter yang diukur adalah *bandwidth* dan *link uptime* saja. Oleh karena itu penelitian SDN yang dilakukan ini melakukan pengujian *static routing* dan *Spanning Tree Protocol* serta mengukur parameter-parameter performansi jaringan yaitu *bandwidth*, *delay*, *jitter*, *packet loss* dan *throughput*.

2. Dasar Teori

2.1 Ryu Controller [18]

Ryu merupakan sebuah *framework* SDN yang berfungsi sebagai OpenFlow *Controller*. Nama Ryu sendiri diambil dari bahasa Jepang yang mempunyai arti “aliran”. *Framework* ini bersifat *open source* dibawah lisensi Apache 2.0 dan didukung dengan API yang memudahkan *developer* untuk membangun sebuah jaringan SDN yang mempunyai kemampuan *network management* dan *control application*.

Ryu juga mendukung beberapa protokol untuk melakukan manajemen perangkat jaringan, diantaranya adalah OpenFlow, NetConf, dan OF-Config. Untuk versi OpenFlow sendiri, Ryu mendukung OpenFlow mulai dari versi 1.0, 1.2, 1.3, dan 1.4.

2.2 Open vSwitch [16]

Open vSwitch adalah sebuah aplikasi yang merepresentasikan *multilayer virtual switch* dibawah lisensi Apache 2.0. Open vSwitch di desain untuk mendukung perubahan arsitektur jaringan dengan teknologi automasi jaringan menuju *programmatic extension*. Open vSwitch juga mendukung *standard management interface* dan protokol seperti NetFlow, sFlow, IPFIX, RSPAN, CLI, LACP, 802.1ag. Selain itu, Open vSwitch juga dapat digunakan sebagai *distribution* antar *platform server* yang berbeda.

2.3 OpenWRT [17]

OpenWrt adalah sebuah sistem operasi embedded berbasis Linux kernel yang digunakan untuk perangkat *embedded* sebagai perutean trafik. Komponen utama OpenWrt adalah Linux kernel, util-linux, uClibc, dan busybox.

Fitur dari OpenWrt adalah konfigurasi dasar secara GUI atau CLI. Selain itu, ada sekitar 3500 *packages*, salah satunya adalah Open vSwitch yang tersedia sebagai aplikasi dan dapat di instal menggunakan *opkg packages management system*.

2.4 Spanning Tree Protocol [11]

Spanning Tree Protocol adalah protokol yang berfungsi untuk mengantisipasi terjadinya *broadcast storm* sehingga terjadi *loop* data secara terus menerus. Fungsi utama adalah mencegah terjadinya *loop* dan digunakan sebagai sarana untuk redundansi jaringan jika terjadi kegagalan jaringan. Ada berbagai jenis STP, termasuk STP, RSTP, PVST+, dan MSTP.

Spanning Tree Protocol (STP: 802.1D IEEE) menangani jaringan sebagai *logic tree* dan dengan menetapkan *port* setiap *switch* (kadang-kadang disebut *bridge* di bagian ini) untuk menghindari *loop*.

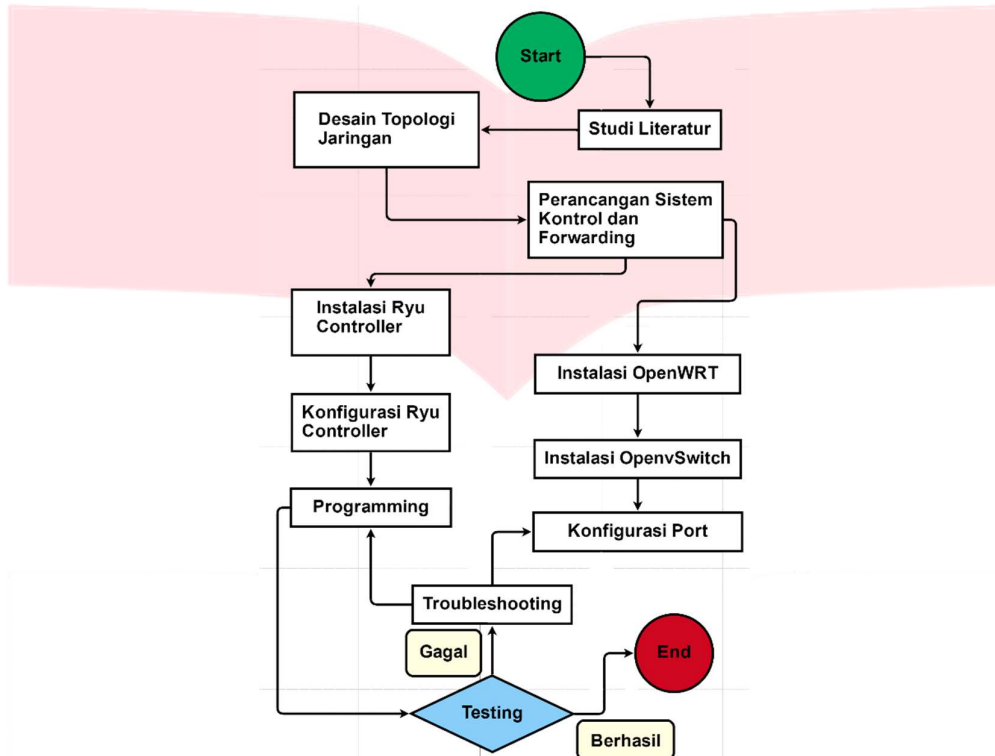
Dengan STP, paket *Bridge Protocol Data Unit* (BPDU) saling bertukar antar *bridge* untuk membandingkan *bridge* dan *port* untuk memutuskan data melalui *port* yang tersedia. Prosedur yang lebih spesifik adalah berikut :

- a. Memilih *Root Bridge*
- b. Memilih peran *port*
 - Root Port*
Port memiliki *cost* terkecil di antara *Bridge* untuk mencapai *Root Bridge*. *Port* ini menerima paket BPDU dari *Root Bridge*.
 - Designated port*
Port di sisi memiliki *cost* kecil untuk mencapai *Root Port* pada setiap *link*. *Port* akan mengirimkan paket BPDU yang diterima dari *Root Bridge*.
 - Non Designated Port*
Port selain *Root Port* dan *Designated Port*. *Port* ini mencegah *frame transfer*.
- c. Perubahan status *Port*
Setelah perhitungan STP selesai, masing-masing *port* dalam status *LISTEN*. Setelah itu, perubahan status sesuai dengan peran masing-masing *port*, akhirnya *port* akan memutuskan untuk *FORWARD* atau *BLOCK*. *Port* dengan status *DISABLE* tidak akan melakukan perubahan apa-apa.

3. Pembahasan

3.1 Tahapan Penelitian

Dibawah ini merupakan tahapan penelitian yang dilakukan dalam menyusun Proyek Akhir ini.

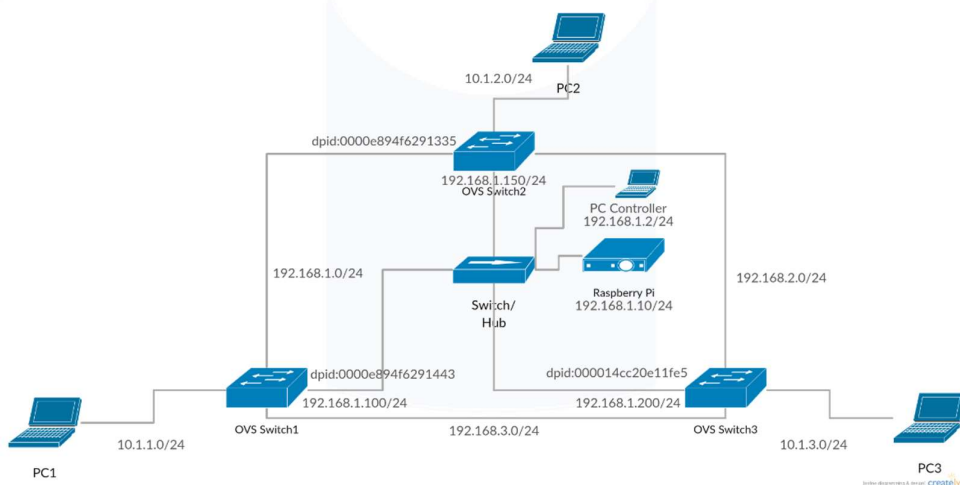


Gambar 1 : Tahapan Penelitian

3.2 Topologi Jaringan

Pada proyek akhir ini, topologi jaringan yang digunakan untuk melakukan pengujian mengikuti penelitian sebelumnya. Hal ini dimaksudkan agar spesifikasi perangkat dan skenario pengujian sama sehingga parameter-parameter yang belum diukur pada penelitian sebelumnya dapat diambil dan diukur performansinya.

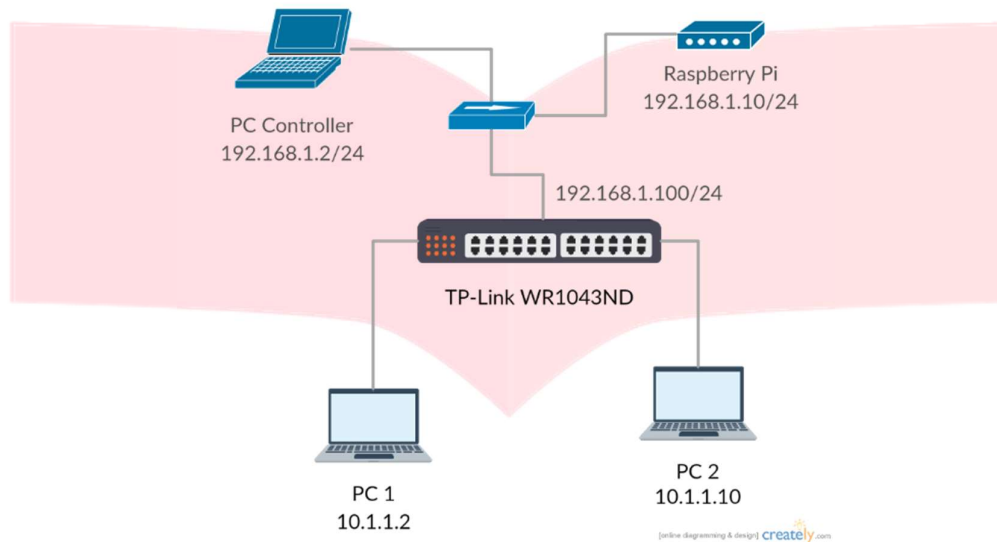
Berikut ini adalah topologi jaringan yang digunakan :



Gambar 2 : Topologi jaringan yang digunakan

3.3 Pengujian Integrasi

Skenario pengujian integrasi pada penelitian ini dilakukan dengan menggunakan satu sampel *switch* yang dapat mewakili semua *switch*. Berikut ini adalah topologi yang digunakan :



Gambar 3 : Topologi jaringan untuk pengujian integrasi SDN

Berdasarkan hasil pengujian, dapat ditarik kesimpulan bahwa Ryu *controller* dan Open vSwitch dapat saling terhubung dan terintegrasi yang dibuktikan dengan `is_connected : true`.

```

192.168.1.1 - PuTTY
root@OpenWrt:/# ovs-vsctl show
b31143a3-0a0f-471c-8420-073db9fcc09d
  Bridge ovs-br
    Controller "tcp:192.168.2.100:6633"
    is_connected: true
    Port "eth0"
      Interface "eth0"
    Port ovs-br
      Interface ovs-br
        type: internal
root@OpenWrt:/#
    
```

Gambar 4 : Jaringan SDN telah terintegrasi

3.4 Pengujian Jaringan SDN

Pengujian jaringan SDN dengan menggunakan dua SDN *Applications*, yaitu fungsi *switching* menggunakan *Spanning Tree Protocol (STP)* dan fungsi *routing* menggunakan *Static Routing*. Topologi yang digunakan sesuai dengan gambar 2.

Spanning Tree Protocol

Tabel 1 : Konfigurasi IP pada skenario STP

No	Perangkat	IP Address	Netmask
1	PC 1	192.168.1.1	255.255.255.0
2	PC 2	192.168.1.2	255.255.255.0
3	PC 3	192.168.1.3	255.255.255.0

Static Routing

Tabel 2 : Konfigurasi IP pada skenario Static Routing

No	Perangkat	IP Address	Netmask
1	PC 1	10.1.1.10	255.255.255.0
2	PC 2	10.1.2.10	255.255.255.0
3	PC 3	10.1.3.10	255.255.255.0

Namun ada beberapa kendala yang terjadi saat melakukan pengujian dengan menjalankan *static routing*. Beberapa kendala yang terjadi saat pengujian adalah:

1. topologi yang digunakan berpotensi adanya *broadcast storm* pada switch sehingga trafik data menjadi padat. Namun, kendala *broadcast storm* ini tidak ditemukan pada saat melakukan simulasi menggunakan *Mininet emulator*
2. Pada pengujian *controller* dengan memutus sementara *link* yang terhubung dengan *switch* mengalami masalah dengan peringatan *dpset : multiple connections* pada *Ryu controller* sehingga *switch* yang terhubung kembali tidak terdaftar dan mengakibatkan jaringan tidak saling terhubung

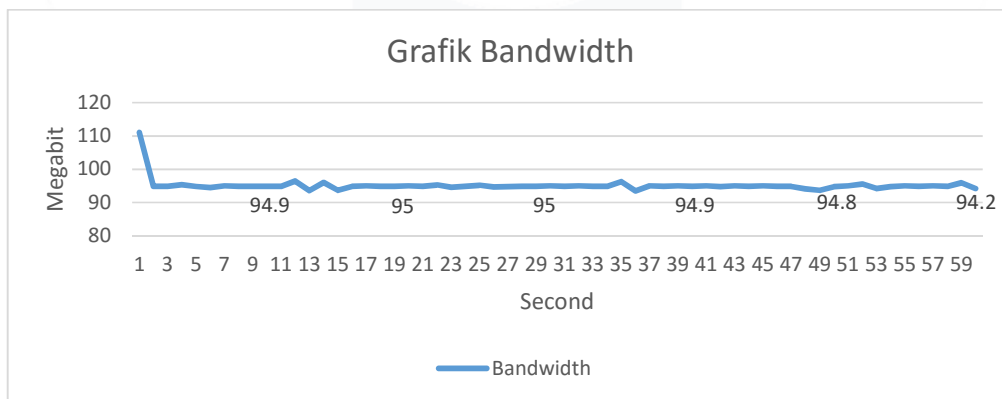
3.5 Pengukuran Performansi Jaringan SDN

Pengukuran performansi jaringan SDN ini diukur dengan topologi jaringan seperti pada gambar 2 dan skenario *Spanning Tree Protocol* dengan melakukan pengiriman data dari PC 1 ke PC 3. Adapun hasil pengukuran yang diperoleh sebagai berikut:

Bandwidth

Pengukuran *bandwidth* dilakukan dengan menggunakan aplikasi *Jperf* dengan mode *server-client*. Pengiriman data dilakukan selama 60 detik sehingga didapatkan hasil pengukuran *bandwidth* rata-rata jaringan SDN adalah 95.164 Mbit/sec.

Berikut ini adalah *screenshot* hasil pengukuran *bandwidth*:

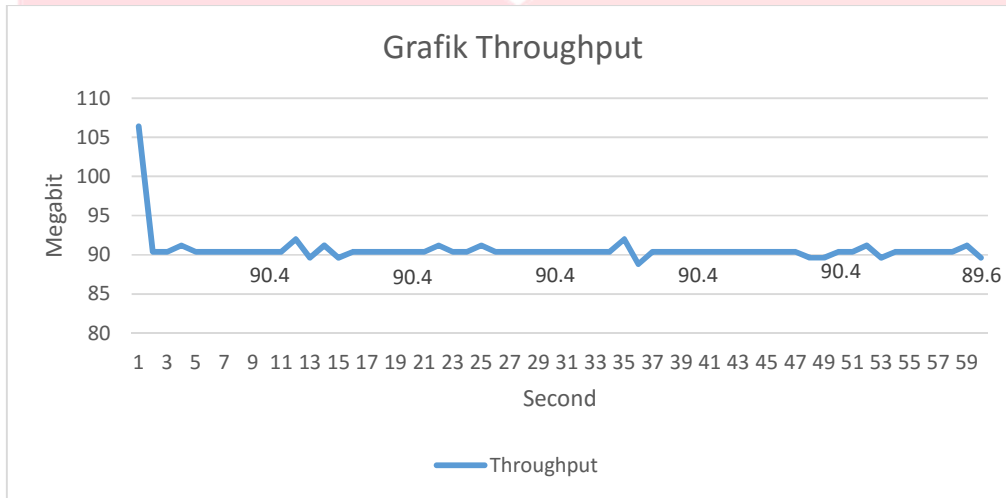


Gambar 5 : Hasil pengukuran *Bandwidth* menggunakan *Jperf*

Throughput

Pengukuran *throughput* dilakukan untuk mengetahui kemampuan jaringan dalam mentransmisikan data secara aktual. Pengujian dilakukan dalam waktu 60 detik, sehingga didapatkan *throughput* pada jaringan SDN adalah 90.693 Mbit/sec.

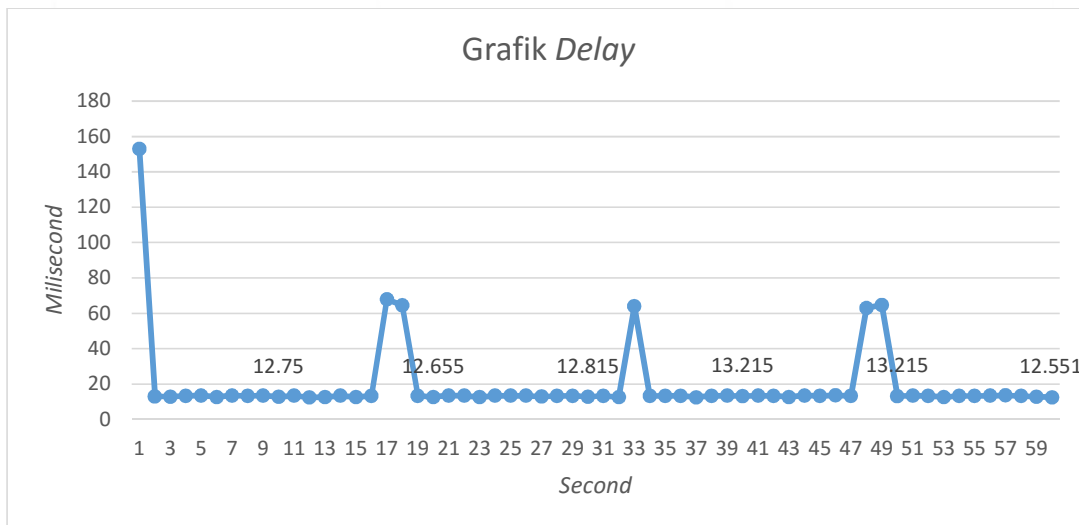
Berikut ini adalah gambar *throughput* yang sudah diolah dalam bentuk grafik :



Gambar 6 : Grafik *Throughput*

Delay

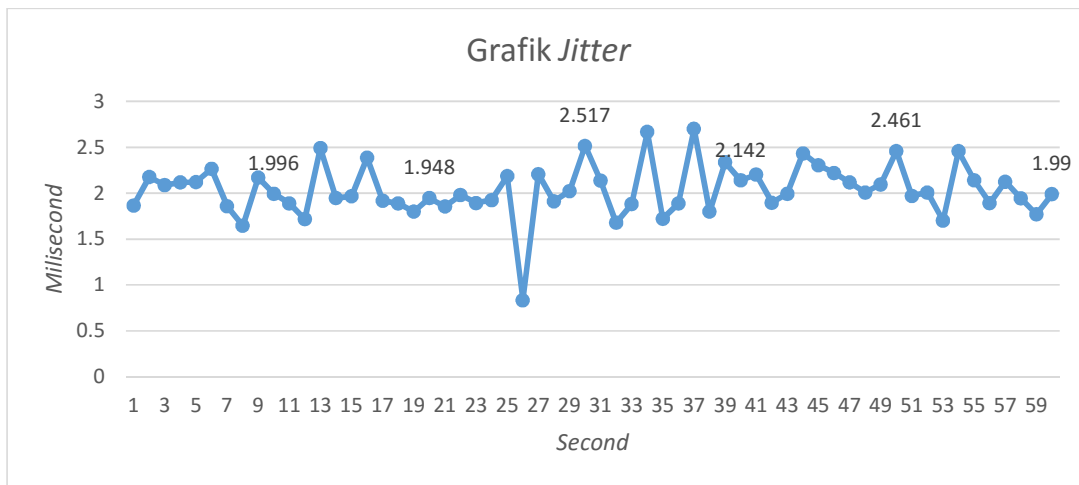
Pengukuran *delay* pada penelitian ini dilakukan menggunakan hrPing dan grPing dengan beban sebesar 65500 bytes dalam waktu 60 detik, sehingga didapatkan hasil sebagai berikut :



Gambar 7 : Grafik *Delay*

Jitter

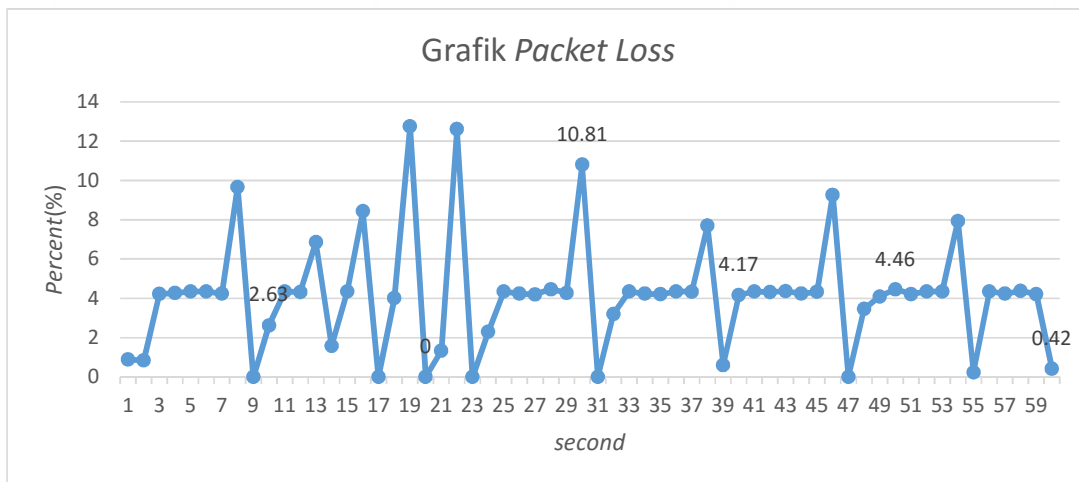
Pengukuran *jitter* pada penelitian ini dilakukan dalam 60 detik sehingga didapatkan hasil sebagai berikut :



Gambar 8 : Grafik Jitter

Packet Loss

Pengukuran *packet loss* pada penelitian ini dilakukan dalam jangka waktu 60 detik sehingga didapatkan hasil sebagai berikut :



Gambar 9 : Grafik Packet Loss

4. Kesimpulan

Berdasarkan hasil dari proyek akhir yang telah dilakukan, maka dapat ditarik kesimpulan sebagai berikut:

1. Implementasi jaringan komputer berbasis SDN yang memisahkan fungsi kontrol dengan fungsi forwarding menggunakan Ryu controller dan Open vSwitch pada perangkat Raspberry Pi dan TP-Link WR-1043ND dapat menjalankan routing dan switching. Namun masih terdapat kendala dalam menjalankan routing, yaitu:
 - a. topologi yang digunakan berpotensi adanya broadcast storm pada switch sehingga traffic data menjadi padat. Namun, kendala broadcast storm ini tidak ditemukan pada saat melakukan simulasi menggunakan Mininet emulator

- b. Pada pengujian controller dengan memutus sementara link yang terhubung dengan switch mengalami masalah dengan peringatan dpset : multiple connections pada Ryu controller sehingga switch yang terhubung kembali tidak terdaftar dan mengakibatkan jaringan tidak saling terhubung
2. Implementasi jaringan komputer berbasis SDN telah selesai dibuat dan diuji performansinya menggunakan parameter bandwidth, throughput, delay, jitter dan packet loss dengan hasil berikut:
 - a. Rata-rata bandwidth dari hasil proyek akhir ini adalah 95.163 Mbit/sec
 - b. Rata-rata throughput dari hasil proyek akhir ini adalah 90.693 Mbit/sec
 - c. Rata-rata delay dari hasil proyek akhir ini adalah 19.212 ms
 - d. Rata-rata jitter dari hasil proyek akhir ini adalah 1.954 ms
 - e. Rata-rata packet loss dari hasil proyek akhir ini adalah 4.3 %

Daftar Pustaka

- [1] Braun, Wolfgang., Michael Menth. (2014). Software-Defined Networking Using OpenFlow: Protocols, Applications and Architectural Design Choice, Germany: Future Internet 2014,6
- [2] "Broadcast storm" [Online] Diakses pada 2 Agustus 2016 Available: <https://www.techopedia.com/definition/6270/broadcast-storm>
- [3] Hariyani. Yuli Sun., Indrarini Dyah Irawati., Danu Dwi S., Mohammad Nuruzzamanirridha(2015). Routing Implementation Based-on Software Defined Network using Ryu Controller and OpenvSwitch. Jurnal Teknologi 78:5, 295-298
- [4] "Jaringan OpenFlow." [Online] Diakses pada 13 November 2014 Available: <http://ardisragen.net/jaringan-openflow.html>
- [5] Kim, Hyunmin., Jaebeom Kim., Young-Bae Ko.(2014).Developing a Cost-Effective OpenFlow Testbed for Small-Scale Software Defined Networking, Korea: ICACT2014.
- [6] "Model B Hardware General Specifications" [Online]. Diakses pada 31 Juli 2016 Available: <http://www.raspberrypi.com/pi/pi-hardware/raspberrypi-model-b/hardware-general-specifications>
- [7] "On data center scale, OpenFlow, and SDN" [Online]. Diakses pada 13 November 2014 Available: <http://bradhedlund.com/2011/04/21/data-center-scale-openflow-sdn/>
- [8] "Open vSwitch Lab 7: Setting OpenWrt" [Online]. Diakses pada 10 Maret 2016 Available: <http://roan.logdown.com/posts/239799-openvswitch-lab-7-setting-openwrt>
- [9] "Re: [Ryu-devel] [maybe a bug] dpset : multiple connections" [Online]. Diakses pada 2 Agustus 2016 Available: <https://www.mail-archive.com/ryu-devel@lists.sourceforge.net/msg10554.html>
- [10] "RPi Hardware" [Online]. Diakses pada 31 Juli 2016 Available: http://elinux.org/RPi_Hardware
- [11] "Ryu SDN Framework." [Online]. Diakses pada 13 November 2014 Available: <http://osrg.github.io/ryu-book/en/Ryubook.pdf>
- [12] "TP-LINK WR1024ND Build OpenvSwitch with OpenWrt". [Online]. Diakses pada 30 Juli 2016 Available: <http://hooootony.logdown.com/posts/259102-tp-link-wr1024nd-build-openvswitch-with-openwrt>
- [13] "TP-Link WR-1043ND v2 datasheet". [Online]. Diakses pada 31 Juli 2016 Available: http://www.tp-link.com/en/download/TL-WR1043ND_V2.html
- [14] "VNC (Virtual Network Computing)". [Online]. Diakses pada 1 Agustus 2016 Available: <https://www.raspberrypi.org/documentation/remote-access/vnc/>
- [15] "What is OpenFlow?." [Online] Diakses pada 13 November 2014 Available: <http://archive.openflow.org/wp/learnmore/>
- [16] "What is OpenvSwitch?." [Online] Diakses pada 13 November 2014 Available: <http://openvswitch.org/>
- [17] "What is OpenWrt?." [Online] Diakses pada 13 November 2014 Available: <https://openwrt.org/>
- [18] "What is Ryu?." [Online] Diakses pada 13 November 2014 Available: <http://osrg.github.io/ryu/>
- [19] Yamahata, Isaku.(2013) Ryu: SDN framework and Python experience, Japan:Pycon APAC 2013, September 14