

TestBed Open vSwitch Raspberry Pi Pada Skala Kecil

Dikla Sasta Wijaksa¹, Rina Mardiaty², Nanang Ismail³, Tutun Juhana⁴

^{1,2,3}Jurusan Teknik Elektro, State Islamic University of Sunan Gunung Djati Bandung

⁴Laboratorium Telematika STEI, Institut Teknologi Bandung
Bandung, Indonesia

e-mail: diklasasta@gmail.com¹, r_mardiaty@uinsgd.ac.id², nanang.is@uinsgd.ac.id³, tutun.j@gmail.com⁴

Abstrak –Testbed Open vSwitch dalam skala kecil pada Raspberry Pi dengan tambahan OpenDaylight sebagai kontroller. Piranti yang digunakan adalah raspberry pi yang telah diinstall dengan aplikasi Open vSwitch. Lalu dikonfigurasi untuk dijadikan sebuah switch OpenFlow yang dapat dikontrol dengan kontroller untuk sesuai kebutuhan. Dengan penambahan usb to eth sejumlah 2 buah untuk menguji apakah sebuah raspberry dapat digunakan sebagai Switch OpenFlow tersebut. Dengan menggunakan 2 client sebagai ujicoba dan 1 kontroller pc untuk digunakan sebagai pengontrol jaringan default. Hasil dari pengujian dengan menggunakan 2 client, 1 switch raspberry dan 1 controller didapatkan dengan menggunakan syntax ping akan didapatkan respon dari client 1 dengan client lainnya. Lalu dilakukan ujicoba pada performansi Open vSwitch dengan menguji response time. Maka dari itu dengan menggunakan 4 devices dan 2 software kita sudah dapat membuat sebuah media ujicoba Software Defined Network sederhana yang dapat dikembangkan lebih lanjut kembali kehidupan masa datang.

Katakunci: Open vSwitch, OpenFlow, Raspberry, Software Defined Network

1. Pengenalan

Pemanfaatan Raspberry Pi sebagai komputer papan yang mempunyai banyak manfaat serta didesain minimalis hanya sebesar kartu kredit. Komputer ini digunakan karena telah banyak memenuhi kebutuhan dari riset yaitu beberapa diantaranya desain yang mini dan memiliki fitur yang cukup untuk digunakan dalam riset. Komputer papan ini sangat terjangkau harganya sehingga dalam pemanfaatannya juga cukup banyak hingga saat ini. Komputer ini umumnya digunakan baik dari semua kalangan yaitu pelajar, mahasiswa, dosen maupun pekerja. Tujuan dari penggunaan ini juga cukup luas sesuai dengan kebutuhan yang diinginkan. Salah satu diantara yang dapat dimaksimalkan adalah menamkan software yang diinginkan untuk melakukan pemecahan masalah yang ada dengan melakukan kombinasi yang diinginkan.

Operating System yang dapat digunakan dalam komputer ini bermacam-macam, sesuai dengan kebutuhan yang diinginkan. Macam-macam dari OS yang ada adalah dibagi menjadi 2 jenis. Pertama adalah sistem operasi yang berbasis linux dan yang kedua sistem operasi berbasis windows. Untuk sistem operasi berbasis linux diantaranya adalah raspbian, ubuntu-mate dll. Sistem operasi berbasis windows untuk raspberry adalah windows 10 IoT. Beragamnya sistem operasi yang tersedia pada Raspberry Pi maka pengujian menggunakan sistem operasi raspbian karena mirip dengan linux distribusi debian dalam syntaks yang digunakan. Jadi dengan begitu akan lebih memudahkan dalam pengujian karena telah terbiasa menggunakan sistem operasi distribusi tersebut.

Software untuk Raspberry Pi yang digunakan sebagai *switch openflow* adalah bermacam-macam. Disini digunakan Open vSwitch sebagai aplikasi *switch* yang mempunyai kemudahan untuk konfigurasi dan instalasi. Aplikasi ini sudah banyak teruji dan masih dikembangkan oleh Open vSwitch Foundation dalam penggunaan kedepannya. Open vSwitch mempunyai berbagai

fasilitas yang dapat digunakan untuk keamanan, *monitoring*, automasi kontrol dan *Quality of Services*. Yaitu sebagai keamanan biasa digunakan untuk VLAN, isolation, traffic filtering. Selanjutnya untuk *monitoring* yaitu Netflow, sFlow, SPAN, RSPAN. Lalu penggunaan untuk Automasi kontrol adalah sebagai OpenFlow, OVSD, mgmt, dan protocol. Terakhir fungsi sebagai QoS adalah untuk *traffic queuing* dan *traffic shapping*.

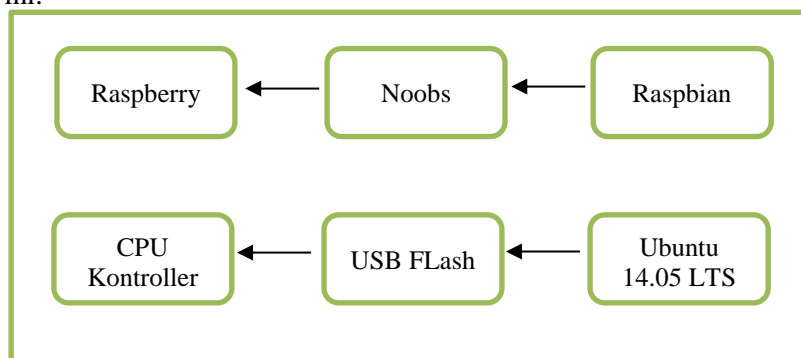
Pengujian Open vSwitch dilakukan dengan menambahkan kontroller untuk mengatur traffic *openflow* yang ditentukan. Kontroller ditambahkan terpisah pada device sendiri dan akan terhubung langsung dengan Raspberry Pi. Kontroller yang digunakan adalah dari OpenDaylight Foundation yaitu dengan lisensi *opensource*. Dipilihnya kontroller ini karena kemudahan dari sumber dan konfigurasi yang ada. Sudah banyak pengembang yang melakukan ujicoba dengan kontroller tersebut. Kontroller tersebut juga terdapat banyak versi sehingga dapat disesuaikan dengan kebutuhan yang ada. Dengan menggunakan karaf distribusi dari kontroller OpenDaylight maka lebih mudah dalam *feature* yang diinginkan sehingga hanya fitur tertentu yang dapat ditambahkan dalam kontroller agar lebih efektif dan efisien.

Kebutuhan akan switch yang dapat dikonfigurasi dengan fleksibel mendasari penulis untuk melakukan pengujian Open vSwitch pada Raspberry Pi tersebut. Dengan melakukan pengujian dengan skala kecil ini, dengan begitu jika dalam pengujian skala kecil tersebut berhasil diharapkan dapat disesuaikan untuk pengujian selanjutnya yaitu pada skala menengah. Masalah yang menjadi fokus pengujian adalah kinerja dari Raspberry sebagai *switch* adalah dapat dilaksanakan dan diimplementasikan serta sesuai kelayakan dari switch yang ada sesuai dengan standard ketentuan dari ITU-T.

2. Metode Riset

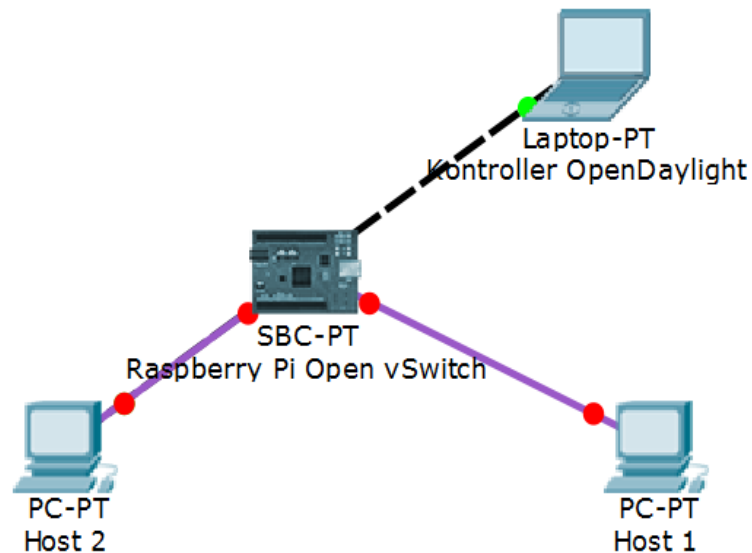
Perancangan dan pengujian dari Open vSwitch dilakukan dengan beberapa tahap yang dilakukan. Tahapan-tahapan tersebut dimaksudkan agar dalam pengujian telah sesuai dengan kondisi yang baik dan benar. Perancangan dari Open vSwitch dilakukan dengan 4 tahapan. Tahapan yang pertama adalah persiapan mengenai alat yang digunakan. Selanjutnya adalah instalasi dari software-software yaitu Raspbian, Open vSwitch, Ubuntu 14.05 *Long Term Services*, OpenDaylight kontroller. Tahapan ketiga yaitu konfigurasi dari Open vSwitch dan OpenDaylight kontroller. Selanjutnya yaitu implementasi dengan topologi skala kecil.

Proses instalasi dari Raspbian menggunakan NOOBS yaitu *instalation assistant* dengan sistem operasi pilihan dari linux. Sistem operasi yang digunakan adalah dari distribusi Raspbian, hal ini mempunyai kesamaan mengenai sintaks dan *command* yang ada di raspbian dengan distribusi debian. Hal ini memudahkan penulis karena dalam sintaks telah lebih biasa menggunakan sistem distribusi dari debian. Selanjutnya instalasi Ubuntu yaitu menggunakan bootloader usb flashdisk untuk memudahkan dalam proses instalasi. Pemilihan sistem operasi Ubuntu dikarenakan sesuai dengan ayah dari sistem operasi ini yaitu dari debian maka sintaks dan *command* yang digunakan lebih familiar. Blok alur kerja dari instalasi dapat dilihat pada blok dibawah ini.



Gambar 1. Blok Kerja Perancangan

Proses instalasi telah usai dilanjut dengan proses perancangan, yaitu perancangan mengenai topologi dan konfigurasi yang diinginkan. Proses perancangan ini dilakukan dengan mendesain topologi terlebih dahulu. Topologi yang di desain terdiri dari 2 host, 1 ovs dan 1 kontroller. Yaitu terdapat pada gambar berikut.



Gambar 2. Topologi Rangkaian Testbed

Konfigurasi pada OVS dilakukan secara individu dengan menggunakan referensi dari sumber website Open vSwitch. Konfigurasi pada OVS dilakukan dengan penambahan port eth0 dan tambahan menggunakan usb to ethernet untuk eth1. Selanjutnya untuk kontroller dihubungkan menggunakan usb to ethernet untuk eth2. Untuk konfigurasi OVS dilakukan dengan sintaks berikut.

```

//setting interface
$ sudo ovs-vsctl add-br ovs0
$ sudo ovs-vsctl add-port eth0
$ sudo ovs-vsctl add-port eth1

//setting controller
$ sudo ovs-vsctl set-controller ovs0 tcp:192.168.1.2:6600
$ sudo ovs-vsctl get-controller

//setting protocols
$ sudo ovs-vsctl set bridge br1 stp_enable=false
$ sudo ovs-vsctl set brige br1 protocols=OpenFlow13
$ sudo ovs-vsctl clear brige br1 protocols

//setting failure mode
$ sudo ovs-vsctl set-fail-mode br1 standalone
$ sudo ovs-vsctl set-fail-mode br1 secure
$ sudo ovs-vsctl del-fail-mode br1
$ sudo ovs-vsctl get-fail-mode br1

//setting flow
$ sudo ovs-ofctl add-flow ovs0

```

Gambar 3. Sintaks untuk Setting OVS

Implementasi dari software defined network dengan menggunakan raspberry dalam skala kecil yaitu dengan melakukan konfigurasi port pada OVS. Selanjutnya ditambahkan mengenai konfigurasi dari OpenDaylight yaitu dengan menambahkan aturan flow yang diharapkan. Untuk lebih mempermudah adalah dengan tabel berikut:

Tabel 1. Flow untuk Port yang Digunakan pada OVS

No	Input	Output	Keterangan
1	Port eth0	Port eth1	Flow untuk host 1 ke host 2
2	Port eth1	Port eth0	Flow untuk host 2 ke host 1

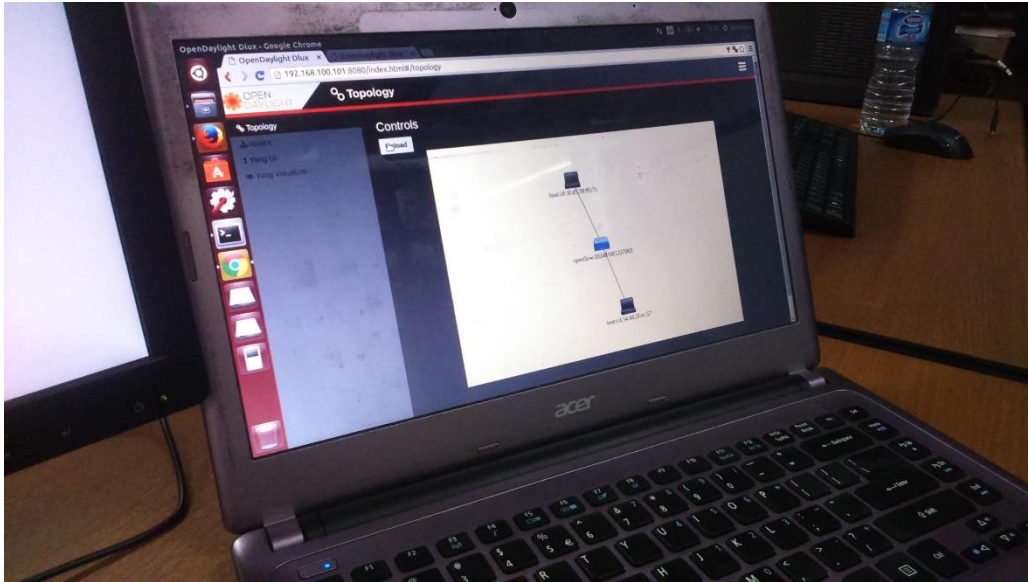
3. Hasil dan Analisis

Perancangan dan pengujian telah siap dilakukan. Pengujian kali ini dilakukan dengan mencoba ping antara host 1 dan host 2 dengan sejumlah byte data tertentu. Pengujian

- Aplikasi dalam Raspberry Pi Hanya bawaan default dan OVS
- Jumlah host dari OVS hanya terdapat 2 buah host

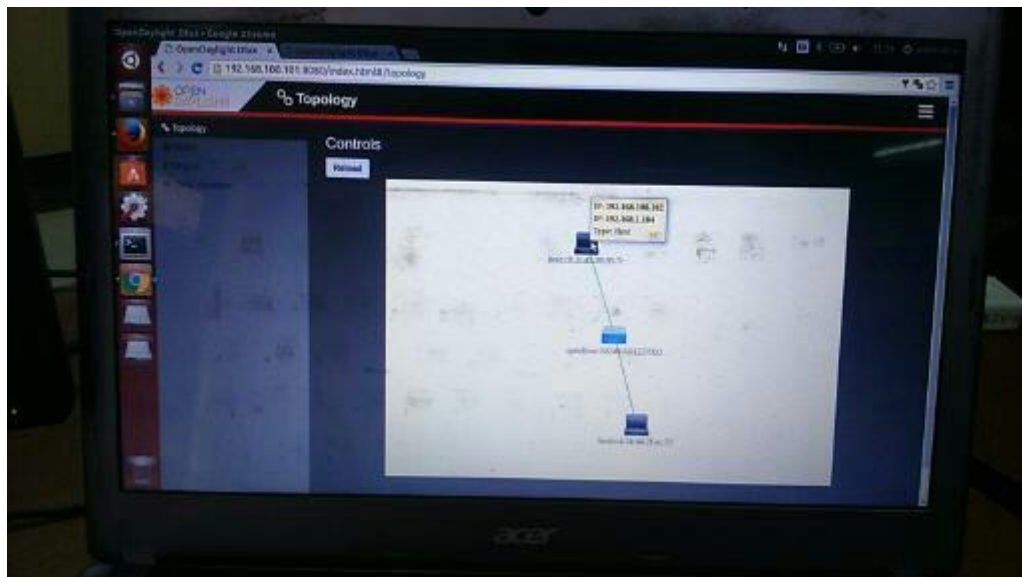
3.1. Hasil

Konfigurasi yang telah dilakukan telah berhasil maka akan dapat ditampilkan gambar dari topologi dalam OpenDaylight controller. Gambar tersebut merupakan desain topologi yang digunakan saat ini untuk melihat host yang tersambung ke dalam jaringan.



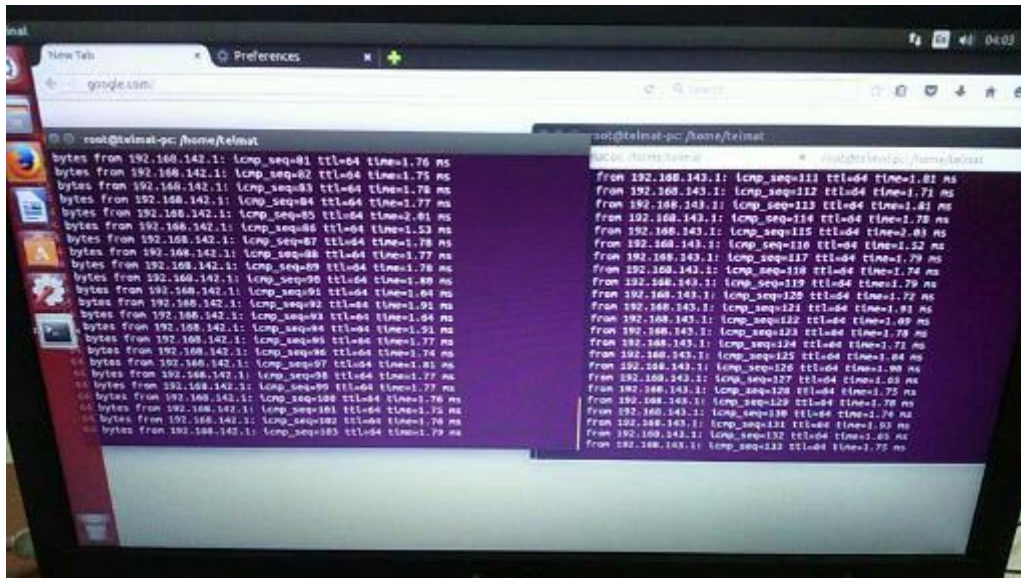
Gambar 4. Topologi pada OpenDaylight controller

Selanjutnya gambar dibawah ini merupakan hasil desain dari topologi uji coba untuk mengetahui apakah host telah terkoneksi dengan baik apakah belum sesuai dengan topologi yang dikehendaki. Tampak 2 host dan 1 switch. Karena untuk controller tidak masuk dalam port ovs. Maka tidak akan tertampil pada jaringan, namun akan tetap dapat digunakan.



Gambar 5. Desai Uji Coba Jaringan

Selanjutnya adalah gambar dibawah ini tampilan dari pengujian response times yang dilakukan untuk mengetahui seberapa cepat response yang diberikan jika ping dilakukan antara host satu ke yang lainnya.



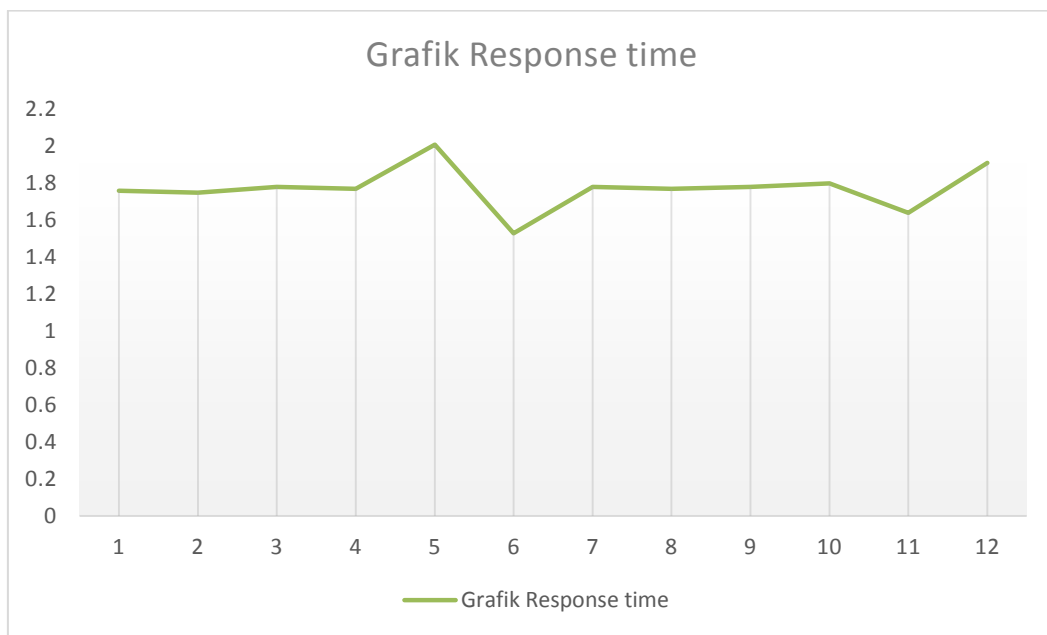
Gambar 6. Pengujian Response Times

Pengujian antara host 1 ke 2 yaitu dengan dilakukannya ping untuk mengetahui koneksi antar host 1 ke 2. Pengujian ini digunakan dengan jumlah byte yang dikirim sejumlah 64, dengan beberapa *sequence* urutan. Untuk lebih mudah nya dapat dilihat dalam bentuk tabel berikut.

Jumlah byte 64 byte

Tabel 2. Response Time Host to Host

Detik	Jumlah byte	responses time
1	64	1.76
2	64	1.75
3	64	1.78
4	64	1.77
5	64	2.01
6	64	1.53
7	64	1.78
8	64	1.77
9	64	1.78
10	64	1.8
11	64	1.64
12	64	1.91



Gambar 7. Grafik Response Time Host to Host

Perhitungan mean response time

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

$$\bar{x} = \frac{1.76 + 1.75 + 1.78 + 1.77 + 2.01 + 1.53 + 1.78 + 1.77 + 1.78 + 1.8 + 1.64 + 1.91}{12}$$

$$\bar{x} = \frac{21.28}{12}$$

$$\bar{x} = 1.7733$$

Rata-rata response time yang diberikan adalah 1.7733 ms

3.2. Analisis

Dari hasil ping antara host 1 ke 2 yang telah dilakukan menunjukkan bahwa grafik untuk response waktu yang diberikan mempunyai grafik yang sedemikian hingga mendekati linier. Dari grafik tersebut terlihat bahwa kinerja merupakan stabil untuk kondisi tersebut. Walaupun terdapat penurunan dan kenaikan pada waktu tertentu, lalu rata-rata dari response yang diberikan adalah 1.773 ms untuk jumlah byte 64. Dari data tersebut dapat dikatakan sesuai dengan beban data sebesar 64 byte telah mencapai 1.773 bahwa sudah cukup dalam kinerja yang diberikan karena dengan delay yang tidak terlalu besar.

Hasil pengujian untuk 2 host dan 1 switch adalah seperti pada hasil yang telah disampaikan. Namun untuk menguji dengan host yang lebih banyak dan dengan menggunakan beberapa switch akan dilakukan dikemudian hari. Karena keterbatasan yang dimiliki oleh peneliti. Maka peneliti telah merasa cukup untuk melakukan uji coba hingga saat ini. Hasil yang

diharapkan telah mencukupi dari harapan karena dalam pengujian telah didapat response time yang cukup. Sehingga dapat dilanjutkan dengan pengujian pada jaringan yang lebih kompleks dan rumit lagi.

4. Kesimpulan

Dari hasil pengujian yang telah dilakukan maka didapatkan hasil yang dapat dilihat bahwa dalam *testbed* Open vSwitch telah dapat dilihat bahwa dapat diimplementasikan dalam skala kecil. Pengujian yang dilakukan didapat delay yang tidak besar dan cukup handal untuk Open vSwitch skala kecil. Dengan menggunakan OpenDaylight sebagai kontroller maka didapatkan kinerja yang cukup untuk mengontrol Open vSwitch dalam skala kecil. Diharapkan kedepannya bahwa dapat digunakan untuk pengembangan lebih jauh lagi karena efektifitas dari alat raspberry yang dirasa cukup untuk dilakukan pengujian yang lebih kembali.

Ucapan Terimakasih

Terimakasih saya ucapkan kepada Staff dan Pengajar Lab Telematika STEI ITB yang mempersilahkan untuk dilakukanya penelitian di tempat tersebut, sehingga dapat terlaksana untuk penelitian yg sedang dilakukan.

Terimakasih saya ucapkan pula atas bimbingan dari bapak dan ibu dosen baik jurusan maupun pembimbing lapangan yang telah membantu dalam proses belajar ini.

Daftar Pustaka

- [1] FIP - UPI, Ilmu & Aplikasi Pendidikan, Bandung: PT. IMTIMA, 2007.
- [2] Kementerian Riset Teknologi dan Perguruan Tinggi, "RISTEKDIKTI," 7 September 2016. [Online]. Available: <http://ristekdikti.go.id>.
- [3] Linux Foundation Collaborative Projects, "Open vSwitch," 7 September 2016. [Online]. Available: <http://openvswitch.org>.
- [4] STEI ITB, "STEI ITB," 7 September 2016. [Online]. Available: <https://stei.itb.ac.id>.
- [5] A. S. W. D. J. Tanenbaum, Computer Networks, New York, Northeastern United States: Prentice Hall, 2011.
- [6] M. Richardson dan S. Wallace, Make: Getting Started With Raspberry Pi, Sebastopol: Maker Media, 2015.