Telecommunications, Networks, Electronics, and Computer Technologies

Vol.3 (1), 2023, 1 - 8

JURNAL UPI

TELNECT

Journal homepage: http://ejournal.upi.edu/index.php/TELNECT/

Simulasi Software Defined Wirelless Network: Skenario Unicasting dan Bicasting

Devi Puspita Dewi[1], Farhan Maulana[2], Titania Emaniar[3], Riyadh Ahmad Faridz[4]*

[1] Universitas Pendidikan Indonesia Kampus Purwakarta, Indonesia
[2] Universitas Pendidikan Indonesia Kampus Purwakarta, Indonesia
[3] Universitas Pendidikan Indonesia Kampus Purwakarta, Indonesia
[4] Universitas Pendidikan Indonesia Kampus Purwakarta, Indonesia

*Devi Puspita Dewi : E-mail: devipuspitadewi@upi.edu.

ABSTRACTS

Software-Define Networking (SDN) merupakan suatu arsitektur yang cocok digunakan untuk aplikasi dan layanan saat ini karena memiliki beberapa kelebihan seperti bersifat dinamis, mudah beradaptasi, dapat dikelola, hemat biaya, dan ideal untuk *bandwidth* yang tinggi. Penelitian ini bertujuan untuk mengukur kecepatan *bandwidth* pada tiga keadaan, yaitu ketika *station* berada pada *range Access Points* 1, *Access Points* 2, dan irisan *Access Points* 1 dan *Access Points* 2. Pada penelitian ini, kami menambahkan ekstensi Mininet-Wi-Fi pada Mininet sehingga dapat ditambahkan AP dan STA. Peneliti membentuk topologi yang terdiri dari beberapa komponen yaitu dua *Access Points* (AP) dan satu *Station* (STA), yang mana AP dihubungkan ke *controller* dan *station* terhubung dengan AP melalui NIC, yangmana pada penelitian ini akan digunakan Iperf untuk mengetest bandwith pada ketiga keadaan. Dari penelitian yang dilakukan, didapatkan hasil ketika STA berada pada range AP 1**,** *bandwidth* maksimum antara 'h1' dengan 'sta1' selama 10 sec yaitu sebesar 8,74 Mbits/s, ketika pada range AP 2, *bandwidth* maksimum antara 'h2' dengan 'sta1' sebesar 8,74 Mbits/s. Dan ketika berada pada irisan AP 1 dan AP 2, *bandwidth* maksimum 18,3 Mbits/s yang mana ketika terjadi bicasting maka bandwidth yang didapatkan menjadi lebih dari 2 kali lipat karena memiliki 2 NIC.

ARTICLE INFO

Article History:
Received 24 Des 2023
Revised 24 Maret 2023
Accepted 01 April 2023
Available online 01 Juni 2023

Keyword:
SDN,
Mininet Wi-Fi,,
Iperf,
Bandwith

1. INTRODUCTION

Software Defined Networking (SDN) merupakan paradigma baru dalam dunia telekomunikasi. *Software Defined Networking* disebut sebagai solusi atas kompleks dan sulit dikendalikannya infrastruktur jaringan saat ini karena sifatnya yang dapat mengukur, mengelola, memprogram, dan melakukan kontrol terpusat infrastruktur jaringan, [1]. SDN juga mampu memberikan solusi atas permasalahan-permasalahan jaringan saat ini, seperti sulitnya mengintegrasikan teknologi baru karena alasan perbedaan perangkat atau platform, kinerja yang buruk karena ada beberapa operasi yang berlebihan pada *protocol layer*, serta sulitnya menyediakan layanan-layanan baru [2] SDN memiliki beberapa kelebihan antara lain dapat dikelola, hemat biaya, mudah beradaptasi, dan bersifat dinamis untuk aplikasi dan layanan saat ini, sehingga dinilai mampu mengikuti perkembangan teknologi, dan ideal untuk *bandwidth* yang tinggi [2]. Arsitektur SDN memisahkan antara *control plane* dan *data plane*, serta sentralisasi jaringan dengan semua pengaturan berada pada *control plane* yang dapat diprogram. Konsep SDN ini mempermudah administrator jaringan dalam mengelola jaringannya.

Baru-baru ini diperkenalkan cabang SDN yaitu *Software Defined Wireless Networking (SDWN)* yang muncul sebagai solusi untuk memenuhi permintaan jaringan nirkabel yang terus meningkat [1]. SDWN digunakan pada jaringan nirkabel untuk menambah fleksibilitas dari meningkatnya lalu lintas data melalui jaringan nirkabel. SDWN melakukan fungsi yang hampir sama dengan SDN, yaitu memindahkan bidang kontrol ke kontrol perangkat lunak eksternal dari bidang data jaringan di luar *Access Points* (APs) [1]. *Access Points* (AP) mengelola *station* (STA) yang terhubung melalui otentikasi dan asosiasi. AP juga menerima instruksi dari *station* dan bertanggung jawab untuk mengirim dan menerima lalu lintas melalui jaringan. SDWN bertujuan untuk menyediakan kontrol terpusat program dari jaringan di luar kotak nirkabel *(Access Points)* yang menegakkan instruksi yang diterima [1]. Dengan adanya kontrol terpusat, maka akan memberikan kemudahan pada administrator jaringan untuk mengelola dan melakukan manajemen trafik dalam jaringan nirkabel [3].

Pada penelitian sebelumnya, sudah terdapat penelitian serupa yang mana membahas mengenai mininet Wi-Fi sebagai alat untuk meniru scenario nirkabel *openflow*/SDN yang mana dilakukan tiga percobaan yaitu *bicasting* melalui jaringan nirkabel, pengintegrasian antarmuka nirkabel fisik, dan mobility [1]. Kemudian terdapat penelitian tentang scenario *bicasting* yang diimplementasikan pada jaringan nirkabel yang mana digunakan untuk mengidentifikasi perbedaan *bandwidth* pada beberapa posisi yang telah ditentukan untuk mengevaluasi kinerja bicasting dan menentukan posisi terbaik saat bicasting terjadi [4]. Penelitian lain yang serupa menggunakan mininet Wi-Fi yaitu penelitian yang dilakukan oleh B.Budi et al yang menggunakan skenario *mobility pada mininet Wi-Fi yang mana dilakukan* Perbedaan penelitian ini dari penelitian-penelitian sebelumnya yaitu pada paper ini….

Pada penelitian ini akan dibahas mengenai simulasi *Software Defined Networking (SDN)* menggunakan Mininet yang telah ditambahkan ekstensi *Mininet-Wi-Fi*, sehingga dapat ditambahkan *access points* (AP) dan *station* (STA) [2]. Protocol yang digunakan pada penelitian ini adalah *protocol openflow.* *Openflow* adalah protokol untuk manajemen mobilitas, pemilihan saluran, mitigasi gangguan dan kontrol lalu lintas untuk beberapa koneksi Wi-Fi [5]. Disini kami menggunakan mininet Wi-Fi karena dianggap memungkinkan untuk dibuat seperti jaringan yang sama seperti sistem jaringan kabel dan jaringan tanpa kabel asli, mininet Wi-Fi berfungsi untuk merancang jaringan dan mengendalikan *end-user* menggunakan satu atau beberapa kontroler. Selain itu, juga akan dilakukan tes pengukuran *bandwidth* dengan *Iperf*.

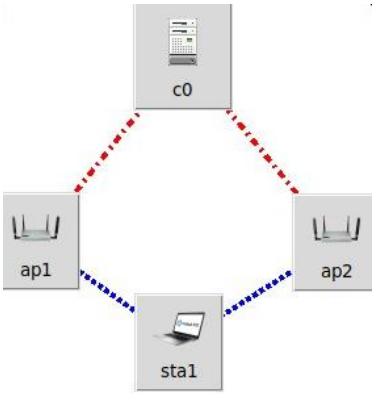
2. MATERIALS AND METHODS

Pada penelitian ini, akan dilakukan pengukuran *bandwith* pada tiga titik sample, yaitu ketika *station* berada di *coverage Access Points* 1 , ketika *station* berada di *coverage Access Points* 2, dan ketika *station* berada pada irisan antara *Access Points* 1 dan 2. Pengukuran *bandwith* dilakukan dengan menggunakan *Iperf*, dimana *Access points* digunakan sebagai *server* dan *Station* digunakan sebagai *client* yang dapat saling terhubung secara *wireless.*

2.1 Desain Topologi Bicasting

Salah satu simulasi yang dilakukan yaitu mensimulasikan konsep bicasting yang mana diimplementasikan pada jaringan wireless. Hasil simulasi nantinya akan mengidentifikasi perbedaan bandwith pada beberapa posisi yang mana digunakan untuk mengevaluasi kinerja

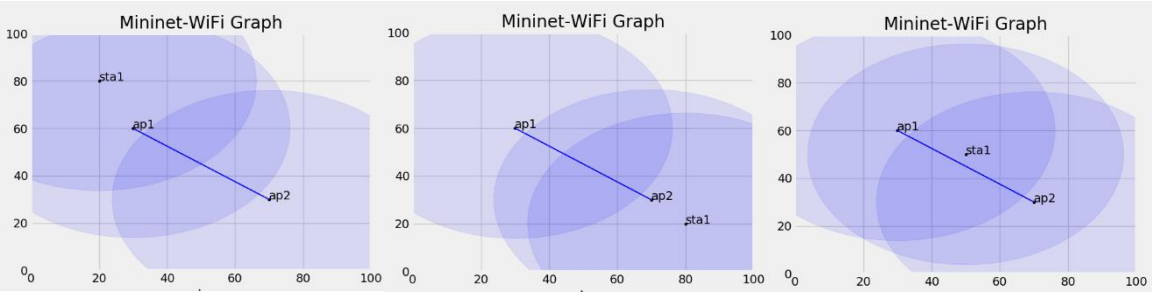
bicasting dan menentukan posisi terbaik ketika terjadinya bicasting. Desain topologi bicasting dapat dilihat pada gambar 1.



Gambar 1. Bicasting Network Topologi

2.2 Desain Simulasi

Topologi yang digunakan dalam percobaan ini memiliki 1 *station* dan 2 *access points.* Dalam penelitian ini, peneliti menggunakan beberapa *sample* untuk diukur keadaan bandwith, diantaranya ketika *station* berada pada *range* AP1, AP2, dan diantara irisan *range* AP1 dan AP2. Gambar 2 merupakan visualisasi dari topologi, yang ditunjukkan pada *Mininet-Wi-Fi Graph*:



Gambar 2. Visualisasi simulasi melalui Mininet-Wi-Fi Graph

Pada penelitian ini, komponen-komponen yang digunakan antara lain : sebuah *controller* ‘c1’, 2 buah *Acces point* yaitu ‘AP1’ dan ‘AP2’, sebuah *station* yaitu ‘sta1’, dan 2 *host* yaitu ‘h1’ dan ‘h2’, yang dimana *host* terhubung dengan masing-masing AP-nya.

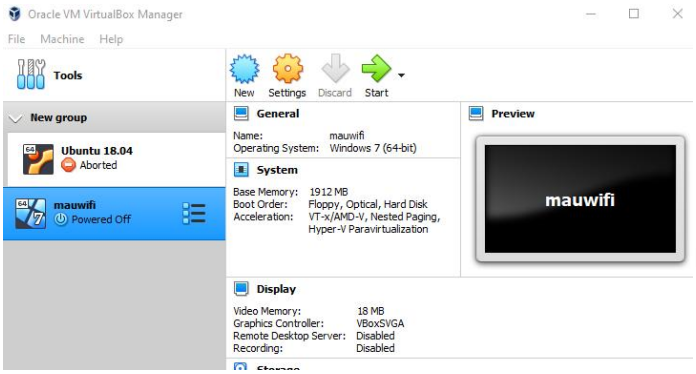
2.3 Parameter Pengujian

Hal yang akan diuji cobakan pada penelitian ini adalah pengukuran *bandwidth* pada station, dimana terdapat tiga pengukuran yaitu ketika *station* berada di *coverage Access Points* 1 , ketika *station* berada di *coverage Access Points* 2, dan ketika *station* berada pada irisan antara *Access Points* 1 dan 2.

2.4 Installasi komponen yang akan digunakan

1. Oracle VM Virtual Box

Kita akan membuat prototype jaringan komputer dengan menggunakan mininet Wi-Fi dan virtualbox sebagai virtual komputer yang baru. Setelah itu, kita jalankan *virtual box* yang telah *terinstall mininet*. Gambar 3 menunjukkan tampilan dari oracle VM VirtualBox



ISSN: 2798-2785

Gambar 3. Oracle VM Virtual box

2. Ubuntu v18.04 dan Mininet-Wi-Fi

Selanjutnya karena pada penelitian ini menggunakan mininet Wi-Fi, maka perlu diinstal terlebih dahulu dengan *command ~$ cd mininet-wifi.* Gambar 4 menunjukkan apabila mininet Wi-Fi telah berhasil di install.

```
farhanm@farhanm-VirtualBox:~$ ls
Desktop    Downloads  mininet-wifi  openflow  Public     Videos
Documents  mininet    Music         Pictures  Templates  wmediumd
farhanm@farhanm-VirtualBox:~$ cd mininet-wifi
farhanm@farhanm-VirtualBox:~/mininet-wifi$
```

Gambar 4. Terminal Ubuntu v18.05 & Mininet-Wi-Fi

2.5 Membangun program simulasi

Setelah semua komponen yang menyokong simulasi sudah terinstall, maka langkah yang dilakukan setelahnya yaitu membuat program. Pada praktiknya, peneliti menggunakan *example* yang ada pada *directory Mininet-Wi-Fi* yaitu *Position.py* dan dimodifikasi sesuai dengan simulasi yang akan diujicobakan, dengan *command: ~/mininet-Wi-Fi/examples$ nano position8.py. Output* yang ditampilkan pada GNU nano 2.9.3

```
GNU nano 2.9.3                                          position8.py

info("*** Creating nodes\n")
ap1 = net.addAccessPoint('ap1', ssid='new-ssid', mode='g', channel='1',
                         failMode="standalone", mac='00:00:00:00:00:02',
                         position='30,60,0')
ap2 = net.addAccessPoint('ap2', ssid='new-ssid', mode='g', channel='2',
                         failMode="standalone", mac='00:00:00:00:00:01',
                         position='70,30,0')

sta1 = net.addStation('sta1', ip='10.0.0.1/8',
                      position='50,50,0')

h1 = net.addHost('h1', ip='10.0.0.2/8')
h2 = net.addHost('h2', ip='10.0.0.3/8')

c1 = net.addController('c1')

info("*** Configuring propagation model\n")
net.setPropagationModel(model="logDistance", exp=4.5)

info("*** Configuring wifi nodes\n")
net.configureWifiNodes()

info("*** Creating links\n")
net.addLink(ap1, h1)
net.addLink(ap2, h2)
```

ditunjukkan seperti pada gambar 5.

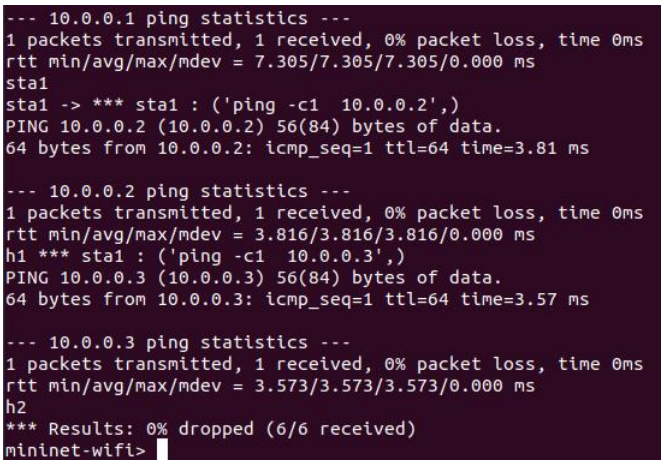
Gambar 5. Program Simulasi Position

Gambar diatas merupakan program yang akan disimulasikan. Program ini dibuat dengan menggunakan *command nano* yang merupakan salah satu fitur pada terminal ubuntu. Pada penelitian yang kami lakukan, kami menambahkan *command net.addLink*, dengan *host* yang terhubung dengan masing-masing AP. AP1 saling terhubung dengan AP2, dan Station terhubung dengan AP 1 dan AP 2 yang akan diujicobakan.

3. RESULTS AND DISCUSSION

3.1 Ping Test

Pada penelitian ini kami mengujicoba program yang telah dibuat dengan menjalankan atau *Run* program. Untuk menjalankan program, dapat dilakukan dengan *command: ~/mininet-Wi-Fi/examples $ sudo pyhton position8.py*. Kemudian untuk mengetahui dan menguji apakah semua *host* telah terhubung ke *Access Point* yang mana sebagai *server*, maka dilakukan *ping test*. *Ping test* ini juga digunakan untuk melihat konektivitas pada jaringan yang telah dibuat. Ping test dapat dilakukan dengan *command: mininet-Wi-Fi> pingall.* Hasil yang didapatkan ditunjukkan pada gambar 6.



Gambar 6. Test konektifitas pingall

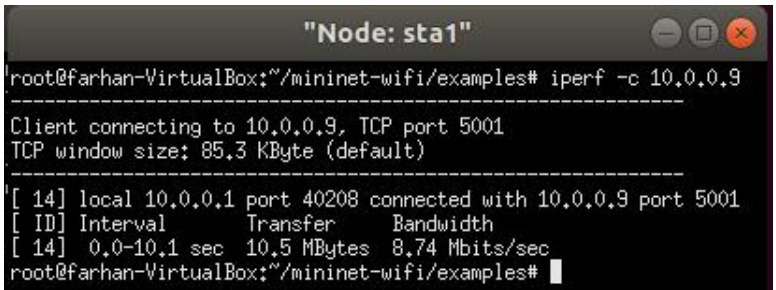
Dari gambar diatas, dapat dilihat bahwa antara *host* dengan *STA* telah terhubung atau *connect* semua, ditandai dengan 0% packet *loss* atau tidak ada paket yang hilang saat pertukaran data seperti gambar diatas. Kemudian *packet* yang ditransmisikan sudah diterima, ditandai dengan adanya *Results : 0 dropped (6/6 received). Received* ini menunjukkan paket telah diterima, dan 0% *dropped* menandakkan tidak ada hambatan pada semua *host.*

3.2 Bandwidth Test dengan Iperf

Bandwidth merupakan kapasitas penghubung jaringan komunikasi yang berfungsi untuk mengirimkan data dari satu titik ke titik yang lain pada koneksi Internet atau jaringan komputer pada suatu waktu. *Bandwidth* biasanya diakumulasikan dengan bentuk Bps atau *bits per second*. Koneksi *bandwidth* dapat terdiri dari 2 jenis, yaitu simetris dan asimetris. *Bandwidth* simetris terjadi apabila *download* dan *upload* kapasitas datanya sama. Sedangkan koneksi asimetris terjadi ketika *download* dan *upload* kapasitas datanya tidak sama, atau salah satu kapasitas datanya baik *download* maupun *upload*, salah satunya ada yang lebih kecil atau lebih besar [7]

Pada simulasi ini peneliti mengujicobakan pengukuran *bandwidth station* yang berada pada 3 titik diantaranya yaitu: ketika station berada pada *range* AP1, AP2, dan diantara irisan *range* AP1 dan AP2 dengan menggunakan *iperf.* Iperf adalah alat untuk mengukur bandwith maksimum TCP, memungkinkan tunning berbagai parameter dan karakteristik UDP. Fitur yang disediakan iperf meliputi TCP dan UDP. [6] Sebelum melakukan ujicoba *iperf*, Langkah pertama yang peneliti harus lakukan adalah membuat terminal pada *example* yang sudah dijalankan, dan kemudian membuat *iperf server* pada *host* yang terhubung dengan AP, dan membuat *iperf client* pada *station*.

Ujicoba pertama, yaitu mengukur *bandwidth* ketika station (STA 1) berada dalam range AP1, dengan menjadikan ‘h1’ (10.0.0.9) yang terhubung dengan AP1 sebagai *iperf server* dan ‘sta1’ (10.0.0.1) sebagai *iperf client. A*ntara *Access Points* dengan *host* dihubungkan oleh *link* dengan *limit bandwith* sebesar 10Mbps, yang mana hasilnya ditunjukkan seperti pada

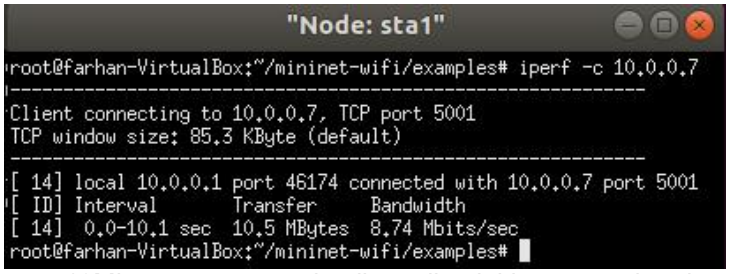


gambar 7.

Gambar 7. Pengukuran *bandwidth ketika '*sta1' berada pada *range* 'AP1'

Dari gambar diatas, didapatkan hasil bahwa ketika dilakukan pengukuran bandwidth ketika STA1 berada pada range AP1,*bandwidth* yang didapatkan yaitu sebesar 8.74 Mbits/sec dengan pengukuran selama 10.1 detik, yang dimana besaran *bandwidth* ini didapatkan karena terhubung dengan AP1 dengan waktu *default* +- 10 detik.

Ujicoba kedua, yaitu mengukur *bandwidth* ketika *station* berada pada range AP2, dengan menjadikan 'h2' (10.0.0.7) yang terhubung dengan AP2 sebagai *iperf server* dan 'sta1' (10.0.0.1) sebagai *iperf client. A*ntara AP dengan *host* dihubungkan oleh *link* dengan limit

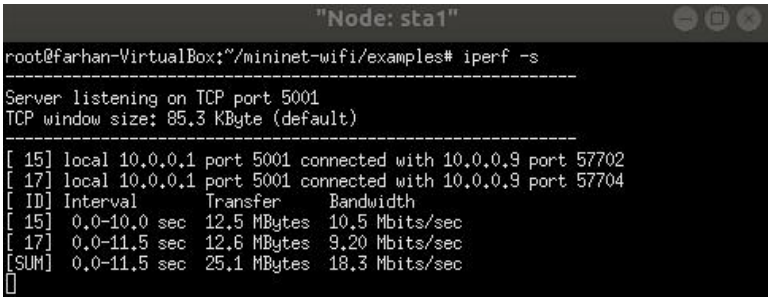


bandwidth sebesar 10Mbps, yang mana hasilnya ditunjukkan seperti pada gambar 8.

Gambar 8. Pengukuran *bandwidth ketika '*sta1' berada pada *range* 'AP2'

Dari gambar diatas, didapatkan hasil bahwa ketika dilakukan pengukuran bandwidth ketika STA1 berada pada range AP2, *bandwidth* yang didapatkan yaitu sebesar 8.74 Mbits/sec dengan pengukuran selama 10.1 detik, yang dimana besaran *bandwidth* ini didapatkan karena terhubung dengan AP1 dengan waktu *default* +- 10 detik.

Ujicoba ketiga, yaitu mengukur *bandwidth* ketika *station* berada pada range irisan *access point*s 1 dan *access points* 2, dengan menjadikan 'h1' (10.0.0.9) dan 'h2' (10.0.07) yang terhubung dengan AP1 dan AP2 sebagai *iperf client* dan 'sta1' (10.0.0.1) yang terhubung dengan AP1 dan AP2 sebagai *iperf server. A*ntara AP dengan *host* dihubungkan oleh *link* dengan *limit bandwidth* sebesar 10Mbps, yang mana hasilnya ditunjukkan seperti pada



gambar 9.

Gambar 9. Pengukuran *bandwidth* ketika 'sta1' berada pada range irisan AP 1 dan AP 2

Dari gambar diatas, didapatkan hasil bahwa sta1 memiliki total *bandwidth* sebesar 18.3 Mbits/sec dengan pengukuran selama 11.5 detik, yang dimana besaran total *bandwidth* ini didapatkan dari kedua *Access Point*. Ketika sta1 terhubung dengan AP1, bandwith yang didapatkan yaitu sebesar 10.5 Mbits/sec dan ketika sta1 terhubung dengan AP2, bandwidth yang didapatkan sebesar 9.20 Mbits/sec.

Keterangan pada terminal Iperf Test [8] :

1. –c = *Client* atau Host yang akan di lihat performanya
2. 10.0.0.1/2/3 = alamat ip yang di cek
3. –u = menggunakan jalur UDP daripada TCP

4. –b = *bandwith* UDP yang dikirim dalam *bit/sec*
5. 100m = ukuran yang di tentukan untuk bandwith UDP
6. –t = waktu untuk dikirim
7. 20= detik yang ditentukan untuk pengiriman data.
8. –p = *port*, yang mengidentifikasikan aplikasi yang menggunakan koneksi di TCP/IP.

Dari ketiga percobaan yang dilakukan, yang mana dilakukan pengukuran bandwith ketika STA 1 berada pada *Access Point* 1, *Access Point* 2, dan irisan antara AP 1 dan AP 2, didapatkan hasil yang ditunjukkan seperti pada table 1 berikut.

TABLE 1. HASIL UJI COBA PENGUKURAN BANDWIDTH

No.	Uji Coba posisi sta1	Waktu (s)	Besaran *Bandwidth*
1.	AP1	10.5	8.74 Mbits/s
2.	AP2	10.5	8.74 Mbits/s
3.	Irisan AP1 dan AP2	11.5	18.3 Mbits/s

Berdasarkan Tabel diatas dapat kita lihat bahwa hasil percobaan pengukuran *bandwidth* sta1 terhadap AP1, AP2, memiliki kesamaan besaran *bandwidth* pada ujicoba pertama dan kedua ketika menyambungkan *station* ke salah satu *Acces Point* yaitu sebesar 8.74 Mbits/s. Sedangkan pada ujicoba ketiga ketika menyambungkan *station* terhadap kedua AP terdapat hasil yang berbeda yaitu memiliki *bandwidth* dua kali lipat lebih dari percobaan pertama dan kedua, dikarenakan pada percobaan ketiga kondisi station memiliki dua NIC yang menyebabkan station terhubung dengan kedua AP dan memiliki besaran *bandwidth* dari kedua AP.

4. DISSCUSSION

Pada penelitian ini, penulis menemukan kendala dikarenakan ternyata spesifikasi laptop yang penulis gunakan memiliki batasan yang membuat pengukuran *bandwidth* menjadi terbatas pada nilai sekitar 50 Mbits/s dan ketika dilakukan eksperimen hasil pengukuran bandwidthnya menjadi tidak sesuai untuk pengukuran STA pada irisan AP 1 dan AP 2 yang seharusnya nilainya lebih dari 2 kali lipat dari pengukuran pada salah satu AP karena memiliki 2 NIC, ketika diuji nilainya hanya bertambah sedikit dari nilai pada salah satu AP. Oleh karena itu penulis membatasi limit *bandwidth* dengan menambahkan command (bw=10M) pada setiap access pointsnya sehingga didappatkan hasil yang valid.

5. CONCLUSION

Hasil dari simulasi ini telah membuktikan bahwa tujuan utama dalam penelitian ini telah tercapai yaitu telah didapatkan hasil pengukuran kecepatan bandwith pada tiga percobaan yang dilakukan ketika STA berada pada range AP1, AP2, dan irisan AP1 dan AP2. Dari ketiga percobaan tersebut nilai bandwith . Percobaan ini dilakukan dengan menggunakan 1 station yang ditempatkan dalam tiga range dan menggunakan waktu default sekitar +/- 10 detik. Percobaan di irisan AP1 dan AP2 meningkat dikarenakan pada posisi tersebut station memiliki 2 NIC sehingga memiliki jumlah bandwith 2 kali lipat lebih besar daripada kedua kondisi sebelumnya. Simulasi ini membuktikan bahwa skenario bicasting dapat menjadikannya salah satu solusi untuk mendapatkan bandwith yang berukuran lebih besar. Skenario dari percobaan diharapakan dapat diterpakan dalam kehidupan nyata pada jaringan telekomunikasi, dimana pada era saat ini alat telekomunikasi telah dibekali dengan sistem radio yang jauh lebih baik

6. ACKNOWLEDGEMENT

Terimakasih kepada seluruh pihak yang terlibat dalam penelitian ini, khususnya kepada Bapak Galura Muhammad Suranegara S.Pd., M.T., selaku pembimbing pada mata kuliah Simulasi dan Pemograman Perangkat Jaringan (SPPJ) pada Program Studi Sistem Telekomunikasi UPI di Kampus Purwakarta.

6. REFERENCES

[1] M. H. R. Jany, N. Islam, R. Khondoker and M. A. Habib, "Performance analysis of OpenFlow based software defined wired and wireless network," 2017 20th International Conference of Computer and Information Technology (ICCIT), 2017, pp. 1-6, doi: 10.1109/ICCITECHN.2017.8281814.

[2] B. Budi and S. Haryadi, "Simulasi Mobility pada Software Defined Networking", *SENTER*, pp. 122–134, Jan. 2018.

[3] R. R. Fontes, S. Afzal, S. H. B. Brito, M. A. S. Santos and C. E. Rothenberg, "Mininet-Wi-Fi: Emulating software-defined wireless networks," 2015 11th International Conference on Network and Service Management (CNSM), 2015, pp. 384-389, doi: 10.1109/CNSM.2015.7367387

[4] D. A. Marenda, G. M. Suranegara, S. Qamar, R. Hakimi and E. Mulyana, "Emulating software-defined wireless network: Bicasting scenario," 2017 3rd International Conference on Wireless and Telematics (ICWT), 2017, pp. 76-80, doi: 10.1109/ICWT.2017.8284142.

[5] P.K Prayogi, M. Orisa,and FX Ariwibisono. "Rancang Bangun Sistem Monitoring Jaringan Access Point Menggunakan Simple Network Management Protokol (SNMP) Berbasis Web". JATI (Jurnal Mahasiswa Teknik Informatika) Vol. 3 No. 2, September 2019

[6] R.Radito, P.H Trisnawan, and K.Amron. "Implementasi Pencarian Jalur berdasarkan Bandwidth dengan menggunakan Algoritme Dijkstra pada Arsitektur Jaringan SoftwareDefined Networking (SDN)". Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, Vol. 4, No. 5, pp.1372-1379, Mei 2020,

[7] E.P Manru and Febrizal." Analisa Kinerja Jaringan W-LAN Pada Perangkat Access Point 802.11/g
(Studi Kasus Fakultas Teknik Universitas Riau)". Jom FTEKNIK Vol. 3, No. 1, pp. 1-11, Februari 2016.

[8] N. Hunaifi and R.F Akbar. "Analisis Kinerja Jaringan Berbaiskan Software Definition Network dengan Protokol Openflow Di RRI Bandung". Jurnal Infotronik Vol. 4, No. 1, pp. 24-32, Juni 2019.

[9] E.P Cintasari. "Analisis Kinerja Jaringan Software Defined Network (SDN) dengan Protokol Open Flow pada Mininet". Online : https://repository.uinjkt.ac.id/. November 2018.

[10] P.R Hardien, R.Primananda, and A.Basuki. "Analisis Mobilitas Node Jaringan Nirkabel pada Software Defined Wireless Network". Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer Vol.2, No.10, pp. 4116-4124 Oktober 2018.