

Simulasi *Handover* pada Jaringan Nirkabel Berbasis *Software Defined Network*

Christoper Siahaan¹, I Made Suartana²

^{1,2}Jurusan Teknik Informatika Fakultas Teknik Universitas Negeri Surabaya

¹christoper.17051204056@mhs.unesa.ac.id

²madesuartana@unesa.ac.id

Abstrak — *Software Defined Network* (SDN) merupakan sebuah metode atau paradigma dalam jaringan komputer dalam mendesain, membuat, dan mengelola sebuah infrastruktur jaringan. SDN memisahkan *control plane* dan *data plane*, SDN ini menjadi solusi untuk perkembangan jaringan yang pesat. SDN memiliki kontrol yang terpusat sehingga mempermudah pengguna dalam mengelola dan mengonfigurasi sebuah jaringan dibandingkan dengan jaringan non-SDN dimana *control plane* dan *data plane* berada pada satu perangkat yang sama. Dengan SDN jaringan menjadi lebih fleksibel dan cerdas. Penerapan SDN pada jaringan wireless juga diharapkan dapat memberikan dampak positif dan meningkatkan kinerja dari jaringan WLAN. *Handover* terminal pada jaringan WLAN berbasis SDN dapat secara efektif meningkatkan QoS dan pemanfaatan sumber daya WLAN. Penelitian ini berfokus pada *handover* berbasis *software defined wireless network* (SDWN), penelitian ini dilakukan dengan menggunakan emulator Mininet-WiFi dengan model mobilitas *random way point*, *random walk*, dan *random direction*. Penelitian ini bertujuan mengetahui pengaruh pergerakan station dengan mobilitas terhadap jarak akses poin dengan jumlah station pada proses *handover* yang sudah ditentukan. Pada setiap model mobility pengujian untuk mendapatkan *throughput*, *jitter*, *packet loss*, dan RSSI. Dari hasil uji coba didapatkan mobilitas pada jaringan WLAN yang mengakibatkan perubahan topologi jaringan, mengakibatkan perubahan nilai parameter uji yang digunakan.

Kata Kunci - Simulasi *Handover*, WLAN, *Software Defined Network*, *Randomway Poin*, *Random Walk*, *Random Direction*

I. PENDAHULUAN

Software Defined Network (SDN) merupakan sebuah metode atau paradigma baru di dunia *networking* dalam mendesain, membuat, dan mengelola sebuah infrastruktur jaringan dimana *control plane* dan *data plane* dikelola secara terpisah. SDN merupakan sebuah solusi atas perkembangan jaringan yang pesat dan terus bertumbuh seiring berkembangnya zaman. SDN memiliki kontrol yang terpusat sehingga mempermudah pengguna dalam mengelola dan mengonfigurasi sebuah jaringan dibandingkan dengan jaringan non-SDN dimana *control plane* dan *data plane* berada pada satu perangkat yang sama [8]. Banyak orang telah mengembangkan SDN ini, salah satunya pada jaringan nirkabel yang disebut *software defined wireless network*

(SDWN). Munculnya *software defined networking* (SDN) menyediakan cara baru untuk menyelesaikan masalah ini secara menyeluruh. Ide utamanya adalah untuk memisahkan fungsi kontrol dari fungsi penerusan jaringan dan menyerahkan kontrol ke pengontrol terpusat. Karena perspektif global dari pengontrol dan antarmuka jaringan yang dapat diprogram terbuka, pengguna dapat memperoleh informasi status pada semua perangkat yang mendasarinya dan mewujudkan penjadwalan lalu lintas sesuai dengan kebutuhan mereka sendiri. Dengan penerapan SDN jaringan menjadi lebih fleksibel dan cerdas. Penerapan SDN pada jaringan wireless juga diharapkan dapat memberikan dampak positif dan meningkatkan kinerja dari jaringan WLAN. *Handover* terminal dalam WLAN berbasis SDN dapat secara efektif meningkatkan QoS dan pemanfaatan sumber daya WLAN, dan sebagai paradigma komputasi baru, jaringan yang ditentukan perangkat lunak juga dapat diterapkan pada jaringan WLAN[9].

WiFi atau *wireless fidelity* merupakan sebuah teknologi yang biasa digunakan untuk akses jaringan publik dan bisa ditemukan di mana saja seperti rumah, kantor, sekolah, stasiun, rumah sakit, kedai kopi, bahkan balai rt sekalipun. Bila melihat perkembangan penggunaan WiFi secara global baik untuk rumahan sampai fasilitas umum, akan ada hampir 628 juta hotspot Wi-Fi publik pada tahun 2023, naik dari 169 juta hotspot pada tahun 2018, meningkat empat kali lipat [1]. IEEE sebagai lembaga standarisasi internasional menetapkan sebuah standar khusus untuk mengatur regulasi penggunaan jaringan nirkabel. Standar WiFi yang ditetapkan IEEE dengan nomor 802.11. Standar 802.11 memiliki beberapa generasi mulai dari generasi a, b, g, n, ac dan standar terbaru ad. Perkembangan standar diciptakan untuk perbaikan standar sebelumnya terutama untuk kecepatan akses. Generasi WiFi terbaru menerapkan teknologi MIMO (*Multiple in Multiple out*) [2].

Pada penelitian [3] dilakukan analisa horisontal *handover* dan pengaruhnya terhadap QoS pada jaringan WLAN, penelitian ini menggunakan layanan multimedia streaming untuk *E-Learning*. Penelitian ini menggunakan perangkat Cisco 3500 sebagai akses poin. Perangkat ini mendukung layanan *handover* dengan sistem *existing*, *ad-hoc* dan tipe *hard handover*. Parameter yang digunakan untuk analisis hasil yaitu waktu tunda (*delay*), waktu tanggapan (*response time*), *throughput*. Simulasi yang dilakukan menggunakan tiga skenario berbeda, skenario pertama bertujuan untuk mengetahui pengaruh horisontal *handover*. Skenario kedua bertujuan untuk mengetahui pengaruh Horizontal *handover*

pada jaringan *ad-hoc* WLAN. Skenario ketiga bertujuan untuk mengetahui pengaruh hard *handover* jaringan *ad-hoc* WLAN. Dari penelitian tersebut didapatkan hasil *delay* pada jaringan ketika terjadi horisontal *handover* bervariasi antara 3 kali waktu pengujian, dimana waktu tunda terbesar mencapai 558 milidetik. Pada skenario kedua, variasi *delay* horisontal *handover* dengan *delay* terbesar sampai 847,4 milidetik. Pada skenario ini, proses waktu horisontal *handover* terbesar terjadi pada proses pencarian AP baru, dengan waktu tunda terbesar sampai 847,4 milidetik. Response time dari proses horisontal *handover* pada jaringan WLAN 802.11 untuk layanan streaming multimedia e-learning bervariasi pada setiap skenario jaringan yang digunakan, tergantung dari posisi node yang terhubung ke akses poin, dan jumlah node yang dilayani oleh akses poin tersebut. Proses *handover* terjadi ketika node mendapat nilai response time yang tinggi. Untuk nilai *throughput* dari tiga skenario yang digunakan menunjukkan nilai *throughput* dari hasil horisontal *handover* untuk layanan streaming multimedia e-learning pada jaringan WLAN 802.11 paling rendah. Hasil tersebut sesuai karena *delay* besar menyebabkan *throughput* menjadi rendah, dan untuk mengatasi itu node harus melakukan proses *handover*. Hasil penelitian ini menunjukkan bahwa proses *handover* mampu memperbaiki QoS secara bertahap, hingga mendapatkan QoS yang kembali normal [3].

Pada penelitian yang dilakukan oleh Hira Manzoor dll, Teknik berbasis SDN untuk mengurangi waktu handoff di jaringan WiFi, penelitian ini memiliki latar belakang pengusulan pendekatan pengurangan waktu deteksi dan penemuan (DeRy), yang menghasilkan solusi berdasarkan jaringan WiFi yang ditentukan perangkat lunak (SD-WiFi). Penelitian ini dilakukan menggunakan simulasi ekstensif yang dilakukan pada simulator jaringan MininetNS3-WiFi dengan parameter yang sudah ditentukan yaitu Area (100.100 m²), kontroler SDN (1, Floodlight), Jumlah AP (3), rentang transmisi setiap AP (33m), Jumlah perangkat nirkabel (1), Mobilitas (5-10 mps), lalu lintas maksimum yang diizinkan (40 Mbps), RSSI threshold (-70 dBm), Unit transmisi maksimum (1500 bytes), simulator (Mininet-NS3-WiFi), sistem operasi (Linux Ubuntu 18.04), standar WiFi (IEEE 802.11 “g”), waktu pengujian (30 menit) dan hasil penelitian dianalisis menggunakan Wireshark. Hasil simulasi menunjukkan bahwa DeRy secara signifikan mengurangi waktu *handoff* sebesar 60–70% dan mengurangi jumlah rata-rata transmisi ulang sebesar 4–49%, sehingga mempertahankan *throughput* untuk aplikasi yang sensitif terhadap penundaan seperti VoIP, jika dibandingkan dengan skema *handoff* berbasis RSSI standar, skema pemilihan akses berbasis pengukuran saluran (CMAS) dan (DL-SINR) sinyal *downlink* ke *interferensi plus rasio noise* skema pemilihan AP (DASA) [4].

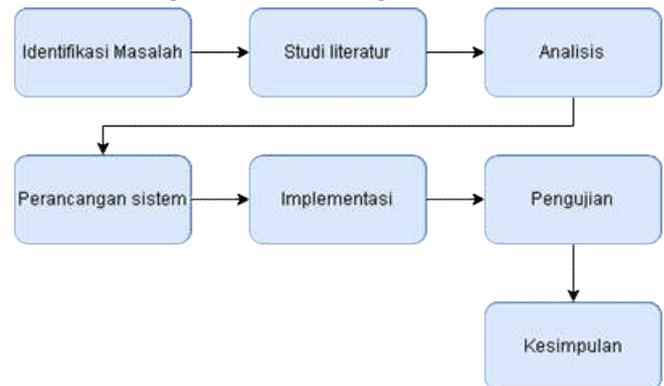
Penelitian ini berfokus pada *handover* berbasis *software defined wireless network* (SDWN), penelitian ini dilakukan dengan menggunakan emulator Mininet-WiFi dengan model mobilitas Random Way Point, Random Walk, dan Random Direction. Penelitian ini dilakukan dengan harapan mengetahui pengaruh pergerakan station dengan mobilitas terhadap jarak

akses poin dengan jumlah station pada proses *handover* yang sudah ditentukan. Terdapat 3 model mobilitas pada penelitian ini yaitu Random Way Point, Random Walk, dan Random Direction. Pada setiap model mobility pengujian untuk mendapatkan *throughput*, jitter, packet loss, dan RSSI.

II. METODOLOGI PENELITIAN

A. Metode Penelitian

Metode yang digunakan pada penelitian ini adalah metode Experimental Design. Metode penelitian Experimental Design adalah rencana rinci untuk mengumpulkan dan menggunakan data untuk mengidentifikasi hubungan sebab akibat.



Gbr 1. Alur metode penelitian

Pada gambar 1 alur metode penelitian merupakan alur metode penelitian dengan penjelasan sebagai berikut:

- 1) **Identifikasi Masalah:** Identifikasi masalah adalah merupakan langkah pertama dalam melakukan penelitian. Pada langkah ini peneliti melakukan identifikasi masalah simulasi *handover* pada software defined network untuk menganalisis model mobility Random Direction, Random Walk, Random Way Point dengan jumlah station yang berbeda untuk mengetahui pengaruh horisontal *handover* yang efisien menggunakan parameter QoS.
- 2) **Studi Literatur:** Penelitian ini menggunakan beberapa landasan literatur salah satunya penelitian [3] yaitu analisa horisontal *handover* dan pengaruhnya terhadap QoS pada jaringan WLAN, penelitian ini menggunakan layanan multimedia streaming untuk E-Learning. Penelitian ini menggunakan perangkat Cisco 3500 sebagai akses poin. Perangkat ini mendukung layanan *handover* dengan sistem existing, *ad-hoc* dan tipe hard *handover*.
- 3) **Analisis:** Peneliti menggunakan literatur yang berhubungan dengan topik penelitian yaitu tentang simulasi *handover* maupun penggunaan Software Defined Network (SDN). Sumber yang digunakan yaitu buku, jurnal, artikel nasional dan internasional.
- 4) **Perancangan Sistem:** Pada tahap ini merupakan proses perancangan system yang akan di implementasikan sesuai dengan hasil Analisa yang telah dilakukan.

- 5) Implementasi: Pada tahap melakukan implementasi atau simulasi dari hasil rancangan pada tahap sebelumnya.
- 6) Pengujian: Pada tahap ini hasil dari implementasi sistem akan diuji sesuai dengan skenario pengujian yang telah ditentukan. Parameter pengujian yang dipakai meliputi : Jitter, Throughput, Packet loss, dan RSII [6][7].
- 7) Kesimpulan: Pada tahap ini dilakukan perbandingan dan analisis hasil dan kesimpulan dari hasil output pengujian yang telah dilakukan.

B. Analisis Kebutuhan

1) Kebutuhan perangkat keras (Hardware)

Perangkat keras yang diperlukan guna tujuan penelitian yaitu Laptop sebagai uji coba dengan spesifikasi berikut :

- Processor : Intel Core I7 11370H
- RAM : 8
- Storage : SSD 512
- Sistem Operasi : Windows 10 Pro 64-bit

2) Kebutuhan perangkat lunak (Software)

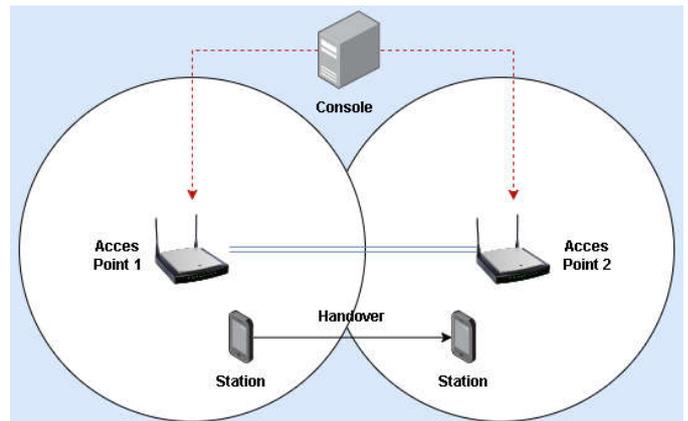
Perangkat lunak berfungsi untuk pengoperasian sistem pada penelitian ini. Pada penelitian ini virtual machine yang digunakan adalah VMware Workstation.

TABEL I
 KEBUTUHAN PERANGKAT LUNAK

No.	Perangkat Lunak	Keterangan
1.	Sistem Operasi Virtual box	Merupakan perangkat virtualisasi yang dapat menjalankan system operasi.
2.	Mininet WiFi	Virtual os Ubuntu 20.04 x64 yang sudah terinstal mininet.

C. Perancangan Sistem

Pada penelitian ini akan digunakan topologi Extended Service Set. Extended Service Set terdiri dari sejumlah IEEE 802.11 BSS (Basic Service Set) dan memungkinkan mobilitas terbatas dalam jaringan WiFi. Station dapat berpindah antar BSS dalam satu ESS namun tetap terhubung ke jaringan. Saat Station pindah ke BSS baru, ia akan melakukan prosedur reasosiasi dengan AP (Akses poin) baru). Pemilihan jenis topologi ini karena ingin menguji simulasi *handover* dengan 3 models berbeda. Untuk jenis Topologi ESS ini memiliki lebih dari satu Acces Point, bertujuan untuk menjangkau area lebih luas.

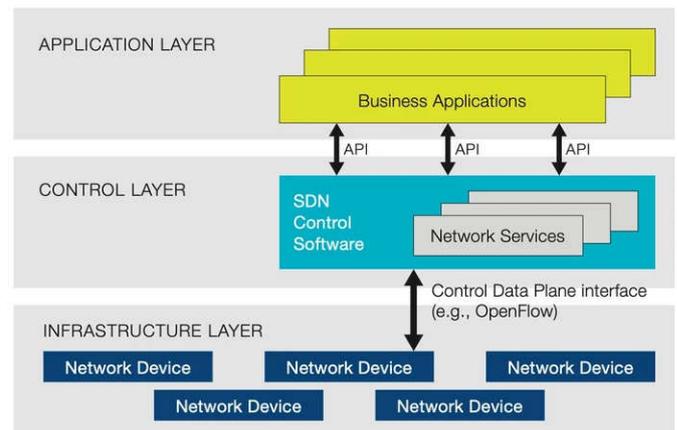


Gbr 2. Perancangan Sistem

Pada gambar 2 merupakan ilustrasi perancangan arsitektur jaringan. Arsitektur jaringan pada penelitian ini menggunakan dua Acces point, dua station dan satu controller . Acces point disini agar station dapat melakukan mobility sehingga dapat melakukan simulasi *handover*[11]. Dengan kecepatan station 8 m/s.

D. Perancangan Software Defined Network Pada WLAN

1) Arsitektur SDN



Gbr 3. Arsitektur SDN

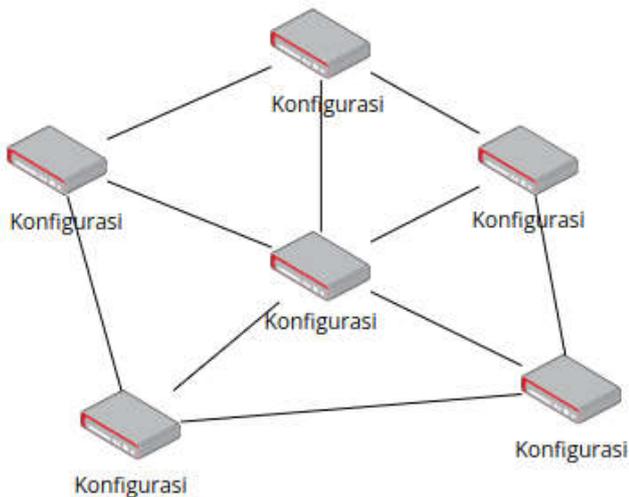
Pada gambar 3 Arsitektur SDN memisahkan kontrol dan komponen data dari jaringan tradisional. Hal ini memungkinkan perusahaan untuk merancang sistem jaringan dan komputasi yang menggunakan teknologi berbasis perangkat lunak untuk mengoptimalkan perangkat keras jaringan [5].

2) Topologi WLAN dengan SDN

Perbedaan topologi jaringan menggunakan SDN dan tidak menggunakan SDN:

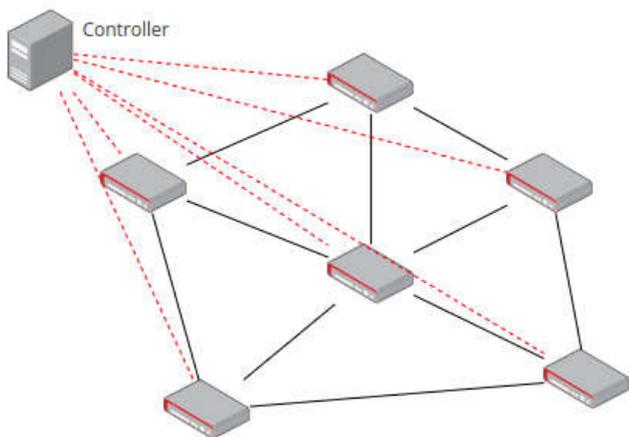
Tidak menggunakan SDN

Pada gambar 4 ilustrasi topologi jaringan ujicoba tanpa menggunakan arsitektur SDN.



Gbr 4. Simulasi tidak menggunakan SDN

Menggunakan SDN



Gbr 5. Simulasi menggunakan SDN

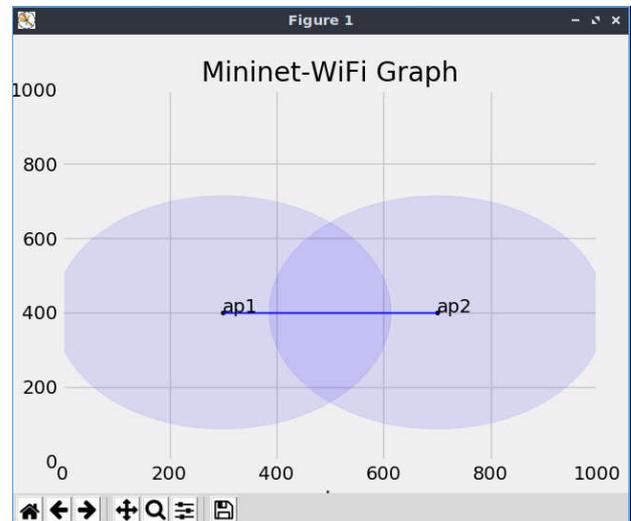
III. HASIL DAN PEMBAHASAN

A. Implementasi

Untuk melakukan proses simulasi hal pertama yang harus dilakukan yaitu instalasi Mininet sebagai simulator SDN software defined network. Mininet diinstal pada virtual box. Pada penelitian ini menggunakan Mininet-WiFi VM. Mininet-WiFi adalah cabang dari emulator jaringan Mininet SDN bertujuan memperluas fungsionalitas dari Mininet dengan menambahkan stasiun WiFi tervirtualisasi dan titik akses berdasarkan nirkabel Linux standar driver dan driver simulasi nirkabel 80211_hwsim. Pada skenario penelitian berikut digunakan image Mininet versi 2.6 yang di unduh pada laman mininet-WiFi.github.io.

1) Implementasi Topologi

Tahapan pertama dalam konfigurasi topologi simulasi pada Mininet yaitu membuat fungsi acces point dengan menentukan kordinatnya guna membentuk jaringan yang saling terhubung seperti Gambar 6 konfigurasi AP.



Gbr 6. Konfigurasi AP

Pada gambar 6 merupakan ilustrasi topologi, pada topologi posisi acces point 1 yaitu $x=300$ dan juga sumbu $y=400$. pada acces point 2 yaitu $x=700$ dan $y=400$.

```
ap1 = net.addAccessPoint('ap1', ssid='ssid-ap1',  
mode='g', channel='1', position='300,400,0')  
ap2 = net.addAccessPoint('ap2', ssid='ssid-ap2',  
mode='g', channel='6', position='700,400,0')  
  
net.setMobilityModel(time=0,model='', max_x=100,  
max_y=100, seed=20)
```

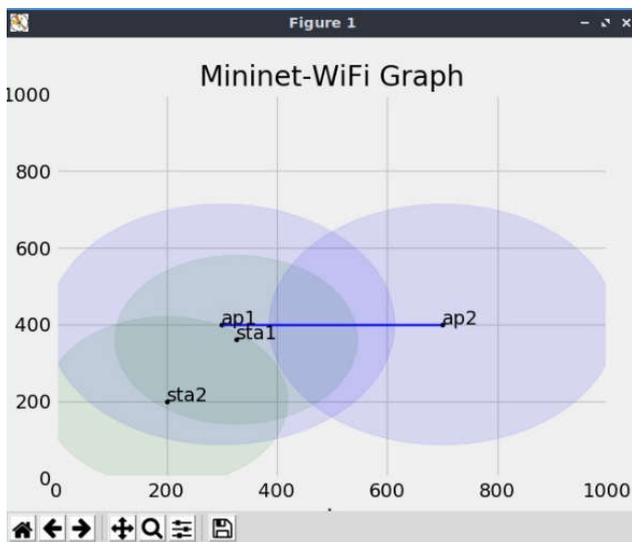
Mobility models dirancang untuk menggambarkan pola pergerakan pengguna seluler (station), dan bagaimana lokasi, kecepatan, dan akselerasi berubah-ubah seiring waktu. Mobility models yang digunakan pada penelitian ini adalah sebagai berikut[12] :

Random Direction

```
net.addStation('sta1', mac='00:00:00:00:00:01',  
ip='10.0.0.1/8', min_x=300, max_x=700,  
min_y=300, max_y=500, min_v=20, max_v=100,  
range=220)
```

min_x = Posisi X minimum # default=0
max_x = Posisi X Maksimum # default=100
min_y = Posisi Y minimum # default=0
max_y = Posisi Y maksimum # default=100
min_v = Nilai minimum untuk kecepatan simpul
max_v = Nilai maksimum untuk kecepatan simpul

```
net.setMobilityModel(time=0,  
model='RandomDirection', max_x=1000,  
max_y=1000, seed=20)
```



Gbr 7. Konfigurasi AP Random Direction

Gambar 7 ilustrasi topologi dengan pergerakan random direction. Topologi terdiri dari dua akses poin (ap1 dan ap2) dan dua node/station (sta1 dan sta2).

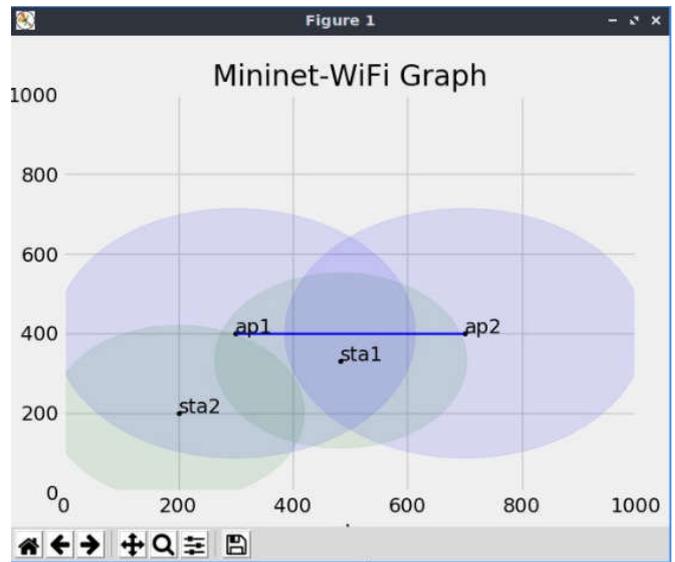
Random Walk

```
net.addStation('sta1', mac='00:00:00:00:00:01',  
ip='10.0.0.1/8', min_x=300, max_x=700, min_y=300,  
max_y=500, constantDistance=10, constantVelocity=10,  
range=220)
```

min_x = Posisi X minimum # default=0
max_x = Posisi X Maksimum # default=100
min_y = Posisi Y minimum # default=0
max_y = Posisi Y maksimum # default=100
constantDistance = nilai jarak konstan yang telah ditempuh dalam setiap langkah. Untuk standarnya adalah 1.0.

constantVelocity = nilai untuk kecepatan node konstan. Standarnya adalah 1.0

```
net.setMobilityModel(time=0, model='RandomWalk',  
max_x=1000, max_y=1000, seed=20)
```



Gbr 8. Konfigurasi AP Random Walk

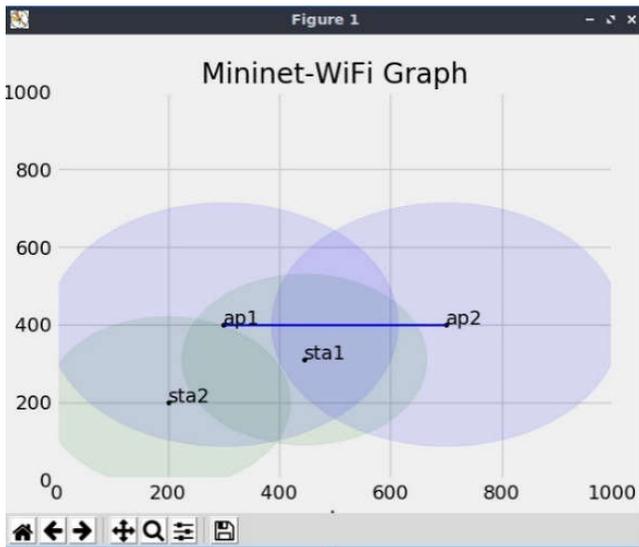
Gambar 8 ilustrasi topologi dengan pergerakan random walk. Topologi terdiri dari dua akses poin (ap1 dan ap2) dan dua node/station (sta1 dan sta2).

Random Way Point

```
net.addStation('sta1', mac='00:00:00:00:00:01',  
ip='10.0.0.1/8', min_x=300, max_x=700, min_y=300,  
max_y=500, min_v=20, max_v=100, min_wt=0,  
max_wt=1000, range=220)
```

min_x = Posisi X minimum # default=0
max_x = Posisi X Maksimum # default=100
min_y = Posisi Y minimum # default=0
max_y = Posisi Y maksimum # default=100
min_v = Nilai minimum untuk kecepatan simpul # default=5
max_v = Nilai maksimum untuk kecepatan node # default=5
min_wt = Waktu tunggu minimum # default=0
max_wt = Waktu tunggu maksimum # default=100

```
net.setMobilityModel(time=0, model='RandomWayPoint',  
max_x=1000, max_y=1000, seed=20)
```



Gbr 9. Konfigurasi AP Random Way Point

Gambar 9 ilustrasi topologi dengan pergerakan random way poin. Topologi terdiri dari dua akses poin (ap1 dan ap2) dan dua node/station (sta1 dan sta2).

B. Pengujian

1) Ping Test

```
mininet-wifi> pingall
*** Ping: testing ping reachability
sta1 -> *** sta1 : ('ping -c1 10.0.0.2',)
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=23.5 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 23.491/23.491/23.491/0.000 ms
sta2 *** sta1 : ('ping -c1 10.0.0.3',)
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=16.4 ms

--- 10.0.0.3 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 16.383/16.383/16.383/0.000 ms
sta3 *** sta1 : ('ping -c1 10.0.0.4',)
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=33.3 ms

--- 10.0.0.4 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 33.348/33.348/33.348/0.000 ms
sta4 *** sta1 : ('ping -c1 10.0.0.5',)
PING 10.0.0.5 (10.0.0.5) 56(84) bytes of data.
64 bytes from 10.0.0.5: icmp_seq=1 ttl=64 time=15.0 ms
```

Gbr 10. Pengujian ping

Ping tes digunakan untuk menguji konvergensi jaringan dan memastikan semua perangkat pada jaringan telah terhubung. Dari gambar 10 dapat dilihat bahwa semua station telah terhubung satu sama lain melalui akses point, ditandai dengan tidak adanya paket yang loss dan tidak terkirim.

2) Throughput

Dalam transmisi data, *throughput* jaringan adalah jumlah data yang berhasil dipindahkan dari satu tempat ke tempat lain dalam periode waktu tertentu. *Throughput* jaringan diukur dalam bit per detik (bps), Kilobit per detik (Kbps) atau gigabit per detik (Gbps). Pada pengujian ini menggunakan wireshark untuk menguji parameter *throughput*. *Throughput* memiliki rumus yaitu (Paket diterima dibagi lama waktu pengalaman). Pengujian dilakukan pada saat setiap station melakukan perpindahan ke satu ap ke ap lain. Hasil *throughput* pada setiap pengujian model terdapat pada Tabel II hasil throughput dibawah.

TABEL I
HASIL THROUGHPUT

Skenario	Random Direction	Random Walk	Random Way Point
1.	49,75259792 1663/ kb	433,3538005 9230005 / kb	81603,140991706 / kb

Hasil *throughput* yang dihasilkan pada setiap model dan skenario tersebut berbeda beda dapat dipengaruhi oleh berbagai faktor salah satunya adalah pergerakan node hasil dari mobilitas yang digunakan sehingga mempengaruhi jarak antar node yang sedang terhubung.

3) Jitter

Jitter adalah istilah dalam jaringan yang menggambarkan beberapa kesalahan interupsi singkat selama transmisi data. Ini dapat menyebabkan kemacetan jaringan, tabrakan, atau gangguan sinyal. Secara teknis, jitter mengacu pada variasi latensi – penundaan antara pengiriman sinyal dan penerimaan. Pada pengujian ini menggunakan wireshark untuk menguji parameter *jitter*. *Jitter* memiliki rumus yaitu (Total variasi delay dibagi (Total paket diterima – 1)). Hasil *jitter* pada setiap pengujian model terdapat pada Tabel III hasil jitter dibawah.

TABEL III
HASIL JITTER

Skenario	Random Direction	Random Walk	Random Way Point
1.	0.0192937128 09646 ms	0.0023392536 21104 ms	0.011966642 ms

Hasil Jitter yang didapatkan dari simulasi didapatkan model randomwalk memiliki nilai tertinggi diantara model pergerakan yang digunakan. Hal ini terjadi karena pada model random walk saat simulasi tidak terjadi proses *handover* hanya bergetar pada garis tengah irisan antar dua AP.

4) Packet Loss

Packet loss adalah kegagalan paket data terkirim pada jaringan. Pada pengujian ini menggunakan wireshark untuk mendapat nilai *packet loss*. *Packet loss* memiliki rumus yaitu

$((\text{Data dikirim} - \text{Data diterima}) / \text{Data dikirim}) \times 100\%$). Hasil *Jitter* pada setiap pengujian model terdapat pada Tabel IV.

TABEL IV
 HASIL PACKET LOSS

Skenario	Random Direction	Random Walk	Random Way Point
1	0%	0%	0%

Hasil loss yang dihasilkan pada setiap model dan skenario tersebut 0% sehingga tidak ada paket yang hilang dan berhasil terkirim.

5) RSSI

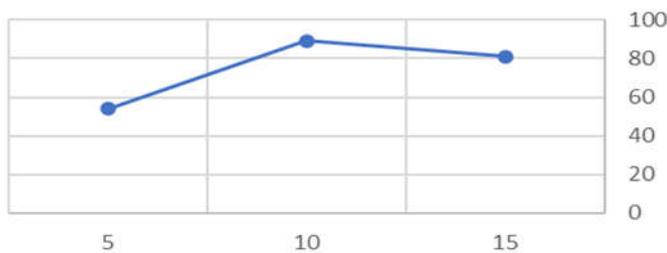
Received signal strength indicator (RSSI) adalah indikator kekuatan sinyal yang diterima. RSSI memiliki makna sebagai ukuran seberapa baik perangkat dapat menerima sinyal dari titik akses atau router [10]. Strategi *handover* berdasarkan RSSI mudah diimplementasikan, tetapi tidak menjamin kinerja klien dan juga keseluruhan *throughput* jaringan. Beberapa skema yang ada memerlukan modifikasi sisi klien, yang tidak cocok untuk diterapkan pada solusi dunia nyata. Pada tabel V menjelaskan mengenai rssp dBm dalam setiap model dan skenario setiap 5s, 10s, 15s.

TABEL V
 HASIL RSSI

Skenario	Random Direction	Random Walk	Random Way Point
1.	(-54, -89, -81) dbm	(-85, -85, -85) dbm	(-84, -88, -89) dbm

Perubahan nilai RSSI yang didapatkan pada pengujian ketiga model pergerakan disajikan pada gambar 10, 11 dan 12.

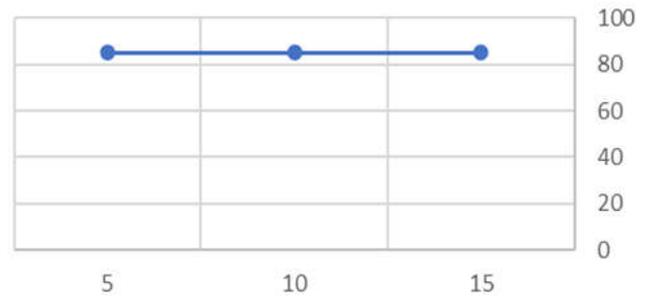
Random Direction



Gbr 11. Pengujian RSSI Random direction

Pada gambar 10 menampilkan perubahan nilai RSSI pada pergerakan Random Direction. Dari grafik terlihat terjadi perubahan pada setiap waktu pengukuran. Karena pada model ini pergerakan station banyak melakukan perpindahan secara acak dan berbeda beda arahnya.

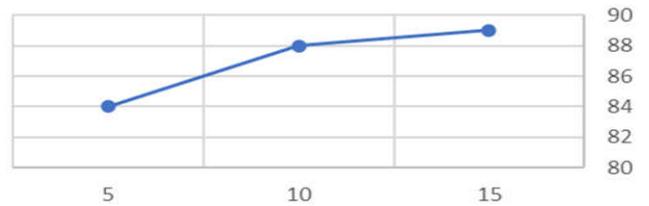
Random Walk



Gbr 12. Pengujian RSSI Random walk

Pada gambar 12 Random Walk terlihat tidak memiliki perubahan sinyal pada setiap waktu pengukuran. Karena pada model ini pergerakan station melakukan perpindahan seperti begetar sehingga sangat tidak efektif dan tidak ada proses *handover*.

Random Way Point



Gbr 12. Pengujian RSSI Random Way Point

Pada gambar 12 Random Way Point terlihat tidak banyak memiliki perubahan sinyal seoerti pada gambar. Karena pada model ini pergerakan station melakukan perpindahan secara acak dan terarah seperti garis.

IV. KESIMPULAN

Berdasarkan hasil penelitian penggunaan Arsitektur WLAN berbasis SDN. untuk mensimulasikan pengaruh mobilitas terhadap *handover* pada jaringan SDN. Hasil percobaan menunjukkan bahwa klien melakukan *handover* yang efisien berdasarkan beban AP dimana total *throughput* jaringan lebih tinggi setelah *handover* terjadi. Terdapat inkonsistensi hasil dari setiap pengujian yang dilakukan pada *throughput*, *jitter*, dan perpindahan station. Seperti pada model mobility random direction yang hasil *throughput*nya kecil karena melakukan perpindahan antar AP dengan sangat cepat berkali kali, sedangkan pada model mobility random walk sedikit lebih besar karena tidak melakukan perpindahan atau *handover* sama sekali, lalu model mobility random way point bergerak sangat lambat tetapi melakukan proses *handover* sehingga hasil *throughput* sangat tinggi. Inkonsistensi tersebut terjadi disebabkan oleh ketidakstabilan model mobility yang digunakan dalam penelitian ini, seperti yang terlihat pada model random walk yang tidak melakukan perpindahan sama sekali terbukti dari hasil RSSI dimana model mobility random

walk tidak mengalami perubahan sinyal setelah diuji dalam kurun waktu 5 detik, 10 detik, dan 15 detik.

REFERENSI

- [1] Cisco, U. (2020). Cisco annual internet report (2018–2023) white paper. Online][accessed March 26, 2021) <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/whitepaper-c11-741490.html>.
- [2] Citra Dewi, L. (2011). WIRELESS TECHNOLOGY DEVELOPMENT: HISTORY, NOW, AND THEN. *Jurnal ComTech*, 2(02).
- [3] Widyantara, I. M. O., Cahyono, B. D., & Setiawan, W. (2015). Analisa Horizontal *Handover* Terhadap QoS Layanan Streaming Multimedia E-Learning Pada Jaringan WLAN 802.11. *Jurnal Teknologi Elektro*, 14.
- [4] Manzoor, H., Manzoor, S., Ali, N., Sajid, M., Menhas, M. I., & Hei, X. (2021). An SDN-based technique for reducing handoff times in WiFi networks. *International Journal of Communication Systems*, 34(16), e4955.
- [5] Gilani, S. M. M., & Jin, W. (2016). SDN-based *Handover* in Future WLAN. *International Journal of Future Generation Communication and Networking*, 9(12), 139-154.
- [6] Cunha, N., Lima, R., Anjos, E., & Matos, F. (2018). A Software Defined Wireless Networking Approach for Managing Handoff in IEEE 802.11 Networks. *Wireless Communications and Mobile Computing*, 2018.
- [7] Sun, M., & Qian, H. (2016). *Handover* management scheme in SDN-based. *Journal of Communications*, 11(3).
- [8] Vasylenko, V., Kuklov, V., & Grynkevych, G. (2016, February). Analysis of SDN for wireless *handover* platform. In 2016 13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET) (pp. 630-633). IEEE.
- [9] Chen, Z., Luo, Z., Duan, X., & Zhang, L. (2020). Terminal *handover* in software-defined WLANs. *EURASIP Journal on Wireless Communications and Networking*, 2020(1), 1-13.
- [10] Nahida, K., Yin, C., Hu, Y., Arain, Z. A., Pan, C., Khan, I., ... & Rahman, G. S. (2017). *Handover* based on AP load in software defined Wi-Fi systems. *Journal of Communications and Networks*, 19(6), 596-604.
- [11] Khan, S. A., Shayea, I., Ergen, M., & Mohamad, H. (2022). *Handover* management over dual connectivity in 5G technology with future ultra-dense mobile heterogeneous networks: A review. *Engineering Science and Technology, an International Journal*, 101172.
- [12] Tama, A. B. Suartana. I.M, (2022), Pengaruh Mobilitas dan Jumlah Pengguna pada Kualitas Layanan Software Defined Wireless Network (SDWN), JINACS (Journal of Informatics and Computer Science), 224-230. <https://doi.org/10.26740/jinacs.v3n03.p224-230>