Analisis Perbandingan Metode VLAN dan MAC-*based* dalam Penerapan Segmentasi Jaringan pada Jaringan OpenFlow

Reza Ainurrachman [1], Widhi Yahya [2]

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: [1]reza.ainur@gmail.com, [2]widhi.yahya@ub.ac.id

Abstrak

Dengan adanya perkembangan teknologi serta peningkatan terhadap kebutuhan layanan yang berjalan secara daring, performa jaringan menjadi salah satu perhatian karena dapat mempengaruhi kinerja dari layanan tersebut. Salah satu komponen yang bertanggung jawab atas performa dari layanan tersebut adalah *Data Center Network* (DCN). DCN merupakan komponen penyusun distribusi layanan yang dapat terdiri dari *switch*, *storage*, *load balancing* yang berjalan mendukung proses layanan secara daring. Untuk meningkatkan performa DCN, dapat dilakukan penerapan segmentasi jaringan yang akan membagi jaringan secara virtual menjadi beberapa *tenant* yang saling terisolasi satu sama lain sehingga dapat digunakan untuk memprioritaskan layanan tertentu. Pada penelitian ini dilakukan penerapan segmentasi jaringan menggunakan 2 metode, yaitu metode VLAN (berdasarkan *port*) dan metode MAC-*based* (berdasarkan MAC *address*). Penggunaan kedua metode tersebut terinspirasi oleh penerapan segmentasi jaringan menggunakan konsep *Virtual Local Area Network* (VLAN) yang akan diterapkan pada paradigma jaringan *Software Defined Network* (SDN). Pengujian pada penelitian ini dilakukan menjadi 2 tahap, yaitu pengujian fungsional untuk melakukan pengecekan keberhasilan isolasi antar *tenant* dengan mengirim paket *ping*, TCP, UDP, *broadcast*, dan pengujian performa untuk melihat performa jaringan dengan mengukur nilai *Round Trip Time* (RTT) dan nilai *throughput* yang dilakukan dengan 3 variasi. Hasilnya pada pengujian fungsional, semua metode berhasil mengisolasi *tenant*, tetapi saat pengujian dengan paket *broadcast*, metode MAC-*based* masih bisa meloloskan paket yang tidak sesuai perancangan sistem. Sedangkan pada pengujian performa, metode MAC-*based* lebih unggul pada semua pengujian kecuali pengujian *throughput* yang ketiga karena pengaruh lolosnya paket *broadcast* yang tidak sesuai perancangan sistem.

Kata kunci: *Software Defined Network (SDN)*, segmentasi jaringan, *Data Center Network (DCN)*

Abstract

With technological developments and increasing demand for services that run online, network performance has become a concern because it can affect the performance of these services. One of the components responsible for the performance of this service is the Data Center Network (DCN). DCN is a component that makes up service distribution which can consist of switches, storage, load balancing which runs to support online service processes. To improve DCN performance, network segmentation can be implemented which will virtually divide the network into several tenants isolated from each other so that they can be used to prioritize certain services. In this research, network segmentation was implemented using 2 methods, namely the VLAN method (based on ports) and the MAC-based method (based on MAC addresses). The use of these two methods is inspired by the application of network segmentation using the Virtual Local Area Network (VLAN) concept which will be applied to the Software Defined Network (SDN) network paradigm. Testing in this research was carried out in 2 stages, namely functional testing to check the success of isolation between tenants by sending ping, TCP, UDP, broadcast packets, and performance testing to see network performance by measuring Round Trip Time (RTT) values and throughput values done with 3 variations. The results in functional testing, all methods succeeded in isolating tenants, but when testing with broadcast packets, the MAC-based method was still able to pass packets that did not match the system design. Meanwhile, in performance testing, the MAC-based method is superior in all tests except the third throughput test due to the influence of passing broadcast packets that do not match the system design.

Keywords: *Software Defined Network (SDN), network segmentation, Data Center Network (DCN)*

1. PENDAHULUAN

Perkembangan teknologi pada jaringan komputer saat ini terus berkembang seiring dilakukannya penelitian untuk menemukan inovasi agar dapat memenuhi kebutuhan manusia di era modern ini baik untuk kegiatan bisnis dalam institusi, organisasi, perusahaan, maupun perseorangan. Selain itu, masa pandemi beberapa saat yang lalu juga memunculkan tantangan baru agar komunikasi dapat tetap berjalan dengan memanfaatkan kemajuan teknologi yang terus berkembang (Gentile et al., 2021). Dengan memanfaatkan teknologi berupa layanan yang tersedia secara daring, proses komunikasi, bisnis, pendidikan, dan kegiatan lainnya dapat tetap berjalan. Salah satu aspek yang bertanggung jawab atas berjalannya layanan-layanan secara daring adalah *data center*.

Data center merupakan infrastruktur jaringan yang terdiri atas sekumpulan *node*, yang berfungsi untuk menjalankan proses komputasi dan penyimpanan data untuk kebelangsungan layanan yang berjalan secara daring. Sekumpulan *node* dalam data center tersebut saling terkoneksi dan membentuk sebuah jaringan yang disebut dengan *Data Center Network* (DCN). DCN saling terhubung untuk mencapai tujuan tertentu seperti menjalankan aplikasi web search, web mail, atau aplikasi lain yang berjalan secara daring (Xia et al., 2017). DCN berperan penting atas berjalannya layanan-layanan yang berjalan secara daring agar layanan dan proses komputasi dapat berjalan dengan lancar (Wang et al., 2014). Jenis topologi serta metode yang digunakan untuk proses *routing*/pencarian jalur distribusi data mempengaruhi layanan-layanan yang berjalan secara daring tersebut (Zhao et al., 2017). Semakin banyak layanan yang berjalan, semakin banyak pula *switch* atau *node* yang berjalan dalam jaringan. Dengan banyaknya *node* yang berjalan, DCN menerapkan konsep *multi-tenant*, yaitu saat *node* satu dan lainnya saling berbagi sumber daya pada satu infrastruktur yang sama (Alam, 2020). Konsep multi-*tenant* ini juga ditemukan pada lingkungan sistem berbasis VLAN (*Virtual Local Area Network*).

VLAN merupakan metode yang digunakan untuk mengurangi cakupan *broadcast* dari jaringan lokal untuk kepentingan keamanan dan penyederhanaan kontrol manajemen jaringan (Nguyen et al., 2016). Pengurangan cakupan *broadcast* tersebut dilakukan dengan membagi jaringan menjadi beberapa kelompok secara virtual, bukan secara fisik (Gentile et al., 2021). Beberapa *host* dikelompokkan secara virtual pada sebuah grup yang dapat disebut sebagai *tenant* (Afolabi et al., 2018). Tenant terdiri dari beberapa *node* yang berada pada domain *broadcast* yang sama, yang mana secara fisik dapat berada pada satu mesin yang sama, maupun tersebar ke beberapa mesin yang berbeda (Taherimonfared et al., 2013). VLAN telah lebih dari 30 tahun digunakan dan masih populer digunakan dalam jaringan pada perusahaan maupun kampus karena dapat menyederhanakan proses manajemen jaringan oleh administrator jaringan (Nguyen et al., 2016).

Virtualisasi yang dilakukan VLAN juga menjadi fondasi lahirnya konsep *network slicing*/segmentasi jaringan. Segmentasi jaringan merupakan pembagian jaringan dengan cara mengisolasi *tenant* satu dengan lainnya untuk menerapkan layanan khusus sesuai konfigurasi yang dibutuhkan (Afolabi et al., 2018). Pada jaringan saat ini, umumnya konsep segmentasi jaringan masih dilakukan menggunakan metode VLAN (Chen et al., 2016). Tetapi, terdapat kekurangan pada VLAN yang konfigurasinya cukup rumit karena konfigurasi harus dilakukan pada setiap *device* pada jaringan. Hal ini akan menyulitkan saat VLAN digunakan pada cakupan jaringan yang luas dan infrastruktur yang terpisah seperti pada bangunan yang berbeda (Nguyen et al., 2016). Untuk mengatasi hal tersebut, digunakan paradigma *Software Defined Network* (SDN) untuk menjalankan konsep segmentasi jaringan (Chen et al., 2016).

SDN merupakan paradigma pada jaringan komputer yang memisahkan pusat logika dari *device* jaringan tradisional yang terdiri *control plane* dan *data plane* untuk membentuk arsitektur jaringan yang fleksibel dan lebih mudah dikelola (Keti et al., 2015). Otak dari jaringan terpusat pada *controller* yang berperan sebagai *control plane*, sedangkan *device* berupa switch menjadi penerus paket disebut sebagai *data plane* (Nunes et al., 2014). Dengan adanya *controller* yang terpusat, administrator jaringan dapat menerapkan instruksi jaringan seperti

mekanisme penerusan paket hingga mekanisme keamanan jaringan (De Oliveira et al., 2014). Dengan adanya fleksibilitas dan kontrol yang lebih sederhana, SDN dapat menerapkan beberapa fungsi di jaringan komputer, salah satunya adalah *network slicing*/segmentasi jaringan.

Beberapa penelitian sebelumnya telah dilakukan tentang penerapan segmentasi jaringan pada SDN menggunakan bantuan dari aplikasi pihak ketiga, yaitu Flowvisor. Muttaqin et al. (2018) menyimpulkan bahwa penggunaan Flowvisor untuk penerapan segmentasi jaringan berhasil meningkatkan performa jaringan dengan memisahkan jaringan TCP dan UDP. Chen et al. (2016) melakukan penelitian dengan membuat lingkungan segmentasi jaringan menggunakan Flowvisor dengan 2 metode berbeda, yaitu dengan metode VLAN dan berdasarkan MAC *Address* (MAC-*based*) yang mendapatkan hasil dengan metode VLAN memiliki *latency* lebih baik daripada metode MAC-*based*. Pada penelitian lain, Kurniawan et al. (2021) juga menyimpulkan bahwa penggunaan Flowvisor untuk penerapan segmentasi jaringan dapat meningkatkan performa jaringan, tetapi penggunaan Flowvisor meningkatkan persentase CPU usage dan memory usage yang cukup tinggi pada percobaan yang ketiga. Dengan beberapa penelitian yang telah dilakukan, serta sifat jaringan berbasis SDN yang mempunyai kontrol terpusat dan adanya kebebasan dalam melakukan konfigurasi jaringan melalui SDN *controller*, dilakukan penelitian untuk menerapkan konsep segmentasi jaringan pada lingkungan jaringan berbasis SDN.

Penelitian dilakukan dengan menjalankan 2 metode yang umumnya digunakan untuk segmentasi jaringan pada konsep VLAN (Nguyen et al., 2016), yaitu *static* VLAN dan *dynamic* VLAN. Metode *static* VLAN memanfaatkan *port* milik *device* dan proses pemasangan VLAN *tag* untuk menandai paket yang dikirim ke host tujuan, sedangkan metode *dynamic* VLAN memanfaatkan MAC *address* dari setiap *host* dalam jaringan untuk kepentingan segmentasi jaringan. Pada penelitian ini, metode *static* VLAN akan disebut sebagai metode VLAN dan *dynamic* VLAN akan disebut sebagai metode berbasis MAC *address* (MAC-*based*). Aplikasi yang telah disediakan oleh *controller* untuk mempelajari dan mencari jalur pada jaringan, akan dimodifikasi untuk menerapkan konsep segmentasi jaringan yang akan membentuk grup/*tenant* secara virtual dengan melakukan isolasi antar *host*. Setelah perancangan dan implementasi segmentasi jaringan dijalankan, akan dilakukan pengujian fungsional untuk mengetahui keberhasilan dari penerapan segmentasi jaringan dan pengujian performa untuk mengetahui perbedaan performa dari 2 metode yang digunakan pada penelitian ini, yaitu metode VLAN dan metode MAC-*based*.

2. TINJAUAN PUSTAKA

Penelitian berjudul *Role-based Campus Network slicing* oleh Chien-Hsin Chen, Chien Chen, Ssu-Hsuan Lu, dan Chien-Chao Tseng, menjelaskan tentang penerapan *network slicing*/segmentasi jaringan pada lingkungan kampus. Segmentasi jaringan diterapkan menggunakan aplikasi FlowVisor yang akan membagi jaringan pada kampus menjadi beberapa jaringan virtual berupa grup/*slice*, sesuai dengan tipe *user* seperti mahasiswa (student), tamu (guest), dan staf kampus (faculty). Jaringan kampus tersebut dijalankan pada jaringan berbasis *Software Defined Network* (SDN) menggunakan beberapa *controller* yang akan menaungi setiap *slice* yang telah dibagi oleh FlowVisor. Penelitian yang dilakukan oleh Chen et al. (2016) ini melibatkan 2 metode, yaitu metode VLAN (port-*based*) MAC-*based* dan VLAN-*based*. Pada metode MAC-*based*, gabungan dari MAC *address* dan *slice* ID akan dimasukkan ke dalam tabel flowspace pada FlowVisor. Tabel flowspace berisi kumpulan dari header paket untuk mengetahui daftar dari *slice* dan tiap anggotanya. Saat ada *node* baru yang masuk pada sebuah *slice*, *controller* yang bertugas sebagai autentikator yaitu authentication *controller*, akan mendaftarkan MAC *address* dan *slice* ID dari *node* baru tersebut ke dalam tabel flowspace pada FlowVisor dan akan dialihkan kepada *controller* yang bertanggungjawab atas *slice* tersebut. Semakin bertambahnya *node* masuk ke dalam jaringan, tabel flowspace pada FlowVisor akan semakin bertambah pula isinya yang akan mempengaruhi *latency* dari proses saat penambahan *node* baru. Untuk mengatasi hal tersebut, diusulkan metode VLAN-*based* yang memanfaatkan VLAN *tag* pada header paket untuk menggantikan peran dari tabel flowspace pada metode MAC-*based*. Tabel *flowspace* pada metode VLAN hanya akan diisi oleh *slice*

ID dari tiap *slice* yang menunjukkan *role*/peran dari *user* pada jaringan. Dari penelitian tersebut, didapatkan hasil bahwa metode VLAN-*based* dapat mengurangi nilai *latency* dibandingkan dengan metode MAC-*based* sebanyak 14% hingga 60% tergantung jumlah *device* yang terdaftar pada FlowVisor. *Latency* dapat berkurang karena pada metode VLAN-*based* tidak perlu melakukan pencarian pada tabel *flowspace* yang berisi gabungan MAC *address* dan *slice ID* dari seluruh *node* dalam jaringan saat ada *node* baru, hanya melakukan pengecekan terhadap *slice ID* saja.

Penelitian selanjutnya berjudul Implementasi *Network slicing* dengan menggunakan FlowVisor untuk Mengontrol Traffic Data *Packet* pada Jaringan *Software Defined Network* oleh Ahmad Rizal Muttaqin, Widhi Yahya, dan Reza Andria Siregar. Penelitian ini menerapkan segmentasi jaringan untuk membagi jaringan sesuai dengan tipe paket data tertentu untuk memaksimalkan performa jaringan sesuai paket tertentu. Penelitian oleh Muttaqin et al. (2018) ini membagi paket dari *Transmission Control Protocol* (TCP) dan *User Datagram Protocol* (UDP) sehingga kedua paket dapat memanfaatkan *throughput* semaksimal mungkin tanpa khawatir adanya collision/tabrakan antar paket dari kedua protokol tersebut. Pembagian jaringan menjadi 2 *slice* untuk protokol TCP dan UDP dilakukan oleh aplikasi FlowVisor dan diterapkan pada jaringan berbasis *Software Defined Network* (SDN). Dari hasil yang didapat, jaringan yang menerapkan segmentasi jaringan menggunakan FlowVisor untuk membagi sumber daya dari protokol TCP dan UDP lebih baik performanya daripada jaringan SDN biasa tanpa diterapkan *network slicing*/segmentasi jaringan.

Penelitian selanjutnya oleh M.T. Kurniawan, Muhammad Fathinuddin, Hilda Aries Widiyanti, Grace R Simanjuntak dengan judul *Network slicing on SDN using FlowVisor and POX Controller to Traffic Isolation Enforcement*, meneliti tentang pembagian jaringan menggunakan FlowVisor pada jaringan berbasis *Software Defined Network* (SDN) untuk mencapai tujuan yaitu menciptakan jaringan *multi-tenant* yang terisolasi antar satu *tenant* dan *tenant* lainnya. Penelitian tersebut berhasil melakukan isolasi antar *tenant*, sehingga *tenant* yang berbeda tidak bisa saling berkomunikasi. Pengujian lainnya adalah pengukuran performa yaitu *throughput*, *delay*, dan *jitter*, dengan hasil *throughput* pada jaringan yang menggunakan FlowVisor sedikit lebih rendah daripada jaringan SDN biasa tanpa penggunaan FlowVisor. Selain itu, penggunaan aplikasi FlowVisor untuk melakukan *network slicing*/segmentasi jaringan, membutuhkan resource tambahan sehingga nilai dari CPU usage dari *controller* lebih tinggi daripada jaringan tanpa menjalankan FlowVisor.

Askar (2017) melakukan penelitian dengan judul *SDN-Based Load Balancing Scheme for Fat-Tree Data Center Networks* yang menjalankan DCN pada jaringan berbasis SDN dan menerapkan *load balancing* untuk meningkatkan performa DCN. Hasilnya nilai *loss rate* mengalami pengurangan dan nilai *throughput* mendapat peningkatan daripada DCN dengan jaringan tradisional tanpa penerapan *load balancing* dan SDN. Penelitian tersebut dijalankan pada topologi *fat-tree* yang umum digunakan pada DCN untuk meningkatkan nilai *bandwidth* karena banyaknya *link* yang terhubung antar *switch*.

3. PERANCANGAN

3.1. Analisis Kebutuhan

Penelitian ini dilakukan dengan tujuan untuk mengetahui mekanisme segmentasi jaringan yang diterapkan pada jaringan berbasis *Software Defined Network* (SDN) menggunakan metode VLAN dan metode berbasis MAC *address* (MAC-*based*). Segmentasi pada kedua metode tersebut memiliki mekanisme yang berbeda. Pada metode VLAN digunakan mekanisme *tagging* atau pemberian tanda pada paket yang dikirim sedangkan pada metode MAC-*based* dilakukan penyaringan paket berdasarkan MAC *address* sebelum paket dikirim ke alamat tujuan. Penerapan algoritma segmentasi jaringan akan memanfaatkan fitur dan kemampuan dari Ryu *controller* sebagai *control plane*. Ryu *controller* menyediakan fungsi *push* VLAN dan *pop* VLAN yang digunakan untuk memasang dan menghapus VLAN *tag* seperti mekanisme VLAN *trunking* pada penerapan VLAN menggunakan *device* tradisional.

Pengujian pertama yang akan dilakukan adalah pengujian fungsional dengan mengirimkan paket *ping*, TCP, UDP, serta paket *broadcast* untuk melakukan pengecekan

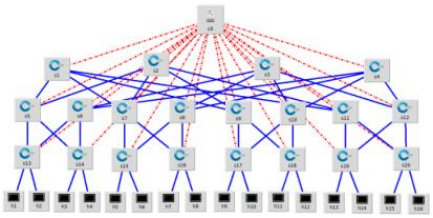
apakah SDN *controller* berhasil membagi jaringan menjadi beberapa *tenant* dan satu *tenant* dengan *tenant* lain dapat terisolasi dengan baik. Pengujian fungsional dilakukan menggunakan aplikasi Ping untuk pengiriman paket *ping* dan aplikasi Iperf untuk pengiriman paket TCP dan UDP, dan pengiriman paket broadcast melalui pemrograman *socket* yang dijalankan melalui shell *script*.

Setelah dilakukan pengujian fungsional, dilakukan pengujian performa untuk mengukur performa jaringan dengan menghitung nilai *Round Trip Time* (RTT) dan *throughput* untuk mengetahui apakah ada perbedaan performa antara metode VLAN dan metode MAC-*based*. Pengujian ini dilakukan menggunakan aplikasi Ping untuk mengetahui nilai RTT dan aplikasi Iperf untuk mengetahui nilai *throughput*.

Semua proses pada penelitian ini dilakukan pada sebuah virtual machine (VM) yang dijalankan pada komputer *host* utama menggunakan aplikasi VirtualBox dengan Ubuntu Server 20.04 sebagai sistem operasi.

3.2. Perancangan Topologi

Topologi yang digunakan pada penelitian ini adalah topologi *fat-tree* yang sering digunakan pada *Data Center Network* (DCN), yang terdiri atas 4 *core switch*, 8 *aggregation switch*, 8 *edge switch*, dan 16 *host*. Topologi disusun menggunakan aplikasi Miniedit *editor*.



Gambar 1. Topologi *Fat-Tree*

Banyaknya *link* yang tersedia pada topologi *fat-tree*, menyebabkan adanya jalur berulang atau *looping* yang dapat menyebabkan kondisi pengiriman paket *broadcast* yang terus berulang, sehingga diperlukan mekanisme *Spanning Tree Protocol* (STP) agar tidak terjadi *looping* pada paket yang dikirim dan membuat jaringan secara otomatis akan mencari jalur baru saat terjadi kegagalan jaringan pada *link*/jalur tertentu (Kubo et al., 2014). Mekanisme STP telah tersedia sebagai aplikasi dari Ryu *controller* sehingga nantinya aplikasi STP dari Ryu *controller* akan digabungkan bersama algoritma segmentasi jaringan yang akan digunakan pada penelitian ini.

Gambar 1 menampilkan topologi yang digunakan pada penelitian ini. Terdapat 4 *core switch*, yaitu *switch* s1, s2, s3, dan s4 yang pada Gambar 1 berada pada baris pertama. Pada baris kedua ada 8 *aggregation switch* yaitu *switch* s5, s6, s7, s8, s9, s10, s11, s12. Sedangkan pada baris ketiga terdapat 8 *edge switch* yaitu *switch* s13, s14, s15, s16, s17, s18, s19, s20 yang terhubung langsung dengan *host*. *Edge switch* akan memiliki tugas untuk melakukan penyaringan paket sesuai algoritma yang digunakan. Pemasangan atau penghapusan VLAN *tag*, pengecekan MAC *address* milik *host*, penerusan paket ke *host* tujuan atau melakukan *drop* pada paket yang tujuannya tidak sesuai, merupakan peran dari *edge switch*.

3.3. Perancangan Algoritma *Network slicing*

Metode pertama untuk segmentasi jaringan yaitu metode VLAN, dibuat untuk menyerupai mekanisme VLAN pada *device* tradisional menggunakan metode *static* VLAN. Ryu *controller* menyediakan mekanisme untuk menerapkan VLAN *tagging* atau pemberian tanda pada paket untuk menerapkan mekanisme pengiriman paket menggunakan VLAN. Terdapat perintah *push* VLAN untuk memasang tanda/*tag* pada paket dan perintah *pop* VLAN untuk melepas *tag* dari paket. Nomor *switch* dan nomor *port* yang akan digunakan dalam jaringan hanya perlu diinisiasi pada sebuah variabel, sehingga variabel tersebut dapat dipanggil saat dibutuhkan di bagian algoritma metode segmentasi jaringan, tanpa perlu melakukan konfigurasi pada setiap *switch* dan *port* yang akan digunakan dalam jaringan.

Metode kedua yang digunakan untuk segmentasi jaringan dijalankan menggunakan metode MAC-*based*. Berbeda dengan penerapan pada device tradisional yang melibatkan VLAN *ID*, pada penelitian ini hanya melibatkan MAC *address* sebagai pengenal paket yang akan dikirim dari *host* satu ke *host* lainnya. MAC *address* akan diinisiasi ke dalam variabel bertipe *dictionary* yang memiliki struktur *key* dan *value*. Setiap MAC *address* dari suatu *host* akan bertindak sebagai *key* yang memiliki *value* berupa MAC *address* dari *host* lain yang terdaftar dalam satu *tenant* yang sama. Dengan inisiasi variabel tersebut, *controller* akan mengetahui alamat MAC *address* yang bertindak sebagai sumber dan tujuan sesuai dengan perancangan sistem dengan melihat bagian *key* sebagai alamat sumber dan *value* sebagai alamat tujuan.

4. PENGUJIAN

Topologi jaringan yang digunakan pada penelitian ini adalah topologi *fat-tree* yang dijalankan pada emulator jaringan Mininet dengan mengeksekusi sintaks pada Gambar 2.

```
rezaa@mininet:~/skrip/net_slice$ sudo mn --controller=remote
   --custom topofatTree.py --topo fatTree4 --switch=default,
protocols=OpenFlow13 --mac --arp
```

Gambar 2. Perintah untuk menjalankan topologi

Kedua metode yang digunakan dalam penelitian ini ditulis pada *source code* menggunakan bahasa Python yang akan dijalankan oleh Ryu *Controller*. Algoritma tersebut ditulis pada *file* dengan nama *slice*VLAN.py untuk metode VLAN dan *slice*byMAC.py untuk metode MAC-*based*. Untuk menjalankan *file* tersebut, digunakan perintah "ryu-manager" dan diikuti nama *file* yang ingin digunakan seperti pada Gambar 3.

```
"Node: c0" (root)@mininet
root@mininet:/home/rezaa/skrip/net_slice# ryu-manager sliceVLAN.py
loading app sliceVLAN.py
Memulai Segmentasi Jaringan menggunakan metode VLAN...
Daftar Tenant dan anggotanya:
Tenant A: h1, h4, h7, h10, h13, h16
```

Gambar 3. Menjalankan aplikasi dari Ryu *Controller*

4.1. Pengujian Fungsional

Pengujian fungsional dilakukan melalui 4 cara, yaitu mengirim paket *ping*, TCP, UDP, dan melakukan *broadcast*. Pengujian ini dilakukan untuk mengecek keberhasilan dari kedua metode yang digunakan (VLAN dan MAC-*based*) dalam melakukan segmentasi jaringan menjadi beberapa *tenant* dan mengisolasi jaringan antar *tenant* tersebut.

Pengujian pertama yaitu pengujian paket *ping*, dilakukan dengan mengetikkan perintah "pingall" pada Mininet CLI yang merupakan perintah dasar dari aplikasi Mininet. Hasil dari pengujian ini, dapat dilihat pada Gambar 4 yang menunjukkan *host* hanya dapat mengirim paket *ping* kepada *host* lain yang terdaftar dalam satu yang sama *tenant*, sedangkan saat mengirim ke *host* yang terdaftar pada *tenant* lain, paket tidak akan sampai yang ditandai dengan karakter 'X'. Kedua metode menghasilkan *output* yang sama seperti pada Gambar 4, menandakan bahwa kedua metode berhasil membentuk *tenant* secara virtual dan melakukan isolasi jaringan pada setiap *tenant* tersebut.

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> X X h4 X X h7 X X h10 X X h13 X X h16
h2 -> X X X h5 X X h8 X X h11 X X h14 X X
h3 -> X X X X h6 X X h9 X X h12 X X h15 X
h4 -> h1 X X X X h7 X X h10 X X h13 X X h16
h5 -> X h2 X X X X h8 X X h11 X X h14 X X
h6 -> X X h3 X X X X h9 X X h12 X X h15 X
h7 -> h1 X X h4 X X X X h10 X X h13 X X h16
h8 -> X h2 X X h5 X X X X h11 X X h14 X X
h9 -> X X h3 X X h6 X X X X h12 X X h15 X
h10 -> h1 X X h4 X X h7 X X X X h13 X X h16
h11 -> X h2 X X h5 X X h8 X X X X h14 X X
h12 -> X X h3 X X h6 X X h9 X X X X h15 X
h13 -> h1 X X h4 X X h7 X X h10 X X X X h16
h14 -> X h2 X X h5 X X h8 X X h11 X X X X
h15 -> X X h3 X X h6 X X h9 X X h12 X X X
h16 -> h1 X X h4 X X h7 X X h10 X X h13 X X
*** Results: 70% dropped (70/240 received)
mininet>
```

Gambar 4. Perintah "*ping*all" pada Mininet CLI

Pengujian selanjutnya adalah pengujian pengiriman paket pada protokol TCP dan UDP, yang dilakukan menggunakan aplikasi Iperf. Pengujian ini melibatkan *host* h1 sebagai *client* atau pengirim paket dan *host* h2, h3, h4 sebagai *server* atau penerima paket pada pengujian protokol TCP. Sedangkan pada pada pengujian protokol UDP, *host* h7 sebagai *client* dan *host* h12, h11, h12 sebagai *server*. *Host* h1, h4, h7, dan h13 terdaftar pada Tenant A, host h2 dan h11 terdaftar pada Tenant B, dan host h3 dan h12 terdaftar pada Tenant C.

Tabel 1. Hasil pengiriman paket pada protocol TCP

Alamat IP pengirim	Alamat IP tujuan	Deskripsi	Hasil
10.0.0.1	10.0.0.4	Sesama Tenant A	Berhasil
10.0.0.1	10.0.0.2	Tenant A dan Tenant B	Gagal
10.0.0.1	10.0.0.3	Tenant A dan Tenant C	Gagal

Tabel 2. Hasil pengiriman paket pada protocol UDP

Alamat IP pengirim	Alamat IP tujuan	Deskripsi	Hasil
10.0.0.7	10.0.0.13	Sesama Tenant A	Berhasil
10.0.0.7	10.0.0.11	Tenant A dan Tenant B	Gagal
10.0.0.7	10.0.0.12	Tenant A dan Tenant C	Gagal

Hasil pengujian pengiriman paket melalui protokol TCP dan UDP saat menjalankan aplikasi segmentasi jaringan menggunakan metode VLAN dan metode MAC-*based* menunjukkan hasil yang sama. Hasil pengujian saat dijalankan pada protokol TCP dapat dilihat pada Tabel 1 sedangkan hasil pengujian saat dijalankan pada protokol UDP dapat dilihat pada Tabel 2. Hasil pada Tabel 1 dan Tabel 2 menunjukkan pengiriman paket dari *host* pengirim menuju *host* penerima yang terdaftar

pada *tenant* yang sama (*host* h1 menuju h4) menunjukkan status berhasil. Sedangkan pengiriman paket dari *host* pengirim menuju ke *host* penerima yang tidak terdaftar pada *tenant* yang sama (*host* h1 menuju h2 dan *host* h1 menuju h3) menunjukkan status gagal.

Selanjutnya, dilakukan pengujian untuk pengiriman paket *broadcast* untuk mengetahui apakah *tenant* satu dan lainnya sudah terisolasi dengan baik. Pengiriman paket *broadcast* dilakukan dengan menjalankan *shell script* untuk mengirim paket UDP secara *broadcast* melalui *socket*.

Tabel 3. Hasil pengiriman paket *broadcast* pada metode VLAN

Alamat IP *client*	Alamat IP *server*	Deskripsi	Hasil
10.0.0.1	10.0.0.4	Sesama Tenant A	Berhasil
10.0.0.1	10.0.0.2	Tenant A dan Tenant B	Gagal
10.0.0.1	10.0.0.3	Tenant A dan Tenant C	Gagal

Tabel 4. Hasil pengiriman paket *broadcast* pada metode MAC-*based*

Alamat IP *client*	Alamat IP *server*	Deskripsi	Hasil
10.0.0.1	10.0.0.4	Sesama Tenant A	Berhasil
10.0.0.1	10.0.0.2	Tenant A dan Tenant B	Berhasil
10.0.0.1	10.0.0.3	Tenant A dan Tenant C	Berhasil

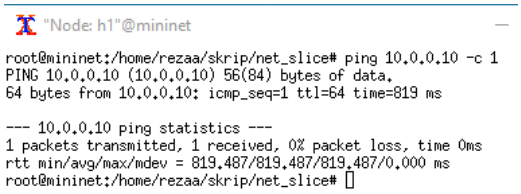
Proses *broadcast* dilakukan oleh host h1 sebagai *client* yang mengirim paket *broadcast*, sedangkan host lain yaitu host h2, h3, dan h4 bertugas sebagai *server* atau penerima paket *broadcast*. Dapat dilihat hasil pengujian pada Tabel 4 yaitu saat menggunakan metode MAC-*based*, paket *broadcast* masih bisa sampai ke *host* dari *tenant* lain. Sedangkan hasil pengujian pada Tabel 3 menunjukkan saat segmentasi jaringan diterapkan dengan metode VLAN, semua *tenant* dapat terisolasi dengan baik karena paket *broadcast* hanya sampai ke *host* yang terdaftar dalam satu *tenant* yang sama. Hasil ini menunjukkan bahwa proses segmentasi jaringan menggunakan metode VLAN lebih baik dalam mengisolasi jaringan daripada metode MAC-*based*.

4.2. Pengujian Performa

Pengujian performa berfungsi untuk mengetahui apakah terdapat perbedaan performa antara metode VLAN dan MAC-*based* dengan mengukur performa kedua metode tersebut melalui pengujian 2 parameter, yaitu *Round Trip Time* (RTT) dan *Throughput*.

Perhitungan nilai RTT dan *throughput* dilakukan dengan mengambil rata-rata dari beberapa *host* yang berasal dari semua *tenant*. Pengambilan rata-rata nilai tersebut diambil sebanyak 4 kali menggunakan beberapa variasi *host*. Perhitungan rata-rata pertama melibatkan 12 *host* yang terdiri dari 4 *host* dari setiap *tenant*, perhitungan kedua melibatkan 9 *host* yang terdiri dari 3 *host* dari setiap *tenant*, perhitungan ketiga melibatkan 6 *host* yang terdiri dari 2 *host* dari setiap *tenant*, dan yang keempat melibatkan 3 *host* yang terdiri dari 1 *host* dari setiap *tenant*. Nilai yang didapat dari pengujian pada beberapa *host* tersebut akan dihitung rata-ratanya dan hasilnya akan digunakan sebagai hasil akhir dari pengukuran nilai RTT dan *throughput*.

Pengujian performa pertama adalah pengujian nilai *Round Trip Time* (RTT) yang merupakan total waktu yang ditempuh saat paket dikirim dari *host* pengirim menuju ke *host* tujuan, sampai *host* tujuan mengirim paket balasan ke *host* pengirim. Nilai RTT akan diukur menggunakan aplikasi Ping dengan perintah "ping -c [alamat *host* tujuan]" seperti pada Gambar 5.



Gambar 5. Contoh pengiriman paket *ping*

Tabel 5. Hasil pengukuran nilai RTT

	3	6	9	12
VLAN	146,3 ms	141,3 ms	136,7 ms	128,0 ms
MAC	79,3 ms	89,1 ms	90,4 ms	62,5 ms

Saat segmentasi jaringan diterapkan menggunakan metode MAC-*based*, paket yang dikirim oleh *host* pengirim akan diseleksi langsung pada *edge switch* yang terdekat dengan *host* tersebut. Apabila MAC *address* milik *host* pengirim dan tujuan dari paket tersebut terdaftar pada *tenant* yang sama, paket akan langsung dikirim tanpa ada proses seleksi lagi. Sedangkan pada segmentasi jaringan

menggunakan metode VLAN, dibutuhkan proses pemasangan dan pelepasan VLAN *tag* pada paket yang menyebabkan nilai RTT lebih tinggi dibandingkan nilai RTT yang didapat saat menjalankan segmentasi jaringan menggunakan metode MAC-*based*.

Selanjutnya adalah pengujian performa dengan melakukan pengujian *throughput*, yang dilakukan dengan 3 variasi pengujian, yaitu: 1) Pengujian nilai *throughput* tanpa ada kondisi tambahan, 2) Pengujian nilai *throughput* serta melakukan *broadcast* pada *tenant* masing-masing, dan 3) Pengujian nilai *throughput* saat seluruh *host* dari salah satu *tenant* melakukan *broadcast* kepada seluruh *host* yang ada di dalam jaringan, baik *host* dari *tenant* yang sama, maupun *tenant* yang berbeda. Pengujian dilakukan menggunakan aplikasi iperf dengan *host* pengirim sebagai *client* dan *host* penerima sebagai *server*. Perintah iperf sebagai *client* dijalankan dengan perintah "iperf -c [nomor ip tujuan] -i 1 -t 10" sedangkan untuk *server* dijalankan dengan perintah "iperf -s -i 1".

Tabel 5. Hasil pengukuran *throughput* pertama

	3	6	9	12
VLAN	7,00 Gbits/sec	3,20 Gbits/sec	1,98 Gbits/sec	2,50 Gbits/sec
MAC	7,12 Gbits/sec	3,42 Gbits/sec	2,14 Gbits/sec	1,59 Gbits/sec

Setelah dilakukan pengujian *throughput* yang pertama, didapatkan hasil rata-rata seperti pada Tabel 5, yang menunjukkan hasil pengujian *throughput* pertama menggunakan metode MAC-*based* lebih unggul dibandingkan dengan metode VLAN. Sama seperti pengujian nilai RTT, nilai *throughput* juga dipengaruhi oleh waktu pemrosesan paket oleh suatu *host*. Sehingga berdasarkan mekanisme yang dilakukan oleh metode VLAN dengan memasang dan melepas VLAN *tag* pada *frame* paket, akan mempengaruhi nilai *throughput* yang didapat saat pengujian.

Tabel 6. Hasil pengukuran *throughput* kedua

	3	6	9	12
VLAN	5,05 Gbits/sec	2,03 Gbits/sec	1,17 Gbits/sec	0,91 Gbits/sec
MAC	5,14 Gbits/sec	2,28 Gbits/sec	1,45 Gbits/sec	1,10 Gbits/sec

Saat dilakukan pengujian *throughput* kedua, masing-masing *host* dari suatu *tenant* akan menjalankan aplikasi Iperf sambil melakukan *broadcast* kepada *host* pada *tenant* itu sendiri, yaitu Tenant A melakukan *broadcast* kepada Tenant A, Tenant B melakukan *broadcast* pada Tenant B, dan Tenant C melakukan *broadcast* kepada Tenant C. Proses *broadcast* dijalankan untuk melihat sejauh mana proses *broadcast* tersebut mempengaruhi hasil pengujian *throughput*.

Dari pengujian *throughput* kedua, didapatkan hasil seperti pada Tabel 6, yang menunjukkan penurunan nilai *throughput* apabila dibandingkan dengan hasil pengujian *throughput* pertama. Penurunan nilai *throughput* pada kedua metode yaitu VLAN dan MAC-*based* disebabkan karena adanya proses *broadcast*. Hasil yang didapat masih menunjukkan bahwa performa metode MAC-*based* lebih unggul dibandingkan performa milik metode VLAN.

Pengujian *throughput* yang ketiga dilakukan dengan cara menjalankan aplikasi Iperf, lalu seluruh *host* pada salah satu *tenant* melakukan *broadcast* menuju semua *host* yang ada dalam jaringan. Pada penelitian ini, diambil salah satu *tenant* yaitu Tenant A untuk melakukan *broadcast* menuju semua *host* pada Tenant A, Tenant B, dan Tenant C. Seluruh *host* pada Tenant A yaitu *host* h1, h4, h7, h10, h13, dan h16 bertindak sebagai pengirim paket *broadcast*. Sedangkan seluruh *host* dalam jaringan termasuk *host* pada Tenant A sendiri menjadi penerima paket *broadcast*.

Tabel 7. Hasil pengukuran *throughput* ketiga

	3	6	9	12
VLAN	4,57 Gbits/sec	1,88 Gbits/sec	1,22 Gbits/sec	0,89 Gbits/sec
MAC	3,85 Gbits/sec	1,67 Gbits/sec	1,05 Gbits/sec	0,82 Gbits/sec

Dari hasil pengujian yang dapat dilihat pada Tabel 7, nilai *throughput* pada pengujian ketiga ini menunjukkan bahwa metode VLAN lebih unggul daripada metode MAC-*based*. Hasil ini membuktikan bahwa ketidakmampuan MAC-*based* untuk menyaring paket *broadcast* membuat performa jaringannya menurun cukup banyak dibandingkan metode VLAN.

5. KESIMPULAN DAN SARAN

Berdasarkan penelitian yang telah

dilakukan, dapat ditarik kesimpulan sebagai berikut:

Pada pengujian fungsional yang dilakukan, implementasi segmentasi jaringan menggunakan metode VLAN dan MAC-based berhasil membagi jaringan secara virtual menjadi 3 grup/tenant dengan melakukan isolasi antar host sesuai dengan perancangan. Komunikasi antar host hanya dapat dilakukan antar host yang terdaftar pada tenant yang sama. Pada pengujian performa yang dilakukan, algoritma metode MAC-based memiliki nilai yang unggul untuk pengiriman paket dibandingkan metode VLAN. Hal ini disebabkan karena adanya mekanisme pemasangan dan pelepasan VLAN tag pada metode VLAN oleh edge switch. Sedangkan pada metode MAC-based, edge switch hanya melakukan pengecekan MAC address milik host pengirim dan tujuan. Dengan adanya jaringan berbasis Software Defined Network, administrator jaringan dapat melakukan konfigurasi sesuai kebutuhan melalui perancangan yang disusun pada sebuah source code dan akan dijalankan oleh SDN Controller seperti Ryu Controller. Tetapi, perlu diteliti lagi saat melakukan perancangan algoritma, karena diperlukan adanya protokol yang jelas untuk pengiriman paket seperti pada metode VLAN yang menggunakan sistem tagging/pemberian tanda pada paket sehingga mekanisme segmentasi jaringan dan pengiriman paket dapat tersampaikan sesuai perancangan. Karena saat segmentasi jaringan dijalankan menggunakan metode MAC-based, paket broadcast yang dikirim antar host yang terdaftar pada tenant berbeda, masih dapat tersampaikan.

Berdasarkan hasil penelitian yang telah dilakukan, juga terdapat saran yang ditujukan untuk penelitian lebih lanjut, yaitu:

1. Dapat dilakukan peningkatan pada algoritma segmentasi jaringan menggunakan metode MAC-*based* agar lebih baik dalam melakukan isolasi jaringan.
2. Dapat ditambahkan mekanisme untuk memudahkan penambahan *host* maupun *switch* yang lebih mudah untuk meningkatkan aspek skalabilitas.
3. Dapat dilakukan penelitian lebih lanjut dalam lingkup jaringan nirkabel yang dapat memanfaatkan aplikasi emulator jaringan Mininet-wifi sebagai simulator sebelum diterapkan pada jaringan nyata.

6. DAFTAR PUSTAKA

Afolabi, I., Taleb, T., Samdanis, K., Ksentini, A., & Flinck, H. (2018). Network slicing and softwarization: A survey on principles, enabling technologies, and solutions. IEEE Communications Surveys and Tutorials, 20(3), 2429–2453. https://doi.org/10.1109/COMST.2018.2815638

Alam, T. (2020). Cloud Computing and Its Role in the Information Technology. SSRN Electronic Journal, May. https://doi.org/10.2139/ssrn.3639063

Asadollahi, S., Goswami, B., & Sameer, M. (2018). Ryu controller's scalability experiment on software defined networks. 2018 IEEE International Conference on Current Trends in Advanced Computing, ICCTAC 2018, 1–5. https://doi.org/10.1109/ICCTAC.2018.8370397

Askar, S. (2017). SDN-Based Load Balancing Scheme for Fat-Tree Data Center Networks. Al-Nahrain Journal for Engineering Sciences, 20(5), 1047–1056. https://nahje.com/index.php/main/article/download/335/270%0Ahttps://nahje.com/index.php/main/article/view/335%0Ahttps://lens.org/055-872-205-669-722

Casado, M., Freedman, M. J., Pettit, J., Luo, J., McKeown, N., & Shenker, S. (2007). Ethane: Taking control of the enterprise. Computer Communication Review, 37(4), 1–12. https://doi.org/10.1145/1282427.1282382

Chen, C. H., Lu, S. H., Tseng, C. C., & Chen, C. (2016). Role-based campus network slicing. Proceedings - International Conference on Network Protocols, ICNP, 2016-Decem(CoolSDN), 1–6. https://doi.org/10.1109/ICNP.2016.7785315

De Oliveira, R. L. S., Schweitzer, C. M., Shinoda, A. A., & Prete, L. R. (2014). Using Mininet for emulation and prototyping Software-Defined Networks. 2014 IEEE Colombian Conference on Communications and Computing, COLCOM 2014 -

Conference Proceedings. https://doi.org/10.1109/ColComCon.2014.6860404

Gentile, A. F., Fazio, P., & Miceli, G. (2021). A Survey on the Implementation and Management of Secure Virtual Private Networks (VPNs) and Virtual LANs (VLANs) in Static and Mobile Scenarios. Telecom, 2(4), 430–445. https://doi.org/10.3390/telecom2040025

Guo, C., Yuan, L., Xiang, D., Dang, Y., Huang, R., Maltz, D., Liu, Z., Wang, V., Pang, B., Chen, H., Lin, Z. W., & Kurien, V. (2015). Pingmesh: A Large-Scale System for Data Center Network Latency Measurement and Analysis. Computer Communication Review, 45(4), 139–152. https://doi.org/10.1145/2785956.2787496

Jimson, E. R., Nisar, K., & Hijazi, M. H. A. (2018). The State of the Art of Software Defined Networking (SDN). International Journal of Information Communication Technologies and Human Development, 10(4), 44–60. https://doi.org/10.4018/ijicthd.2018100104

Jo, E., Pan, D., Liu, J., & Butler, L. (2015). A simulation and emulation study of SDN-based multipath routing for fat-tree data center networks. Proceedings - Winter Simulation Conference, 2015-Janua, 3072–3083. https://doi.org/10.1109/WSC.2014.7020145

Keti, F., & Askar, S. (2015). Emulation of Software Defined Networks Using Mininet in Different Simulation Environments. Proceedings - International Conference on Intelligent Systems, Modelling and Simulation, ISMS, 2015-Octob, 205–210. https://doi.org/10.1109/ISMS.2015.46

Kurniawan, M. T., Fathinuddin, M., Widiyanti, H. A., & Simanjuntak, G. R. (2021). Network Slicing on SDN using FlowVisor and POX Controller to Traffic Isolation Enforcement. 7th International Conference on Engineering and Emerging Technologies, ICEET 2021, October, 1–6. https://doi.org/10.1109/ICEET53442.2021.9659765

Muttaqin, A. R., Yahya, W., & Siregar, R. A. (2018). Implementasi Network Slicing dengan menggunakan Flowvisor untuk Mengontrol Traffic Data Packet pada Jaringan Software Defined Network. Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer, 2(2), 793–801. https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/960

Nguyen, V. G., & Kim, Y. H. (2016). SDN-based enterprise and campus networks: A case of VLAN management. Journal of Information Processing Systems, 12(3), 511–524. https://doi.org/10.3745/JIPS.03.0039

Nunes, B. A. A., Mendonca, M., Nguyen, X. N., Obraczka, K., & Turletti, T. (2014). A survey of software-defined networking: Past, present, and future of programmable networks. IEEE Communications Surveys and Tutorials, 16(3), 1617–1634. https://doi.org/10.1109/SURV.2014.012214.00180

Rehman, A., Siddiqui, F. A., Khan, J. R., & Saeed, M. (2019). Spanning tree protocol for preventing loops and saving energy in software defined networks along with its vulnerability and threat analyses. Advances in Intelligent Systems and Computing, 857, 1166–1180. https://doi.org/10.1007/978-3-030-01177-2_84

Taherimonfared, A., & Rong, C. (2013). Multi-tenant Network Monitoring. Multi-Tenant Network Monitoring Based on Software Defined Networking, 327–341.

Wang, T., Su, Z., Xia, Y., and Hamdi, M. (2014). Rethinking the Data Center Networking: Architecture, Network Protocols, and Resource Sharing. In IEEE Access, vol. 2, pp. 1481-1496. doi: 10.1109/ACCESS.2014.2383439.

Wulandari, R. (2016). Analisis QoS (Quality of Service) Pada Jaringan Internet. Jurnal Teknik Informatika Dan Sistem Informasi, 2(2), 162–172.

Xia, W., Zhao, P., Wen, Y., & Xie, H. (2017). A Survey on Data Center Networking (DCN): Infrastructure and Operations. In IEEE Communications Surveys and Tutorials (Vol. 19, Issue 1, pp. 640–656). IEEE. https://doi.org/10.1109/COMST.2016.2626784

Yu, M., Rexford, J., Sun, X., Rao, S., & Feamster, N. (2011). A survey of virtual LAN usage in campus networks. IEEE Communications Magazine, 49(7), 98–103. https://doi.org/10.1109/MCOM.2011.5936161

Zhao, A., Liu, Z., Pan, J., & Liang, M. (2017). A simple, cost-effective addressing and routing architecture for fat-tree based datacenter networks. 2017 IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS 2017, 36–41. https://doi.org/10.1109/INFCOMW.2017.8116349

Kubo, R., Fujita, T., & Agawa, Y. (2014). Ryu SDN Framework : Open Source SDN Infrastructure Software (Special feature: Current Status of Technology Development for Network Virtualization). In NTT Technology Journal (Vol. 26, Issue 5). RYU project team. http://ci.nii.ac.jp/naid/40020078728/