

Analisis dan Implementasi Software Defined Networking (SDN) untuk Automasi Perangkat Jaringan

Riski Amalia¹, Toibah Umi Kalsum², Riska³

^{1 2 3}Program Studi Rekayasa Sistem Komputer, Universitas Dehasen Bengkulu
*riskiaj354@gmail.com

Abstrak

Penelitian ini bertujuan untuk membangun infrastruktur automasi jaringan dengan menerapkan aplikasi web berbasis *Software Defined Networking* (SDN) untuk memudahkan administrator jaringan dalam menerapkan sistem kontroller berbasis web terhadap perangkat jaringan dengan mengoptimalkan waktu konfigurasi, manajemen dan operasi terhadap perangkat jaringan. Penelitian ini menggunakan metode action research. Di dalam metode penelitian action research, peneliti dapat mendeskripsikan, menginterpretasi dan menjelaskan suatu kondisi pada saat yang bersamaan dengan melakukan intervensi yang bertujuan untuk perbaikan atau partisipasi. Hasil dari penelitian ini menunjukkan bahwa perancangan dan implementasi aplikasi web berbasis *Software Defined Networking* (SDN) untuk automasi perangkat jaringan menggunakan bahasa pemrograman python dengan library paramiko dan framework django dapat diterapkan pada infrastruktur jaringan, sehingga dengan penerapan SDN pada aplikasi automasi perangkat jaringan dapat melakukan kontroller perangkat jaringan secara terpusat, sehingga dapat melakukan konfigurasi terhadap banyak router dapat dilakukan dengan optimal pada satu aplikasi web berbasis *Software Defined Networking* (SDN).

Kata kunci : Automasi, Jaringan, Python, SDN

Abstract

This study aims to build a network automation infrastructure by implementing a Software Defined Networking (SDN)-based web application to facilitate network administrators in implementing a web-based controller system for network devices by optimizing the configuration, management and operation time of network devices. This research uses action research method. In the action research method, researchers can describe, interpret and explain a condition at the same time as carrying out interventions aimed at improvement or participation. The results of this study indicate that the design and implementation of a Software Defined Networking (SDN)-based web application for network device automation using the python programming language with the paramiko library and the django framework can be applied to network infrastructure, so that with the application of SDN on network device automation applications can perform controllers. centralized network devices, so that configurations of many routers can be optimally carried out on a single Software Defined Networking (SDN) based web application.

Keywords : Automation, Network, Python, SDN

1. Pendahuluan

Penggunaan jaringan komputer saat ini sudah semakin pesat, dapat dilihat dari semua segi kehidupan tidak terlepas dari penggunaan jaringan komputer baik yang menggunakan kabel ataupun tidak menggunakan kabel bahkan mobile. Dalam penggunaan jaringan komputer

tentu ada komponen atau perangkat jaringan yang digunakan dalam membangun jaringan komputer tersebut seperti, router, switch, dan perangkat lainnya yang mempunyai berbagai fungsi dan protokol, terlepas apakah jaringan komputer yang digunakan menggunakan kabel ataupun tidak menggunakan kabel.

Dalam pengelolaan ataupun konfigurasi terhadap perangkat jaringan komputer, biasanya seorang administrator jaringan melakukan konfigurasi secara manual pada masing – masing perangkat yang digunakan. Kendala yang sering dialami oleh seorang administrator jaringan adalah kondisi geografis yang cukup luas dengan banyaknya penggunaan perangkat jaringan yang harus di konfigurasi secara manual pada masing – masing perangkat jaringan tersebut terutama pada router di dalam jaringan. Sebab router menjadi pusat kendali terhadap konektivitas di dalam jaringan.

Dari masalah yang sudah dituliskan diatas tadi, ada sebuah teknik yang dapat digunakan untuk memudahkan pekerjaan seorang administrator jaringan yaitu automasi jaringan atau *network automation*. *Network automation* merupakan proses automasi perangkat jaringan baik dalam segi konfigurasi, manajemen, dan operasi jaringan komputer[1][2]. Dalam pengembangan *network automation* ini, dapat menggunakan sebuah konsep ataupun arsitektur industri yang disebut dengan Software Defined Networking (SDN)[3][4]. *Software Defined Networking* (SDN) adalah sebuah paradigma arsitektur baru dalam bidang jaringan komputer, yang memiliki karakteristik dinamis, *manageable*, *cost effective*, dan *adaptable*, sehingga sangat ideal untuk kebutuhan aplikasi saat ini yang bersifat dinamis[5][6]. SDN juga bisa disebut sebagai controller untuk perangkat jaringan sehingga

dapat melakukan konfigurasi, manajemen serta operasi terhadap perangkat jaringan secara otomatis.

2. Tinjauan Pustaka

2.1 Penelitian Terkait

Dengan mengacu pada penelitian yang sebelumnya dilakukan [7] dengan judul penelitian “pengembangan manajemen jaringan berbasis software-defined network di politeknik sekayu” dengan hasil penelitian berupa aplikasi berbasis *Command Line* (CLI) menggunakan mininet menunjukkan bahwa Jaringan berbasis SDN lebih unggul dari pada jaringan konvensional karena menggunakan system terdistribusi (controller) dan sistem tersebut dapat langsung mengkalkulasi jalur dari setiap node berdasarkan bobot sisi pada topologi yang di pasang.

Penelitian lainnya dilakukan [6] dengan judul “Perancangan Simulasi Jaringan Virtual Berbasis Software-Define Networking” dengan hasil penelitian berupa controller berbasis CLI menggunakan mininet yang menunjukkan bahwa jaringan virtual berbasis SDN dapat dirancang dan dapat dijalankan dengan maksimum topologi 16 switch menggunakan sumber daya dengan spesifikasi RAM 32 GB dan CPU 16 Core. Selanjutnya ada juga penelitian yang dilakukan [8] dengan Judul penelitian “ Pengembangan Aplikasi Otomatisasi Jaringan Berbasis Website Menggunakan Bahasa Pemrograman Python” yang menghasilkan sebuah aplikasi berbasis

web yang dapat digunakan sebagai otomatisasi jaringan dalam hal konfigurasi routing static, dynamic, pembuatan VLAN, maintenance berupa backup dan restore secara terpusat sehingga akan lebih termanajemen lebih baik[8].

Penelitian berikutnya dilakukan [9] dengan judul penelitian “Pengujian Konfigurasi Otomatis Penambahan Gateway Pada Virtual Router Menggunakan Aplikasi Otomatisasi Jaringan Berbasis Web” menghasilkan sebuah aplikasi web yang dapat digunakan untuk melakukan konfigurasi penambahan gateway otomatis pada sebuah topologi jaringan yang terhubung dengan aplikasi GNS3.

Penelitian berikutnya dilakukan [10] dengan judul penelitian “Otomatisasi Perangkat Jaringan Komputer Menggunakan Ansible Pada Laboratorium Komputer” yang menghasilkan sebuah penelitian otomatisasi jaringan dengan memanfaatkan Ansible sangat memungkinkan untuk dilakukan dengan tujuan untuk mengotomatisasi perangkat jaringan seperti router, server dan lain – lain.

2.2 Landasan Teori

1. *Software Defined Networking* (SDN)

Software-Defined Networking (SDN) adalah sebuah paradigma arsitektur baru dalam bidang jaringan komputer, yang memiliki karakteristik *dinamis*, *manageable*, *cost-effective*, dan *adaptable*, sehingga sangat *ideal* untuk kebutuhan aplikasi saat ini yang bersifat dinamis

dan *high-bandwidth*[6]. Arsitektur ini memisahkan antara *network control* dan fungsi *forwarding*, sehingga *network control* tersebut menjadi *directly programmable* (dapat diprogram secara langsung), sedangkan infrastruktur yang mendasarinya dapat diabstraksikan untuk layer aplikasi dan *network services*. Dalam [11] *Software-Defined Networking* (SDN) atau split arsitektur adalah sebuah konsep yang memungkinkan/ memperbolehkan operator jaringan untuk mengelola *router* dan *switch* secara fleksibel menggunakan *software* yang berjalan di *server* eksternal.

2. Automasi Jaringan

Network automation atau automasi jaringan adalah sebuah metodologi yang mana perangkat lunak secara otomatis mengelola, mengkonfigurasi dan menguji perangkat jaringan (*network devices*). Biasanya *network automation* ini digunakan oleh perusahaan untuk mengurangi kesalahan manusia (*human error*), meningkatkan efisiensi dan juga biaya operasinya[2]. Sebagai contoh, misal kita memiliki ratusan atau bahkan ribuan router, jika tanpa *network automation*, kita harus mengakses setiap device satu persatu jika ingin melakukan monitoring atau konfigurasi pada router. Disinilah peran *Network Automation* sangat dibutuhkan, dengan *Network Automation*, kita bisa mengelola banyak sekali device hanya pada satu dashboard. Sehingga jika kita ingin melakukan monitoring atau konfigurasi pada suatu *device*,

kita tidak perlu login ke setiap *device*, kita cukup melakukannya dari satu *dashboard*[1].

3. Aplikasi Web

Aplikasi web adalah aplikasi yang disimpan dan dieksekusi di lingkungan web *server*. Setiap permintaan yang dilakukan oleh *client* melalui aplikasi web *browser* akan diproses oleh aplikasi web dan hasilnya akan dikembalikan lagi dihadapan *user*. Dengan aplikasi web, halaman yang tampil dilayar web *browser* dapat bersifat dinamis, tergantung dari nilai data atau parameter yang dimasukkan oleh *user*[12].

4. GNS 3

GNS3 adalah sebuah program *graphical network simulator* yang dapat mensimulasikan topologi jaringan yang lebih kompleks dibandingkan dengan simulator lainnya. Program ini dapat dijalankan di berbagai sistem operasi, seperti Windows, Linux, atau Mac OS X [13].

5. Python

Python adalah bahasa pemrograman interpretatif yang dianggap mudah dipelajari serta berfokus pada keterbacaan kode. Dengan kata lain, Python diklaim sebagai bahasa pemrograman yang memiliki kode – kode pemrograman yang sangat jelas, lengkap, dan mudah untuk dipahami[14].

6. Paramiko

Paramiko adalah *library* Python (2.7, 3.4+) dari protokol SSHv2 yang menyediakan fungsionalitas *client* dan *server*. Sementara itu memanfaatkan ekstensi Python C untuk

kriptografi tingkat rendah (Kriptografi), Paramiko sendiri adalah antarmuka Python murni di sekitar konsep jaringan SSH[15].

7. Django

Django adalah *web framework* berbasis bahasa pemrograman Python. Django adalah *web framework* Python yang didesain untuk membuat aplikasi web yang dinamis, kaya fitur dan aman. Django yang dikembangkan oleh Django *Software Foundation* terus mendapatkan perbaikan sehingga membuat *web framework* yang satu ini menjadi pilihan utama bagi banyak pengembang aplikasi web[16].

8. Mikrotik

Mikrotik adalah suatu sistem operasi dan perangkat lunak yang dapat digunakan untuk menjadikan komputer menjadi *router network* yang handal, mencakup berbagai fitur yang dibuat untuk ip network dan jaringan *wireless*, cocok digunakan oleh ISP dan *provider hotspot*. Untuk instalasi Mikrotik tidak dibutuhkan piranti lunak tambahan atau komponen tambahan lain. Mikrotik didesain untuk mudah digunakan dan sangat baik digunakan untuk keperluan administrasi jaringan komputer seperti merancang dan membangun sebuah sistem jaringan komputer skala kecil hingga yang kompleks sekalipun[17]. Kemampuan yang di tawarkan oleh mikrotik sebagai *router* yang handal juga hampir tidak bisa di percaya, dari mulai management bandwidth, data user, *firewall*, *wireless access point* (WiFi), *backhaul*

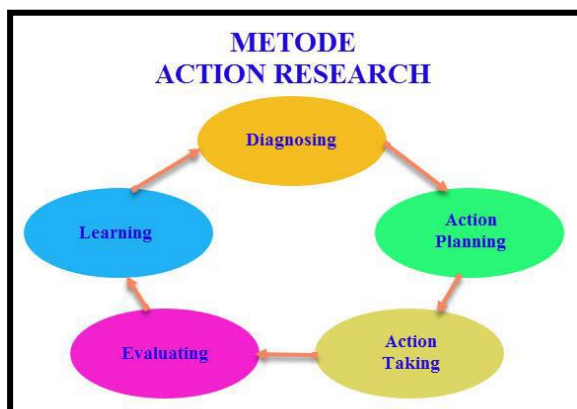
link, sistem *hotspot*, *Virtual Private Network* (VPN) *server* dan bahkan mikrotik mampu menggabungkan 2 internet gateway atau lebih ke dalam satu *router*[18].

9. Cisco

Cisco adalah peralatan utama yang banyak digunakan pada Jaringan Area Luas atau *Wide Area Network* (WAN). Dengan *cisco router*, informasi dapat diteruskan ke alamat-alamat yang berjauhan dan berada di jaringan computer yang berlainan[19].

3. Metode Penelitian

Metode penelitian yang digunakan adalah metode *action research*. Di dalam metode penelitian *action research*, peneliti dapat mendeskripsikan, menginterpretasi dan menjelaskan suatu kondisi pada saat yang bersamaan dengan melakukan intervensi yang bertujuan untuk perbaikan atau partisipasi. Berikut merupakan gambar dan tahapan dari metode penelitian *action research*, yaitu:



Gambar 1 Metode Action Research

1. *Diagsoning*

Pada tahapan ini, dilakukan identifikasi terhadap masalah yang berkaitan langsung pada infrastruktur jaringan yang akan diteliti guna menjadi dasar dan pertimbangan penelitian dalam melakukan pengembangan dan pemaksimalan kinerja automasi jaringan atau *controller* jaringan berbasis *Software Defined Networking*.

2. *Action Planning*

Pada tahapan ini, dilakukan perencanaan solusi agar dapat menyelesaikan masalah yang terjadi. Pada penelitian ini, *action planning* dilakukan mulai dari membuat daftar dari kebutuhan sistem baik dari segi perangkat keras maupun perangkat lunak yang akan digunakan serta membuat rencana kerja yang akan dilakukan.

3. *Action Taking*

Pada tahapan ini, akan dilakukan implementasi dari rencana kerja yang sudah dibuat yang selanjutnya akan dilakukan sesuai dengan tujuan dari penelitian ini.

4. *Evaluating*

Pada tahapan ini, dilakukan pengujian terhadap automasi jaringan berbasis SDN yang sudah diimplementasikan pada tahapan *action taking*.

5. *Learning*

Pada tahapan ini, akan dilakukan pembelajaran dari penelitian yang sudah dilakukan untuk mengetahui hasil dari penelitian tersebut dan kemudian akan dibuat kesimpulan mengenai penelitian yang sudah dilakukan.

3.1 Teknik Pengumpulan Data

Untuk memperoleh data yang diperlukan dalam penyusunan skripsi nanti penulis menggunakan beberapa metode dalam pengumpulan data yaitu:

1. Observasi

Merupakan metode pengumpulan data yang digunakan dengan cara melakukan pengamatan langsung pada jaringan yang ada di laboratorium komputer UPT. Puskom Universitas Dehasen Bengkulu.

2. Studi Pustaka

Merupakan metode pengumpulan data yang dilakukan dengan cara membaca buku-buku di perpustakaan kampus maupun perpustakaan daerah dan artikel di *internet* yang ada hubungannya dengan masalah yang dibahas dalam penelitian ini.

3. Studi Laboratorium

Data penelitian dikumpulkan dengan melakukan percobaan di laboratorium komputer UPT. Puskom Universitas Dehasen Bengkulu, mengenai cara perancangan infrastruktur jaringan.

3.2 Lokasi Penelitian

Penelitian dilaksanakan di Laboratorium Jaringan UPT. Puskom Universitas Dehasen Bengkulu yang beralamatkan di Jl. Meranti Raya No. 32 Sawah Lebar Bengkulu.

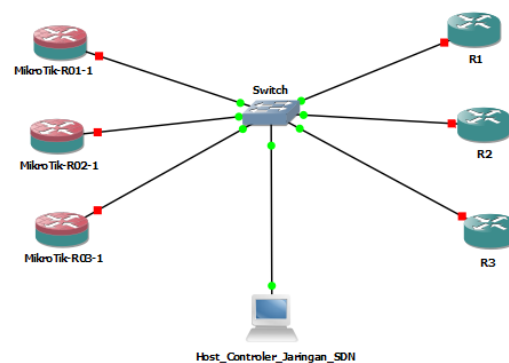
4. Hasil dan Pembahasan

4.1 Prinsip Kerja Sistem

Prinsip kerja dari automasi perangkat jaringan berbasis *Software Defined Networking* (SDN) atau bisa juga disebut dengan *SDN Controller* ini adalah mentranslasikan kebutuhan antara aplikasi dan infrastruktur jaringan dengan memberikan instruksi yang sesuai dengan SDN dapatkan dan relevan dengan *SDN Application*. Dimana proses itu dilakukan sesuai dengan apa yang dituliskan pada program yang dirancang sesuai dengan kebutuhan dari infrastruktur dalam jaringan.

4.2 Desain Topologi

Pada penelitian ini akan dilakukan perancangan arsitektur jaringan secara virtual menggunakan aplikasi GNS 3 dengan menerapkan sebuah *controller* berbasis *Software Defined Networking* (SDN) dalam melakukan automasi terhadap perangkat jaringan. Adapun topologi yang akan digunakan adalah sebagai berikut.

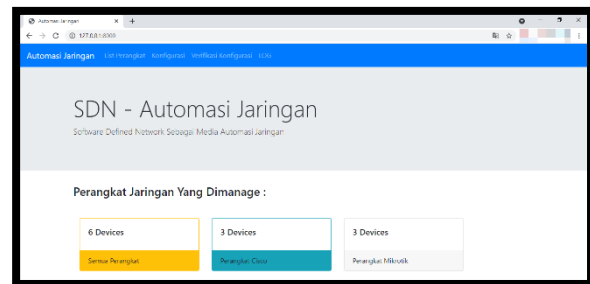


Gambar 3 Topologi Jaringan Virtual

Pada Gambar 3 diatas, dapat dilihat infrastruktur ataupun topologi dari jaringan komputer berbasis SDN untuk automasi perangkat jaringan, dimana terdapat tiga *router* Mikrotik dan tiga *router* cisco yang akan di automasi dengan teknologi SDN. Untuk menerapkan automasi perangkat jaringan tersebut akan dilakukan melalui *host controller* berbasis *cloud* yang terhubung langsung dengan program berbasis SDN yang dapat di kelola melalui apliasi yang sudah penulis rancang dengan bahasa pemrograman python menggunakan library paramiko dan framework web django. Adapun hasil dari aplikasi yang sudah penulis terapkan adalah sebagai berikut.

4.3 Perancangan Aplikasi Web Berbasis SDN

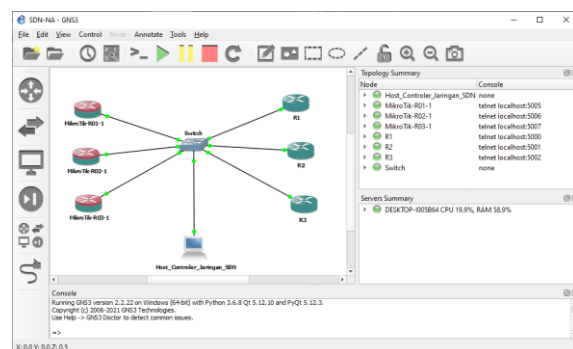
Aplikasi web berbasis *Software Defined Networking* (SDN) dalam penelitian ini penulis bangun menggunakan bahasa pemrograman python dengan menggunakan *library* paramiko dan *framework* web django dan juga *framework* bootstrap. Diaman dalam proses membangun aplikasi tersebut ada beberapa hal penting yang harus dilakukan, seperti membuat project pada python, menambahkan tema bootstrap, membuat models (basis data), membuat tampilan list perangkat, membuat menu konfigurasi, membuat menu verifikasi konfigurasi, membuat menu log. Sehingga hasil dari perancangan aplikasi web berbasis SDN menjadi seperti berikut ini.



Gambar 4 Hasil Perancangan Aplikasi Web

4.4 Infrastruktur Jaringan di GNS3

Berikut ini implementasi dari infrastruktur yang sudah penulis rancang sebelumnya.



Gambar 5 Infrastruktur Jaringan pada GNS3

4.5 Pengujian Koneksi ke Infrastruktur Jaringan Virtual

Pengujian ini digunakan untuk memastikan bahwa infrastruktur jaringan yang sudah dibuat pada aplikasi GNS3 dapat terhubung dengan aplikasi *Software Defined Networking* (SDN) untuk automasi jaringan melalui port SSH. Pengujian ini dilakukan pada menu verifikasi konfigurasi. Adapun hasil yang di dapatkan dapat dilihat seperti berikut ini.

a. Hasil Verifikasi Router Mikrotik

Untuk melakukan verifikasi koneksi dari router mikrotik dapat dilakukan dengan memasukkan command mikrotik "ip address print" pada menu verifikasi konfigurasi. Berikut ini merupakan hasil

verifikasi dari router mikrotik melalui menu verifikasi konfigurasi.



```

Hasil Verifikasi

Result on 192.168.11.2
Flags: X - disabled, I - invalid, D - dynamic
# ADDRESS NETWORK INTERFACE
0 192.168.11.2/24 192.168.11.0 ether1

Result on 192.168.11.3
Flags: X - disabled, I - invalid, D - dynamic
# ADDRESS NETWORK INTERFACE
0 192.168.11.3/24 192.168.11.0 ether1
1 172.16.1.1/32 172.16.1.1 loopback0

Result on 192.168.11.4
Flags: X - disabled, I - invalid, D - dynamic
# ADDRESS NETWORK INTERFACE
0 192.168.11.4/24 192.168.11.0 ether1
  
```

Gambar 6 Hasil Verifikasi Router Mikrotik

b. Hasil Verifikasi Router Cisco

Untuk melakukan verifikasi koneksi dari *router* cisco dapat dilakukan dengan memasukkan command cisco “show ip interface brief” pada menu verifikasi konfigurasi. Berikut ini merupakan hasil verifikasi dari *router* cisco melalui menu verifikasi konfigurasi.



```

Hasil Verifikasi

Result on 192.168.11.5

R1#terminal length 0
R1#show ip interface brief
Interface IP-Address OK? Method Status Protocol
FastEthernet0/0 192.168.11.5 YES manual up
FastEthernet0/1 unassigned YES unset administratively down down
R1#

Result on 192.168.11.6

R2#terminal length 0
R2#show ip interface brief
Interface IP-Address OK? Method Status Protocol
FastEthernet0/0 192.168.11.6 YES manual up
FastEthernet0/1 unassigned YES unset administratively down down
R2#

Result on 192.168.11.7

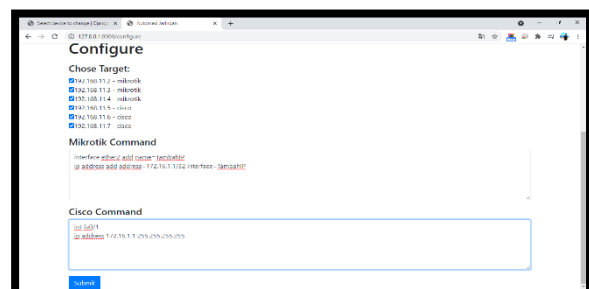
R3#terminal length 0
R3#show ip interface brief
Interface IP-Address OK? Method Status Protocol
FastEthernet0/0 192.168.11.7 YES manual up
FastEthernet0/1 unassigned YES unset administratively down down
R3#
  
```

Gambar 7 Hasil Verifikasi Router Cisco

Dari kedua pengujian yang sudah dilakukan, dapat dilihat bahwa aplikasi web berbasis SDN sebagai media untuk melakukan automasi terhadap perangkat jaringan sudah dapat terhubung atau dengan kata lain sudah ada konektifitas dengan ditampilkannya IP *address*, *hostname*, *interface* yang digunakan pada masing – masing router yang terhubung.

4.6 Pengujian Penambahan IP Address

Pengujian ini dilakukan untuk memastikan bahwa aplikasi automasi perangkat jaringan dapat melakukan konfigurasi melalui satu controller. Penambahan IP address pada *router* mikrotik dapat dilakukan dengan cara mengetikkan command “ip address add address=192.168.22.1/24 interface=ether2”, sedangkan untuk penambahan IP address pada *router* cisco dengan menggunakan command “int fa0/1 \n ip address 192.168.22.1 255.255.255.0” seperti yang terlihat seperti gambar berikut.



Gambar 8 Penambahan IP address

Setelah mengetikkan perintah pada masing – masing command seperti Gambar 8, maka hasilnya dapat dilihat seperti berikut ini.


```

Hasil Verifikasi
Result on 192.168.11.2
Flags: X - disabled, I - invalid, D - dynamic
# ADDRESS NETWORK INTERFACE
0 192.168.11.2/24 192.168.11.0 ether1

Result on 192.168.11.3
Flags: X - disabled, I - invalid, D - dynamic
# ADDRESS NETWORK INTERFACE
0 192.168.11.3/24 192.168.11.0 ether1
1 172.16.1.1/32 172.16.1.1 loopback0

Result on 192.168.11.4
Flags: X - disabled, I - invalid, D - dynamic
# ADDRESS NETWORK INTERFACE
0 192.168.11.4/24 192.168.11.0 ether1

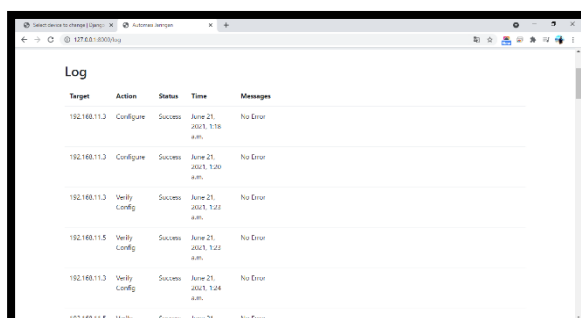
Result on 192.168.11.5
R1#terminal length 0
R1#show ip interface brief
Interface IP-Address OK? Method Status Protocol
FastEthernet0/0 192.168.11.5 YES manual up up
FastEthernet0/1 unassigned YES unset administratively down down
R1#
Result on 192.168.11.6
R2#terminal length 0
R2#show ip interface brief
Interface IP-Address OK? Method Status Protocol
FastEthernet0/0 192.168.11.6 YES manual up up
FastEthernet0/1 unassigned YES unset administratively down down
R2#
Result on 192.168.11.7
R3#terminal length 0
R3#show ip interface brief
Interface IP-Address OK? Method Status Protocol
FastEthernet0/0 192.168.11.7 YES manual up up
FastEthernet0/1 unassigned YES unset administratively down down

```

Gambar 9 Hasil Penambahan IP address

4.7 Pengujian Pemantauan Log

Pengujian pemantauan log dilakukan untuk mengamati hasil konfigurasi dari aplikasi SDN untuk automasi perangkat jaringan yang sudah dilakukan. Adapun hasil dari log hasil konfigurasi automasi perangkat jaringan dapat dilihat seperti berikut ini.



Target	Action	Status	Time	Messages
192.168.11.3	Configure	Success	June 21, 2021, 1:18 am.	No Error
192.168.11.3	Configure	Success	June 21, 2021, 1:20 am.	No Error
192.168.11.3	Verify Config	Success	June 21, 2021, 1:22 am.	No Error
192.168.11.5	Verify Config	Success	June 21, 2021, 1:23 am.	No Error
192.168.11.3	Verify Config	Success	June 21, 2021, 1:24 am.	No Error
192.168.11.7	Verify Config	Success	June 21, 2021, 1:24 am.	No Error

Gambar 10 Hasil Pemantauan Log

Dari Gambar 10 diatas, dapat dilihat log dari hasil konfigurasi yang sudah dilakukan, dimana pada log ditampilkan informasi berupa Ip address sebagai target, action, status sukses, time dan juga messages no error yang menunjukkan bahwa aplikasi automasi perangkat jaringan

dapat melakukan konfigurasi secara terpusat dengan menggunakan aplikasi web berbasis SDN yang sudah penulis terapkan.

5. Kesimpulan

Berdasarkan penelitian yang sudah penulis lakukan, maka dapat disimpulkan bahwa aplikasi web berbasis *Software Defined Networking* (SDN) untuk automasi perangkat jaringan dapat diterapkan pada infrastruktur jaringan virtual yang sudah di implementasikan pada aplikasi GNS3, sehingga konfigurasi terhadap banyak router yang ada pada aplikasi GNS3 dapat dilakukan dengan optimal pada satu aplikasi berbasis web dengan menerapkan metode *Software Defined Networking* (SDN).

6. Daftar Pustaka

- [1] A. R. Komarudin, "Belajar Network Automation berbasis Web dengan Django," *Udemy.com*, 2020. [Online]. Available: <https://www.udemy.com/course/belajar-network-automation-berbasis-web-dengan-django/>.
- [2] Netmonk, "5 Alasan Utama untuk Network Automation," *netmonk.id*, 2020. [Online]. Available: <https://netmonk.id/5-alasan-utama-untuk-network-automation>.
- [3] T. I. Bayu and E. E. Tahan, "Software Defined Network (Sdn) Simulation Concept Using Raspberry Pi," *J. Terap. Teknol. Inf.*, vol. 2, no. 2, pp. 1–11, 2018.

- [4] K. NUGROHO and D. P. SETYANUGROHO, "Analisis Kinerja RouteFlow pada Jaringan SDN (Software Defined Network) menggunakan Topologi Full-Mesh," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 7, no. 3, p. 585, 2019.
- [5] I. Hidayat and B. A. Perdana, "Arsitektur Software Defined Network : Implementasi Pada Small Network," vol. 01, no. 01, pp. 1–13, 2020.
- [6] I. Ummah and D. Abdillah, "Perancangan Simulasi Jaringan Virtual Berbasis Software-Define Networking," *Indones. J. Comput.*, vol. 1, no. 1, pp. 95–106, 2016.
- [7] R. Annisa, "Pengembangan Manajemen Jaringan Berbasis Software-Defined Network Di Politeknik Sekayu Resty Annisa Program Studi Teknik Informatika Politeknik Sekayu Email styannisa@gmail.com," *J. Tek. Inform. Politek. Sekayu*, vol. VII, no. 2, pp. 33–43, 2017.
- [8] R. A. Wiryawan and N. R. Rosyid, "Pengembangan Aplikasi Otomatisasi Administrasi Jaringan Berbasis Website Menggunakan Bahasa Pemrograman Python," *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 10, no. 2, pp. 741–752, 2019.
- [9] E. S. Ginting and I. Hadi, "Pengujian Konfigurasi Otomatis Penambahan Gateway Pada Virtual Router Menggunakan Aplikasi Otomatisasi Jaringan Berbasis Web," vol. 4, pp. 1126–1131, 2020.
- [10] M. Rifki Afandi, P. Hatta, A. Efendi, K. Kunci-Otomatisasi Jaringan, L. Komputer, and P. Jaringan, "Otomatisasi Perangkat Jaringan Komputer Menggunakan Ansible Pada Laboratorium Komputer," *SMARTICS J.*, vol. 6, no. 2, pp. 48–53, 2020.
- [11] R. Kartadie, E. Utami, and E. Pramono, "Prototipe Infrastruktur Software-Defined Network Dengan Protokol Openflow Menggunakan Ubuntu Sebagai Kontroler," *Progr. Magister Tek. Inform. Progr. Pascasarj. STMIK AMIKOM Yogyakarta*, vol. 15, no. 1, pp. 24–32, 2014.
- [12] I. Gunawan and T. Akbar, "Prototipe Sistem Monitoring Tegangan Panel Surya (Solar Cell) Pada Lampu Penerang Jalan Berbasis Web Aplikasi," *Infotek J. Inform. dan Teknol.*, vol. 2, no. 2, pp. 70–78, 2019.
- [13] A. Nurhayati and E. Indriyani, "Simulasi Pengiriman Paket VoIP Menggunakan Simulator GNS3 Versi 0.8.6," *J. ICT Penelit. dan Penerapan Teknol.*, pp. 91–98, 2017.
- [14] J. Enterprise, *Python untuk Programmer Pemula*. Elex media komputindo, 2019.
- [15] J. Forcier, "Welcome to Paramiko!," *paramiko.org*, 2020. [Online]. Available: <http://www.paramiko.org/#id1>.
- [16] D. Saputra and R. F. Aji, "Analisis Perbandingan Performa Web Service Rest

- Menggunakan Framework Laravel, Django dan Ruby On Rails Untuk Akses Data Dengan Aplikasi Mobile (Studi Kasus: Portal E-Kampus STT Indonesia Tanjungpinang),” *J. Ilm. Bangkit Indones.*, vol. 2, no. VII, pp. 17–22, 2018.
- [17] M. I. Septian, “Analisa Perbandingan Metode Per Connection Classifier (Pcc) Dengan Equal Cost Multi Path (Ecmp) Dalam Pemanfaatan Teknik Load Balancing Pada Mikrotik,” *BILTEK*, vol. 12, no. 3, 2017.
- [18] Y. Kuspani Putra, M. Sadali, and M. Mahpuz, “Penerapan Mikrotik Dalam Mengembangkan Infrastruktur Jaringan Pada Kantor Desa Rumbuk Kecamatan Sakra,” *Infotek J. Inform. dan Teknol.*, vol. 3, no. 2, pp. 182–193, 2020.
- [19] H. Hermawan and M. Siddik, “Rancang Bangun Aplikasi Pembelajaran Interaktif Materi Cisco di Sekolah Tinggi Ilmu Komputer Pelita Indonesia,” ... *Teknol. Komput. dan ...*, vol. 2, no. 1, pp. 62–67, 2020.