Volume 4, Nomor 4, Oktober 2020, Page 1126-1131 ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online) Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib DOI 10.30865/mib.v4i4.2485



Pengujian Konfigurasi Otomatis Penambahan *Gateway* Pada *Virtual Router* Menggunakan Aplikasi Otomatisasi Jaringan Berbasis *Web*

Elin Sylvania Ginting*, Suroso, Irawan Hadi

Jurusan Teknik Elektro, Program Studi Teknik Telekomunikasi, Politeknik Negeri Sriwijaya, Palembang, Indonesia Email: ^{1,*}elinsylvania@gmail.com, ²osorus11@gmail.com, ³irawanhadi@polsri.ac.id Email Penulis Korespondensi: elinsylvania@gmail.com

Abstrak—Konfigurasi pada router saat ini masih dilakukan secara konvensional, yang berarti bahwa untuk melakukan konfigurasi router dalam suatu jaringan yang kompleks, seorang administrator jaringan melakukan konfigurasi router secara satu-persatu. Hal ini sangat tidak efisien, sebab apabila router yang ingin dikonfigurasi dilakukan dalam jumlah ratusan atau ribuan router, maka waktu yang dibutuhkan oleh seorang administrator jaringan sangat lama. Oleh sebab itu diperlukan otomatisasi. Penelitian ini membuat aplikasi otomatisasi jaringan berbasis web dengan python, library paramiko yang berfungsi sebagai jembatan penghubung antara server dengan perangkat jaringan menggunakan protokol SSH (Secure Shell), dan framework django yang dapat melakukan konfigurasi IP Gateway secara otomatis. Otomatisasi jaringan dilakukan secara simulasi menggunakan aplikasi GNS3 pada topologi jaringan yang telah didesain terlebih dahulu. Metode pengujian aplikasi otomatisasi jaringan yang digunakan adalah metode black-box testing. Adapun keluaran dari penelitian ini adalah sebuah website yang terdiri dari halaman pengguna dan halaman admin yang dapat melakukan konfigurasi router virtual secara otomatis.

Kata Kunci: Otomatisasi Jaringan, Python, IP Gateway, Paramiko, Django, GNS3

Abstract—Current router configuration is still done conventionally, which means that to configure routers in a complex network, a network administrator configures the routers one by one. This is very inefficient, because if the router you want to configure is done in hundreds or thousands of routers, then the time needed by a network administrator is very long. Therefore, automation is needed. This study creates a web-based network automation application with python, the paramiko library which functions as a bridge between the server and network devices using the SSH (Secure Shell) protocol, and the django framework that can configure IP Gateway automatically. Network automation is carried out in a simulation using the GNS3 application on a previously designed network topology. The network automation application testing method used is the black-box testing method. The output of this research is a website consisting of a user page and an admin page that can configure virtual routers automatically.

Keywords: Network Automation, Python, IP Gateway, Paramiko, Django, GNS3

1. PENDAHULUAN

Konfigurasi *router* saat ini masih dilakukan secara konvensional, yang berarti bahwa untuk melakukan konfigurasi *router* dalam suatu jaringan yang kompleks, seorang administrator jaringan harus melakukan konfigurasi *router* secara satu-persatu. Hal ini sangat tidak efisien, sebab apabila *router* yang ingin dikonfigurasi dilakukan dalam jumlah ratusan atau ribuan *router*, maka waktu yang dibutuhkan oleh seorang administrator jaringan sangat lama. Hal ini juga menyulitkan administrator jaringan apabila *router* yang dikonfigurasi memiliki merek yang berbeda sehingga langkah-langkah untuk melakukan konfigurasi pada setiap merek *router* juga memakan waktu yang sangat lama yang menimbulkan banyaknya biaya yang keluar. Dengan begitu, penting bagi perusahaan-perusahaan untuk melakukan otomatisasi jaringan. Otomatisasi jaringan adalah proses mengotomatisasi konfigurasi, pengujian, operasi perangkat virtual dalam jaringan, dan manajemen.

Pengontrolan konfigurasi *router* pada otomatisasi jaringan dapat dilakukan melalui *script* yang menggunakan bahasa pemrograman *python* ataupun aplikasi berbasis web. Bahasa pemrograman yang digunakan untuk melakukan otomatisasi jaringan adalah bahasa pemrograman python dengan menggunakan *library* paramiko. Paramiko merupakan implementasi Python (2,7, 3,4+) dari protokol SSHv2 yang menyediakan fungsionalitas klien dan server. Paramiko juga dapat disebut sebagai antarmuka *python* murni di sekitar konsep jaringan SSH. [1]

Dengan mengacu pada penelitian yang telah dilakukan sebelumnya oleh Rheza Adhyatmaka Wiryawan dan Nur Rohman Rosyid pada jurnal yang berjudul "Pengembangan Aplikasi Otomatisasi Jaringan Berbasis Website Menggunakan Bahasa Pemrograman Python" menerapkan otomatisasi pada aplikasi berbasis website menggunakan bahasa python pada library paramiko dengan metode RAD (Rapid Application Development) dan diterapkan pada perangkat Mikrotik dan Cisco yang menghasilkan beberapa fitur, antara lain: konfigurasi routing, restore, backup, setting, dan vlan. Pengujian pada kelima fitur tersebut dilakukan dengan menggunakan metode black box testing pada semua fungsi aplikasi yang dikembangkan berhasil diterapkan pada perangkat Mikrotik dan Cisco. [2]

Penelitian lain dilakukan oleh Ahmad Rosid Komarudin pada bukunya yang berjudul "Otomatisasi Administrasi Jaringan Dengan Script *Python*", ia menjelaskan tentang bagaimana menerapkan otomatisasi administrasi jaringan menggunakan *library* netmiko, napalm, paramiko, ansible, dan pytnc yang diterapkan pada perangkat Cisco. Dari percobaan yang telah dilakukan, Ahmad Rosid Komarudin menyimpulkan bahwa *library* napalm dan *library* ansible merupakan *library* dengan fitur yang paling *powerfull* yang dapat diterapkan pada

Elin Sylvania Ginting, Copyright ©2020, MIB, Page 1126 Submitted: 08/09/2020; Accepted: 24/09/2020; Published: 20/10/2020

Volume 4, Nomor 4, Oktober 2020, Page 1126-1131

ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online) Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib DOI 10.30865/mib.v4i4.2485



perangkat Cisco. Dukungan pada *library* napalm masih terbatas seperti: Cisco IOS-XR, Juniper JunOS, Cisco NX-OS dan Arista EOS. Sedangkan pada *library* ansible, sudah dapat mendukung banyak jenis perangkat jaringan dan memiliki modul tersendiri untuk jaringan. [3]

Penelitian lain dilakukan oleh I Made Bayu Swatika dan I Gede Oa Gartria Atitama pada jurnal yang berjudul "Otomatisasi Konfigurasi Mikrotik *Router* Menggunakan *Software* Ansible" tentang implementasi otomatisasi jaringan pada *Router* OS Mikrotik menggunakan Ansible dengan mengatur *bandwidth* pada fitur *Queue Tree* di Mikrotik. Pengimplementasian dilakukan dengan menambahkan modul *Router*OS API pada Ansible agar *router* mikrotik dapat terkonfigurasi melalui API saat *software* Ansible telah terhubung. Pengujian pada penelitian ini juga menggunakan metode *Black-Box Testing* dengan hasil bahwa proses 19 konfigurasi *queue tree* dapat dilakukan secara berurutan pada satu eksekusi file dalam waktu kurang lebih satu menit dengan menggunakan *software* Ansible. [4]

Penelitian lain dilakukan oleh Paul MIHĂILĂ, Titus BĂLAN, Radu CURPEN, dan Florin SANDU pada jurnal yang berjudul "Network Automation and Abstraction using Python Programming Methods" menjelaskan bahwa pada jaringan yang tidak mendukung protokol OpenFlow SDN sangat penting untuk dilakukan otomatisasi. Sehingga pada tiga switch cisco dilakukan otomatisasi pembuatan VLAN yang dikontrol dengan menggunakan script python. Dengan menggunakan python, administrator jaringan tidak perlu melakukan konfigurasi secara satu-persatu pada setiap perangkat jaringan. Sehingga administrator jaringan hanya perlu menerapkan scripting otomatisasi dan membuat infrastuktur yang tetap. Otomatisasi dapat menggunakan koneksi Secure Shell dan python. [5]

Dengan beberapa acuan dari jurnal yang telah disebutkan di atas, penulis tertarik melakukan konfigurasi otomatis penambahan *gateway* pada *router* pada aplikasi otomatisasi jaringan berbasis web yang dibangun melalui *script* dengan menggunakan bahasa pemrograman *python*, *library* paramiko, dan *framework* django. Metode pengujian yang digunakan untuk konfigurasi otomatis penambahan *gateway* pada *router* di aplikasi otomatisasi jaringan berbasis web ialah pengujian koneksi dan metode *black-box testing* untuk pengujian fungsional aplikasi otomatisasi jaringan berbasis web. Semua tindakan konfigurasi *router* dilakukan secara simulasi menggunakan aplikasi GNS3 dan melibatkan dua merek *router* berbeda, yaitu mikrotik dan cisco.

2. METODOLOGI PENELITIAN

2.1 Alat dan Bahan

Dalam melakukan penelitian ini dibutuhkan alat yang berupa perangkat keras dan bahan berupa perangkat lunak diantaranya:

a) Perangkat keras

Spefisikasi perangkat keras untuk pengujian termuat pada Tabel 1.

Tabel 1. Spefikasi perangkat keras

Perangkat	Nama Perangkat	Spesifikasi	
1 Laptop	Samsung Model NP355V4X 64 bit	Processor AMD A6-4400M APU with	
	Samsung Model NP333 V4X 04 bit	Radeon(tm) HP Graphics 2.70 GHz	
Memori DDR3	-	8 GB	
Hardisk	-	1 TB	

b) Perangkat lunak

- 1. Aplikasi Python yang digunakan untuk pengembangan aplikasi
- 2. Library Paramiko yang digunakan untuk otomatisasi
- 3. Django yang digunakan untuk framework full-stack
- 4. GNS3 yang digunakan untuk simulasi topologi jaringan
- 5. Google Chrome yang digunakan untuk web browser
- 6. Putty yang digunakan untuk koneksi SSH

2.2 Metode Penelitian

Metode yang digunakan dalam penelitian ini adalah:

- a. Pengumpulan data yang dilakukan melalui observasi, wawancara dan studi literatur.
- b. Perancangan yang dilakukan merupakan perancangan topologi jaringan secara simulasi menggunakan aplikasi GNS3.
- c. Implementasi merupakan realisasi dari tahap perancangan dan akan dilakukan pengujian pada tahap akhir.
- d. Pengujian konfigurasi otomatis penambahan *gateway* pada *router* menggunakan *metode black box testing* untuk melihat apakah proses konfigurasi otomatis penambahan *gateway* pada *router* berhasil dilakukan pada aplikasi otomatisasi jaringan berbasis web.
- e. Analisis yang dilakukan berupa analisis terhadap konfigurasi otomatis penambahan *gateway* pada *multi virtual router*.

Elin Sylvania Ginting, Copyright ©2020, MIB, Page 1127 Submitted: 08/09/2020; Accepted: 24/09/2020; Published: 20/10/2020

Volume 4, Nomor 4, Oktober 2020, Page 1126-1131
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib

3. HASIL DAN PEMBAHASAN

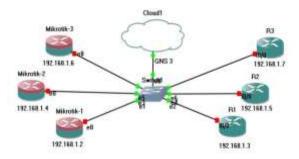
3.1 Perancangan Topologi Jaringan

DOI 10.30865/mib.v4i4.2485

Sebelum melakukan pengujian konfigurasi otomatis penambahan *gateway* pada *router* menggunakan aplikasi otomatisasi jaringan berbasis web, dibutuhkan topologi jaringan untuk menentukan bentuk jaringan yang dibangun dan jumlah perangkat yang akan dilakukan konfigurasi.

Penulis menggunakan aplikasi simulasi GNS3 untuk membuat topologi jaringan. Pada aplikasi simulasi GNS3, ketika membuat topologi jaringan, terdapat *command* pada setiap perangkat jaringan yang akan dibuat. *Command* tersebut berfungsi untuk memastikan apakah jalur yang digunakan benar atau tidak. Hal ini sangat menguntungkan bagi penulis, sebab ketika terdapat perangkat jaringan yang gagal mengirim atau menerima data, simulasi pada topologi jaringan tersebut mati.

Di bawah ini merupakan perancangan topologi jaringan yang telah didesain. Topologi jaringan tersebut dibentuk dengan topologi *star*, yang terdiri dari tiga perangkat *router* dengan merek mikrotik dan tiga perangkat *router* dengan merek cisco yang saling terhubung ke perangkat jaringan *switch* agar berada dalam satu jaringan. *Cloud* berfungsi sebagai *web application* yang akan mengirimkan perintah otomatisasi konfigurasi terhadap seluruh perangkat router di GNS 3.



Gambar 1. Perancangan Topologi Jaringan

Pengalamatan IP *Address* pada jaringan ini hanya ditinjau berdasarkan perangkat *switch* yang terhubung ke setiap perangkat yang berisi nama perangkat, *interface* yang dipakai, dan IP *Adress* yang telah didaftarkan pada aplikasi simulasi GNS 3.

Nama Perangkat Interface Alamat IP Ether 0 192.168.1.1/24 1. Laptop 2. Router Mikrotik 1 Ether 1 192.168.1.2/24 3. Router Mikrotik 2 Ether 2 192.168.1.3/24 4. Router Mikrotik 3 Ether 3 192.168.1.4/24 5. Router Cisco 1 Ether 4 192.168.1.5/24 6. Router Cisco 2 Ether 5 192.168.1.6/24 Router Cisco 3 7. Ether 7 192.168.1.7/24

Tabel 2. Alokasi IP Address pada GNS3

3.2 Tampilan Aplikasi Otomatisasi Jaringan Berbasis Web

Sistem aplikasi otomatisasi jaringan berbasis web dikembangkan menggunakan framework django yang dapat diakses dengan jaringan localhost. Aplikasi ini terdiri dari halaman pengguna dan halaman admin. Pada tampilan halaman admin terdiri dari fitur menu login, add device, dan delete device. Pada tampilan halaman pengguna terdiri dari fitur menu home, menu device list, menu configure, menu verify configure, dan menu log. Akan tetapi, konfigurasi otomatis penambahan gateway pada router dilakukan di halaman pengguna yang terdapat pada aplikasi otomatisasi jaringan berbasis web. Sehingga, penulis hanya menampilkan tampilan aplikasi otomatisasi berbasis web pada halaman pengguna.

Halaman awal website merupakan halaman yang pertama kali muncul ketika admin mengakses alamat web. Halaman ini berisi jumlah perangkat yang terhubung dengan aplikasi GNS3 sesuai dengan merek router, yaitu tiga router mikrotik dan tiga router cisco. Jumlah perangkat tersebut ditampilkan dengan bentuk kotak yang dibedakan menjadi tiga jenis, yaitu total keseluruhan perangkat, jumlah router mikrotik, dan jumlah router cisco. Setelah itu, halaman home juga berisi last event yang memperlihatkan aktifitas yang dilakukan menggunakan web, seperti aktifitas konfigurasi yang dapat dilakukan dengan menggunakan menu configure ataupun aktifitas monitoring yang dapat dilakukan dengan menggunakan menu verify configure. Adapun tampilan halaman awal website akan terlihat seperti gambar berikut.

Elin Sylvania Ginting, Copyright ©2020, MIB, Page 1128 Submitted: **08/09/2020**; Accepted: **24/09/2020**; Published: **20/10/2020**

Volume 4, Nomor 4, Oktober 2020, Page 1126-1131

ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)

Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib

DOI 10.30865/mib.v4i4.2485



| Barrier | Control | Cont

Gambar 2. Tampilan halaman awal aplikasi otomatisasi jaringan berbasis web

3.3 Konfigurasi Penambahan Gateway

Untuk melakukan konfigurasi otomatis penambahan *gateway* pada *router* menggunakan aplikasi otomatisasi jaringan berbasis web, dapat dilakukan dengan melakukan klik pada menu *configure* setelah aplikasi otomatisasi jaringan berbasis web berhasil menampilkan halaman awal atau bisa juga disebut halaman pada fitur menu *home*. Gambar di bawah ini menunjukkan bahwa menu *configure* dapat berfungsi dengan baik menampilkan *choose target* untuk setiap perangkat *router* yang terhubung pada topologi jaringan yang ada di aplikasi GNS3 yang berarti bahwa konfigurasi penambahan *gateway* pada *router* secara otomatis dapat dilakukan.



Gambar 3. Tampilan menu configure pada aplikasi otomatisasi jaringan berbasis web

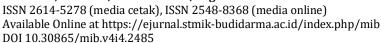
Penambahan *gateway* pada *router* dilakukan di menu *configure* yang kemudian akan muncul tampilan *choose target. Choose target* berarti dapat memilih perangkat apa saja yang ingin dilakukan penambahan *gateway* yang terhubung pada topologi jaringan di GNS3. Dalam hal ini, penulis memilih semua perangkat jaringan yang terdapat pada tampilan menu *configure*, kemudian pada tab **Mikrotik Command** dapat diisi dengan *command* **ip route add dst-address=0.0.0.0/0 gateway=192.168.1.1 ip dns set servers=8.8.8.8 allow-remote-requests=yes.**

Kemudian pada tab **Cisco Command** dapat diisi dengan *command* **ip route 0.0.0.0 0.0.0.0 192.168.1.1 ip name-server 8.8.8.** Lalu, setelah memastikan perangkat apa saja yang ingin diujikan, tekan **Submit**. Tampilan keseluruhan proses penambahan *gateway* juga dapat dilihat pada gambar di bawah ini.



Gambar 4. Penambahan gateway

Volume 4, Nomor 4, Oktober 2020, Page 1126-1131





Kemudian, setelah penulis melakukan klik **Submit**, maka akan muncul tampilan seperti pada gambar di bawah ini. Muncul *last event* yang berarti penambahan *gateway* pada seluruh perangkat jaringan yang terhubung dengan topologi jaringan berjalan sukses, yang berarti bahwa setiap *router* telah terhubung ke internet. Setiap IP *address* pada perangkat jaringan yang terhubung dengan topologi jaringan yang ada pada GNS3 sudah terkonfigurasi dengan baik. Hal ini menunjukkan bahwa pengujian konfigurasi otomatis penambahan *gateway* pada *router* juga berhasil.

160,1661.2	Configure	Serve	Avg.25. 2015 1115 p.m.	No Stror
140.100.13	Contgore	Scow	Aug 25, 2036 1177 p.m.	No Descr
160,160,14	Configure	Scott	Aug 25 2021 1115 a.m.	Also Singar
150,000.1.0	Carrigure	See	Aug 25. 2020. 11.16 p.m.	So tree
785,566,1.8	Configure	Sees	Aug. 25. 2020 11,76 p.m.	No bear
96.965.7	Certain	See	Aug 23. 2020. 217 Nove.	No. Serve

Gambar 5. Hasil penambahan *gateway*

3.4 Pengujian Koneksi

Otomatisasi konfigurasi perangkat *router* terletak pada keseluruhan perangkat yang terhubung dengan topologi jaringan yang ada pada GNS3 dilakukan hanya dengan satu kali klik sehingga penulis tidak lagi melakukan konfigurasi *router* secara manual dimana konfigurasi tersebut harus dilakukan secara satu-persatu dengan menggunakan aplikasi PuTTY agar *router* dapat terhubung ke internet. Keberhasilan pengujian pada penambahan *gateway* juga dapat divalidasi dengan melakukan ping ke ip address 8.8.8.8 atau ke google.com dari salah satu *router* mikrotik seperti yang terdapat pada gambar di bawah ini. Hasil yang didapatkan bahwa ping ke 8.8.8.8 maupun google.com berhasil dilakukan yang berarti bahwa *router* mikrotik sudah terhubung ke internet.

```
| Company | Comp
```

Gambar 6. Hasil tes ping dari salah satu *router* mikrotik

Hal ini juga dilakukan pada *router* cisco untuk melakukan validasi bahwa pengujian konfigurasi otomatis penambahan *gateway* pada *router* cisco berjalan dengan sukses. Validasi tersebut dilakukan dengan melakukan ping 8.8.8.8 dari salah satu *router* cisco untuk membuktikan bahwa *router* cisco yang terhubung ke GNS3 juga telah terhubung ke internet. Hasil yang didapatkan dapat dilihat pada gambar di bawah ini, yang menjelaskan bahwa ketika melakukan ping 8.8.8.8 dari salah satu *router* cisco memiliki *success rate* 100%, yang berarti bahwa penambahan *gateway* pada *router* cisco berhasil dan *router* cisco telah terhubung ke internet.

```
#IMping 0.0.0.0

Type escape sequence to abort.

Sending 5, 180-byte ECMF Echos to 8.0.5.0, timeout is 2 seconds:

1111

Success rate is 180 percent (5/5), round-trip min/avg/max = 32/46/56 ms
#IMPING 0.0.0.0

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 0.0.0, timeout is 2 seconds:

11111

Success rate is 100 percent (5/5), round-trip min/avg/max = 180/191/444 ms
#IMPING 0.0.0
```

Gambar 7. Hasil tes ping dari salah satu *router* cisco

Dengan keberhasilan pada pengujian konfigurasi otomatis ini, diharapkan dapat memudahkan para administrator jaringan dalam melakukan konfigurasi *router* pada jaringan yang lebih kompleks sehingga waktu yang digunakan juga lebih efisien.

Elin Sylvania Ginting, Copyright ©2020, MIB, Page 1130 Submitted: **08/09/2020**; Accepted: **24/09/2020**; Published: **20/10/2020**

Volume 4, Nomor 4, Oktober 2020, Page 1126-1131

ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online) Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib DOI 10.30865/mib.v4i4.2485



3.5 Pengujian Fungsional

Kemudian, dilakukan pula pengujian pada halaman pengguna dengan menggunakan metode *black box testing* dengan tujuan untuk mengetahui apakah fungsionalitas tiap fitur pada aplikasi otomatisasi jaringan berbasis web ini dapat digunakan dengan baik atau tidak. Fitur yang terdapat pada halaman pengguna lebih banyak dibandingkan fitur yang terdapat pada halaman admin. Akan tetapi, penulis hanya menggunakan dua fitur menu, yaitu fitur menu *home* dan fitur menu *configure*, sehingga hasil pengujian dapat dilihat pada tabel di bawah ini.

Tabel 3. Pengujian Fitur pada Halaman Pengguna

Fitur	Skenario		
Home	Dapat melihat jumlah perangkat yang terhubung berdasarkan total maupun berdasarkan merek <i>router</i> yang tersedia.		
	Dapat melihat <i>last event</i> pada aktivitas yang terjadi dalam kurun waktu dekat.		
Configure	Pada bagian <i>choose target</i> , dapat diklik. Pada bagian <i>mikrotik command</i> dan <i>cisco command</i> dapat diisi. Pada bagian <i>submit</i> dapat diklik.		
	Dapat melakukan konfigurasi penambahan <i>gateway</i> terhadap perangkat jaringan yang tersedia secara keseluruhan maupun satu persatu.	Berhasil.	

Seperti yang terdapat pada tabel di atas, pengujian fungsional menggunakan metode *black box testing* mendapatkan hasil yang sempurna, yang berarti bahwa pada aplikasi otomatisasi jaringan berbasis web memiliki fungsional yang sesuai dengan fungsi fitur menu masing-masing dan tidak terjadi kesalahan atau *error*.

4. KESIMPULAN

Dari hasil penelitian yang telah dilakukan, diperoleh kesimpulan:

- Aplikasi otomatisasi jaringan berbasis web ini dapat membantu administrator jaringan dalam melakukan konfigurasi penambahan gateway otomatis pada sebuah topologi jaringan yang terhubung dengan aplikasi GNS3.
- 2. Konfigurasi otomatis penambahan *gateway* pada *virtual router* menggunakan aplikasi otomatisasi jaringan berhasil dilakukan yang dibuktikan dengan pengujian koneksi.
- 3. Aplikasi otomatisasi jaringan berbasis web ini dapat berfungsi dengan baik yang dibuktikan dengan melakukan pengujian metode *black-box testing*.

REFERENCES

- [1] J. Forcier, "Welcome to Paramiko!," 2020. [Online]. Available: http://www.paramiko.org/.
- [2] 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.
- [3] A. R. Komarudin, Otomatisasi Administrasi Jaringan Dengan Script Python. Jakarta: Jasakom, 2018.
- [4] I. M. B. Swastika and I. G. O. G. Atitama, "Otomatisasi Konfigurasi Mikrotik Router Menggunakan Software Ansible," *SNATIA (Seminar Nas. Teknol. Inf. Apl.*, pp. 495–502, 2017.
- [5] P. Mihăilă, T. Bălan, R. Curpen, and F. Sandu, "Network Automation and Abstraction using Python Programming Methods," *MACRo 2015*, vol. 2, no. 1, pp. 95–103, 2017.
- [6] A. N. Rahimah, D. S. Rusdianto, and M. T. Ananta, "Pengembangan Sistem Pengelolaan Ruang Baca Berbasis Web Dengan Menggunakan Django Framework (Studi Kasus: Ruang Baca Fakultas Ilmu Komputer Universitas Brawijaya)," vol. 3, no. 5, pp. 4439–4446, 2019.
- [7] A. A. Arwaz, T. Kusumawijaya, R. Putra, K. Putra, and A. Saifudin, "Pengujian Black Box pada Aplikasi Sistem Seleksi Pemenang Tender Menggunakan Teknik Equivalence Partitions," *J. Teknol. Sist. Inf. dan Apl.*, vol. 2, no. 4, p. 130, 2019.
- [8] J. Arifin, "Perancangan Jaringan LAN dan WLAN Berbasis Mikrotik Pada Sekolah Menengah Kejuruan," J. Sisfokom (Sistem Inf. dan Komputer), vol. 2, no. 2, pp. 17–22, 2013.
- [9] E. D. Fariliana, "Analisis Penggunaan IP Publik Pada Broadband Network," Simetris J. Tek. Mesin, Elektro dan Ilmu Komput., vol. 10, no. 1, pp. 341–356, 2019.
- [10] M. F. Asnawi, "APLIKASI KONFIGURASI MIKROTIK SEBAGAI MANAJEMEN BANDWIDTH DAN INTERNET GATEWAY BERBASIS WEB," Penelit. dan Pengabdi. Kpd. Masy. UNSIQ, vol. 5, no. 1, pp. 42–48, 2018.
- [11] L. Galaxy Technologies, "GNS3: The software that empowers network professionals," 2020. [Online]. Available: https://gns3.com/.

Elin Sylvania Ginting, Copyright ©2020, MIB, Page 1131 Submitted: 08/09/2020; Accepted: 24/09/2020; Published: 20/10/2020