

## Implementasi *Network Automation* Untuk Konfigurasi Jaringan Baru Dengan Netmiko

Donny Rahardika<sup>1</sup>, Niki Ratama<sup>2</sup>

<sup>1,2</sup>Teknik Informatika, Universitas Pamulang, Jl. Surya Kencana No.1, Indonesia, 154117

e-mail: <sup>1</sup>[donnyrahardika@gmail.com](mailto:donnyrahardika@gmail.com), <sup>2</sup>[nickyratama@gmail.com](mailto:nickyratama@gmail.com)

### Abstract

*PT. Maxindo Mitra Solusi is a company engaged in telecommunications services (internet). Every customer who wants to enjoy internet service, need to do network device installation first. Each network device needs to be configured to be able to connect to the internet. Configuration of network devices is still done conventionally, where devices are configured one by one which requires more time and effort, especially when there is a risk of mistakes that are usually made by humans. In the development of this network automation system using the Network Development Life Cycle (NDLC) method. This network automation system is built using the Python programming language and uses the help of the Netmiko library. Network automation can help network administrators make the installation of new networks easier and faster. network administrator just need to run this system then every device will be configured automatically.*

*Keywords: Internet, NDLC, Network, Automation, Configuration, Netmiko, Python.*

### Abstrak

PT. Maxindo Mitra Solusi adalah sebuah perusahaan bergerak dibidang jasa telekomunikasi (internet). Setiap *customer* yang ingin menikmati layanan internet, akan dilakukan pemasangan perangkat jaringan terlebih dahulu. Setiap perangkat jaringan perlu dilakukan konfigurasi agar dapat terhubung dengan internet. Konfigurasi perangkat jaringan masih dilakukan secara konvensional, dimana perangkat dilakukan konfigurasi satu persatu yang membutuhkan waktu dan usaha lebih terlebih terdapat risiko kekeliruan yang biasa dilakukan manusia. Dalam pengembangan sistem *network automation* ini menggunakan metode *Network Development Life Cycle (NDLC)*. Sistem *network automation* ini dibangun menggunakan bahasa pemrograman Python dan menggunakan bantuan *library* Netmiko. *Network automation* dapat membantu *network administrator* dalam melakukan pemasangan jaringan baru menjadi lebih mudah dan cepat. *network administrator* hanya perlu menjalankan sistem ini lalu setiap perangkat akan terkonfigurasi secara otomatis.

Kata Kunci: Internet, NDLC, Network, Automation, Konfigurasi, Netmiko, Python.

### 1. PENDAHULUAN

Internet adalah adalah singkatan dari *international network*, yang didefinisikan juga sebagai suatu jaringan komputer yang sangat besar, dimana jaringan komputer tersebut terdiri dari beberapa jaringan-jaringan kecil yang saling terhubung satu sama lain.

*Internet Service Provider (ISP)* merupakan perusahaan yang menyediakan layanan internet untuk di distribusikan kepada pengguna. Saat ini konfigurasi perangkat jaringan baru yang dilakukan oleh *network administrator* dilakukan dengan cara manual, yaitu dengan melakukan konfigurasi terhadap masing-masing perangkat.

Dengan latar belakang ini maka dibuatlah *network automation* dengan Netmiko menggunakan topologi *star* sebagai solusi agar dapat membantu menyelesaikan permasalahan yang ada di PT. Maxindo Mitra Solusi. Netmiko adalah *library* Python yang berfungsi sebagai alat bantu penghubung antara komputer dengan perangkat jaringan [1].

Tujuan dari penelitian ini adalah melakukan “Implementasi *Network Automation* Untuk Konfigurasi Jaringan Baru Dengan Netmiko” sebagai solusi dalam melakukan konfigurasi perangkat jaringan baru secara otomatis dengan harapan dapat menghemat waktu dan sumber daya serta mengurangi faktor kesalahan atau kekeliruan yang bisa terjadi bila dilakukan secara manual.

## 2. PENELITIAN YANG TERKAIT

Berikut ini adalah beberapa artikel jurnal yang berhubungan dengan materi penelitian:

- a. Penelitian yang dilakukan oleh Paul MIHAILĂ, Titus BĂLAN, Radu CURPEN, Florin SANDU (Transilvania University 2017) yang berjudul “*Network Automation and Abstraction using Python Programming Methods*” membahas tentang perancangan sistem *network automation* dengan *Software-Define Network* (SDN) menggunakan metode SDLC (*System Development Life Cycle*) dengan tujuan menghasilkan sistem otomatis dalam melakukan konfigurasi perangkat jaringan sehingga *network engineers* tidak perlu mengkonfigurasi satu-persatu perangkat serta memberikan kemudahan dalam melakukan penerapan konfigurasi baru secara otomatis[2].
- b. Penelitian yang dilakukan oleh Kukuh Nugroho, Anggi Dzikri Abrariansyah, Syariful Ikhwan (Institut Teknologi Telkom Purwokerto 2020) yang berjudul “Perbandingan Kinerja *Library* Paramiko dan Netmiko Dalam Proses Otomasi Jaringan” membahas tentang perbedaan kinerja antara *library* Paramiko dan Netmiko dalam membuat sistem otomatis jaringan menggunakan metode analisis sistem dengan tujuan mengetahui perbedaan performansi dari *library* Paramiko dan Netmiko dalam melakukan konfigurasi perangkat jaringan[1].
- c. Penelitian yang dilakukan oleh Muhammad Fauzi Islami, Purnawarman Musa and Missa Lamsani (Gunadarma

University 2020) dengan judul “*Implementation of Network Automation Using Ansible to Configure Routing Protocol in Cisco and Mikrotik Router with Raspberry PI*” membahas tentang implementasi *network automation* dengan Ansible menggunakan metode *waterfall* dengan tujuan menghasilkan sistem yang dapat melakukan konfigurasi perangkat jaringan secara otomatis sehingga mengurangi usaha dan kerumitan dalam mengatur perangkat jaringan[3].

- d. Penelitian yang dilakukan oleh Rheza Adhyatmaka Wiryawan dan Nur Rohman Rosyid (Universitas Gajah Mada 2019) dengan judul “Pengembangan Aplikasi Otomatisasi Administrasi Jaringan Berbasis Website Menggunakan Bahasa Pemrograman Python” membahas tentang perancangan aplikasi administrasi jaringan secara otomatis menggunakan metode *Rapid Application Development* (RAD) dengan tujuan menghasilkan sistem berbasis web yang dapat digunakan sebagai administrasi jaringan dengan fitur konfigurasi *routing*, *vlan*, *backup*, *restore*, dan *setting*. Dimana pada fitur-fitur tersebut dapat dilakukan fungsi utama aplikasi dalam melakukan konfigurasi administrasi jaringan berupa *routing static*, *dynamic OSPF*, *RIPv1*, *RIPv2*, *BGP*, *backup* dan *restore* konfigurasi[4].
- e. Penelitian yang dilakukan oleh Elin Sylvania Ginting, Suroso dan Irawan Hadi (Politeknik Negeri Sriwijaya 2020) dengan judul “Pengujian Konfigurasi Otomatis Penambahan *Gateway* Pada *Virtual Router* Menggunakan Aplikasi Otomatisasi Jaringan Berbasis Web” membahas tentang penerapan aplikasi otomatisasi jaringan menggunakan metode RAD (*Rapid Application Development*) dengan tujuan menghasilkan sistem berbasis web yang dapat melakukan konfigurasi penambahan *gateway* secara otomatis terhadap perangkat jaringan keseluruhan maupun satu persatu[5].
- f. Penelitian yang dilakukan oleh Danish Rafique dan Luis Velasco (*Optical Society of America* 2018) dengan judul “*Machine Learning for Network Automation: Overview, Architecture, and Applications*” membahas tentang optimalisasi jaringan telekomunikasi berbasis *optic* dalam

memenuhi segala kebutuhan layanan dengan tujuan mempermudah dalam operasional skala besar dan optimalisasi keterbatasan dalam hal skalabilitas dan efisiensi[6].

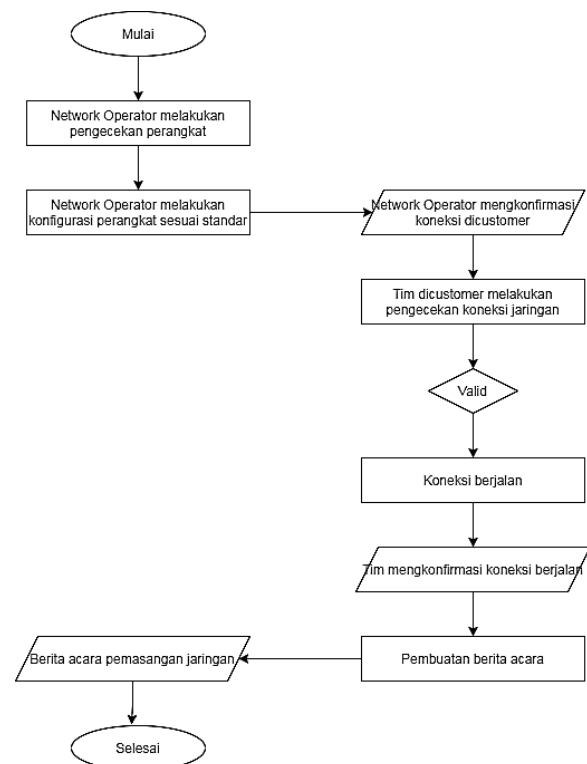
- g. Penelitian yang dilakukan oleh Mr. Shubham Yadav dan Brijesh Kumar Dubey (*International Journal for Research in Applied Science & Engineering Technology (IJRASET)* 2019) dengan judul “*Research Paper on Network Automation*” membahas tentang perancangan Smart Grid yaitu sebuah sistem jaringan yang sepenuhnya otomatis dengan tujuan untuk menyelesaikan sambungan jaringan energi terdistribusi, sambil membangun sistem tenaga listrik yang lebih aman, andal, ramah lingkungan, dan ekonomis, banyak negara mengajukan rencana pengembangan otomatisasi jaringan[7].
- h. Penelitian yang dilakukan oleh Adian Fatchur Rochim et al (Universitas Muhammadiyah Malang 2020) dengan judul “*As-RaD System as a Design Model of the Network Automation Configuration System Based on the REST API and Django Framework*” membahas tentang perancangan sistem manajemen jaringan otomatis menggunakan metode RAD (*Rapid Application Development*) dengan tujuan menghasilkan aplikasi berbasis web yang dapat digunakan untuk melakukan konfigurasi perangkat jaringan dan memiliki performa lebih baik daripada *library* Paramiko dan NAPALM[8].
- i. Penelitian yang dilakukan oleh Yuli Sun Hariyani et al (Universitas Telkom 2016) dengan judul “*Routing Implementation Based-on Software Defined Network Using RYU Controller And OpenVSwitch*” membahas tentang perancangan sistem kontrol jaringan berdasarkan *Software-Define Network (SDN)* menggunakan metode *waterfall* dengan tujuan membangun sistem kontrol fungsi untuk manajemen jaringan berbasis SDN dengan biaya yang terjangkau dengan efisiensi menggunakan RYU Controller dan OpenVSwitch[9].
- j. Penelitian yang dilakukan oleh Santyadiputra et al (Universitas Pendidikan Ganesha 2021) dengan judul “*The effectiveness of Automatic Network Administration (ANA) in network automation simulation at Universitas Pendidikan Ganesha*” membahas tentang perancangan sistem otomatisasi untuk

pengaturan dan pencadangan konfigurasi perangkat jaringan menggunakan metode RAD (*Rapid Application Development*) dengan tujuan membangun sistem yang dapat melakukan konfigurasi perangkat jaringan secara otomatis dengan mudah menggunakan aplikasi berbasis web[10].

### 3. METODE PENELITIAN

Penelitian dilakukan pada salah satu perusahaan ISP di wilayah Jakarta Barat. Dalam perancangan sistem ini penelitian menggunakan metode pengembangan *Network Development Life Cycle (NDLC)* yang meliputi 6 tahap yaitu analisis, desain, simulasi, implementasi, pemantauan, dan manajemen.

Tahapan dalam melakukan pemasangan jaringan baru berawal dari pemasangan perangkat radio point-to-point dari sisi Point of Presence ke arah tempat customer. Untuk alur pemasangannya dapat dilihat pada Gambar 1.



Gambar 1. Flow Pemasangan Jaringan Baru

- 1. Studi Literatur
  - a. Studi Pustaka
    - Penulis mencari informasi mengenai topik yang sedang dibahas dari jurnal atau literatur terkait [11].
  - b. Observasi
    - Pengamatan langsung terhadap cara konfigurasi dari perangkat jaringan yang

digunakan, untuk mendapatkan data dan informasi yang berkaitan dengan permasalahan yang ada pada salah satu ISP di Jakarta Barat.

c. Wawancara

Kegiatan wawancara dilakukan kepada bapak Alim Bachtiar yang menjabat sebagai *Manager Business Support* di salah satu ISP di Jakarta Barat. Informasi yang didapat berkaitan dengan metode konfigurasi perangkat jaringan, *hardware* yang diperlukan, *system* dan infrastruktur jaringan.

2. Analisis Masalah

Tahapan ini dilakukan dengan memantau cara konfigurasi perangkat jaringan masih menggunakan cara manual, oleh karena itu dibuatlah sistem *network automation* dengan Netmiko untuk membantu *network operator* dalam melakukan pemasangan jaringan baru [12].

3. Perancangan Sistem

Pembuatan sistem memerlukan perancangan untuk mengetahui perangkat keras dan skema jaringan yang dibutuhkan untuk membangun sistem *network automation* secara lengkap .

4. Pengerjaan Sistem (Implementasi)

Penjelasan pengerjaan sistem yang dilakukan adalah untuk membuat suatu sistem otomatis pada pemasangan jaringan baru dengan cara yang cepat dan efisien.

5. Pengujian (*Testing*)

Pada tahap ini di lakukan proses pengujian sistem guna mengetahui apakah sistem bekerja secara baik, jika bermasalah maka sistem perlu dilakukan perancangan sistem ulang untuk memperbaiki permasalahan yang terjadi [13].

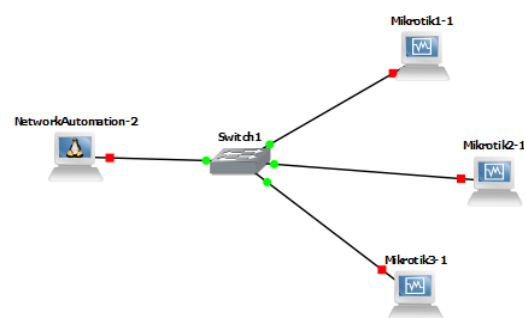
a. Tampilan Konfigurasi IP *Static Server*

## 4. HASIL DAN PEMBAHASAN

Penyusunan prosedur dalam pembuatan sistem *network automation* dibagi menjadi 2 tahap, yaitu persiapan skema jaringan dan persiapan sisten *network automation*.

### 4.1 Topologi Jaringan

Berdasarkan analisis dan perancangan sistem menghasilkan skema jaringan yang menggunakan topologi *star*. Topologi ini dipilih karena bentuk topologi jaringan yang berupa konvergensi dari *node* tengah ke setiap *node* atau pengguna.



Gambar 2 Topologi *Star*

Pada Gambar 2 dijelaskan satu buah *switch* menghubungkan antara perangkat sistem *network automation* dengan perangkat *router* yang akan dilakukan konfigurasi secara otomatis.

### 4.2 Implementasi Sistem *Network Automation*

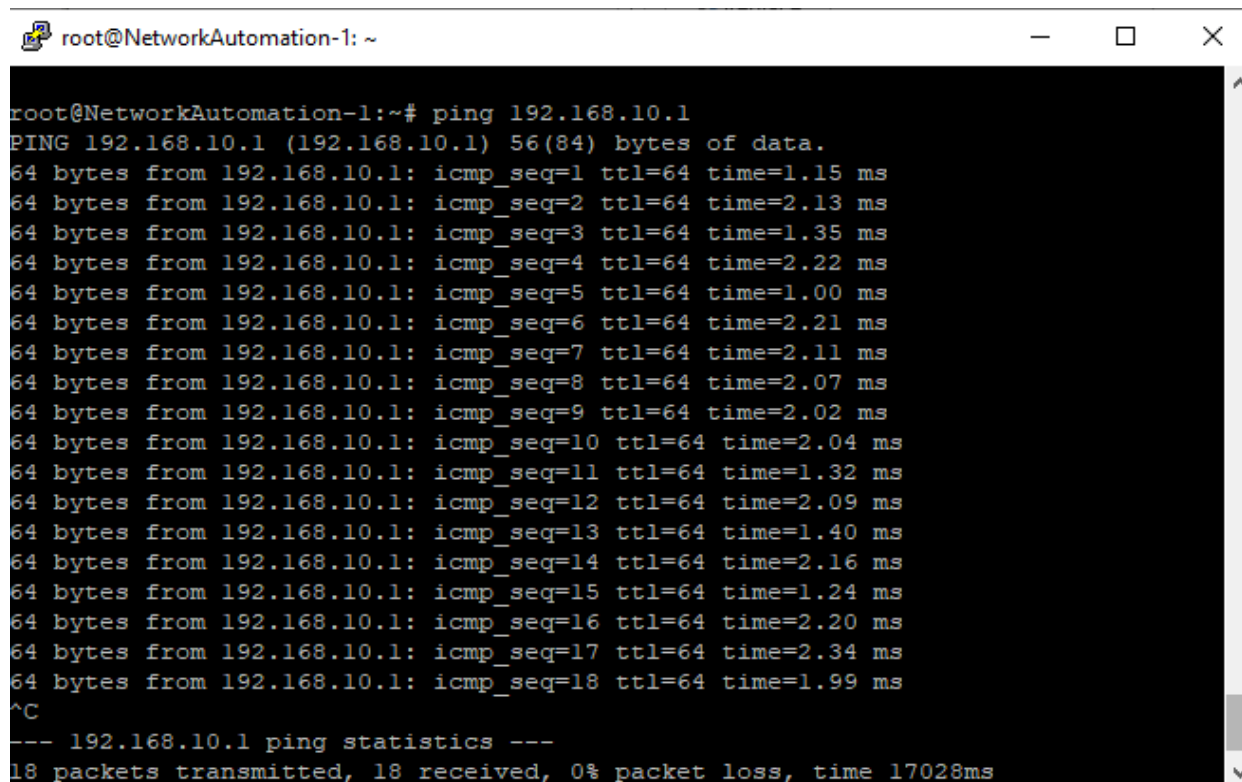
Sistem *network automation* dibuat menggunakan bahasa pemrograman Python dengan bantuan *library* Netmiko.

Berikut ini adalah beberapa tampilan dalam mempersiapkan sistem *network automation* menggunakan sistem operasi linux ubuntu sebagai berikut:



Pada tampilan diatas, perangkat *router* Mikrotik menggunakan IP *static* supaya dapat membuat komunikasi antara sistem *network automation* dengan *router*. Untuk merubah konfigurasi IP *static* dapat menggunakan perintah berikut “ip address add address=(IP address) interface=(interface network card)”.

c. Tampilan Pengujian Koneksi Antara Sistem Dengan Perangkat *Router*

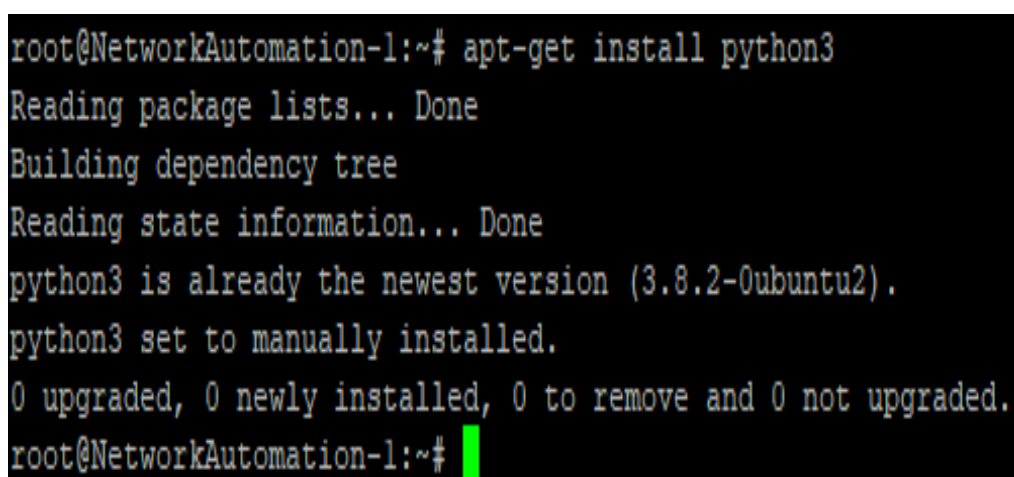


```
root@NetworkAutomation-1: ~  
root@NetworkAutomation-1:~# ping 192.168.10.1  
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.  
64 bytes from 192.168.10.1: icmp_seq=1 ttl=64 time=1.15 ms  
64 bytes from 192.168.10.1: icmp_seq=2 ttl=64 time=2.13 ms  
64 bytes from 192.168.10.1: icmp_seq=3 ttl=64 time=1.35 ms  
64 bytes from 192.168.10.1: icmp_seq=4 ttl=64 time=2.22 ms  
64 bytes from 192.168.10.1: icmp_seq=5 ttl=64 time=1.00 ms  
64 bytes from 192.168.10.1: icmp_seq=6 ttl=64 time=2.21 ms  
64 bytes from 192.168.10.1: icmp_seq=7 ttl=64 time=2.11 ms  
64 bytes from 192.168.10.1: icmp_seq=8 ttl=64 time=2.07 ms  
64 bytes from 192.168.10.1: icmp_seq=9 ttl=64 time=2.02 ms  
64 bytes from 192.168.10.1: icmp_seq=10 ttl=64 time=2.04 ms  
64 bytes from 192.168.10.1: icmp_seq=11 ttl=64 time=1.32 ms  
64 bytes from 192.168.10.1: icmp_seq=12 ttl=64 time=2.09 ms  
64 bytes from 192.168.10.1: icmp_seq=13 ttl=64 time=1.40 ms  
64 bytes from 192.168.10.1: icmp_seq=14 ttl=64 time=2.16 ms  
64 bytes from 192.168.10.1: icmp_seq=15 ttl=64 time=1.24 ms  
64 bytes from 192.168.10.1: icmp_seq=16 ttl=64 time=2.20 ms  
64 bytes from 192.168.10.1: icmp_seq=17 ttl=64 time=2.34 ms  
64 bytes from 192.168.10.1: icmp_seq=18 ttl=64 time=1.99 ms  
^C  
--- 192.168.10.1 ping statistics ---  
18 packets transmitted, 18 received, 0% packet loss, time 17028ms
```

Gambar 5. Tampilan Pengujian Koneksi Antara Sistem dan *Router*

Tampilan diatas, menunjukkan koneksi antara sistem *network automation* dengan perangkat *router* berjalan dengan baik.

d. Instalasi Python Pada Sistem



```
root@NetworkAutomation-1:~# apt-get install python3  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
python3 is already the newest version (3.8.2-0ubuntu2).  
python3 set to manually installed.  
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.  
root@NetworkAutomation-1:~#
```

Gambar 6. Instalasi Python Pada Sistem

Gambar diatas adalah langkah dalam melakukan instalasi bahasa pemrograman Python. Instalasi Python dapat menggunakan perintah “sudo apt-get install python3”.



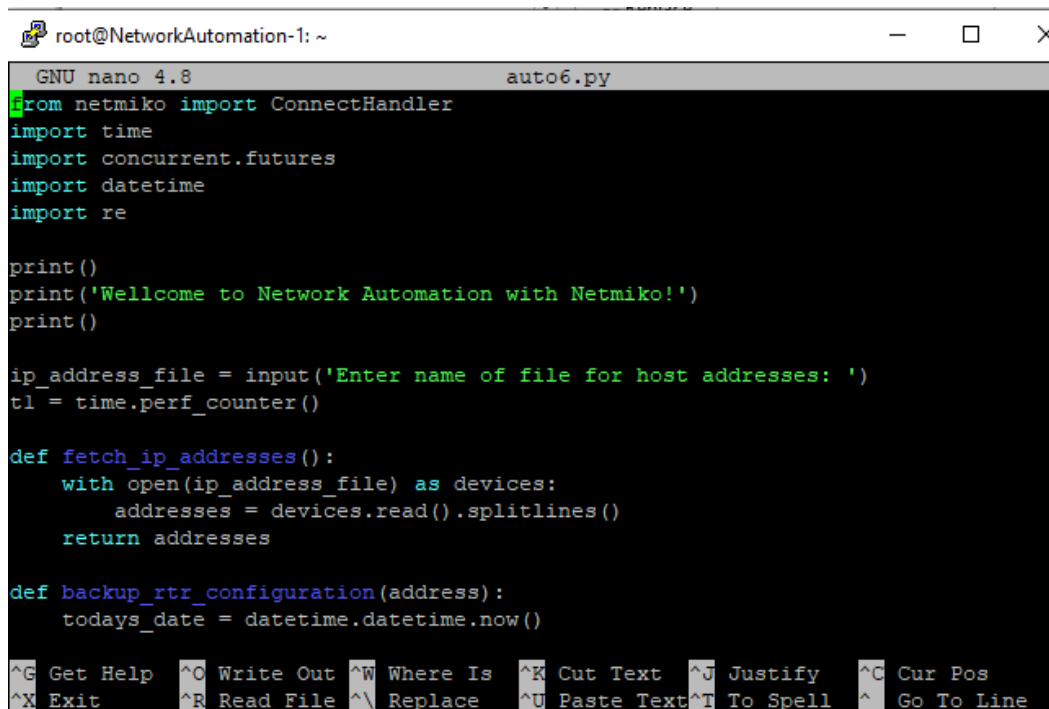
e. Instalasi *Library* Netmiko

```
root@NetworkAutomation-1:~# pip3 install netmiko==2.4.2
Requirement already satisfied: netmiko==2.4.2 in /usr/lib/python3/dist-packages
(2.4.2)
root@NetworkAutomation-1:~# pip3 show netmiko
Name: netmiko
Version: 2.4.2
Summary: Multi-vendor library to simplify Paramiko SSH connections to network de
vices
Home-page: https://github.com/ktbyers/netmiko
Author: Kirk Byers
Author-email: ktbyers@twb-tech.com
License: MIT
Location: /usr/lib/python3/dist-packages
Requires:
Required-by: pyntc, napalm
root@NetworkAutomation-1:~#
```

Gambar 7. Instalasi *Library* Netmiko

Gambar diatas merupakan langkah dalam instalasi *library* Netmiko pada sistem. Netmiko ini sangat berpengaruh penting karena dapat memudahkan pengguna dalam membuat koneksi dengan perangkat jaringan. Instalasi dapat dilakukan dengan perintah “pip3 install netmiko==2.4.2”.

f. Pembuatan Sistem *Network Automation*



```
root@NetworkAutomation-1: ~
GNU nano 4.8 auto6.py
from netmiko import ConnectHandler
import time
import concurrent.futures
import datetime
import re

print()
print('Wellcome to Network Automation with Netmiko!')
print()

ip_address_file = input('Enter name of file for host addresses: ')
t1 = time.perf_counter()

def fetch_ip_addresses():
    with open(ip_address_file) as devices:
        addresses = devices.read().splitlines()
    return addresses

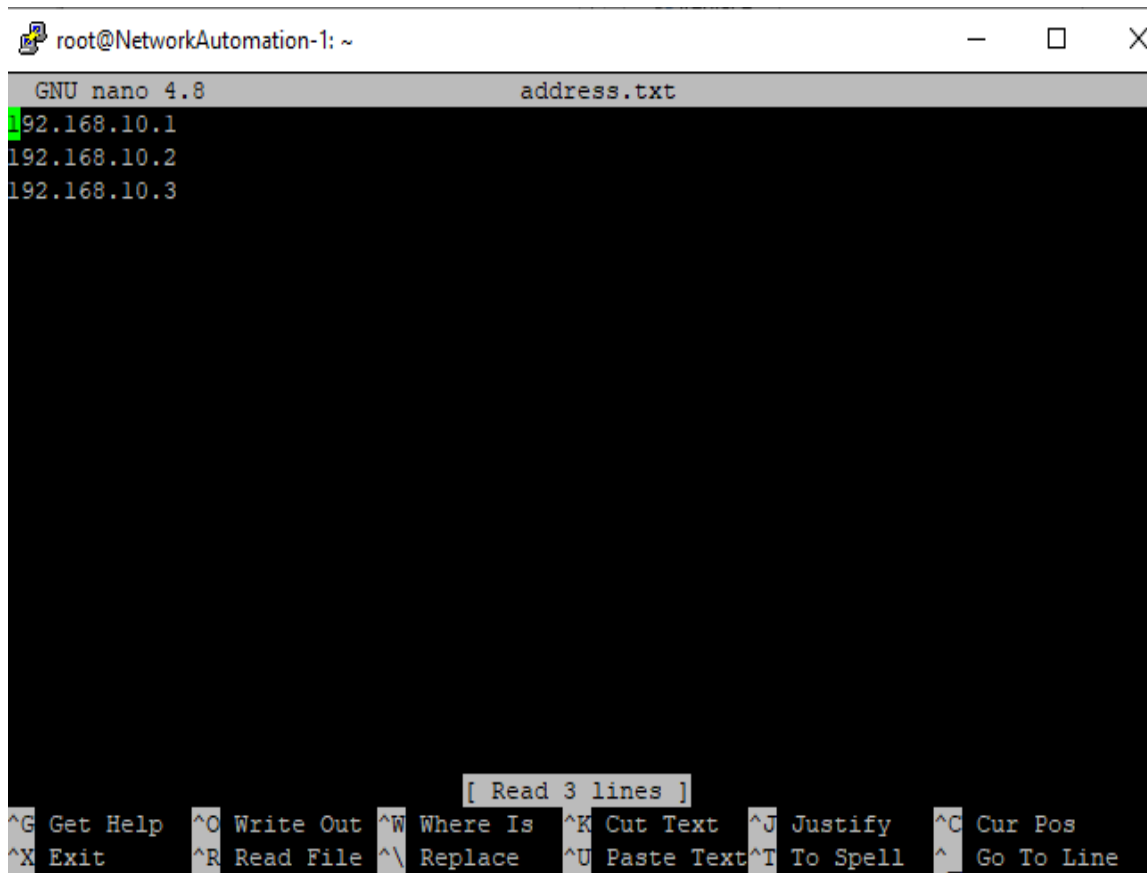
def backup_rtr_configuration(address):
    todays_date = datetime.datetime.now()

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line
```

Gambar 8. Pembuatan Sistem *Network Automation*

Gambar diatas menampilkan program *network automation* yang menggunakan bahasa Python dengan bantuan *library* Netmiko.

g. Tampilan Daftar IP Address File



```
root@NetworkAutomation-1: ~
GNU nano 4.8 address.txt
192.168.10.1
192.168.10.2
192.168.10.3
[ Read 3 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Paste Text ^T To Spell  ^_ Go To Line
```

Gambar 9. Tampilan Daftar IP Router

Gambar diatas adalah daftar dari IP *address* perangkat jaringan yang akan dilakukan konfigurasi secara otomatis. Dengan perintah “nano address.text”.

h. Tampilan File Konfigurasi Mikrotik

```
#Config Default Account
user set 0 password=password comment="Default account - BACKUP ONLY"
user add name=noc pass=myspassword group=full comment="Account NOC"

#Config Banner
system note set show-at-login=yes
system note set note="Networks Automation - Authorized administrators only. Access
to this device is monitored."

#Config LAN
interface bridge add name=LAN
#interface bridge port
#add bridge=bridge1 interface=ether2
#add bridge=bridge1 interface=ether3
#add bridge=bridge1 interface=ether4
ip address add address 192.168.100.1/24 int LAN
ip firewall nat add chain=srcnat src-address=192.168.100.0/24 action=masquerade

#Config DNS
ip dns set servers=8.8.8.8,1.1.1.1
```

Gambar 10. Tampilan File Konfigurasi Mikrotik



Gambar diatas merupakan file konfigurasi perangkat yang nantinya akan diterapkan di setiap perangkat yang terhubung secara otomatis. Dengan perintah “nano conf.text”.

i. Proses Menjalankan Sistem *Network Automation*

```
root@NetworkAutomation-1:~# python3 auto6.py

Wellcome to Network Automation with Netmiko!

Enter name of file for host addresses: address.txt
Connecting to host 192.168.10.1...
Connecting to host 192.168.10.2...
Connecting to host 192.168.10.3...
SSH connection established to 192.168.10.3:22
Interactive SSH session established
Doing configuration for host 192.168.10.3...
SSH connection established to 192.168.10.1:22
Interactive SSH session established
Doing configuration for host 192.168.10.1...
Generating backup configuration for host 192.168.10.3...
Generating backup configuration for host 192.168.10.1...

The script finished executing in 7.77 seconds.

root@NetworkAutomation-1:~#
```

Gambar 11. Proses Menjalankan Sistem *Network Automation*

Gambar diatas merupakan tampilan dalam proses dalam menjalankan sistem *network automation*. *Network administrator* diminta untuk memasukkan nama file yang berisi data IP *address* perangkat. Proses eksekusi menggunakan perintah “python3 auto6.py”.

j. Tampilan Perubahan Konfigurasi Setelah Menjalankan *Network Automation*

```
Flags: X - disabled, I - invalid, D - dynamic
# ADDRESS NETWORK INTERFACE
0 192.168.10.1/24 192.168.10.0 ether1-r1
1 192.168.100.1/24 192.168.100.0 LAN
[admin@Router_1] > int pr
Flags: D - dynamic, X - disabled, R - running, S - slave
# NAME TYPE ACTUAL-MTU L2MTU
0 R ether1-r1 ether 1500
1 R LAN bridge 1500 65535
[admin@Router_1] > ip ro pr
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
# DST-ADDRESS PREF-SRC GATEWAY DISTANCE
0 A S 0.0.0.0/0 192.168.10.254 1
1 S 0.0.0.0/0 192.168.10.254 1
2 S 0.0.0.0/0 192.168.10.254 1
3 S 0.0.0.0/0 192.168.10.254 1
4 S 0.0.0.0/0 192.168.10.254 1
5 S 0.0.0.0/0 192.168.10.254 1
6 S 0.0.0.0/0 192.168.10.254 1
7 S 0.0.0.0/0 192.168.10.254 1
8 ADC 192.168.10.0/24 192.168.10.1 ether1-r1 0
9 ADC 192.168.100.0/24 192.168.100.1 LAN 0
[admin@Router_1] >
```

Gambar 12. Tampilan Perubahan Konfigurasi Pada *Router*

Gambar diatas memperlihatkan perubahan yang sudah selesai menjalankan sistem *network automation* sesuai dengan konfigurasi file yang sudah disiapkan sebelumnya.

## 5. KESIMPULAN

Dari hasil studi pustaka, merancang sistem *network automation*, pengimplementasian dan pengujian sistem otomatis diperoleh poin – poin untuk menjawab masalah – masalah yang ada pada pemasangan jaringan baru, berikut ini adalah kesimpulan dari penelitian yang peneliti lakukan:

- a. Untuk meningkatkan produktivitas *network operator center*, peneliti merancang dan membangun sebuah sistem *network automation* berbasis *command-line* dengan metode pengembangan sistem *Network Development Life Cycle*. Sistem ini dapat membantu *network operator center* dalam melakukan konfigurasi perangkat jaringan baru sekaligus hanya dengan beberapa langkah mudah.
- b. Kekeliruan yang terjadi bila melakukan konfigurasi perangkat jaringan secara manual dapat dikurangi, setelah ada sistem *network automation* ini konfigurasi perangkat berjalan sesuai kehendak.

Melihat banyaknya kemungkinan untuk menambahkan fitur – fitur yang dapat ditambahkan pada sistem ini, saran peneliti untuk peneliti berikutnya adalah sebagai berikut:

- a. Sistem *network automation* ini masih menggunakan tampilan *command-line* based dimana dari segi tampilan masih bisa ditingkatkan lagi yaitu dengan menggunakan *graphical (web-based)* yang mungkin lebih memudahkan bagi beberapa orang.
- b. Sistem *network automation* ini juga masih bisa ditambahkan sebuah fitur untuk melakukan pengecekan dari perangkat jaringan yang sudah terpasang. Sehingga memungkinkan sistem ini untuk dapat menjadi sebuah sistem *monitoring* sekaligus dapat digunakan untuk konfigurasi perangkat jaringan baru.

## UCAPAN TERIMAKASIH

Ucapan terima kasih penulis kepada pihak yang membantu ataupun memberikan dukungan

terkait dengan penelitian yang dilakukan seperti bantuan fasilitas penelitian, dana hibah, dan lainnya.

## DAFTAR PUSTAKA

- [1] K. Nugroho, A. D. Abrariansyah, and S. Ikhwan, “Perbandingan Kinerja Library Paramiko dan Netmiko Dalam Proses Otomasi Jaringan,” vol. 1, 2020.
- [2] P. Mihäilä, T. Bälän, R. Curpen, and F. Sandu, “Network Automation and Abstraction using Python Programming Methods,” *MACRo 2015*, vol. 2, no. 1, pp. 95–103, 2017, doi: 10.1515/macro-2017-0011.
- [3] M. F. Islami, P. Musa, and M. Lamsani, “Implementation of Network Automation using Ansible to Configure Routing Protocol in Cisco and Mikrotik Router with Raspberry PI,” *J. Ilm. Komputasi*, vol. 19, no. 2, pp. 127–134, 2020, doi: 10.32409/jikstik.19.2.80.
- [4] 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.
- [5] 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, doi: 10.30865/mib.v4i4.2485.
- [6] D. Rafique and L. Velasco, “Machine Learning for Network Automation: Overview, Architecture, and Applications [Invited Tutorial],” *J. Opt. Commun. Netw.*, vol. 10, no. 10, p. D126, Oct. 2018, doi: 10.1364/JOCN.10.00D126.
- [7] S. Yadav, “Research Paper on Network Automation,” *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 7, pp. 1446–1450, Apr. 2019, doi: 10.22214/ijrasnet.2019.4261.
- [8] A. F. Rochim, A. Rafi, A. Fauzi, and K. T. Martono, “As-RaD System as a Design Model of the Network Automation Configuration System Based on the REST-API and Django Framework,” *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 4, pp. 291–298, 2020, doi:

- 10.22219/kinetik.v5i4.1093.
- [9] Y. S. Hariyani, I. D. Irawati, S. Danu Dwi, and M. Nuruzzamanirridha, "Routing implementation based-on software defined network using Ryu controller and openvswitch," *J. Teknol.*, vol. 78, no. 5, pp. 295–298, 2016, doi: 10.11113/jt.v78.8315.
- [10] G. S. Santyadiputra, I. M. E. Listartha, and G. A. J. Saskara, "The effectiveness of Automatic Network Administration (ANA) in network automation simulation at Universitas Pendidikan Ganesha," *J. Phys. Conf. Ser.*, vol. 1810, no. 1, p. 012028, 2021, doi: 10.1088/1742-6596/1810/1/012028.
- [11] N. Ratama, "Analisa Dan Perbandingan Sistem Aplikasi Diagnosa Penyakit Asma Dengan Algoritma Certainty Factor Dan Algoritma Decision Tree Berbasis Android," *J. Inform. J. Pengemb. IT*, vol. 3, no. 2, pp. 177–183, 2018, doi: 10.30591/jpit.v3i2.848.
- [12] N. Ratama and Munawaroh, "Perancangan Sistem Informasi Sosial Learning untuk Mendukung Pembangunan Kota Tangerang dalam Meningkatkan Smart city Berbasis Android," *SATIN – Sains dan Teknol. Inf.*, vol. 5, no. 2, pp. 59–67, 2019.
- [13] Munawaroh, "Penerapan Metode Fuzzy Inference System Dengan Algoritma Tsukamoto," *J. Inform. J. Pengemb. IT Poltek Tegal*, vol. 03, no. 02, pp. 184–189, 2018.