

# SETTING BONDING MENGGUNAKAN LIBRARY PARAMIKO PADA SWITCH MIKROTIK

Kaneo Dria Perkasa, Aris Sudaryanto  
Fakultas Teknik Informatika, Universitas 17 Agustus 1945  
email : [kaneoperkasa@gmail.com](mailto:kaneoperkasa@gmail.com), [aris@untag-sby.ac.id](mailto:aris@untag-sby.ac.id)

## **Abstract**

Nowadays Automation technology are rapidly evolve, one of the new method is *Network Automation* which is more focused on networking device. *Network Automation* uses python language for the scripting and Took *Software Defined Network* as Inspiration, however *Network Automation* level is below *Software Defined Network*. With this *Network Automation* Method, time needed for configuring device can be shorted. With testing occurred in paramiko for 10 command, 20 command, 30 command, 40 command, 50 command, time sum for send each of one command are 1,25 second

**Keywords** : *Network Automation, Paramiko, Python Scripting*

## **Abstrak**

Teknologi otomasi semakin berkembang pada zaman sekarang, Salah satu metode baru dalam otomasi adalah *Network Automation* yang difokuskan di ranah perangkat jaringan. *Network Automation* menggunakan Bahasa pemrograman *python* yang mengambil inspirasi dari *Software Defined Network*, namun tingkatan *Network Automation* berada di bawah *Software Defined Network*. Dengan metode *Network Automation* ini akan mempersingkat waktu dalam mengkonfigurasi perangkat jaringan. Dengan pengujian yang dilakukan di paramiko waktu yang dibutuhkan pada pengujian 10 baris command, 20 baris command, 30 baris command, 40 baris command dan 50 baris command, waktu yang dibutuhkan untuk mengirim command rata rata adalah 1,25 detik.

**Kata kunci** : *Otomasi Jaringan, Paramiko, Pemrograman Python*

## **1. PENDAHULUAN**

Seiring berkembangnya suatu perusahaan maka aspek jaringan pada suatu perusahaan akan menjadi hal yang penting. Akan tetapi bagi seorang admin jaringan mengelola suatu jaringan yang kompleks akan terasa rumit dan akan memakan waktu [1]. Integrasi standar pada beberapa vendor juga menjadi masalah komabilitas dengan menggunakan protokol standar seperti OpenFlow, Sejak adanya *Software Defined Network* (SDN) vendor mencoba menghilangkan ketergantungan akan protokol standar [2] . Hampir semua vendor mulai menerapkan *network programming* yang digunakan untuk mempermudah dalam

proses mengkonfigurasi perangkat jaringan, serta mengurangi *human error* ketika melakukan konfigurasi.

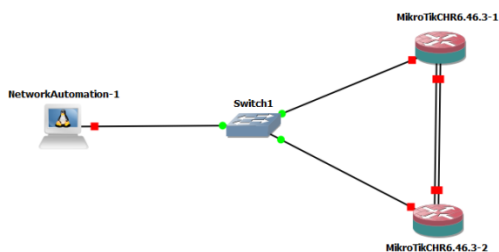
Akan tetapi tidak semua perangkat kompatibel dengan *Software Defined Network*. Maka dari itu ada alternatifnya yaitu *network automation* yang menggunakan bahasa pemrograman python. Pada penelitian yang telah dilakukan oleh Afrianto [3] membuat sistem backup konfigurasi router dengan shell script dimana file konfigurasi yang ada di router akan otomatis masuk ke komputer backup sesuai jadwal yang telah ditentukan.

Dari penelitian yang sudah ada, peneliti melakukan pengiriman konfigurasi otomatis ke router mikrotik menggunakan

salah satu library *network automation* di python. Paramiko [4] merupakan library yang ada di python digunakan untuk mengontrol perangkat jaringan melalui Secure Shell SSH.

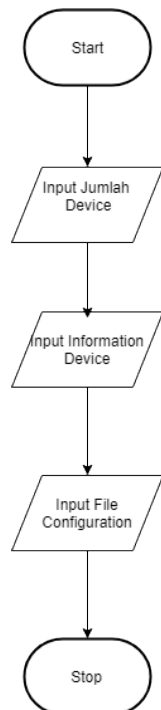
## 2. METODE PENELITIAN

Software yang digunakan untuk Implementasi pada jurnal ini menggunakan software *Graphical Network Simulator 3* (GNS3), yang menggunakan images Mikrotik CHR 6.46.3 untuk topologi yang digunakan pada penelitian ini sebagai berikut.



Gambar 2. 1 Topologi

Untuk Flowchart yang digunakan pada penelitian kali ini sebagai berikut.



Gambar 2. 2 Flowchart

Flowchart diatas menjelaskan cara kerja sistem yang digunakan pada penelitian ini sebagai berikut.

1. Seorang admin jaringan akan menjalankan script python.
2. Seorang admin jaringan akan diminta untuk menginputkan jumlah perangkat yang akan dilakukan konfigurasi. Maksimum konfigurasi bonding yang di jalankan bersamaan adalah 4 channel
3. Seorang admin jaringan akan diminta menginputkan informasi perangkat dalam format JSON yang berisikan ip address, username, password.
4. Terakhir seorang admin jaringan diminta menginputkan file txt yang berisikan command dari mikrotik chr yang digunakan untuk konfigurasi.

## 3. PEMBAHASAN

Pada pembahasan ini penulis menguji berhasil atau tidaknya konfigurasi bonding terkirim ke mikrotik.

```

root@NetworkAutomation-1:~# python final.py
-----
1. Config Devices
2. Cek Konfigurasi Mikrotik
0. Quit
-----
Choice Menu : █
  
```

Gambar 4. 1 menu cek konfigurasi device

Pada tampilan gambar 4.1 adalah menu utama dari script paramiko yang dibuat penulis, untuk mengecek apakah konfigurasi terkirim dapat memilih menu nomor 2.

```

root@NetworkAutomation-1:~# ifconfig eth0 10.1.1.3/24
root@NetworkAutomation-1:~# python final.py
-----
1. Config Devices
2. Cek Konfigurasi Mikrotik
0. Quit
-----
Choice Menu : 2
IP : 10.1.1.1
Username : admin
Password:
Flags: X - disabled, R - running
-----
1. Config Devices
2. Cek Konfigurasi Mikrotik
0. Quit
-----
Choice Menu : █
  
```

Gambar 4. 2 tampilan jika belum terkonfigurasi

Seperti pada gambar 4.2 akan muncul tampilan kosong jika mikrotik belum terkonfigurasi.

```

1. Config Devices
2. Cek Konfigurasi Mikrotik
3. Quit
-----
Choice Menu : 2
IP : 10.1.1.1
Username : admin
Password:
Flags: X - disabled, R - running
0 R name="bond21" mtu=1500 mac-address=0C:4C:0C:A7:8D:01 arp-enabled
arp-timeout=auto slaves=ether2,ether3 mode=balance-rx primary=none
link-monitoring=mi arp-interval=10ms arp-ip-target=""
mi-interval=10ms down-delay=5ms up-delay=5ms lscp-rate=30secs
transmit-hash-policy=layer-2 min-links=0
-----

```

Gambar 4. 3 tampilan jika sudah terkonfigurasi

Bisa dilihat dari gambar 4.3 menunjukkan interface bonding yang telah dikirim melalui script paramiko yang penulis buat berhasil masuk ke perangkat mikrotik 1.

UJI COBA	UJI COBA SINGLE			
	CHR 1		CHR 2	
	rx-current	rx-10s-average	rx-current	rx-10s-average
1	14.0Gbps	13.5Gbps	14.6Gbps	14.0Gbps
2	12.9Gbps	13.4Gbps	13.1Gbps	13.6Gbps
3	13.4Gbps	13.9Gbps	15.8Gbps	14.6Gbps
4	14.1Gbps	13.6Gbps	14.0Gbps	13.5Gbps
5	12.5Gbps	13.1Gbps	16.0Gbps	14.6Gbps
6	14.0Gbps	13.5Gbps	14.5Gbps	13.8Gbps
7	14.2Gbps	14.0Gbps	14.8Gbps	14.2Gbps
8	13.9Gbps	13.3Gbps	14.1Gbps	13.6Gbps
9	14.3Gbps	13.3Gbps	14.4Gbps	13.2Gbps
10	13.8Gbps	13.6Gbps	12.6Gbps	13.5Gbps
11	14.2Gbps	13.9Gbps	14.1Gbps	13.6Gbps
12	15.8Gbps	14.9Gbps	14.0Gbps	13.4Gbps
13	13.2Gbps	12.9Gbps	15.3Gbps	14.6Gbps
14	14.7Gbps	14.7Gbps	16.1Gbps	15.0Gbps
15	15.4Gbps	15.0Gbps	14.5Gbps	14.2Gbps
16	11.6Gbps	13.4Gbps	14.1Gbps	13.6Gbps
17	14.0Gbps	13.6Gbps	15.3Gbps	15.0Gbps
18	13.9Gbps	13.3Gbps	13.4Gbps	13.7Gbps
19	13.4Gbps	13.8Gbps	14.4Gbps	13.4Gbps
20	12.6Gbps	13.7Gbps	13.1Gbps	13.3Gbps

Gambar 4. 4 uji coba single sebelum bonding

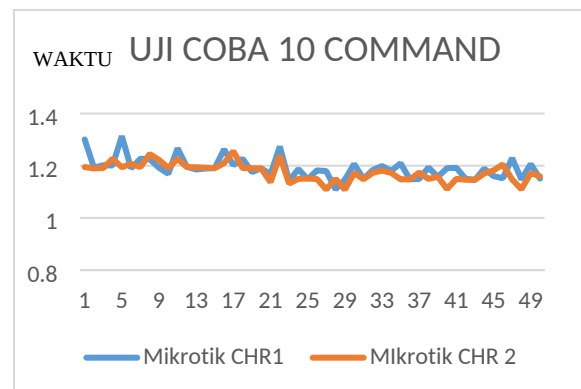
Pada pembahasan selanjutnya penulis mencoba melakukan uji coba bandwidth perangkat sebelum dan sesudah dilakukan bonding, dapat dilihat pada gambar 4.4 adalah hasil uji coba sebelum dilakukan konfigurasi bonding.

UJI COBA	UJI COBA BONDING			
	CHR 1		CHR 2	
	rx-current	rx-10s-average	rx-current	rx-10s-average
1	14.4Gbps	14.1Gbps	14.8Gbps	13.8Gbps
2	15.5Gbps	15.4Gbps	14.4Gbps	14.2Gbps
3	14.4Gbps	14.2Gbps	14.5Gbps	13.9Gbps
4	14.9Gbps	14.0Gbps	13.0Gbps	13.5Gbps
5	14.0Gbps	13.2Gbps	14.3Gbps	13.7Gbps
6	14.7Gbps	13.0Gbps	14.2Gbps	13.9Gbps
7	15.1Gbps	14.3Gbps	13.5Gbps	13.2Gbps
8	15.3Gbps	14.4Gbps	14.8Gbps	15.0Gbps
9	15.3Gbps	14.5Gbps	13.5Gbps	13.9Gbps
10	14.6Gbps	14.0Gbps	14.1Gbps	14.5Gbps
11	15.2Gbps	14.6Gbps	14.7Gbps	14.9Gbps
12	14.3Gbps	14.0Gbps	14.1Gbps	13.2Gbps
13	13.4Gbps	13.2Gbps	13.9Gbps	13.9Gbps
14	14.0Gbps	13.4Gbps	14.1Gbps	13.4Gbps
15	13.8Gbps	13.1Gbps	13.3Gbps	14.1Gbps
16	13.8Gbps	13.4Gbps	16.4Gbps	15.0Gbps
17	14.1Gbps	13.2Gbps	13.2Gbps	13.6Gbps
18	14.4Gbps	13.7Gbps	14.0Gbps	13.3Gbps
19	14.0Gbps	13.7Gbps	12.1Gbps	13.2Gbps
20	14.5Gbps	13.8Gbps	14.1Gbps	13.6Gbps

Gambar 4. 5 uji coba setelah terkonfigurasi bonding

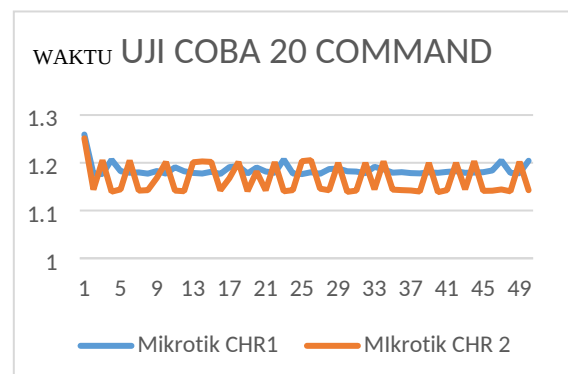
Lalu dapat dilihat pada gambar 4.5 adalah uji coba setelah konfigurasi bonding berhasil terkirim, perlu diketahui karena pengujian dilakukan menggunakan software virtual maka hasil pengujian akan berbeda beda dan bergantung pada spesifikasi hardware pengujian.

Pada pembahasan kali ini penulis mencoba melakukan percobaan menggunakan paramiko dengan cara mengirimkan sejumlah command pada masing-masing switch mikrotik chr untuk mencari tau berapa jumlah waktu yang dibutuhkan.



Gambar 4. 6 perbandingan hasil uji coba 10 command

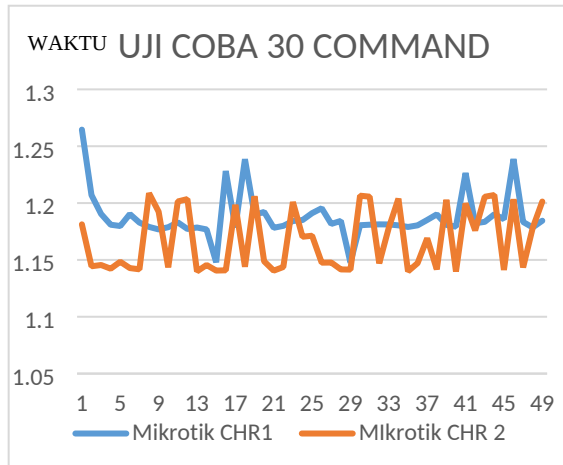
Pada pengujian yang dilakukan dengan mengirim 10 command, hasil yang terlihat di atas rata rata waktu chr 2 lebih cepat yaitu di 1,2 seperkian detik sampai 1,1 seperkian detik.



Gambar 4. 7 perbandingan hasil uji coba 20 command

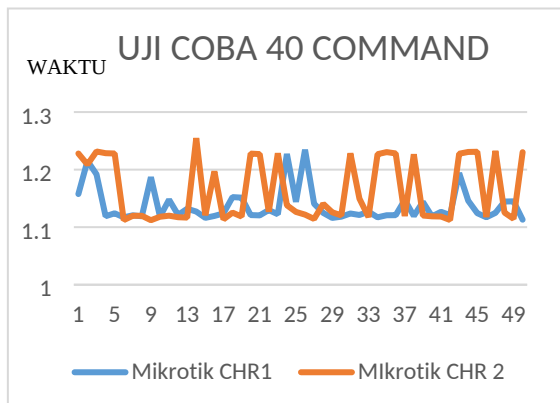
Pada pengujian yang dilakukan dengan mengirim 20 command, hasil yang terlihat di

atas rata rata waktu chr 2 lebih cepat yaitu di 1,25 seperkian detik sampai 1,1 seperkian detik.



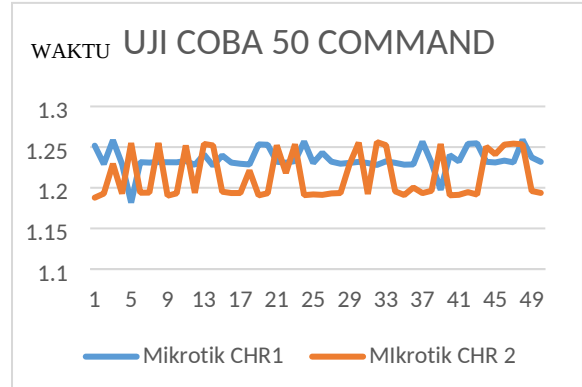
Gambar 4. 8 perbandingan hasil uji coba 30 command

Pada pengujian yang dilakukan dengan mengirim 30 command, hasil yang terlihat di atas rata rata waktu chr 2 lebih cepat yaitu di 1,14 seperkian detik.



Gambar 4. 2 perbandingan hasil uji coba 40 command

Pada pengujian yang dilakukan dengan mengirim 40 command, hasil yang terlihat di atas rata rata waktu chr 1 lebih cepat yaitu di 1,12 seperkian detik. sedangkan chr 2 menembus 1,25 detik.



Gambar 4. 9 perbandingan hasil uji coba 50 command

Pada pengujian yang dilakukan dengan mengirim 50 command, hasil yang terlihat di atas rata rata waktu chr 2 lebih cepat yaitu di 1,19 seperkian detik.

#### 4. KESIMPULAN DAN SARAN

Berdasarkan tahapan yang telah dilakukan, kesimpulan yang didapatkan sebagai berikut :

1. Dari pengujian yang dilakukan di kedua mikrotik chr masing-masing hanya berbeda beberapa milidetik dimana chr 2 memiliki rata rata waktu lebih cepat.

Supaya aplikasi yang akan datang menjadi lebih baik, maka penulis menyarankan :

1. Karena tampilan program masih berbasis CLI (Command Line Interface), maka untuk kedepannya bisa menggunakan tampilan GUI (Graphical User Interface)
2. Bisa juga diuji dengan konfigurasi selain bonding

#### DAFTAR PUSTAKA

- [1] A. R. Komarudin, Otomatisasi Administrasi Jaringan Dengan Script Python, Jakarta: Jasakom, 2018.
- [2] Furqan A. New Networking Era: Software Defined Networking. 2013; 3(1): 349-353
- [3] Mashuda A. Sistem Backup Konfigurasi Router Secara Otomatis Dengan Shell Script. 2019; 15(1): 57-69

- [4] paramiko. (2021, January 12). Paramiko.  
Retrieved from docs.paramiko.org:  
<http://docs.paramiko.org/en/stable/>