

Otomatisasi Jaringan Menggunakan Script Python Untuk Penyediaan Konfigurasi Internet Dan Manajemen Mikrotik

Muhammad Fahmi¹, Maisyaroh Maisyaroh^{2*}, Ishak Komarudin³, Siti Faizah¹, Irfan Fadhillah¹

¹ Teknik Informatika; Universitas Nusa Mandiri; Jl. Jatiwaringin Raya No.02 Cipinang Melayu Jakarta Timur, 13620, Jakarta, Indonesia; e-mail: fahmi.mmf@nusamandiri.ac.id, siti.sfz@nusamandiri.ac.id, 12190363@nusamandiri.ac.id

² Teknologi Komputer; Universitas Bina Sarana Informatika; Jl. Kramat Raya No.98, Kwitang, Jakarta Pusat, 10450, Jakarta, Indonesia; e-mail: maysaroh.msy@bsi.ac.id

³ Sistem Informasi; Universitas Bina Sarana Informatika; Jl. Kramat Raya No.98, Kwitang, Jakarta Pusat, 10450, Jakarta, Indonesia; e-mail: ishak.komarudin@bsi.ac.id

* Korespondensi: e-mail: maysaroh.msy@bsi.ac.id

Diterima: 28 Mei 2021; Review: 16 Juni 2021; Disetujui: 21 Juni 2021

Cara sitasi: Fahmi M, Maisyaroh M, Ishak K, Faizah S, Fadhillah I. 2021. Otomatisasi Jaringan Menggunakan Script Python Untuk Penyediaan Konfigurasi Internet Dan Manajemen Mikrotik. Bina Insani ICT Journal. Vol. 8 (1): 53-62.

Abstrak: Sistem penyediaan konfigurasi internet awal atau "*Provisioning*" untuk pelanggan baru pada PT. Link Net Tbk, masih menggunakan cara manual yaitu dengan masuk dan konfigurasi satu per satu ke perangkat jaringan. Begitu juga untuk pengelolaan perangkat-perangkat router Mikrotik eksisting yang berjumlah cukup banyak, apabila ada perubahan masih dilakukan secara manual. Setelah dilakukan penelitian dengan metode observasi, wawancara ke pihak terkait, dan studi pustaka melalui buku dan jurnal internet didapatkan salah satu solusi untuk mengatasi masalah-masalah tersebut. Otomatisasi jaringan merupakan solusinya. Otomatisasi jaringan menggunakan *script Python* dan *library Paramiko* dapat melakukan pekerjaan untuk penyediaan internet dan perubahan konfigurasi masal ke router Mikrotik *eksisting*. Hal ini dapat mempersingkat proses pekerjaan secara signifikan serta dapat menghindari adanya kesalahan konfigurasi oleh manusia.

Kata kunci: konfigurasi, mikrotik, otomatisasi jaringan, paramiko, python.

Abstract: *Initial internet configuration system or Provisioning for new customers at PT. Link Net Tbk, still uses the manual method, namely by logging in and configuring one by one to network devices. Like wise for the management of existing Mikrotik router devices which amount to quite a lot, if there are changes, they are still done manually. After conducting research using observation methods, interviews with related parties, and literature studies through books and internet journals, one solution was found to overcome these problems. Network automation is the solution. Network automation using Python scripts and the Paramiko library can do the job of providing internet and mass configuration changes to existing Mikrotik routers. This can significantly shorten the work process and can avoid misconfiguration by humans.*

Keywords: *provisioning, mikrotik, network automation, paramiko, python.*

1. Pendahuluan

Sebagai perusahaan yang bergerak di bidang layanan teknologi informasi dan internet, PT. Link Net, Tbk (Link Net) saat ini telah mempunyai *Point of Presence* (PoP), kantor pusat dan beberapa cabang yang tersebar di beberapa wilayah Indonesia yang sudah terintegrasi, hal ini untuk menghadirkan ketersediaan jaringan komunikasi data dan internet untuk masyarakat perumahan (*residential*) maupun korporasi (*corporate*). Untuk menunjang ketersediaan layanan untuk pelanggan korporasi tersebut dibutuhkan infrastruktur jaringan yang handal dan sistem redundansi jaringan yang optimal. Dalam praktiknya, Link Net menggunakan infrastruktur jaringan serat optik atau *Fiber Optic* (FO) sebagai jaringan akses untuk pelanggan korporasi, juga menyediakan layanan tambahan dengan infrastruktur jaringan *Wireless* sebagai jaringan akses alternatif di beberapa daerah yang dapat terjangkau.

Jaringan akses *wireless* menjadi sebuah alternatif untuk memenuhi permintaan pelanggan yang memerlukan koneksi internet dengan proses pemasangan yang relatif cepat, atau lokasi pelanggan belum terjangkau oleh jaringan FO. Perangkat jaringan yang digunakan saat ini untuk infrastruktur jaringan *wireless* sebagian besar adalah *Mikrotik*. Saat ini Link Net mengelola lebih dari 100 (seratus) unit *Mikrotik* yang difungsikan sebagai *Bandwidth Management*, *Firewall*, *Internet Distribution Router*, dan *Customer Edge Router*.

Konfigurasi router pada Link Net 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. Karena itu penting bagi perusahaan-perusahaan untuk melakukan otomatisasi jaringan.

Sebelum ada *network automation*, administrator jaringan perlu melakukan semuanya secara manual dan memerlukan koordinasi yang rumit. Namun sekarang, administrator jaringan bisa menggunakan *network automation tool* untuk melakukan pekerjaan-pekerjaan yang rumit tersebut, sehingga pekerjaan bisa terselesaikan dengan jauh lebih cepat [1].

Otomatisasi jaringan merupakan sebuah solusi dalam mengimplementasikan beberapa pekerjaan yang rumit atau berulang. Dalam metode tradisional konfigurasi pada perangkat jaringan dilakukan dengan masuk ke dalam perangkat jaringan untuk mengkonfigurasi perangkat jaringan tersebut, pekerjaan tersebut terlihat mudah bila yang dikonfigurasi masih satu atau dua perangkat jaringan, hal ini akan terlihat rumit jika terdapat banyak perangkat jaringan seperti router dan server yang harus dikonfigurasi [2].

Otomatisasi jaringan juga adalah proses mengotomatisasi konfigurasi, pengujian, operasi perangkat virtual dalam jaringan, dan manajemen [3]. Dalam implementasi sistem otomatisasi jaringan dibutuhkan sebuah software atau program untuk menjalankan perintah ke dalam sebuah perangkat. Salah satu program yang dapat digunakan adalah *Python*. *Python* sendiri memiliki beberapa jenis *library* yang digunakan untuk implementasi sistem otomatisasi jaringan diantaranya adalah *Paramiko* dan *Netmiko* [4].

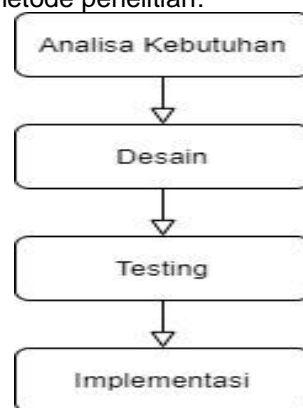
Seiring dengan bertambahnya cakupan daerah jaringan Link Net dan kebutuhan koneksi internet pelanggan korporasi meningkat, permintaan pemasangan internet juga semakin meningkat, berbanding lurus dengan perangkat jaringan yang dikelola jumlahnya juga akan terus bertambah. Berdasarkan latar belakang masalah ini muncul ide untuk melakukan otomatisasi jaringan untuk *provisioning* atau penyediaan konfigurasi awal layanan internet untuk pelanggan baru pada perangkat jaringan *wireless* dan manajemen perangkat *Mikrotik* ISP Link Net menggunakan *script Python*. Telah dilakukan penelitian dan pembahasan mengenai "Otomatisasi Jaringan Menggunakan *Script Python* untuk Penyediaan Konfigurasi Internet dan Manajemen *Mikrotik*".

2. Metode Penelitian

Dalam penelitian ini pengumpulan data dan informasi dilakukan dengan cara menggunakan beberapa metode pengumpulan data. Adapun metode pengumpulan data yang digunakan adalah: a) Observasi, secara langsung mengamati, mencoba, merekam data, menganalisa secara aktual di PT. Link Net, Tbk Department *Access Network*, serta menguji rencana otomatisasi jaringan menggunakan aplikasi simulasi jaringan dan mesin virtual sebelum dilakukan implementasi langsung ke jaringan *eksisting*, b) Wawancara: menanyakan secara verbal dan berdiskusi tentang informasi yang dibutuhkan kepada Bapak Mochammad

Yasin selaku *Access Network Department Head* dan Bapak Agus Supriyanto selaku *System Engineer* di PT. Link Net, Tbk., c) Studi Pustaka: melakukan studi literatur dan mengumpulkan beberapa referensi yang berkaitan dengan topik bahasan melalui buku, jurnal dan internet sedangkan metode penelitian yang digunakan dalam penelitian ini adalah *Network Development Life Cycle (NDLC)*. *NDLC* merupakan sebuah metode dalam membangun suatu jaringan dalam suatu tempat untuk menggambarkan atau merancang topologi jaringan untuk mengetahui statistik dan kinerja jaringan dengan pemantauan jaringan [5]. *NDLC* adalah salah satu metode yang dilakukan dalam pengembangan metode dalam jaringan. Dimana *NDLC* memiliki enam (6) tahapan, tahapan-tahapan yang dimaksud adalah sebagai berikut: 1) *Analysis*, 2) *Design*, 3) *Simulation Prototyping*, 4) *Implementation*, 5) *Monitoring*, 6) *Management* [6].

Tahapan *NDLC* dalam metode penelitian yang digunakan dalam penelitian ini adalah: 1) *Analisa Kebutuhan*: untuk implementasi otomatisasi jaringan ini dibutuhkan 1 (satu) buah server network automation berbasis linux ubuntu sebagai pusat pengontrol konfigurasi, paket instalasi Python dan library Paramiko, 2) *Desain*: *server network automation* dihubungkan ke *switch core* dan dikonfigurasi agar dapat terhubung ke jaringan eksisting agar bisa *me-remote* atau mengontrol semua perangkat jaringan, 3) *Testing*: pengetesan dapat dilakukan setelah instalasi dan konfigurasi server untuk otomatisasi jaringan selesai. Tahap awal akan dilakukan pengetesan login ke perangkat yang dibutuhkan untuk di-otomatisasi (Mikrotik Firewall dan *Bandwidth Management* serta *Internet Distribution Router*) menggunakan protokol SSH dan menjalankan beberapa perintah dasar melihat status router menggunakan *script Python*, 4) *Implementasi*: setelah berhasil login dan dapat menjalankan perintah dasar dengan benar dan hasilnya sesuai ekspektasi, selanjutnya akan dilakukan penambahan konfigurasi yang dibutuhkan untuk provisioning internet, diantaranya pendaftaran *IP public* pada perangkat Firewall, melakukan limitasi bandwidth sesuai paket berlangganan pelanggan pada perangkat *Bandwidth Management*, serta penambahan konfigurasi *gateway IP public* pelanggan pada perangkat *Internet Distribution Router*. Selain itu juga dilakukan konfigurasi radius dan *Simple Network Management Protocol (SNMP)* pada ratusan perangkat *Mikrotik eksisting* untuk menerapkan sistem manajemen jaringan. *SNMP* adalah sebuah protokol yang digunakan untuk memonitor perangkat pada jaringan, seperti memonitor peralatan jaringan (seperti *switch*), peralatan komputer, dan perangkat lainnya [7]. Protokol *SNMP* digunakan untuk mengumpulkan data perangkat dengan melakukan permintaan informasi pada perangkat. Informasi yang dikumpulkan digunakan untuk menghitung nilai *availability* pada perangkat. Basis data digunakan untuk menampung hasil dari permintaan *SNMP* [8]. *SNMP* menggunakan sistem log yang menyimpan segala informasi dari perangkat jaringan yang terdaftar. Sehingga keuntungan dari menggunakan *SNMP* dapat mengambil segala kebutuhan informasi perangkat yang digunakan dan dapat ditampilkan untuk memenuhi kebutuhan sistem monitoring [9]. Berikut ini gambar kerangka pemikiran dari metode penelitian:



Sumber: Hasil Penelitian (2021)

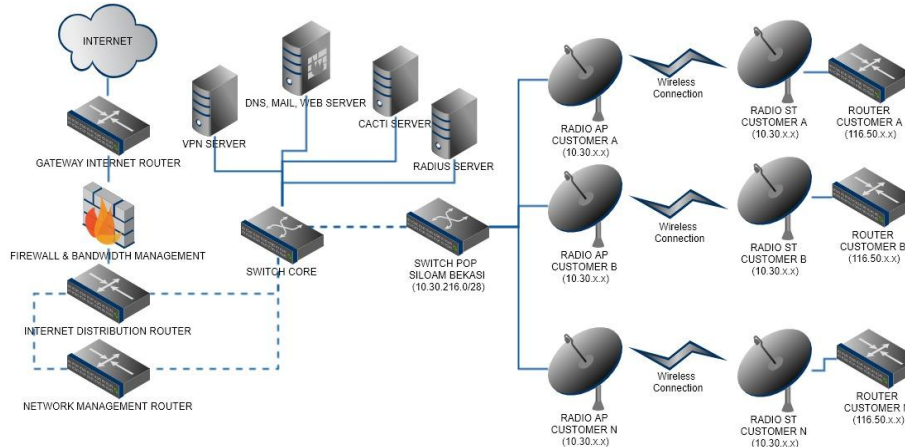
Gambar 1. Kerangka Pemikiran

Pada penelitian sebelumnya [10] telah berhasil dilakukan pengujian otomatisasi jaringan. Dalam penelitian tersebut dibutuhkan beberapa perangkat keras dan perangkat lunak untuk melakukan pengujian. Perangkat keras yang digunakan seperti PC atau laptop, server, router Mikrotik, router Cisco dan switch unmanageable. Sedangkan perangkat lunak yang

digunakan adalah sistem operasi Ubuntu yang digunakan sebagai server, paket instalasi Python dan library Paramiko. Penelitian yang dilakukan sebelumnya juga [3], aplikasi otomatisasi jaringan dapat membantu *administrator* jaringan dalam melakukan konfigurasi penambahan *gateway* otomatis pada sebuah topologi jaringan yang terhubung dengan simulator jaringan. Telah berhasil dilakukan konfigurasi otomatis penambahan *gateway* pada virtual *router* menggunakan aplikasi otomatisasi jaringan berbasis *web*. Kemudian penelitian selanjutnya [4], dilakukan pengujian perbandingan kinerja *Library Paramiko* dan *Netmiko* dalam proses otomatisasi jaringan menggunakan *Python*. Hasil dari penelitian tersebut performansi *library Paramiko* dalam proses otomatisasi jaringan lebih baik dari *Netmiko*. *Library Paramiko* menggunakan satu tahap proses eksekusi perintah, sedangkan pada penggunaan *library Netmiko* terdapat dua tahap eksekusi perintah. Dengan adanya dua kali tahap eksekusi perintah tersebut maka waktu proses kirim data *script Python* ke perangkat *router* dengan menggunakan *library Netmiko* 4,14 kali lebih lambat dibandingkan jika menggunakan *Paramiko*.

3. Hasil dan Pembahasan Topologi Jaringan Berjalan

Topologi yang digunakan di Link Net untuk distribusi ke pelanggan-pelanggan yang menggunakan akses *wireless* adalah menggunakan topologi *star*. Topologi *star* merupakan topologi yang fleksibel karena pemasangan perangkat untuk pelanggan baru cukup mudah, dan tidak mengganggu jaringan pelanggan lainnya, kontrol terpusat karena masing-masing terhubung ke *port switch*, sehingga memudahkan pengecekan kinerja jaringan dan dalam mendeteksi kesalahan atau gangguan pada jaringan. Arsitektur jaringan akses yang digunakan di Link Net agar pelanggan dapat terkoneksi internet dengan memasang perangkat radio di kedua sisi, baik di sisi POP yang disebut *Radio Access Point*, maupun di sisi lokasi pelanggan yang disebut *Radio Station*. Frekuensi radio yang digunakan untuk berkomunikasi adalah 5725-5825 MHz. Sedangkan untuk jaringan tulang punggung dari POP ke Data Center pusat menggunakan infrastruktur *Fiber Optic* (FO) Link Net yang handal. Topologi jaringan yang berjalan saat ini dapat dijelaskan pada gambar di bawah ini.



Sumber: Hasil Penelitian (2021)

Gambar 2. Topologi Jaringan Berjalan

Pada Gambar 2 diambil sampel salah satu POP. Setiap pelanggan akan dialokasikan atau diberikan *IP address public /30* yang dipasang di *router* pelanggan. Saat ini apabila terdapat pemasangan jaringan pelanggan baru, *administrator* jaringan memasukkan manual informasi identitas pelanggan dan alokasi *IP address public* untuk pelanggan di perangkat *Internet Distribution Router*, selanjutnya mendaftarkan *IP public* pelanggan dan membuat limitasi *bandwidth* sesuai paket berlangganan pelanggan di perangkat *Firewall-Bandwidth Management*.

Selain itu, kondisi saat ini router di pelanggan belum semua terpantau dengan baik, seperti konfigurasi *SNMP* yang belum diterapkan, dikarenakan belum terdapat prosedur tertulis untuk standar konfigurasi *router* pelanggan yang dikerjakan oleh tim instalasi dan aktivasi lapangan.

Permasalahan

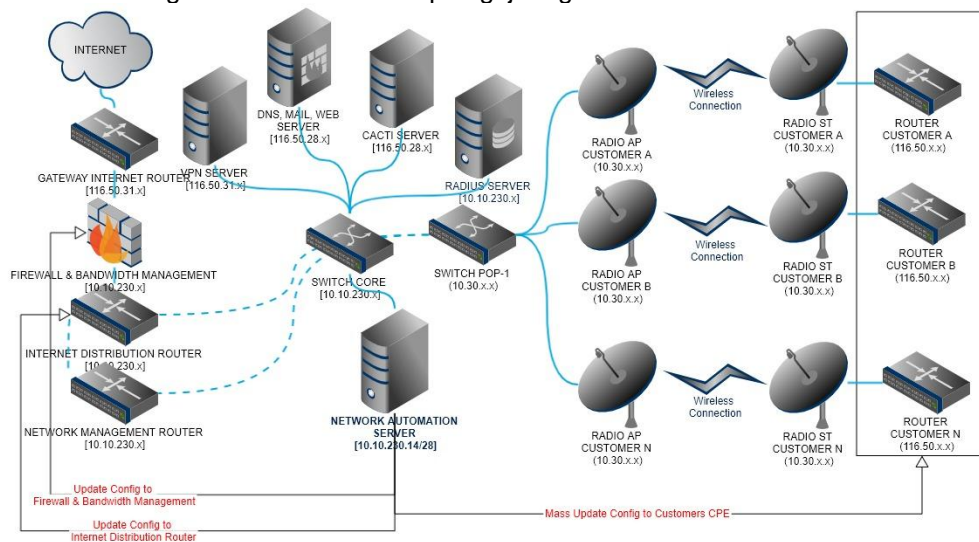
Berdasarkan penelitian yang dilakukan, ditemukan beberapa hal terkait pekerjaan sehari-hari staf *engineer* Link Net, yaitu masih dilakukan secara manual dalam melakukan konfigurasi awal atau *provisioning* layanan internet pelanggan baru, sehingga ada kemungkinan terjadinya *human error*. Masalah lainnya adalah belum seragamnya konfigurasi yang dipakai di sisi perangkat *router* CPE, seperti konfigurasi *SNMP* dan *radius* pada masing-masing perangkat, sehingga perangkat-perangkat tersebut belum dapat terpantau dengan baik di *Network Monitoring System* (NMS).

Alternatif Pemecahan Masalah

Berdasarkan analisa permasalahan yang sebelumnya diuraikan, diperlukan untuk menerapkan otomatisasi jaringan atau *Network Automation* menggunakan *script Python* pada lingkungan jaringan eksisting, yang diharapkan dapat mempersingkat waktu *provisioning* dan meminimalisir *human error*. Selain itu otomatisasi jaringan juga bisa diterapkan ke semua perangkat-perangkat *router* CPE eksisting, untuk melakukan manajemen perubahan langsung ke semua perangkat tersebut tanpa harus masuk ke satu persatu perangkat yang mana tidak efisien dan memakan waktu cukup lama karena berjumlah ratusan perangkat.

Topologi Jaringan Usulan

Topologi jaringan yang diusulkan tetap memanfaatkan semua infrastruktur dan topologi jaringan yang berjalan, hanya saja perlu penambahan 1 (satu) unit *server* yang terhubung ke *switch core* eksisting. Berikut ini adalah topologi jaringan usulan:



Sumber : Hasil Penelitian (2021)

Gambar 3. Topologi Jaringan Usulan

Berdasarkan topologi di atas, terdapat 1 (satu) unit server tambahan yang dihubungkan langsung ke *switch core* dan menggunakan blok IP address seperti server lainnya, hal ini agar server baru tersebut dapat mengakses ke semua *network* atau *IP address* eksisting. Server tersebut diberi label *Network Automation Server*, server perlu diinstal *Operating System Linux Ubuntu* dan paket instalasi *Python*. *Network Automation Server* berfungsi untuk menjalankan semua perintah secara terpusat dengan *script Python* untuk *provisioning* internet pelanggan baru, serta dapat melakukan perubahan konfigurasi secara masal ke semua perangkat Mikrotik eksisting untuk penerapan manajemen jaringan. *Network Automation Server* dapat di-remote menggunakan protokol *SSH* menggunakan aplikasi *SSH Client*, misalnya *Putty* dari jaringan internal Link Net, atau apabila ingin mengakses dari luar harus melakukan koneksi *VPN* terlebih dahulu. *Network Automation Server* dapat melakukan update konfigurasi ke perangkat-perangkat jaringan diantaranya *Firewall-Bandwidth Management*, *Internet Distribution Router*, dan *router-router* CPE pelanggan. *Script Python* dimanfaatkan untuk keperluan *provisioning*

internet pelanggan baru dan perubahan konfigurasi masal ke router-router CPE pelanggan *eksisting*.

Rancangan Aplikasi

Rancangan aplikasi otomatisasi jaringan menggunakan *script* dengan bahasa pemrograman *Python* yang dijalankan pada *server* berbasis *Linux Ubuntu*. Berikut adalah langkah-langkah yang dilakukan untuk perancangan aplikasi otomatisasi jaringan yang diusulkan: 1) Persiapan *Server* dan Instalasi OS *Ubuntu*, *server* yang digunakan untuk implementasi otomatisasi jaringan menggunakan *Virtual Machine* (VM) dengan spesifikasi *Memory* 2GB, *Processor* 2 Core, *Harddisk* 40GB, dan OS *Linux Ubuntu* 20.04.1 LTS. File instalasi sistem operasi *Ubuntu* bisa didapatkan dari situs resmi *Ubuntu* (<https://ubuntu.com/download/server>). Untuk instalasi di *server* fisik dapat menggunakan CD atau melalui USB yang sudah diisi *installer* *Ubuntu*. Untuk instalasi menggunakan *Virtual Machine* seperti *VMware* dapat langsung menggunakan file ISO *Ubuntu*, 2) Instalasi *Python3* dan *Python3-pip*. Setelah OS *Ubuntu* terinstal bisa cek apakah *python* sudah terpasang di *server*, secara *default* saat ini *Python* sudah terpasang di *Ubuntu*. Selanjutnya instal *python3-pip*, dengan perintah berikut:

```
#apt install python3
#apt install python3-pip
```

3) Instalasi *Paramiko*, Setelah paket *python3* dan *python3-pip* terinstal, selanjutnya instalasi *library Paramiko* dengan perintah "*pip3 install paramiko*", 4) Membuat *Script Python* untuk *Provisioning*. Untuk membuat *script python* diperlukan *text editor*, dalam hal ini digunakan *text editor nano* dan disimpan dengan format ".py". Berikut merupakan *script python* yang digunakan untuk *provisioning* internet pelanggan baru, dinamakan *file script* dengan nama *provisioning1.py*:

```
import paramiko
import time
import getpass
print ("\n-----")
print ("Welcome to Network Automation for IP Provisioning Wireless")
print ("-----\n")
username = input("User Login: ")
password = getpass.getpass()
ssh = paramiko.SSHClient()
ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
ssh.connect(hostname='10.10.230.2', username=username, password=password)
print ("Login berhasil!\n")
customer_name = input("Masukkan CID dan nama customer (contoh A00001-001_PT-ABC) : ")
ip_allocation = input("Masukkan alokasi IP gateway dan prefix lenght customer (contoh 116.50.24.1/29) : ")
interface = input("Masukkan interface (contoh vlan123) : ")
print ("Sedang melakukan konfigurasi...")
print ("[DONE]")
ssh.exec_command("ip address add address={} interface={} comment={}".format(ip_allocation,interface,customer_name))
time.sleep(0.5)
print ("Sukses provisioning IP {}\n".format(customer_name))
ip_allocation2 = input("Masukkan alokasi IP network dan prefix length customer (contoh 116.50.24.0/29) : ")
bw_allocation1 = input("Masukkan bandwidth internasional customer (contoh 10M) : ")
bw_allocation2 = input("Masukkan bandwidth lokal customer (contoh 10M) : ")
ssh.connect(hostname='10.10.230.7', username=username, password=password)
print ("Sedang melakukan konfigurasi lagi...")
print ("[DONE]")
ssh.exec_command("ip firewall address-list add list=reg-cst-public address={} comment={}".format(ip_allocation2,customer_name))
```

```
ssh.exec_command("queue simple add name={} target={} max-limit={}/{}".format(customer_name,ip_allocation2,bw_allocation2,bw_allocation2))
ssh.exec_command("queue simple add name={}_INT target={} max-limit={}/{} parent={} packet-marks=packet-int".format(customer_name,ip_allocation2,bw_allocation1,bw_allocation1,customer_name))
ssh.exec_command("queue simple add name={}_IIX target={} max-limit={}/{} parent={} packet-marks=packet-iix".format(customer_name,ip_allocation2,bw_allocation2,bw_allocation2,customer_name))
time.sleep(1)
print ("Sukses mendaftarkan IP {} di Firewall dan membuat profile di Bandwidth Management\n".format(customer_name))
ssh.close()
```

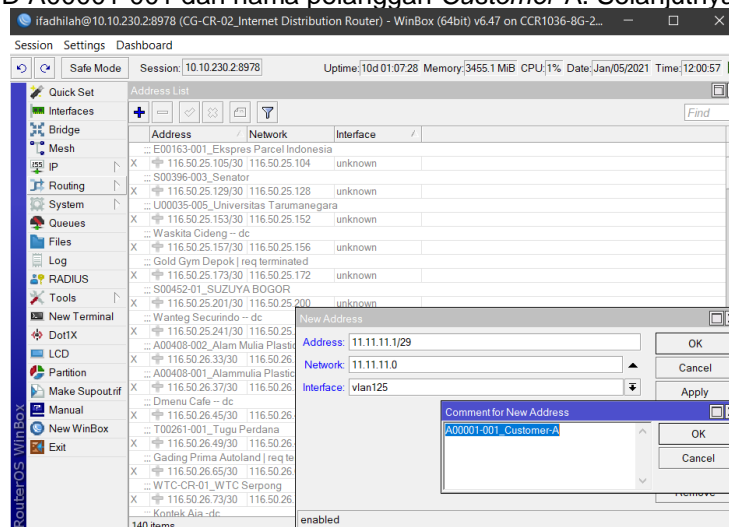
5) Otomatisasi Jaringan *Script Python*, untuk melakukan otomatisasi jaringan yang telah dibahas sebelumnya, *script python* yang sudah dibuat harus dijalankan pada server. Berikut adalah perintah untuk menjalankan *script python*:

```
#python3 provisioning1.py
#python3 mass-update-config1.py
```

Setelah dijalankan, maka akan muncul interaktif untuk memasukkan detail yang dibutuhkan, seperti ID pelanggan, alokasi IP untuk pelanggan, dan *bandwidth* berlangganan pelanggan. Sebagai contoh kita akan *provisioning Customer A* dengan ID pelanggan A00001-001, alokasi IP address 11.11.11.0/29, dan *bandwidth* berlangganan 1 (satu) Mbps internasional dan 10 (sepuluh) Mbps lokal. Selanjutnya dilakukan *update* konfig *Simple Management Network Protocol (SNMP)*, *Radius* dan *Network Time Protocol (NTP)* router secara bersamaan untuk *router* eksisting *Customer A* dengan IP 11.11.11.2 dan *Customer B* dengan IP 12.12.12.2.

Pengujian Jaringan Awal

Pengujian awal dicoba untuk melakukan *provisioning* internet pelanggan baru dan *update* konfigurasi secara manual pada setiap *router* pelanggan eksisting. Sebagai contoh kita akan *provisioning Customer A* dengan ID pelanggan A00001-001, alokasi IP address *dummy* 11.11.11.0/29, dan *bandwidth* berlangganan 1 (satu) Mbps internasional serta 10 (sepuluh) Mbps lokal. Berikut adalah langkah-langkah untuk konfigurasi *provisioning* secara manual: 1) Masuk via *Winbox* ke perangkat *Internet Distribution Router (IDR)*, 2) Setelah masuk ke *IDR*, klik menu *IP – Addresses – New Address symbol (+)*, lalu masukkan pada menu *Address IP gateway* beserta *subnet* pelanggan sesuai alokasi sebelumnya (11.11.11.1/29), *Network* 11.11.11.0, *Interface* sesuai *interface vlan* atau lokasi *POP* contoh *vlan125*, dan *Comment* untuk deskripsi ID A00001-001 dan nama pelanggan *Customer-A*. Selanjutnya klik *OK*.



Sumber: Hasil Penelitian (2021)

Gambar 4. Menambahkan IP Address Pelanggan Baru Secara Manual

3) Masuk via *Winbox* ke perangkat *Firewall – Bandwidth Management (FW_BWM)*, 4) Setelah masuk, daftarkan IP address pelanggan baru di IP – *Firewall – Address List* seperti pada gambar berikut, 5) Setelah mendaftarkan IP Address di *Firewall*, selanjutnya melakukan limitasi *bandwidth* pelanggan baru, sesuai paket berlangganan pelanggan, contoh 1 Mbps Internasional dan 10 Mbps Lokal. Masuk ke menu *Queue – Simple Queues – Add (+)*. Pada tab *General*, *Name* diisi ID dan nama pelanggan “A00001-001_Customer-A”, *Target* diisi IP address pelanggan “11.11.11.0/29”, *Max Limit Upload* 10M, *Max Limit Download* 10M.

Selanjutnya dilakukan update konfigurasi *SNMP*, *Radius* dan *NTP router* secara bersamaan untuk *router* pelanggan eksisting. Jumlah *router* pelanggan saat ini sekitar 113 unit, tidak mungkin dilakukan *update* konfigurasi secara manual satu persatu ke perangkat tersebut karena akan memakan waktu yang sangat lama. Dicontohkan langkah-langkah konfigurasi manual pada salah satu *router* pelanggan dengan IP 10.30.216.25 sebagai berikut: 1) Masuk via *Winbox* ke perangkat *router* IP 10.30.216.25, 2) Masuk menu IP – *SNMP*, centang *Enabled* untuk mengaktifkan *SNMP*. Klik menu *Communities*, klik *New (+)*, masukkan *Name* “kvcn”, dan klik *OK*. Selanjutnya di menu awal *SNMP* isi *Contact Info* diisi dengan “LinkNet”, *Trap Community* pilih “kvcn”, dan *Trap Version* pilih “2”, 3) Untuk setting *radius*, masuk menu *Radius*, klik *Add (+)*, *Service* centang *login*, *Address* isi *server radius* “10.10.230.5”, *secret* di isi sesuai kredensial, 4) Untuk setting *NTP*, masuk menu *System – SNTP Client*, centang *Enabled*, *Primary NTP Server* diisi “192.168.222.2”, *Secondary NTP Server* diisi “192.168.222.3”, setelah selesai klik *OK*.

Pengujian Jaringan Akhir

Pengujian jaringan akhir merupakan pengujian jaringan usulan menggunakan *Server Otomatisasi Jaringan (Network Automation Server)* untuk update konfigurasi ke perangkat-perangkat terkait. Staf *engineer* perlu masuk ke *server* via *SSH* terlebih dahulu, lalu menjalankan *script python* yang sudah dibuat. Terdapat 2 (dua) *script* yang sudah dibuat untuk keperluan *provisioning* dan *update* konfigurasi masal, yaitu “*provisioning1.py*” dan “*mass-update-config1.py*”. Berikut merupakan langkah-langkah *provisioning* dan *mass-update configuration* dengan penerapan otomatisasi jaringan sesuai desain jaringan usulan: 1) Masuk ke *Server Otomatisasi Jaringan* via *SSH client*.

support@10.10.230.14's password:

Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-52-generic x86_64)

*Documentation: <https://help.ubuntu.com>

*Management : <https://landscape.canonical.com>

*Support : <https://ubuntu.com/advantage>

System information as of Tue 05 Jan 2021 02:11:05 PM UTC

System load : 0.0 Process : 208

Usage of / : 51.5% of 8.78 GB User logged in : 1

Memoru usage : 11% Ipv4 address for ens33 : 10.10.230.14

Swap usage : 0%

0 update can be installed immediately

0 of these updates are security updates.

Last login: Tue Jan 5 11:12:55 2021

Sebelum menjalankan *script python* perlu akses *root* terlebih dahulu dengan menjalankan perintah “*sudo su*” dan masukkan *password*. Untuk pekerjaan *provisioning* pelanggan baru, jalankan *script* dengan perintah “*python3 provisioning1.py*”. Contoh terdapat aktivasi pelanggan baru dengan nama *Customer-Z*, ID pelanggan Z00001-001 dan alokasi IP address 100.100.100.0/29. Welcome to *Network Automation for IP Provisioning Wireless* dengan *User Login*: ifadhilah login berhasil. Masukkan *CID* dan nama *customer* (contoh A00001-001_PT-ABC): Z00001-001_Customer-Z. Masukkan alokasi *IP gateway* dan *prefix length customer* (contoh 116.50.24.1/29). Masukkan *interface* (contoh vlan123): vlan125, sedang melakukan konfigurasi dan sukses *provisioning* IP Z0001-001_Customer-Z. Masukkan alokasi IP network dan *prefix length customer* (contoh 116.50.24.0/29): 100.100.100.0/29. Masukkan *bandwith internasional customer* (contoh 10M): 5M. Masukkan *bandwith lokal customer* (contoh 10M): 1M sedang melakukan konfigurasi lagi dan sukses mendaftarkan IP Z00001-001_Customer-Z di *Firewall* dan membuat *profile* di *Bandwith Management*. Untuk pekerjaan *update* konfigurasi secara masal ke perangkat-perangkat *router* eksisting, cukup

menjalankan *script* “*mass-update-config1.py*”, sebelumnya pada *script* perlu di-*define* IP perangkat-perangkat *router* pelanggan eksisting. Dilakukan sampel konfigurasi masal ke 10 (sepuluh) perangkat *router* eksisting, hanya dibutuhkan waktu sekitar 25 detik.

```
root@networkautomationvm: /home/support
root@networkautomationvm:/home/support# python3 mass-update-config1.py
-----
      SET SNMP, RADIUS, and NTP CONFIGURATION ON MIKROTIK
-----
User Login: support
Password:
Sukses konfigurasi SNMP, Radius, dan NTP pada 10.30.216.25
Sukses konfigurasi SNMP, Radius, dan NTP pada 10.30.216.29
Sukses konfigurasi SNMP, Radius, dan NTP pada 10.30.216.38
Sukses konfigurasi SNMP, Radius, dan NTP pada 10.30.216.41
Sukses konfigurasi SNMP, Radius, dan NTP pada 10.30.216.77
Sukses konfigurasi SNMP, Radius, dan NTP pada 10.30.216.110
Sukses konfigurasi SNMP, Radius, dan NTP pada 10.30.216.138
Sukses konfigurasi SNMP, Radius, dan NTP pada 10.30.216.141
Sukses konfigurasi SNMP, Radius, dan NTP pada 10.30.216.189
Sukses konfigurasi SNMP, Radius, dan NTP pada 10.30.216.190
root@networkautomationvm: /home/support#
```

Sumber: Hasil Penelitian (2021)

Gambar 5. *Mass-Update Config* dengan *Network Automation*

4. Kesimpulan

Sebelum diterapkan otomatisasi jaringan, semua konfigurasi di perangkat dilakukan secara manual, baik untuk pekerjaan provisioning maupun update konfigurasi perangkat secara masal. Konfigurasi awal atau provisioning internet pelanggan baru dapat dilakukan secara lebih cepat, juga meminimalisir terjadinya kesalahan konfigurasi karena human error dengan memanfaatkan otomatisasi jaringan dengan *script Python*. Perubahan konfigurasi secara masal ke perangkat-perangkat *router* eksisting yang berjumlah banyak apabila dilakukan secara manual akan memakan waktu sangat lama, namun dengan memanfaatkan otomatisasi jaringan dengan *script Python* dapat mempersingkat waktu pekerjaan secara signifikan. Setelah diterapkannya otomatisasi jaringan pada perangkat-perangkat *router* eksisting, maka akan tercipta manajemen jaringan yang lebih baik karena terjadi keseragaman dan ter-manage dengan baik di *tools Network Management System*.

Referensi

- [1] R. Komarudin, *Otomatisasi Administrasi Jaringan Dengan Script Python*. Jakarta: Jasakom, 2018.
- [2] M. R. Affandi, Puspanda Hatta, and Agus Efendi, “Otomatisasi Perangkat Jaringan Komputer Menggunakan Ansible Pada Laboratorium Komputer”, *SMARTICS*, vol. 6, no. 2, pp. 48-53, Oct. 2020.
- [3] E. S. Ginting and I. Hadi, “Penguujian Konfigurasi Otomatis Penambahan Gateway Pada Virtual Router Menggunakan Aplikasi Otomatisasi Jaringan Berbasis Web,” vol. 4, pp. 1126–1131, 2020.
- [4] K. Nugroho, A. D. Abrariansyah, and S. Ikhwan, “Perbandingan Kinerja Library Paramiko dan Netmiko Dalam Proses Otomasi Jaringan,” vol. 1, 2020.
- [5] K. Rianafirin and M. T. Kurniawan, “Design network security infrastructure cabling using network development life cycle methodology and ISO/IEC 27000 series in Yayasan Kesehatan (Yakes) Telkom Bandung,” *Proc. 2017 4th Int. Conf. Comput. Appl. Inf. Process. Technol. CAIPT 2017*, vol. 2018-Januari, pp. 1–6, 2018.
- [6] H.Sujadi, A.Mutaqin, "Rancang Bangun Arsitektur Jaringan Komputer Teknologi Metropolitan Area Network (MAN) Dengan Menggunakan Metode Network Development Life Cycle (NDLC)", *Jurnal J-Ensitec*, Vol.4, no.1, pp. 142-146, November 2017.

- [7] S. Taftazanie, A. B. Prasetijo, and E. D. Widiyanto, "Aplikasi Pemantau Perangkat Jaringan Berbasis Web Menggunakan Protokol SNMP dan Notifikasi SMS," *Jurnal Teknologi dan Sistem Komputer*, vol. 5, no. 2, pp. 62-68, Apr. 2017
- [8] R. Pradikta, A. Affandi, and E. Setijadi, "Rancang Bangun Aplikasi Monitoring Jaringan dengan Menggunakan Simple Network Management Protocol," *J. Tek. ITS*, vol. 2, no. 1, pp. A154--A159, 2013.
- [9] A. Michael, H. Hermawan, H. I. Pratiwi, "Sistem Monitoring Server Dengan Menggunakan
- [10] SNMP," *Jurnal Widyakala*, Vol. 6, no. 2, pp. 163-166, September 2019
- [11] 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.