

IMPLEMENTASI KONSEP *SOFTWARE DEFINED NETWORKING* (SDN) WIDE AREA NETWORK (WAN) PADA MIKROTIK DENGAN PYTHON 3

Sabar Hanadwiputra¹, Subandri^{1*}, Delyfan Prawinarko¹

¹Program Studi Teknik Informatika, Universitas Bani Saleh, sabar.hanadwiputra@gmail.com, andrisubandri@ymail.com, zekyaifan@gmail.com

ABSTRAK

Software Defined Networking (SDN) merupakan suatu teknologi jaringan terbaru untuk mengatasi tantangan yang ada pada TE. Konsep dari *Software Defined Networking* (SDN) menggunakan cara kerja secara terpusat (centralized control) dan lebih fleksibel, SDN secara signifikan dapat memangkas penggunaan sumber daya jaringan komunikasi yang dibutuhkan untuk kebermanfaatan sumber daya jaringan secara keseluruhan. Ketika perusahaan ingin mengembangkan jaringan komputernya ke area yang lebih luas, termasuk menghubungkan kantor cabang di remote area dengan data di kantor pusat dan kantor lainnya, perlu dilakukan setup pada puluhan hingga ratusan *Router* dan switch yang memakan waktu lama dan biaya mahal belum lagi bila terjadi kesalahan manusia (Human Error). Oleh karena itu diperlukan teknologi virtualisasi agar konfigurasi/setup pada banyak *Router* bisa dilakukan bersamaan untuk itu dengan mengadopsi konsep kerja *Software Defined Networking* (SDN) yang akan diterapkan pada teknologi Otomasi Jaringan. Nantinya Otomasi Jaringan ini akan diterapkan pada Mikrotik yang dipadu padankan dengan Linux Ubuntu Server dan juga menggunakan bahasa Python 3. Tujuan penelitian ini diharapkan mampu membantu dan mempermudah dalam penanganan konfigurasi/setup pada banyak *Router*. Sehingga dalam penanganan banyak *Router* tidak perlu kesulitan melakukan konfigurasi/setup satu persatu pada banyak *Router*. Hasil analisis implementasi konsep *Software Defined Networking* (SDN) pada teknologi Otomasi Jaringan adalah terkonfigurasinya *Router* Mikrotik Jogja dan Bandung secara bersamaan dalam hal penambahan IP Address.

Kata Kunci: Otomasi Jaringan, Mikrotik, Python 3, Linux Ubuntu Server, *Software Defined Networking* (SDN).

ABSTRACT

Software Defined Networking (SDN) is the latest network technology to address the challenges that exist in TE. The concept of *Software Defined Networking* (SDN) uses a centralized and more flexible way of working, SDN can significantly cut the use of communication network resources needed for the utilization of network resources as a whole. When a company wants to expand its computer network to a wider area, including connecting branch offices in remote areas with data at the head office and other offices, it is necessary to setup tens to hundreds of Routers and switches which take a long time and are expensive, not to mention if something goes wrong, human (Human Error). Therefore, virtualization technology is needed so that configuration/setup on many Routers can be done simultaneously for this reason by adopting the working concept of *Software Defined Networking* (SDN) which will be applied to Network Automation technology. Later this Network Automation will be applied to Mikrotik which is combined with Linux Ubuntu Server and also uses the Python 3 language. So that in handling many Routers you don't need to have trouble configuring/setting up one by one on many Routers. The result of the analysis of the implementation of the *Software Defined Networking* (SDN)

concept in Network Automation technology is that the Jogja and Bandung Mikrotik Routers are configured simultaneously in terms of adding IP addresses.

Keywords: *Otomasi Jaringan, Mikrotik, Python 3, Linux Ubuntu Server, Software Defined Networking (SDN).*

PENDAHULUAN

Pada Jurnal Ilmiah Ilmu Komputer (Estu Rizky Huddiniah, Eristya Maya Safitri, Satrio Adi Priyambodo, Muhammad Nasrullah, Nisa Dwi Angresti, menjelaskan bahwa “*Software Defined Networking (SDN)*, merupakan suatu teknologi jaringan terbaru untuk mengatasi tantangan yang ada pada TE. Dengan pengaturan secara terpusat (centralized control) dan lebih fleksible, *Software Defined Networking (SDN)*, secara signifikan dapat memangkas penggunaan sumber daya jaringan komunikasi yang dibutuhkan untuk kebermanfaatan sumber daya jaringan secara keseluruhan”.

Mikrotik memiliki perangkat keras sendiri yaitu salah satunya *RouterBoard*, perangkat ini bisa dijadikan PC atau komputer untuk *Router network* yang handal. Didalam *Router PC* terdapat banyak fitur unggulan seperti IP Network, jaringan wireless, *Firewall* dan Nat, Routing, Hotspot, Point to Point Tunneling Protocol, DNS server, DHCP server, Manajemen Bandwidth, Konfigurasi Keamanan dan masih banyak fitur lainnya.

Python dapat mengakses dukungan socket dasar dalam sistem operasi yang mendasarinya, yang memungkinkan untuk mengimplementasikan klien dan server untuk kedua protokol berorientasi koneksi dan tanpa sambungan. Python juga memiliki pustaka yang menyediakan akses tingkat lebih tinggi ke protokol jaringan tingkat aplikasi tertentu, seperti FTP, HTTP, dan seterusnya.

Ketika perusahaan ingin mengembangkan jaringan komputernya ke area yang lebih luas, termasuk menghubungkan kantor cabang di remote area dengan data di kantor pusat dan kantor lainnya, perlu dilakukan setup pada puluhan hingga ratusan *Router* dan switch yang memakan waktu lama dan biaya mahal belum lagi bila terjadi kesalahan manusia (Human Error). Oleh karena itu diperlukan teknologi virtualisasi agar konfigurasi/setup pada banyak *Router* bisa dilakukan bersamaan untuk itu dengan menerapkan teknologi Otomasi Jaringan ini bermanfaat untuk menghilangkan sebagian besar pekerjaan manual dan memakan waktu yang rentan kesalahan manusia.

Oleh karena itu tugas akhir ini akan membuat prototype implementasi Otomasi Jaringan dengan menerapkan metode Otomasi Jaringan pada mikrotik yang dipadukan dengan linux ubuntu server untuk mengolah bahasa python dan juga sebagai wadah konfigurasi otomatis yang ditujukan ke mikrotik.

Dengan diterapkannya Otomasi Jaringan ini penulis berharap dapat membantu dan mempermudah kinerja para praktisi IT dalam membangun jaringan secara jarak jauh, sehingga dapat mengurangi kesalahan manusia, biaya operasional, dan meningkatkan efisien, untuk memenuhi penelitian tersebut penulis mengangkat tema skripsi dengan judul “Implementasi Konsep *Software Defined Networking (SDN) Wide Area Network (WAN)* Pada Mikrotik Dengan Python 3”.

TINJAUAN PUSTAKA

Tinjauan Penelitian diambil dari beberapa artikel dan jurnal yang terkait oleh masalah yang diambil penulis untuk menjadikan suatu referensi pada penelitian ini.

Software Defined Networking (SDN) *Software Defined Networking (SDN)* merupakan komponen perangkat lunak untuk mengontrol operasi jaringan. (Yang et al., 2019) Perangkat lunak manajemen khusus memvirtualisasikan perangkat keras jaringan dengan cara yang sama seperti hypervisors dan komponen lain memvirtualisasikan operasi pusat data. *Software Defined Networking (SDN)* merupakan sebuah teknik untuk menggunakan perangkat lunak untuk membuat jaringan area luas lebih cerdas dan fleksibel.

Software Defined Networking (SDN) dapat membantu jaringan area luas untuk menangani lalu lintas jaringan dengan protokol tertentu, serta menyediakan antarmuka yang intuitif bagi pengguna. Tidak hanya itu, *Software Defined Networking (SDN)* juga dapat mendukung fitur seperti *Firewall*, gateway, dan alat jaringan pribadi virtual untuk privasi. Selain itu, *Software Defined Networking (SDN)* dapat membantu redundansi, percadangan dan pemulihan, serta pemecahan masalah. Otomasi Jaringan (*Network Automation*) *Network Automation* atau Otomasi Jaringan adalah proses mengotomatiskan konfigurasi, manajemen, dan operasi jaringan komputer. Ini adalah istilah luas yang mencakup sejumlah alat, teknologi, dan metodologi yang digunakan untuk mengotomatiskan proses jaringan (Toschi, Campos and Cugnasca, 2017).

Otomatisasi jaringan diimplementasikan melalui kombinasi solusi berbasis perangkat keras dan perangkat lunak yang secara otomatis menjalankan dan mengelola proses lingkungan jaringan yang berulang. Biasanya proses digunakan oleh perusahaan dan penyedia layanan atau service provider untuk meningkatkan efisiensi dan mengurangi kesalahan manusia dan biaya operasi. *Network Automation* memainkan peran kunci dalam network virtualization, network orchestration, dan juga jaringan yang ditentukan oleh perangkat lunak *Software Defined Networking (SDN)*.

MikroTik adalah perusahaan Latvia yang didirikan pada tahun 1996 untuk mengembangkan *Router* dan sistem ISP nirkabel (Asnawi, 2018). Secara umum mikrotik merupakan sistem operasi dan perangkat lunak yang bisa digunakan untuk menjadikan sebuah PC atau komputer untuk *Router* network yang handal. Didalam *Routernya* tadi, PC juga bisa mencakup banyak fitur unggulan seperti IP Network jaringan wireless, provider warnet dan hotspot. Jadi, mikrotik benar-benar membantu para penggunanya. Mikrotik memang didesain khusus untuk memudahkan berbagai keperluan jaringan komputer.

Linux, pada tahun 1969, Ken Thompson dan Dennis Ritchie (juga adalah developer bahasa C), para peneliti di AT&T Bell Laboratorium Amerika, membuat sistem operasi UNIX, cikal bakal dari Linux. UNIX mendapatkan perhatian besar karena merupakan sistem operasi pertama yang dibuat bukan oleh hardware maker. Selain itu juga karena seluruh source code-nya dibuat dengan bahasa C, sehingga mempermudah pemindahannya ke berbagai platform (Rusmato, 2016). Linux adalah suatu sistem operasi yang bersifat multi user dan multitasking, yang dapat berjalan di berbagai platform, termasuk prosesor INTEL 386 dan yang lebih tinggi. Sistem operasi ini mengimplementasikan standard POSIX. Linux dapat berinteroperasi secara baik

dengan sistem operasi yang lain, termasuk Apple, Microsoft dan Novell. Bahasa Pemrograman Python Bahasa pemrograman Python (Lutz, 2007) adalah bahasa pemrograman yang dibuat oleh Guido van Rossum dari Amsterdam, Belanda. Pada awalnya, motivasi pembuatan bahasa pemrograman ini adalah untuk bahasa skrip tingkat tinggi pada sistem operasi terdistribusi Amoeba. Bahasa pemrograman ini menjadi umum digunakan untuk kalangan engineer seluruh dunia dalam pembuatan perangkat lunaknya, bahkan beberapa perusahaan seperti Google, NASA, Instagram, YouTube, dan Spotify menggunakan Python sebagai pembuat perangkat lunak komersial. Python banyak digunakan untuk membuat berbagai macam program, seperti program CLI, Program GUI (desktop), Aplikasi Mobile, Web, IoT, Game, Program untuk Hacking, dan sebagainya.

Paramiko adalah implementasi Python (2.7, 3.4+) dari protokol SSHv2, menyediakan fungsionalitas klien dan server. Meskipun memanfaatkan ekstensi Python C untuk kriptografi tingkat rendah (Kriptografi), Paramiko sendiri adalah antarmuka Python murni di sekitar konsep jaringan SSH (Wiryanan and Rosyid, 2019). Virtual Box, Oracle VM VirtualBox (Larosa, 2016) adalah aplikasi virtualisasi lintas platform. Aplikasi ini diinstal pada komputer berbasis Intel atau AMD yang sudah ada. Aplikasi ini menjalankan sistem operasi (OS) Windows, Mac OS X, Linux, atau Oracle Solaris dan juga memperluas kemampuan komputer agar dapat menjalankan banyak OS, di dalam beberapa mesin virtual, pada waktu yang bersamaan. Sebagai contoh, ketika menjalankan Windows dan Linux di Mac, menjalankan Windows Server 2016 di server Linux, menjalankan Linux di PC Windows, dan seterusnya, semua di samping aplikasi. Aplikais ini memungkinkan untuk dapat menginstal dan menjalankan mesin virtual sebanyak mungkin. Satu-satunya batasan praktis adalah ruang disk dan memori.

Putty adalah klien SSH dan telnet, awalnya dikembangkan oleh Simon Tatham untuk platform Windows (Simon Tatham, 2016). Putty adalah perangkat lunak sumber terbuka yang tersedia dengan kode sumber dan dikembangkan serta didukung oleh sekelompok sukarelawan.

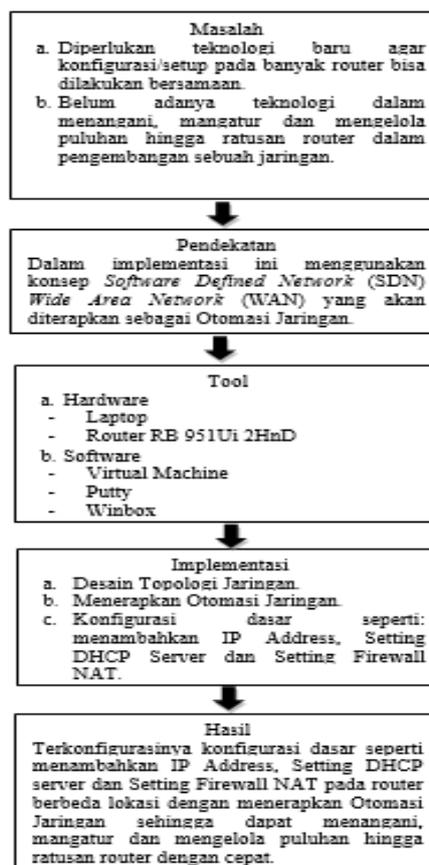
Perutean IP adalah proses memindahkan paket data antara jaringan yang berbeda. Secara default, dua jaringan IP yang berbeda tidak dapat berkomunikasi satu sama lain. Dua jaringan IP yang berbeda membutuhkan perangkat mediator yang dapat mengalihkan paket di antara Dua jaringan IP berbeda tersebut. *Router* mengambil tanggung jawab ini. Antarmuka *Router* dikaitkan dengan jaringan yang berbeda. Asosiasi ini disimpan dalam tabel routing. *Router* menggunakannya untuk mengambil keputusan pengalihan (Li et al., 2020).

Perutean Statis, dalam perutean statis kita harus menambahkan semua lokasi jaringan secara manual. Jika terjadi perubahan dalam jaringan, administrator bertanggung jawab untuk memperbaruinya secara manual di semua *Router*. Keuntungan dari perutean statis yaitu: mudah di implementasikan, tidak ada biaya tambahan pada sumber daya seperti CPU atau memori dan lain-lain. Sedangkan kerugiannya yaitu: hanya cocok untuk jaringan skala kecil, jika link gagal, rute statis tidak dapat mengubah

rute lalu lintas dan lain-lain. Perutean Dinamis, dalam *Router* perutean dinamis menambahkan lokasi jaringan secara otomatis membentuk informasi perutean. Jika ada perubahan yang terjadi dalam jaringan, *Router* yang terpengaruh memperbarui yang lain melalui informasi perutean. Keuntungan dari perutean dinamis yaitu: cocok untuk semua jenis jaringan, buat tabel perutean secara otomatis dan lain-lain. Sedangkan kerugiannya yaitu: sulit untuk diterapkan, kurang aman karena berbagi pembaruan perutean dengan *Router* lain dan sebagainya.

Sistem Otonom, Autonomous System (AS) adalah kumpulan *Router* yang berbagi informasi tabel routing yang sama. AS adalah garis batas untuk protokol routing. Untuk membedakan antara AS yang berbeda, Internet Assigned Numbers Authority (IANA) menyediakan rentang dari 1 hingga 65535. Ada dua jenis AS, privat dan publik. Nomor AS pribadi digunakan untuk jaringan internal. Nomor AS publik digunakan untuk tulang punggung internet.

Wide Area Network (WAN) merupakan jaringan komputer yang mencakup area yang besar sebagai contoh yaitu jaringan komputer antar wilayah, kota atau bahkan negara, atau dapat didefinisikan juga sebagai jaringan komputer yang membutuhkan *Router* dan saluran komunikasi public. Dalam membangun jaringan Wide Area Network (WAN) diperlukan beberapa jenis alat antara lain: Network Interfaced Card (NIC), Repeater, Hub/Switch, Bridge, *Router*, Modem, Kabel Jaringan.



Gambar 1. Kerangka Pemikiran

METODE PENELITIAN

Teknik pengumpulan data yang digunakan adalah studi pustaka dalam menganalisa kebutuhan yang meliputi teknik pengumpulan data, metode penelitian, kebutuhan perangkat keras (*hardware*) disajikan pada tabel 1 dan 2 dan kebutuhan perangkat lunak (*software*).

Tabel 1. Spesifikasi Laptop Hp EliteBook 2570p

Nama	<i>Hp EliteBook 2570p</i>
Cpu	<i>Processor Intel Core i5 3320M speed 2,6 Ghz</i>
Gpu	<i>Intel HD Graphics 40000</i>
Storage	<i>SSD SATA 128 GB</i>
RAM	<i>DDR3 8 GB</i>

Dalam tabel 1 dan 2 merupakan spesifikasi kebutuhan perangkat keras (*hardware*) yang dibutuhkan sehingga penelitian dapat berjalan dengan baik.

Tabel 2. Spesifikasi RouterBoard 951Ui 2HnD

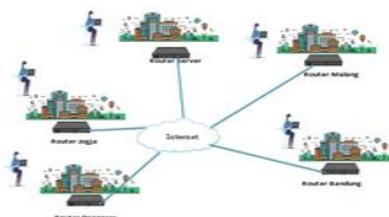
Nama	<i>RouterBoard 951Ui 2HnD</i>
CPU	<i>AR9344 600 MHz</i>
RAM	<i>128 MB</i>

Perangkat lunak yang dibutuhkan yaitu: *Winbox, Virtual Machine, Sistem Operasi (Windows 10 dan Ubuntu Server 16.04)*. Dalam penelitian ini penulis menggunakan Metode PPDIIO (*Prepare, Plan, Design, Implement, Operate, Optimize*), sebagai metode pengembangan sistemnya. Dengan metode ini dapat mempresentasikan apa yang dibutuhkan dalam penelitian.



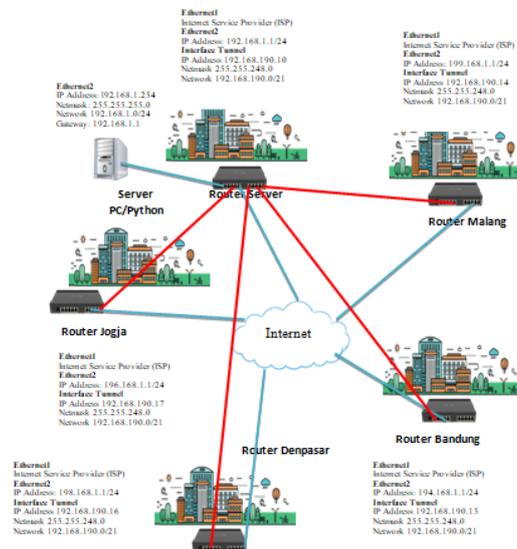
Gambar 2. PPDIIO

Pada tahap perencanaan melakukan persiapan dengan menganalisis kebutuhan jaringan yang akan dibangun dalam implementasi konsep kerja Software Defined Networking (SDN) yang akan dituangkan pada Otomasi Jaringan. Dalam tahap ini dibutuhkan perangkat keras dan lunak.



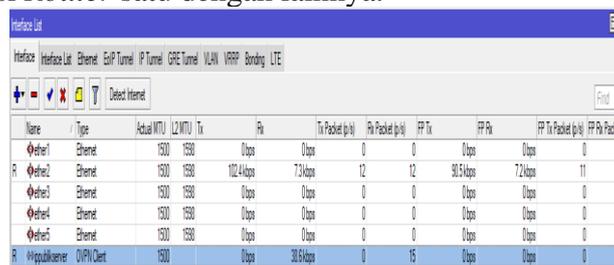
Gambar 3. Contoh Topologi

Pada tahap *design* ini akan membuat gambar *design* topologi jaringan interkoneksi yang akan di bangun dengan mengolah data-data yang diperoleh sebelumnya, diharapkan dengan gambar ini akan memberikan gambaran seutuhnya dari kebutuhan yang ada. *Design* yang akan digunakan pada tulisan ini merupakan *design* struktur topologi yang akan memberikan gambaran jelas tentang implementasi konsep kerja Software Defined Networking (SDN) pada Otomasi Jaringan dibangun.



Gambar 4. Rancangan Topologi

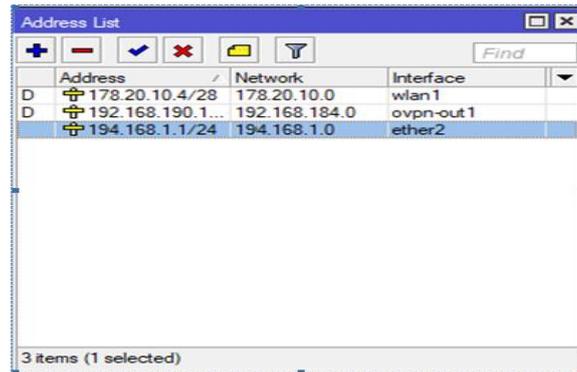
Pada fase implementasi, peralatan-peralatan baru dilakukan instalasi dan di konfigurasi, sesuai spesifikasi desain. Berikut langkah implementasinya: a) Install Virtual Machine pada Windows 10, untuk melakukan konfigurasi pada server python perlu dilakukan pada ubuntu server tetapi sebelum itu install terlebih dahulu Virtual Machine. Disini penulis menggunakan Virtual Machine Oracle VM VirtualBox 6.1.14. b) Install Sistem Operasi Ubuntu Server 16.04 pada Virtual Machine Oracle VM VirtualBox 6.1.14. c) Instal Putty pada Windows 10. d) Memberikan Akses Internet pada Server PC melalui *Router* Server. e) Konfigurasi OpenVPN pada *Router* Server, Jogja, Bandung, Malang dan Denpasar. OpenVPN ini berfungsi sebagai penghubung antara *Router* Jogja, Bandung, Malang dan Denpasar menggunakan IP Tunnel agar dapat berkomunikasi *Router* satu dengan lainnya.



Name	Type	Actual MTU	L2 MTU	Tx	Rx	Tx Packet (p/s)	Rx Packet (p/s)	PP Tx	PP Rx	PP Tx Packet (p/s)	PP Rx Packet
eth1	Ethernet	1500	1500	0bps	0bps	0	0	0bps	0bps	0	0
eth2	Ethernet	1500	1500	102.4kbps	7.3kbps	12	12	95.5kbps	7.2kbps	11	11
eth3	Ethernet	1500	1500	0bps	0bps	0	0	0bps	0bps	0	0
eth4	Ethernet	1500	1500	0bps	0bps	0	0	0bps	0bps	0	0
eth5	Ethernet	1500	1500	0bps	0bps	0	0	0bps	0bps	0	0
OpenVPN Client	OpenVPN Client	1500		0bps	30.6kbps	0	15	0bps	0bps	0	0

Gambar 5. OVPN berhasil ditambahkan

Pada tahap *operate* akan di lakukan pengujian sistem Otomasi Jaringan yang telah di buat dan akan dievaluasi apakah sistem yang dibangun berjalan dengan baik atau tidak.

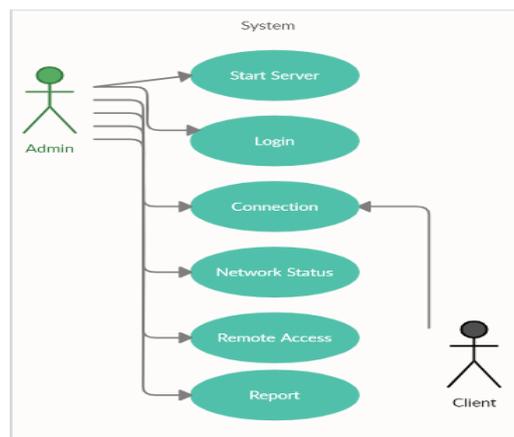


Gambar 6. Hasil Tampilan Penambahan IP Address di *Router* Client secara otomatis

Pada tahapan optimasi ini akan melakukan perubahan dari teknologi lama ke teknologi baru dimana dalam konfigurasi *Router* tidak perlu dilakukan satu-persatu melainkan konfigurasi secara terpusat. Jika sistem yang baru dirancang ternyata tidak sesuai dan menimbulkan banyak error/permasalahan maka diperlukan merancang kembali sistem. Syarat – syarat yang dimodifikasi ulang mengarah kepada awal siklus metode PPDIOO.

HASIL DAN PEMBAHASAN

Setelah melakukan rancangan dan impementasi pada sistem yang di bangun, selanjutnya akan membahas mengenai tahap uji *testing* terhadap *system*, apakah *system* yang telah di bangun sesuai dengan yang di butuhkan. Diharapkan hasilnya sesuai dengan apa yang diharapkan oleh peneliti dan bermanfaat bagi setiap pengguna. Perancangan penelitian yang dilakukan disajikan pada gambar 7 Use Case sebagai berikut:



Gambar 7. Use Case Perancangan Admin dan *Client*

Implementasi Otomasi Jaringan pada Server PC untuk konfigurasi *Router* Jogja, Bandung, Malang dan Denpasar. Pastikan Server PC sudah bisa akses internet disajikan pada gambar 8.

```
root@ubuntu:/home/ubuntu/Python-MikrotikAutomation# ifconfig
ethp0s3  Link encap:Ethernet  HWaddr 08:00:27:70:46:dc
         inet addr:192.168.1.253  Bcast:192.168.1.255  Mask:255.255.0
         inet6 addr: fe80::a00:27ff:fe70:46dc/64 Scope:Link
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:104949 errors:0 dropped:0 overruns:0 frame:0
         TX packets:623 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:9025010 (9.0 MB)  TX bytes:74102 (74.1 KB)

lo       Link encap:Local Loopback
         inet addr:127.0.0.1  Mask:255.0.0.0
         inet6 addr: ::1/128 Scope:Host
         UP LOOPBACK RUNNING  MTU:65536  Metric:1
         RX packets:160 errors:0 dropped:0 overruns:0 frame:0
         TX packets:160 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1
         RX bytes:11840 (11.8 KB)  TX bytes:11840 (11.8 KB)
```

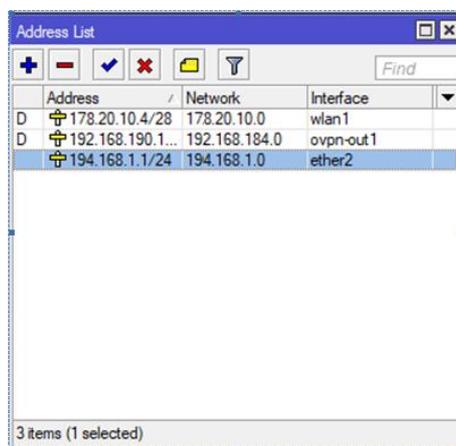
Gambar 8. Melihat IP pada Ubuntu Server

Terlihat pada gambar 8 Ubuntu Server mendapatkan IP secara otomatis dari Router Server 192.168.1.253. Lakukan konfigurasi untuk Router Jogja, Bandung, Malang dan Denpasar melalui Server PC disajikan pada gambar 9.

```
root@ubuntu:/home/ubuntu/Python-MikrotikAutomation# python3 skrip.py
MIKROTIK AUTOMATION
Masukan File IP      : ip.txt
Masukan File Konfigurasi : konfigurasi.txt
Masukan Username    : admin
Masukan Password    :
/usr/lib/python3/dist-packages/Crypto/Cipher/blockalgo.py:141: FutureWarning: CTR mode needs counter
parameters, not IV
  self._cipher = factory.new(key, mode, **kwargs)
Berhasil Login 192.168.190.14
Konfigurasi Sukses 192.168.190.14
Berhasil Login 192.168.190.15
Konfigurasi Sukses 192.168.190.15
Berhasil Login 192.168.190.16
Konfigurasi Sukses 192.168.190.16
Berhasil Login 192.168.190.17
Konfigurasi Sukses 192.168.190.17
Selesai
root@ubuntu:/home/ubuntu/Python-MikrotikAutomation#
```

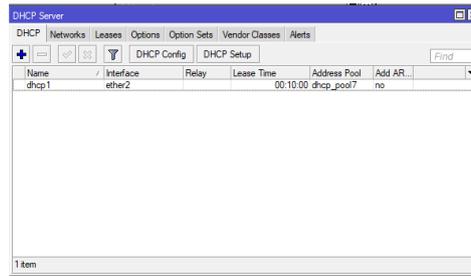
Gambar 9. Konfigurasi Router Mikrotik pada Server PC

Pada gambar 9 dilakukan konfigurasi keempat Router Mikrotik seperti, penambahan IP Address, Setting DHCP Server dan Setting Firewall. Skrip konfigurasi Router Mikrotik sudah di input pada “konfigurasi.txt” Skrip “konfigurasi.txt” secara otomatis akan bekerja sesuai dengan IP Tunnel dari masing Router Mikrotik yang sudah diinput di “ip.txt”. Terlihat pada gambar 10 bahwa konfigurasi untuk keempat Router Mikrotik berhasil semua.



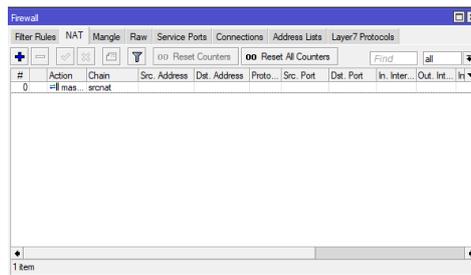
Gambar 10. Hasil penambahan IP pada Router Bandung

Pada gambar 10 penambahan IP secara otomatis pada ethernet 2 Router Bandung berhasil, terlihat IP yang terinput 194.168.1.1/24 dengan network 194.168.1.0/24.



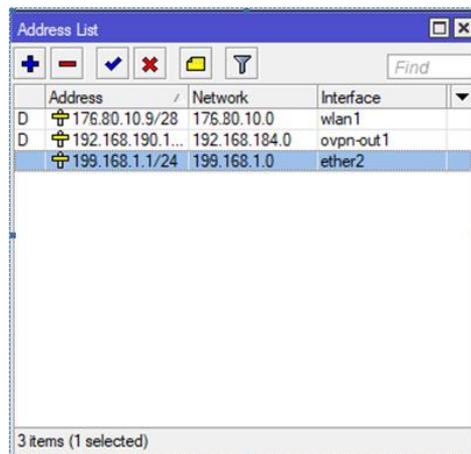
Gambar 11. Hasil *Setting* DHCP Server pada *Router* Jogja

Pada gambar 11 *Setting* DHCP Server secara otomatis pada *Router* Jogja berhasil terlihat pada menu DHCP sudah terinput dhcp1.



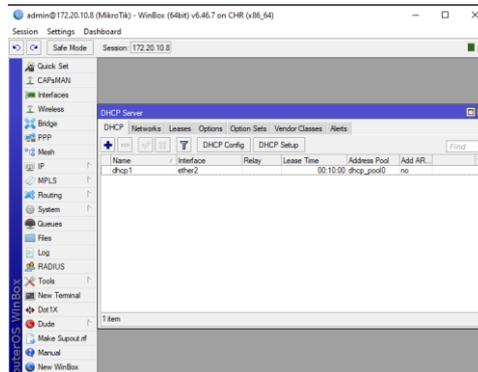
Gambar 12. Hasil *Setting* Firewall NAT pada *Router* Jogja

Pada gambar 12 *Setting* Firewall NAT secara otomatis pada *Router* Jogja Berhasil terlihat pada menu NAT sudah teridentifikasi.



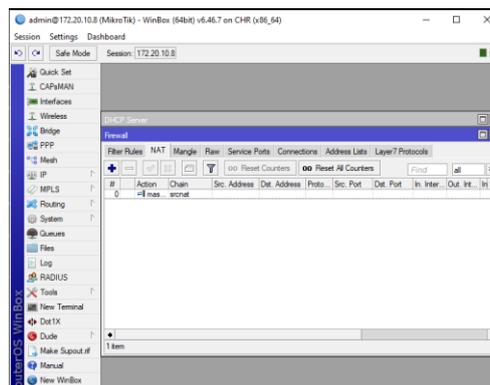
Gambar 13. Hasil penambahan IP pada *Router* Malang

Pada gambar 13 penambahan IP secara otomatis pada ethernet 2 *Router* Malang berhasil, terlihat IP yang terinput 199.168.1.1/24 dengan network 199.168.1.0/24.



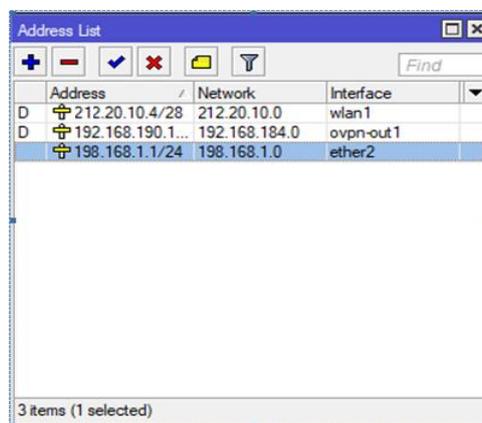
Gambar 14. Hasil *Setting* DHCP Server pada *Router* Malang

Pada gambar 14 *Setting* DHCP Server secara otomatis pada *Router* Malang berhasil terlihat pada menu DHCP sudah terimput dhcp1.



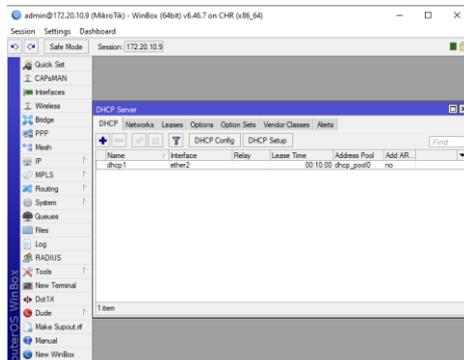
Gambar 15. Hasil *Setting* Firewall NAT pada *Router* Malang

Pada gambar 15 *Setting* Firewall NAT secara otomatis pada *Router* Malang Berhasil terlihat pada menu NAT sudah teridentifikasi.



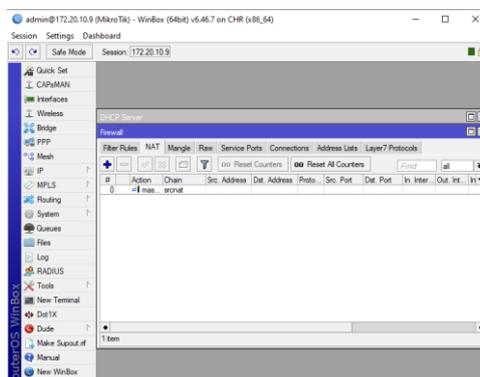
Gambar 16. Hasil penambahan IP pada *Router* Denpasar

Pada gambar 16 penambahan IP secara otomatis pada ethernet 2 *Router* Malang berhasil, terlihat IP yang terimput 198.168.1.1/24 dengan network 198.168.1.0/24.



Gambar 17. Hasil *Setting* DHCP Server pada *Router* Denpasar

Pada gambar 17 *Setting* DHCP Server secara otomatis pada *Router* Denpasar berhasil terlihat pada menu DHCP sudah terinput dhcp1.



Gambar 18. Hasil *Setting* Firewall NAT pada *Router* Denpasar

Pada gambar 18 *Setting* Firewall NAT secara otomatis pada *Router* Denpasar Berhasil terlihat pada menu NAT sudah teridentifikasi.

Setelah melakukan Uji Coba dalam pembahasan ini akan membandingkan konfigurasi manual dan konfigurasi secara otomatis berjalan dengan baik atau tidak. Tabel di bawah ini akan menunjukkan hasil dari konfigurasi manual dan konfigurasi otomatis.

Tabel 3. Tabel Perbandingan

No.	Cara	Media	Setup yang dilakukan	Tingkat Keberhasilan
1	Konfigurasi Manual	Microtic RB 951Ui 2HnD	Penambahan IP Address	Sukses
	Winbox	<i>Setting</i> DHCP Server	<i>Setting</i> DHCP Server	Sukses
2	Konfigurasi Otomatis	Microtic RB 951Ui 2HnD	Penambahan IP Address	Sukses
	Winbox	<i>Setting</i> Firewall NAT	<i>Setting</i> Firewall NAT	Sukses

No.	Cara	Media	Setup yang dilakukan	Tingkat Keberhasilan
	Server PC (Python)	Setting Firewall NAT	Setting Firewall NAT	Sukses

Berdasarkan dari perbandingan hasil setup antara konfigurasi manual dan konfigurasi otomatis terlihat tingkat keberhasilan sukses semua, karena skrip setup pada konfigurasi manual dipakai pada konfigurasi otomatis yang diinput pada skrip Python. Dengan adanya teknologi Otomasi Jaringan ini diharapkan ketika skrip konfigurasi lebih lengkap tidak hanya penambahan IP, *Setting DHCP Server* dan *Setting Firewall* saja yang bisa dilakukan melainkan seluruh konfigurasi pada mikrotik secara manual bisa dikkonfigurasi secara otomatis.

PENUTUP

Berdasarkan penelitian yang di lakukan penulis dapat mengambil kesimpulan sebagai berikut: dengan adanya Teknologi Otomasi Jaringan ini memudahkan dalam konfigurasi secara bersamaan pada banyak *Router* Mikrotik secara jarak jauh dan dengan adanya Teknologi Otomasi Jaringan ini memudahkan dalam dalam menangani, mengatur dan mengelola puluhan hingga ratusan *Router* dalam pengembangan sebuah jaringan. Penelitian ini tentunya masih memiliki beberapa kekurangan. Dari hasil penelitian penulis memberikan beberapa saran agar tahap selanjutnya bisa menjadi evaluasi Teknologi Otomasi Jaringan. Teknologi Otomasi Jaringan diharapkan bisa melakukan berbagai macam konfigurasi-konfigurasi yang dibutuhkan dalam pembangunan atau pengembangan jaringan. Pada pembangunan atau pengembangan selanjutnya diharapkan Teknologi Otomasi jaringan ini tidak hanya dapat diimplementasikan pada *Virtual Machine* melainkan bisa implementasikan pada diperangkat keras jaringan.

REFERENSI

- Asnawi, M. F. (2018) 'Aplikasi Konfigurasi Mikrotik Sebagai Manajemen Bandwidth Dan Internet Gateway Berbasis Web', *Jurnal Penelitian dan Pengabdian Kepada Masyarakat UNSIQ*. doi: 10.32699/ppkm.v5i1.437.
- Huddiniah, E. R. et al. (2018) 'Optimasi Rute Untuk Software Defined Networking-Wide Area Network (SDN-WAN) Dengan Openflow Protocol', *Informatika Mulawarman: Jurnal Ilmiah Ilmu Komputer*, 13(1), p. 7. doi: 10.30872/jim.v13i1.1006.
- Larosa, F. G. N. (2016) 'Pemanfaatan Virtual Box Dalam Praktikum Administrasi Server', *Pemanfaatan Virtual Box Dalam Praktikum Administrasi Server Menggunakan Teknik Dhcp Pada Mikrotik Router Os*.
- Li, Y. et al. (2020) 'Learning dynamic routing for semantic segmentation', in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. doi: 10.1109/CVPR42600.2020.00858.



-
- Lutz, M. (2007) Learning Python, Icarus. doi: 10.1016/0019-1035(89)90077-8.
- Marzuqi, O., Virgono, A. and Negara, R. M. (2019) 'Implementation model architecture Software Defined Networking (SDN) Wide Area Network (WAN) using raspberry Pi: A review paper', *Telkomnika (Telecommunication Computing Electronics and Control)*, 17(3), pp. 1136–1141. doi: 10.12928/TELKOMNIKA.V17I3.8859.
- Rusmato (2016) Sejarah Dan Perkembangan Linux, Linux.or.id.
- Tatham, S., (2016) 'PuTTY: a free SSH and Telnet client', Simon Tatham.
- Toschi, G. M., Campos, L. B. and Cugnasca, C. E. (2017) 'Home automation networks: A survey', *Computer Standards and Interfaces*. doi: 10.1016/j.csi.2016.08.008.
- Tsai, C. C. et al. (2016) 'A study of modern Linux API usage and compatibility: What to support when you're supporting', in *Proceedings of the 11th European Conference on Computer Systems, EuroSys 2016*. doi: 10.1145/2901318.2901341.
- Wiryawan, R. A. and Rosyid, N. R. (2019) 'Pengembangan Aplikasi Otomatisasi Administrasi Jaringan Berbasis Website Menggunakan Bahasa Pemrograman Python', *Simetris: Jurnal Teknik Mesin, Elektro dan Ilmu Komputer*.
- Yang, Z. et al. (2019) 'Software-defined wide area network (Software Defined Network): architecture, advances and opportunities', in *Proceedings - International Conference on Computer Communications and Networks, ICCCN*. doi: 10.1109/ICCCN.2019.8847124.