IMPLEMENTASI NAT IPv4 DENGAN NETWORK AUTOMATION PADA CISCO ROUTER

[1]Ragiel Hadi Prayitno
[2]Bayu Kumoro Yakti

[1] Universitas Gunadarma, ragielhp@staff.gunadarma.ac.id
[2]Universitas Gunadarma, bayuyakti@staff.gunadarma.ac.id

ABSTRAK

Perkembangan teknologi khususnya dalam bidang komunikasi melalui jaringan komputer mengalami peningkatan yang sangat pesat. Komunikasi antar komputer dapat dilakukan dengan mudah dan cepat sehingga dapat meningkatkan efektifitas dan efisiensi yang menghasilkan produktifitas yang tinggi. Perusahaan – perusahaan yang bergerak dalam bidang komunikasi memiliki banyak perangkat jaringan yang saling terintegrasi antara jaringan lokal dan jaringan internet. Untuk dapat mengintegrasikan jaringan lokal dengan internet dibutuhkan services NAT, sedangkan dengan banyaknya perangkat jaringan dapat memakan waktu yang lama pada proses konfigurasi dan perawatan perangkat. Network automation merupakan salah satu solusi untuk menghindari masalah tersebut. Penelitian ini membuat sebuah network automation untuk konfigurasi ip address, dhcp server dan NAT. Perancangan aplikasi menggunakan metode waterfall dan bahasa pemrograman python. Sedangkan untuk implementasi sistem aplikasi menggunakan simulator GNS3. Hasil dari ujicoba sistem aplikasi menunjukkan bahwa aplikasi yang telah dibuat dapat berfungsi sesuai dengan skenario yang ada yaitu jaringan lokal dapat berkomunikasi dengan jaringan internet.
Kata kunci: Cisco Router, GNS3, Python, Network Automation

PENDAHULUAN

Perkembangan teknologi dalam satu dekade ini mengalami peningkatan yang sangat pesat, khususnya dalam bidang komunikasi melalui jaringan komputer. Komunikasi antar komputer dapat dilakukan dengan mudah dan cepat sehingga dapat meningkatkan efektifitas dan efisiensi yang menghasilkan produktifitas yang tinggi. Saat ini, perusahaan – perusahaan yang bergerak dalam bidang komunikasi memiliki banyak perangkat jaringan yang saling terintegrasi antara jaringan lokal dan jaringan internet. Hal tersebut menjadi tantangan tersendiri untuk perusahaan dalam melakukan konfigurasi dan perawatan perangkat jaringan. Metode tradisional dengan melakukan proses remote tiap perangkat akan memakan waktu yang lama. Otomatisasi jaringan merupakan solusi untuk melakukan pekerjaan-pekerjaan yang rumit tersebut dan dapat diimpelementasikan ke perangkat yang mendukung protocol SSH sehingga pekerjaan bisa diselesaikan jauh lebih cepat dan juga efisien dalam pemeliharaan jaringan dengan prosedur yang lebih mudah diikuti dan diimplementasikan di dalam jaringan berskala besar (Wiryawan & Rosyid, 2019).

Pada penelitian Wijaya dan Silviana. (2020) menghasilkan sebuah aplikasi otomatisasi jaringan pada perangkat cisco dan mikrotik untuk melakukan konfigurasi IP Address, DHCP dan Routing. Pada penelitian tersebut perancangan sistem dibuat dengan menggunakan Bahasa pemrograman python.

Pada penelitian Wiryawan dan Rosyid (2019) menghasilkan sebuah

pengembangan aplikasi otomatisasi administrasi jaringan berbasis website dengan lima buah fitur yaitu 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 dilakukan secara terpusat sehingga akan lebih termanajemen lebih baik. Metode Black-box Testing sebagai pengujian aplikasi menunjukkan bahwa semua fungsi pada aplikasi yang dikembangkan pada penelitian ini berfungsi dengan baik dan berhasil diterapkan pada vendor yang berbeda yaitu Cisco dan Mikrotik.

Pada penelitian Wijaya. (2018) menghasilkan metode dalam mengkonfigurasi perangkat jaringan dengan menggunakan otomatisasi, mengurangi waktu untuk konfigurasi peralatan dan perawatan yang lebih mudah.

Berdasarkan penelitian – penelitian terdahulu, penelitian ini bertujuan untuk membuat sebuah perancangan *network automation* khususnya dalam melakukan konfigurasi NAT (*Network Address Translation)* pada router cisco. Hal tersebut didasarkan pada kebutuhan perusahaan agar dapat saling terintegrasi antara jaringan lokal dan jaringan internet, maka peneliti memilih untuk fokus pada konfigurasi NAT. NAT itu sendiri berfungsi untuk menterjemahkan alamat ip jaringan lokal (*private*) menjadi alamat ip *public* sehingga jaringan lokal dapat berkomunikasi dengan jaringan internet.

METODE PENELITIAN

Penelitian ini dimulai dari studi literatur pada beberapa jurnal terkait sebagai sumber referensi penelitian. Perancangan sistem menggunakan metode waterfall, dimana metode tersebut melakukan pendekatan secara sistematis dan terurut dalam membangun suatu sistem. Adapun tahapan dalam metode waterfall pada penelitian ini adalah studi literatur, Analisa kebutuhan, perancangan topologi, perancangan sistem dan implementasi sistem. Fase dari metode waterfall dapat dilihat pada gambar 1.

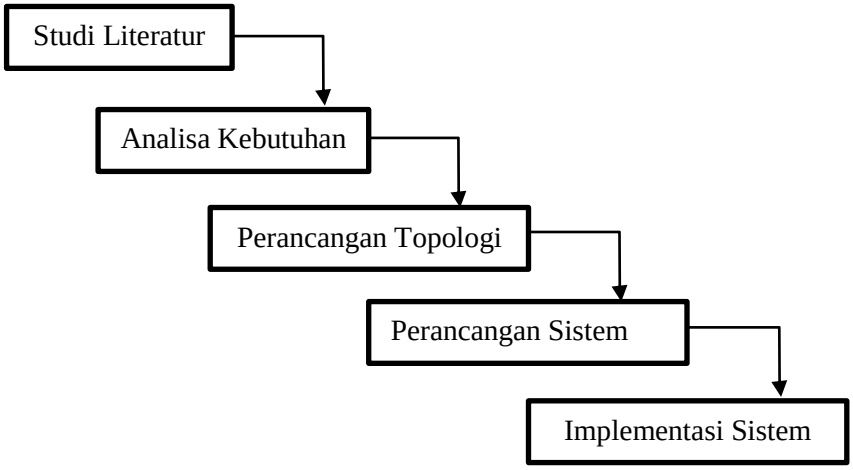
Pada tahap studi literatur, peneliti mencari beberapa referensi jurnal yang dapat mendukung dalam proses penelitian. Pada tahap ini, peneliti melakukan analisis terhadap penelitian terdahulu untuk menentukan arah dari penelitian.

Tahap Analisa kebutuhan, peneliti mengumpulkan data perangkat lunak dan perangkat keras yang diperlukan dalam proses penelitian. Adapun perangkat lunak yang dibutuhkan antara lain : Python 3, intepreter Python 3, notepad ++, dan GNS3. Sedangkan perangkat keras yang dibutuhkan antara lain : PC/Laptop.

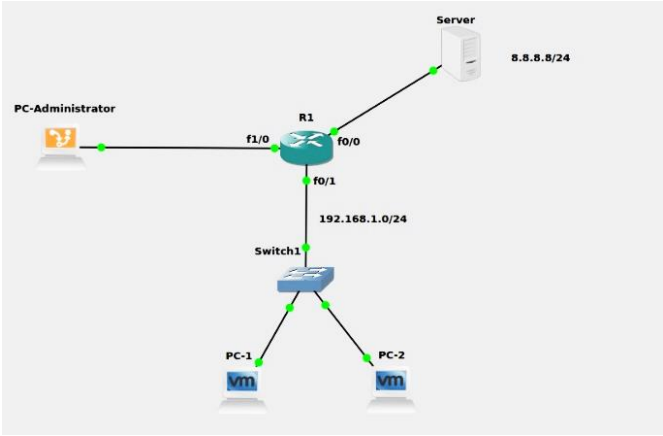
Tahap perancangan topologi, peneliti menggunakan perangkat lunak GNS3 sebagai *tools* simulasi jaringan. Adapun komponen yang digunakan antara lain : 1 buah router cisco, 1 buah switch, 1 buah server, 1 buah pc adiministrator untuk menjalankan aplikasi *network automation* dan 2 buah pc *client*. Berikut ini merupakan topologi yang digunakan pada penelitian.

Tahap perancangan sistem, peneliti membuat sebuah desain aplikasi yang akan menjadi interface antara admin jaringan (user) dengan program. Perancangan sistem dalam penelitian ini menggunakan Bahasa pemrograman python 3 dan notepad ++ sebagai text editor. Pada tahap ini peneliti menggunakan *use case diagram* dan *activity diagram* untuk merepresentasikan interaksi antara user

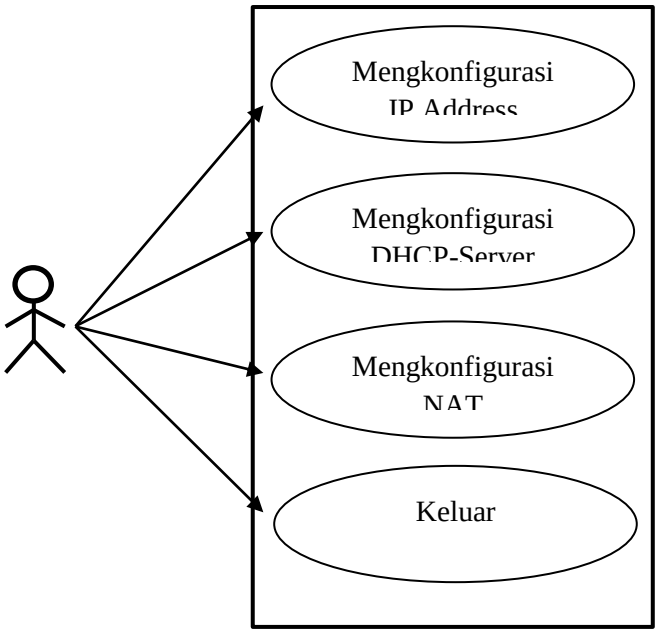
dengan aplikasi dan menggambarkan alur kerja dari aplikasi



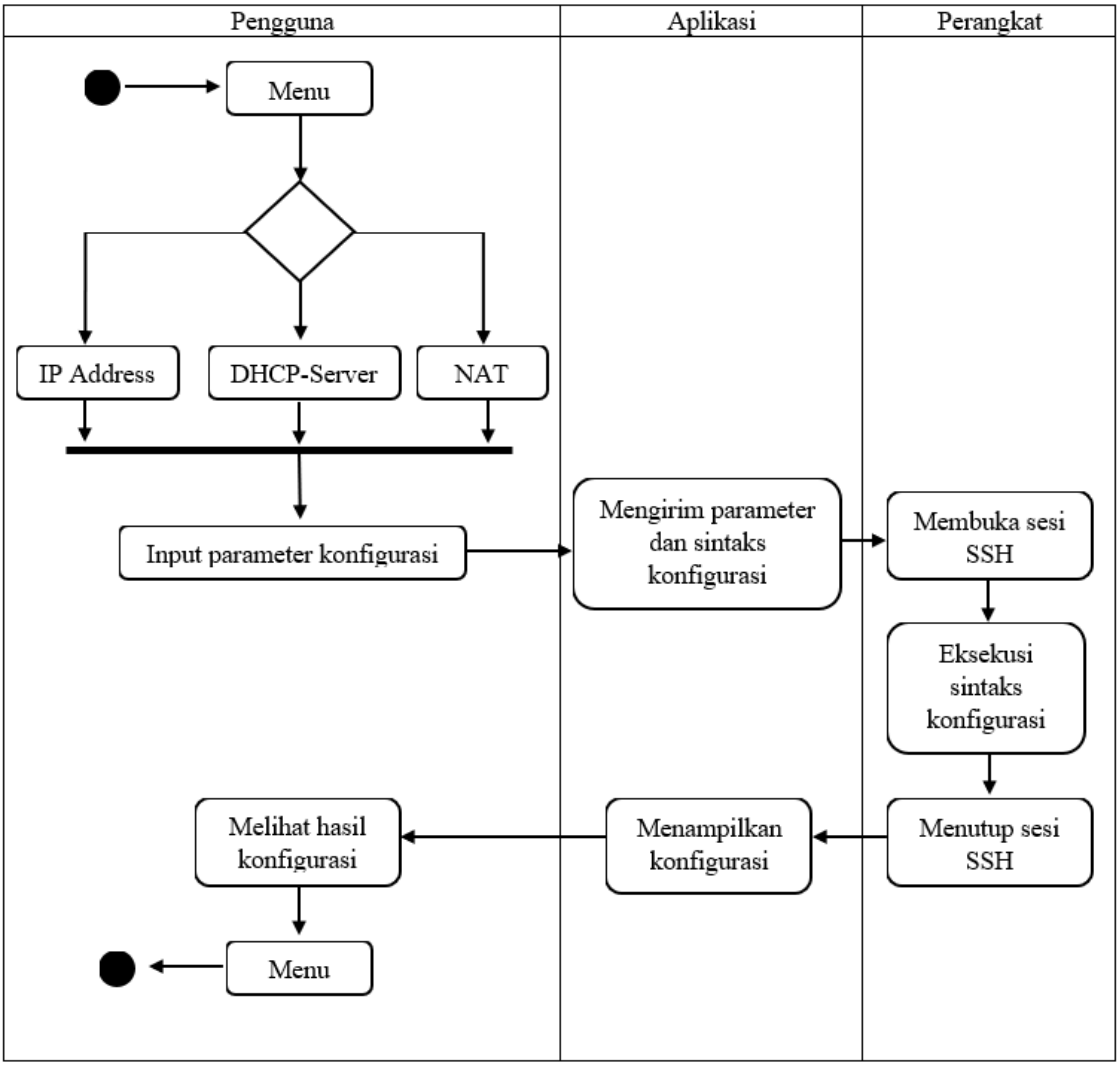
Gambar 1 Metode Penelitian



Gambar 2 Topologi Jaringan Aplikasi



Gambar 3 *Use Case Diagram* Aplikasi



Gambar 4 *Activity Diagram* Aplikasi

Tahap terakhir dalam penelitian ini adalah implementasi sistem dengan melakukan ujicoba dengan simulator GNS3 dan interpreter python 3 untuk menjalankan aplikasi. Ujicoba dilakukan sesuai dengan skenario / topologi yang dibuat.

HASIL DAN PEMBAHASAN

Penelitian ini menghasilkan sebuah aplikasi *network automation* yang memiliki fitur untuk melakukan konfigurasi IP Address, DHCP dan NAT serta fitur keluar. Adapun tampilan antarmuka sistem aplikasi sebagai berikut (Gambar 5).

A. Ujicoba Aplikasi

1. Fitur IP Address

Fitur IP address merupakan fitur untuk melakukan konfigurasi IP Address pada *interface* yang tersedia dalam perangkat. Fitur ini akan menjalankan *script* yang berisi perintah – perintah untuk melakukan konfigurasi IP Address. Adapun *script* yang dijalankan pada fitur ini adalah sebagai berikut (Gambar 6).

Alamat IP yang digunakan pada aplikasi ini sesuai dengan *script* yang ada pada program yaitu 192.168.1.1 untuk *interface fast ethernet 0/1* dan 8.8.8.8.1 untuk *interface fast ethernet 0/0* (Gambar 7).

1. Fitur DHCP Server

Fitur DHCP-Server merupakan fitur untuk melakukan konfigurasi DHCP agar PC *client* mendapatkan IP Address dari router sehingga tidak memerlukan konfigurasi IP address secara manual pada PC *client*. Fitur ini akan menjalankan *script* yang berisi perintah – perintah untuk melakukan konfigurasi DHCP Server. Adapun *script* yang dijalankan pada fitur ini adalah sebagai berikut (Gambar 8 & 9).

2. Fitur NAT

Fitur NAT merupakan fitur untuk melakukan konfigurasi NAT agar jaringan lokal akan dapat berkomunikasi dengan jaringan internet. Fitur NAT ini akan menterjemahkan alamat IP *private* pada jaringan lokal menjadi IP *public* sehingga memungkinkan untuk melakukan komunikasi antara jaringan lokal dan internet. Fitur ini akan menjalankan *script* yang berisi perintah – perintah untuk melakukan konfigurasi NAT. Adapun *script* yang dijalankan pada fitur ini adalah sebagai berikut (Gambar 10 & 11).

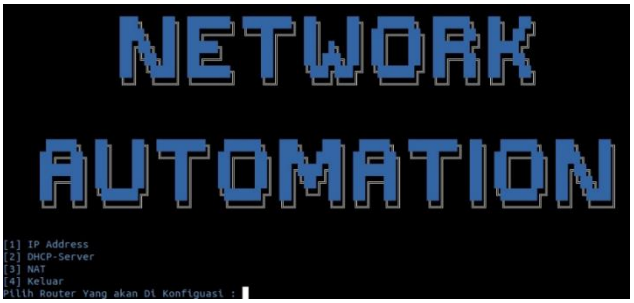
A. Hasil Ujicoba Aplikasi

1. Konfigurasi IP Address (Gambar 12).
2. Konfigurasi DHCP (Gambar 13).

Berdasarkan hasil pada gambar 10, *router* menyediakan fasilitas DHCP Server untuk *client* dengan IP *network* 192.168.1.0 dan *subnetmask* 255.255.255.0 sehingga PC *client* tidak perlu melakukan konfigurasi IP secara manual. Berikut hasil IP yang di dapat oleh PC *client* (Gambar 14).

3. Konfigurasi NAT (Gambar 15).

Berdasarkan hasil pada gambar 16, maka PC *client* dapat melakukan komunikasi dengan jaringan internet. Gambar 16 menunjukkan bahwa sistem aplikasi network automation berhasil dijalankan sesuai dengan skenario topologi yang dibuat. Penelitian ini dibuat berdasarkan penelitian yang dilakukan Wijaya dan Silviana, (2020) yang menghasilkan sebuah sistem aplikasi otomatisasi jaringan pada perangkat cisco dan mikrotik. Pada penelitian tersebut telah dibahas metode dan langkah-langkah melakukan proses otomatisasi pada perangkat jaringan. Penelitian tersebut memiliki beberapa fitur seperti : konfigurasi IP Address, Konfigurasi DHCP Server dan Konfigurasi Routing OSPF (*Open Shortest Path First*) untuk perangkat Cisco dan Mikrotik. Penelitian Wijaya dan Silviana(2020) mengatakan pengembangan aplikasi otomatisasi jaringan dapat dilakukan dengan menambahkan fitur-fitur konfigurasi jaringan yang lain seperti konfigurasi NAT, ACL, dll. Oleh karena itu, peneliti membuat sebuah sistem aplikasi *network automation* dengan menambahkan fitur konfigurasi NAT.



Gambar 5 Antarmuka Sistem Aplikasi

```
#fungsi yang menjalankan fitur IP Address
def ipaddress():
    #varibel ip
    ip_address_server = "8.8.8.1"
    ip_address_client = "192.168.1.1"
        mask = "255.255.255.0"

    #membuka koneksi ssh
    ssh_client = paramiko.SSHClient()
    ssh_client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
    ssh_client.connect(hostname=host,username=user, password=password)

    conn = ssh_client.invoke_shell()

    #perintah - perintah yang dikirim
    conn.send("conf t\n")
    conn.send("int fa0/0\n")
    conn.send("ip add {} {}\n".format(ip_address_server,mask))
    conn.send("no sh\n")
    conn.send("exit\n")
    conn.send("int fa0/1\n")
    conn.send("ip add {} {}\n".format(ip_address_client,mask))
    conn.send("no sh\n")
    conn.send("exit\n")

    #waktu prosesnya
    time.sleep(1)

    #menampilkan perintah yang sudah dikirim ke router
    output = conn.recv(65535)
    print (output.decode("ascii"))

    #menutup koneksi
    ssh_client.close()

    print (colorize("Konfigurasi IP berhasil!!!","success"))
```

Gambar 6 Kode Program Untuk Fitur IP Address

```
[1] IP Address
[2] DHCP-Server
[3] NAT
[4] Keluar
Pilih Router Yang akan Di Konfiguasi : 1

R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#int fa0/0
R1(config-if)#ip add 8.8.8.1 255.255.255.0
R1(config-if)#no sh
R1(config-if)#exit
R1(config)#int fa0/1
R1(config-if)#ip add 192.168.1.1 255.255.255.0
R1(config-if)#no sh
R1(config-if)#exit
R1(config)#
Konfigurasi IP berhasil!!!
Apakah Anda Ingin Keluar?(y/t)
```

Gambar 7 Konfigurasi IP Address Router

```
#fungsi yang menjalankan fitur dhcp-server
def dhcp():
    #varibel yang berisi ip network dhcp
    ip_dhcp = "192.168.1.0"
    mask_dhcp = "255.255.255.0"
    gateway = "192.168.1.1"
    ip_excluded = "192.168.1.1"

    #membuka koneksi ssh
    ssh_client = paramiko.SSHClient()
    ssh_client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
    ssh_client.connect(hostname=host,username=user, password=password)

    conn = ssh_client.invoke_shell()

    #perintah - perintah yang dikirim
    conn.send("conf t\n")
    conn.send("ip dhcp pool dhcp_pool1\n")
    conn.send("network {} {}\n".format(ip_dhcp,mask_dhcp))
    conn.send("default-router {}\n".format(gateway))
    conn.send("exit\n")
    conn.send("ip dhcp excluded-address {}\n".format(ip_excluded))

    #waktu prosesnya
    time.sleep(1)

    #menampilkan perintah yang sudah dikirim ke router
    output = conn.recv(65535)
    print (output.decode("ascii"))

    #menutup koneksi
    ssh_client.close()

    print (colorize("Konfigurasi DHCP berhasil!!!","success"))
```

Gambar 8 Kode Program Untuk Fitur DHCP

```
[1] IP Address
[2] DHCP-Server
[3] NAT
[4] Keluar
Pilih Router Yang akan Di Konfiguasi : 2

R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#ip dhcp pool dhcp_pool1
R1(dhcp-config)#network 192.168.1.0 255.255.255.0
R1(dhcp-config)#default-router 192.168.1.1
R1(dhcp-config)#exit
R1(config)#ip dhcp excluded-address 192.168.1.1
R1(config)#
Konfigurasi DHCP berhasil!!!
Apakah Anda Ingin Keluar?(y/t)
```

Gambar 9 Konfigurasi DHCP Server

```
#fungsi yang menjalankan fitur nat
def nat():

    ip_route = "0.0.0.0 0.0.0.0 8.8.8.8"
    #membuka koneksi ssh
    ssh_client = paramiko.SSHClient()
    ssh_client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
    ssh_client.connect(hostname=host,username=user, password=password)

    conn = ssh_client.invoke_shell()

    #perintah - perintah yang dikirim
    conn.send("conf t\n")
    conn.send("ip route {}\n".format(ip_route))
    conn.send("ip nat inside source list 1 interface fa0/0 overload\n")
    conn.send("access-list 1 permit any\n")
    conn.send("int fa0/0\n")
    conn.send("ip nat outside\n")
    conn.send("exit\n")
    conn.send("int fa0/1\n")
    conn.send("ip nat inside\n")
    conn.send("exit\n")

    #waktu prosesnya
    time.sleep(1)

    #menampilkan perintah yang sudah dikirim ke router
    output = conn.recv(65535)
    print (output.decode("ascii"))

    #menutup koneksi
    ssh_client.close()

    print (colorize("Konfigurasi NAT berhasil!!!","success"))
```

Gambar 10 Kode Program Untuk Fitur NAT

```
[1] IP Address
[2] DHCP-Server
[3] NAT
[4] Keluar
Pilih Router Yang akan Di Konfiguasi : 3

R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#ip route 0.0.0.0 0.0.0.0 8.8.8.8
R1(config)#ip nat inside source list 1 interface fa0/0 overload
R1(config)#access-list 1 permit any
R1(config)#int fa0/0
R1(config-if)#ip nat outside
R1(config-if)#exit
R1(config)#int fa0/1
R1(config-if)#ip nat inside
R1(config-if)#exit
R1(config)#
Konfigurasi NAT berhasil!!!
Apakah Anda Ingin Keluar?(y/t)
```

Gambar 11 Konfigurasi NAT

```
!
interface FastEthernet0/0
 ip address 8.8.8.1 255.255.255.0
 ip nat outside
 ip virtual-reassembly
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 192.168.1.1 255.255.255.0
 ip nat inside
 ip virtual-reassembly
 duplex auto
 speed auto
!
```

Gambar 12 Hasil Konfigurasi IP Address Router

```
!
ip dhcp pool dhcp_pool1
   network 192.168.1.0 255.255.255.0
   default-router 192.168.1.1
!
```

Gambar 13 Hasil Konfigurasi DHCP-Server Pada Router

```
PC1> ip dhcp
DDORA IP 192.168.1.2/24 GW 192.168.1.1

PC1> show ip

NAME        : PC1[1]
IP/MASK     : 192.168.1.2/24
GATEWAY     : 192.168.1.1
DNS         :
DHCP SERVER : 192.168.1.1
DHCP LEASE  : 86396, 86400/43200/75600
MAC         : 00:50:79:66:68:00
LPORT       : 20015
RHOST:PORT  : 127.0.0.1:20016
MTU:        : 1500
```

Gambar 14 Hasil DHCP Pada *Client*

```
ip route 0.0.0.0 0.0.0.0 8.8.8.8
!
!
ip nat inside source list 1 interface FastEthernet0/0 overload
!
access-list 1 permit any
no cdp log mismatch duplex
!
```

Gambar 15 Hasil Konfigurasi NAT

Gambar 16 Hasil Ujicoba NAT

KESIMPULAN DAN SARAN

Penelitian ini menghasilkan sebuah sistem aplikasi *network automation*. Berdasarkan hasil ujicoba, sistem ini telah berhasil dibuat dan dapat berjalan sesuai dengan fitur yang ada. Selain itu, sistem ini juga memudahkan administrator untuk melakukan konfigurasi NAT agar jaringan lokal dapat berkomunikasi dengan jaringan internet.

Sistem dapat dikembangkan dengan menambahkan fitur keamanan, dimana pada fitur tersebut berisi rules – rules access list pada perangkat.

DAFTAR PUSTAKA

Wijaya, I. & Silviana, A. B. (2020). Aplikasi Otomatisasi Jaringan Berbasis Command Line Interface Pada Router Cisco Dan Mikrotik, *Library Gunadarma*, [Abstract]. Available: repository, http://www.library.gunadarma.ac.id/repository/. [Accessed April 14, 2020].

Wiryawan, R. A. & Rosyid, N. R. (2019). Pengembangan Aplikasi Otomatisasi Administrasi Jaringan Berbasis Website Menggunakan Bahasa Pemrograman Python, *SIMETRIS*, 10, 741-752.

Wijaya, J. (2018). Network Automation using Ansible for Cisco Router Basic Configuration, *Teknik Elektro dan Informatika, Institut Teknologi Bandung*.