

Rancang Bangun Sistem Monitoring Menggunakan Network Automation Netmiko Python

Dimas Ardiansyah

Teknologi Informasi, Universitas Bina Sarana Informatika
Jl.Kramat Raya No.98, Senen, Jakarta Pusat, Indonesia

Email : dimas.a.0408@gmail.com

Submit: 30-09-2024 | Revisi : 06-12-2025 | Terima : 17-12-2025 | Publikasi: 19-12-2025

Abstrak

PT Astra Otoparts Tbk, adalah perusahaan yang bergerak di bidang otomotif yang menggunakan jaringan komputer untuk mendukung kegiatan operasional perusahaan. Dengan jaringan yang terhubung satu sama lain, PT Astra Otoparts Tbk, dapat bertukar data dan informasi melalui Server. Namun banyaknya jaringan yang terhubung seringkali PT Astra Otoparts kesulitan dalam melakukan perbaikan seperti perangkat yang terhubung terdapat kesalahan konfigurasi. Untuk mengatasi masalah itu maka, dibutuhkan sebuah sistem monitoring, dimana sistem ini dapat membantu pemantauan hinggaantisipasi dini masalah yang akan datang. Teknik pengumpulan data dalam penelitian ini adalah metode observasi, dan wawancara. Dengan menggunakan library Netmiko pada Python. Berdasarkan hasil dari pengujian sistem monitoring menggunakan library Netmiko. Teknisi PT Astra Otoparts Tbk dapat mempersingkat pengerjaan pemeliharaan perangkat jaringan.

Kata Kunci : Sistem Monitoring, Network Automation, Netmiko.

Abstract

PT Astra Otoparts Tbk, is a company operating in the automotive sector that uses computer networks to support the company's operational activities. With networks connected to each other, PT Astra Otoparts Tbk, can exchange data and information via servers. However, with the large number of connected networks, PT Astra Otoparts often has difficulty carrying out repairs, such as when the connected device has a configuration error. To overcome this problem, a monitoring system is needed, where this system can help monitor and anticipate future problems early. Data collection techniques in this research are observation and interview methods. By using the Netmiko library in Python. Based on the results of testing the monitoring system using the Netmiko library. PT Astra Otoparts Tbk technicians can shorten network device maintenance work.

Keywords: Monitoring System, Network Automation, Netmiko.

1. Pendahuluan

Teknologi di dunia berkembang dengan pesat, semakin besar dan banyak pula jalur jaringan yang tersedia baik untuk penggunaan rumahan hingga suatu perusahaan. Banyaknya jalur jaringan itupun tidak lepas dari pemeliharaan untuk selalu terkoneksi satu sama lainnya agar tetap terhubung dan dapat berkomunikasi. Namun, pemeliharaan yang dilakukan tidak lepas dari sebuah kesalahan yang mengakibatkan gagalnya suatu jaringan tidak dapat tersambung atau tidak bekerja sebagaimana seharusnya.

Penggunaan otomatisasi jaringan dapat menjadi solusi untuk melakukan pekerjaan yang berulang. Terlebih jika sistem jaringan yang dimiliki memiliki banyak perangkat yang tersebar, membuat metode tradisional konfigurasi perangkat membutuhkan waktu yang cukup lama, dengan otomatisasi jaringan lama waktu pengerjaan dapat lebih singkat.

Untuk mengimplementasikan otomatisasi jaringan dibutuhkan program yang dapat menjalankan sebuah perintah ke dalam perangkat jaringan. 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 (Fahmi et al., 2021).



PT Astra Otopart Tbk. adalah perusahaan besar yang bergerak di bidang otomotif dan tersebar di banyak titik, membuatnya memiliki banyak jalur jaringan, dan rentan akan koneksi yang tidak stabil akibat kesalahan konfigurasi. Sehingga implementasi otomatisasi jaringan sangat diperlukan untuk menangani pekerjaan yang berulang-ulang maupun terstruktur.

Berdasarkan latar belakang masalah yang diuraikan maka peneliti mengidentifikasi masalah yaitu: 1) Pekerjaan konfigurasi yang masih rentan mengalami kesalahan; 2) Perlunya sistem yang dapat menjalankan konfigurasi secara otomatis.

Tujuan dari penelitian ini untuk mempermudah melakukan konfigurasi secara otomatis terhadap perangkat jaringan sehingga meminimalisir kegagalan atau kesalahan pada konfigurasi perangkat jaringan yang digunakan.

2. Metode

2.1 Teknik Pengumpulan Data

Teknik pengumpulan data yang dilakukan meliputi

a. Observasi

Melakukan observasi terhadap topologi yang digunakan untuk memperoleh informasi visual yang dapat membantu dalam logika jalur konektivitas jaringan.

b. Wawancara

Wawancara dilakukan untuk memahami kegunaan suatu jaringan atau sistem serta krusialnya suatu jaringan jika tidak bekerja.

2.2. Landasan Teori

A. Jaringan Komputer

Jaringan komputer adalah sekumpulan komputer yang terhubung satu sama lain untuk berkomunikasi. Media yang digunakan untuk melakukan komunikasi tersebut dapat melalui kabel atau nirkabel (tanpa kabel). “Adapun data yang dikomunikasikan dapat berupa data teks, suara, gambar, atau video. Media jaringan komputer dapat melalui kabel-kabel atau tanpa kabel sehingga memungkinkan pengguna jaringan komputer dapat saling melakukan pertukaran informasi” (Papaceda et al., 2023).

B. Router

Router adalah sebuah perangkat keras jaringan komputer yang berfungsi menghubungkan beberapa jaringan. Routing adalah proses menghubungkan jaringan. “Fungsi utama router yaitu untuk membagi atau mendistribusikan IP (Internet Protocol) address, baik itu secara statis ataupun dhcp kepada semua komputer yang terhubung ke router tersebut” (Gunawan & Kurniawan, 2020).

C. Server

Server merupakan perangkat jaringan yang menyediakan berbagai macam fasilitas seperti sumber data, layanan, dan program yang dapat digunakan oleh pengguna lain yang disebut sebagai client. “Server adalah suatu perangkat keras atau perangkat lunak yang berfungsi sebagai pusat pengelolaan dan penyimpanan data dalam suatu jaringan komputer. Server berperan sebagai penyedia layanan dan sumber daya kepada client atau perangkat lainnya dalam jaringan” (Rahma et al., 2023).

D. Secure Shell (SSH)

Secure Shell (SSH) adalah protokol jaringan yang menggunakan port 22 dan berada di layer model OSI dan TCP/IP. SSH berfungsi sebagai sistem komunikasi yang aman untuk client dan server. “Meski demikian, internet tidak sepenuhnya aman. SSH menyediakan integritas dan kerahasiaan data melewati teknik enkripsi dan dekripsi dilakukan secara otomatis pada koneksinya” (Tohirin, 2020).

E. Framework Django

“Django adalah sebuah framework full-stack untuk membuat aplikasi web dengan bahasa pemrograman Python. Framework akan membantu membuat web lebih cepat, dibandingkan menulis kode dari nol” (Risya & Nuryamin, 2023). Full-stack memiliki arti, Django dapat melakukan kedua sisi dalam pengaplikasian program, yaitu front-end dan back-end. Front-end merupakan sisi yang dapat dilihat oleh pengguna seperti halaman-halaman yang ada pada suatu web. Sedangkan Back-end merupakan sisi yang tidak dapat dilihat oleh pengguna dikarenakan terdapat alur logika pemrograman ke sebuah server, agar permintaan dari pengguna dapat dijalankan.

F. Netmiko

<http://jurnal.bsi.ac.id/index.php/mti>

Netmiko adalah library python yang menggunakan koneksi SSH, akan tetapi Netmiko tidak dapat mendeteksi perangkat jaringan yang terhubung secara otomatis, Karena Netmiko tidak dapat mendeteksi perangkat jaringan maka memerlukan informasi dari perangkat jaringan seperti IP address, username, password, dan tipe perangkat agar dapat masuk ke dalam perangkat jaringan. “Tipe perangkat jaringan diperlukan karena Netmiko tidak dapat mendeteksi secara otomatis suatu tipe perangkat jaringan, contoh tipe perangkat jaringan yaitu cisco_ios yang merupakan jenis perangkat jaringan router dari vendor Cisco” (Muchamad Mashudi, 2022).

G. Sistem Monitoring

Sistem Monitoring merupakan sistem yang berfungsi untuk memantau perangkat jaringan yang terhubung, guna mengetahui keadaan perangkat jaringan tersebut secara real-time. “Monitoring digunakan untuk mengetahui perangkat jaringan mana yang mati dan hidup. Sistem monitoring digunakan untuk memantau, mengawasi, dan mengontrol jalan atau tidaknya suatu perangkat jaringan” (Husna & Rosyani, 2021). Pentingnya sistem ini tidak dapat diabaikan karena adanya gangguan pada jalur komunikasi yang digunakan dapat memiliki dampak yang signifikan “Sistem monitoring jaringan yang efektif dan efisien merupakan hal penting dalam menjaga 5 ketersediaan dan kinerja optimal jaringan dalam suatu organisasi” (Ishaq & Firmansyah, 2023).

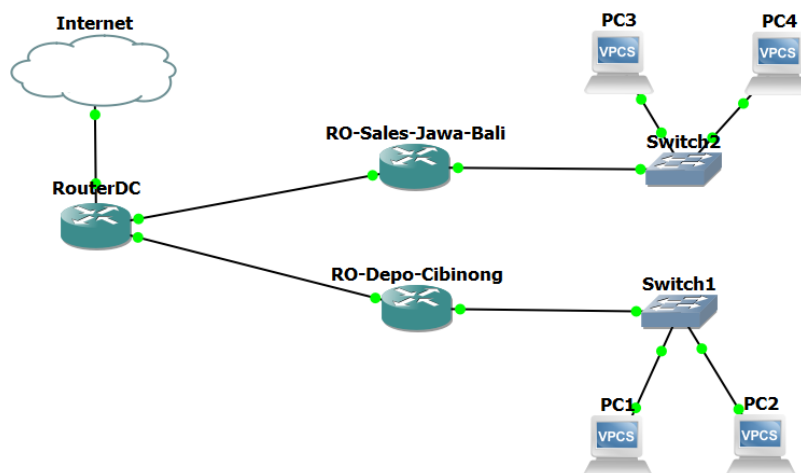
H. Network Automation

Network automation merupakan proses mengotomatiskan konfigurasi, manajemen, pengujian serta pengoperasian perangkat jaringan baik berupa fisik maupun virtual. “Proses otomatisasi jaringan ini menggunakan logika pemrograman (programmable logic) untuk mengatur sumber daya (resources) dan layanan jaringan sehingga menjadi lebih efisien dibandingkan dengan proses konfigurasi secara manual” (Kurniawan et al., 2023).

3. Hasil dan Pembahasan

3.1. Topologi Jaringan Berjalan

Topologi yang digunakan pada PT Astra Otoparts Tbk adalah topologi *tree*. Topologi *tree* merupakan topologi gabungan dari topologi bus dan star. Topologi *tree* sangat cocok digunakan PT Astra Otoparts Tbk karena banyaknya cabang yang tersebar dan membutuhkan jalur *backbone* untuk menghubungkan perangkat lainnya. Topologi ini juga efektif jika ada jalur jaringan selain jalur *backbone* yang mati maka tidak akan mengganggu jalur jaringan yang lain.



Gambar 1. Topologi jaringan

3.2. Permasalahan

Masih sering dilakukan secara konvensional yang membuatnya kurang efisien, terlebih jika konfigurasi yang dimasukkan berulang-ulang yang memungkinkan masih terjadinya kerusakan, dan perbaikan yang meliputi pergantian perangkat atau penambahan konfigurasi.

3.3. Alternatif Pemecahan Masalah

Penggunaan otomatisasi jaringan dapat menjadi solusi untuk melakukan pekerjaan yang berulang. Terlebih jika sistem jaringan yang dimiliki memiliki banyak perangkat yang tersebar, membuat metode tradisional

konfigurasi perangkat membutuhkan waktu yang cukup lama, dengan otomatisasi jaringan lama waktu pengerjaan dapat lebih singkat.

3.4. Rancangan Aplikasi

Rancangan aplikasi untuk membuat *network automation* membutuhkan bahasa pemrograman serta sistem operasi untuk menjalankannya. Berikut langkah-langkah yang dilakukan:

- Install framework Django dan library Netmiko melalui CMD dengan command **pip install Django** dan **pip install netmiko**.
- Buka *text editor* seperti *Visual Studio Code* dan buka terminal lalu buat folder proyek dengan ketik **Django admin startproject [nama_folder_projek]**, lalu buat 2 buah folder aplikasi untuk login dan manajemen perangkat dengan ketik **python manage.py startapp [nama_folder_aplikasi]**.
- Buatlah struktur database pada file **models.py** pada folder aplikasi login dan folder aplikasi manajemen perangkat.
- Script file **models.py** pada folder aplikasi login:

```
from django.contrib.auth.models import AbstractBaseUser, BaseUserManager
from django.db import models
class UserManager(BaseUserManager):
    def create_user(self, username, password=None, **extra_fields):
        if not username:
            raise ValueError('The Username field must be set')
        user = self.model(username=username, **extra_fields)
        user.set_password(password)
        user.save(using=self._db)
        return user
    def create_superuser(self, username, password=None, **extra_fields):
        extra_fields.setdefault('is_staff', True)
        extra_fields.setdefault('is_superuser', True)
        return self.create_user(username, password, **extra_fields)
class CustomUser(AbstractBaseUser):
    username = models.CharField(max_length=150, unique=True)
    is_active = models.BooleanField(default=True)
    is_staff = models.BooleanField(default=False)
    objects = UserManager()
    USERNAME_FIELD = 'username'
    REQUIRED_FIELDS = []
    def __str__(self):
        return self.username
```

Tabel 1. Struktur Database Aplikasi Login

id	password	last_login	username	active	staff
1	TechDash@2024	2024-07-26 9:19:56	Teknisi	1	1
2	FulanDash@2024	2024-08-11 14:42:27	Fulan	1	0

- Script file **models.py** pada folder aplikasi manajemen perangkat:
- ```
from django.db import models
from portal.models import CustomUser
class Device(models.Model):
 DEVICE_TYPES = 'mikrotik_routeros'
 name = models.CharField(max_length=100)
 device_type = models.CharField(max_length=50, default=DEVICE_TYPES)
 host = models.CharField(max_length=100)
 username = models.CharField(max_length=100)
 password = models.CharField(max_length=100)
 status = models.CharField(max_length=10, default='Unknown')
 def __str__(self):
```

```

return self.name
def get_device_type_display(self):
return 'MikroTik'

```

Tabel 2. Struktur Database Aplikasi Manajemen Perangkat

| id | name              | device_type       | host           | username | password | status    |
|----|-------------------|-------------------|----------------|----------|----------|-----------|
| 1  | RODC_AGIT         | mikrotik_routeros | 192.168.56.101 | admin    | admin    | connected |
| 2  | RO_Sales_Cibinong | mikrotik_routeros | 192.168.56.103 | admin    | admin    | connected |

- f. Buat folder templates yang akan digunakan untuk halaman file.html, pada folder aplikasi login tambah 1 file html untuk login, dan pada folder aplikasi manajemen perangkat tambah 3 file untuk penambahan perangkat, daftar perangkat, dan manajemen perangkat.

- g. Buat file **create\_users.py** pada folder proyek untuk membuat user yang dapat digunakan untuk *login*:

```

import os
import django
os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'subject7.settings')
django.setup()

from portal.models import CustomUser
def create_users(users):
 for username, password in users:
 if not CustomUser.objects.filter(username=username).exists():
 CustomUser.objects.create_user(username=username, password=password)
 print(f"User {username} created")
 else:
 print(f"User {username} already exists")

users = [
 ('SPV', 'Subject@NA7'),
 ('Teknisi', 'TechDash@2024'),
 ('Fulan', 'FulanDash@2024'),
]
create_users(users)

```

- h. Buat folder baru dan buat file **JSON** pada folder aplikasi manajemen perangkat untuk mengisi perintah yang akan dijalankan:

```

[
 {
 "name": "DHCP IT-Infra",
 "command": [
 "/interface vlan add name=IT_Infra vlan-id=24 interface=ether2",
 "/ip address add address=192.168.10.1/24 interface=IT_Infra",
 "/ip pool add name=dhcp-pool-IT-Infra ranges=192.168.10.2
192.168.10.254",
 "/ip dhcp-server add name=dhcp-Infra interface=IT_Infra lease-time=1800
address-pool=dhcp-pool-IT-Infra",
 "/ip dhcp-server network add address=192.168.10.0/24
gateway=192.168.10.1 dns-server=8.8.8.8"
]
 },
 {
 "name": "Set Firewall NAT",
 "command": [
 "/ip firewall nat add chain=srcnat out-interface=ether1 action=masquerade"
]
 }
]

```

```

51
 },
 {
 "name": "Routing Ke RODC_AGIT",
 "command": [
 "/ip route add dst-address=20.20.20.0/24 gateway=192.168.56.103"
]
 },
 {
 "name" : "Show IP Address",
 "command": [
 "/ip address print"
]
 },
 {
 "name" : "Add DNS 8.8.8.8",
 "command": [
 "/ip dns set servers=8.8.8.8 allow-remote-requests=yes"
]
 }
]

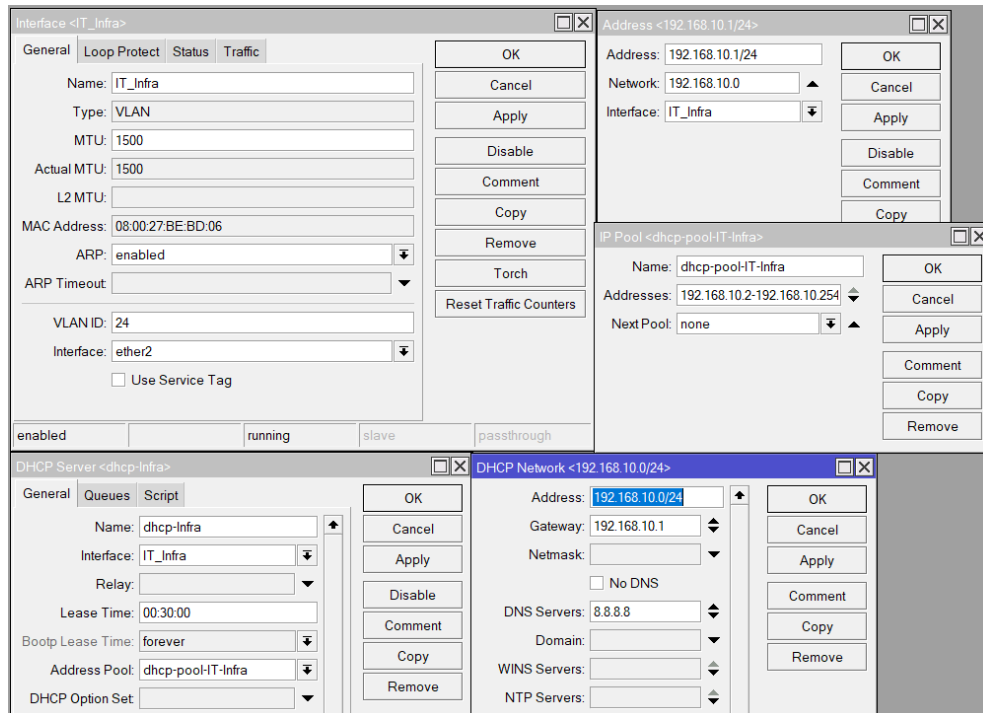
```

- i. Buka terminal dan lakukan *migrate* untuk menyimpan atau memperbarui tabel pada database sesuai dengan perubahan tersebut dengan ketik **python manage.py makemigrations** lalu **python manage.py migrate**.
- j. Ketik **python create\_users.py** untuk membuat pengguna yang sudah ditetapkan sebelumnya.
- k. Siapkan ISO seperti Windows atau Ubuntu (linux) untuk server, dan RouterOS CHR untuk perangkat jaringan.
- l. Ketik perintah **python manage.py runserver** untuk menjalankan server Django, dan buka browser dan ketik <http://127.0.0.1:8000/>.
- m. Lakukan login dan tambahkan perangkat jaringan.

### 3.5. Pengujian Jaringan Awal

Pada tahap pengujian jaringan awal, dilakukan beberapa tes seperti *ping* untuk mengetahui apakah jaringan tersebut terkoneksi dan melakukan konfigurasi *DHCP server* pada perangkat jaringan untuk divisi IT-Infra. Berikut adalah langkah-langkah pembuatan *DHCP Server*:

- a. Masuk ke perangkat jaringan melalui winbox.
- b. Klik menu Interface -> klik tanda + -> VLAN -> ubah *Name* vlan1 menjadi IT\_Infra -> vlan id 24 -> *Interface ether2* -> *Apply*.
- c. Klik menu IP -> *Addresses* -> klik tanda + -> *Address* 192.168.10.1/24 -> *Interface* IT\_Infra -> *Apply*.
- d. Klik menu IP -> *Pool* -> klik tanda + -> *Name* dhcp-pool-IT-Infra -> *Addresses* 192.168.10.2-192.168.10.254 -> *Apply*.
- e. Klik menu IP -> *DHCP Server* -> *DHCP* -> klik tanda + -> *Name* dhcp-Infra -> *Interface* IT\_Infra -> *Lease Time* 00:30:00 -> *Address pool* dhcp-pool-IT-Infra -> *Apply*.
- f. Klik menu IP -> *DHCP Server* -> *Network* -> Klik tanda + -> *Address* 182.168.10.0/24 -> *Gateway* 192.168.10.1 -> *DNS Server* 8.8.8.8 -> *Apply*.



Gambar 2. Konfigurasi perangkat pada Winbox

### 3.6. Pengujian Jaringan Akhir

Pada pengujian jaringan akhir dilakukan uji coba program *network automation* untuk melakukan perintah konfigurasi yang sama, yaitu membuat *IP DHCP Server* melalui browser, berikut langkah-langkah yang dilakukan:

- Jalankan perintah **python manage.py runserver**, dan buka browser dan ketik <http://127.0.0.1:8000/>.
- Login menggunakan pengguna yang sudah ditetapkan.
- Daftarkan terlebih dahulu perangkat jaringan dengan klik **Add New Device**.
- Pastikan *IP Address* perangkat sudah dikonfigurasi sebelumnya dan isi data sesuai dengan perangkat jaringan yang dituju.

### Add Device

Name:

Host:

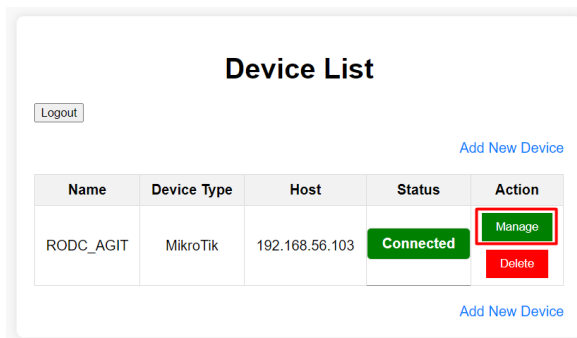
Username:

Password:

[Back to Device List](#)

Gambar 3. Mendaftarkan perangkat jaringan

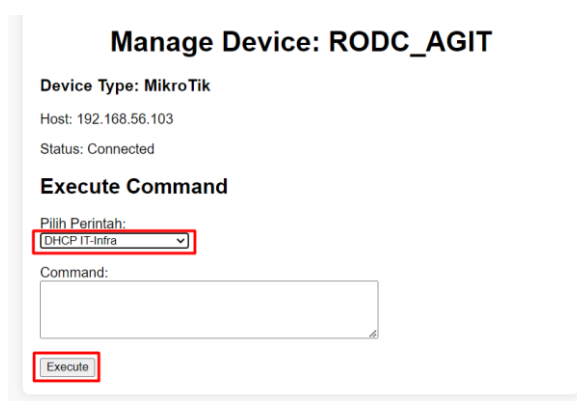
- Klik menu **Manage** pada perangkat jaringan yang sudah terdaftar untuk melakukan konfigurasi.



| Name      | Device Type | Host           | Status    | Action                                     |
|-----------|-------------|----------------|-----------|--------------------------------------------|
| RODC_AGIT | MikroTik    | 192.168.56.103 | Connected | <span>Manage</span><br><span>Delete</span> |

Gambar 4. Daftar perangkat jaringan yang terdaftar

- f. Pada menu Pilih Perintah, pilih DHCP IT-Infra, lalu klik Execute.



**Manage Device: RODC\_AGIT**

Device Type: MikroTik  
Host: 192.168.56.103  
Status: Connected

**Execute Command**

Pilih Perintah:  
[DHCP IT-Infra]

Command:

[Execute]

Gambar 5. Menu pilihan perintah

#### 4. Kesimpulan

Sistem pemantauan sangat berguna untuk membantu pemantauan jaringan, kita dapat mengetahui ada atau tidak adanya jalur jaringan yang bermasalah, sehingga kita dapat melakukan antisipasi saat jalur jaringan mengalami masalah.

Sistem pemantauan yang telah dibuat memiliki kemampuan untuk mengelola perangkat jaringan dari jarak jauh (remote) sehingga memiliki potensi sebagai alternatif sistem pemantauan lain, baik open source maupun berbayar.

#### Referensi

- Fahmi, M., Maisyaroh, M., Komarudin, I., Faizah, S., & Fadhilah, I. (2021). Otomatisasi Jaringan Menggunakan Script Python Untuk Penyediaan Konfigurasi Internet Dan Manajemen Mikrotik. *Bina Insani Ict Journal*, 8(1), 53. <https://doi.org/10.51211/biict.v8i1.1517>
- Gunawan, T., & Kurniawan, D. F. (2020). Rancang Bangun Jaringan Wireless Local Area Network ( WLAN ) Menggunakan Metode Routing Statik Pada Smpn 7 Pesawaran. *Jurnal Informatika Software Dan Network*, 01(01), 41–47.
- Husna, M. A., & Rosyani, P. (2021). Implementasi Sistem Monitoring Jaringan dan Server Menggunakan Zabbix yang Terintegrasi dengan Grafana dan Telegram. *JURIKOM (Jurnal Riset Komputer)*, 8(6), 247. <https://doi.org/10.30865/jurikom.v8i6.3631>
- Ishaq, M. Y., & Firmansyah, F. (2023). Implementasi Sistem Monitoring Menggunakan Zabbix Dan Notifikasi Realtime Telegram. *Jurnal INSAN Journal of Information System Management Innovation*, 3(2), 72–77. <https://doi.org/10.31294/jinsan.v3i2.2432>
- Kurniawan, A., Murad, F. A., AUFAR, T. Q., Aziz, M. T., Idzhaarulhaq, M. N., & Agustin, M. (2023). Network Automation Menggunakan Ansible di Laboratorium Komputer. *Seminar Nasional Inovasi Vokasi*, 2(1), 283–290.
- Muchamad Mashudi. (2022). Network Automation Menggunakan Bahasa Pemrograman Python. *Jurnal Teknik Industri, Sistem Informasi Dan Teknik Informatika*, December, 120–122.
- Papaceda, D. D., Mewengkang, A., & Pratasik, S. (2023). Analisis dan Pengembangan Jaringan Komputer di SMK

- Negeri 8 Weda Halmahera Tengah. *Edutik : Jurnal Pendidikan Teknologi Informasi Dan Komunikasi*, 3(1), 1–13. <https://doi.org/10.53682/edutik.v3i1.6465>
- Rahma, A., Indriyani, F., & Sandi, T. A. A. (2023). Perancangan Dan Implementasi Monitoring Perangkat Server Menggunakan Zabbix Pada PT. Rizki Tujuh Belas Kelola. *Jurnal INSAN Journal of Information System Management Innovation*, 3(2), 85–95. <https://doi.org/10.31294/jinsan.v3i2.3009>
- Risyda, F., & Nuryamin, Y. (2023). Perancangan Sistem Informasi Manajemen Invoice Menggunakan Generator Freamwork Django-Python Berbasis Website Pada Pt. Lampuind Tekno Elektrik. *Jurnal Sistem Informasi Universitas Suryadarma*, 10(1). <https://doi.org/10.35968/jsi.v10i1.999>
- Tohirin, T. (2020). Penerapan Keamanan Remote Server Melalui Ssh Dengan Kombinasi Kriptografi Asimetris Dan Autentikasi Dua Langkah. *Jurnal Teknologi Informasi*, 4(1), 133–138. <https://doi.org/10.36294/jurti.v4i1.1262>