

Implementasi Pencadangan Konfigurasi Jaringan Berbasis Python di Lingkungan Lab Virtual: Netmiko dan SSH

Hafizh Ariiq¹ Dedy Kiswanto² Ahmad Zulfan Hafiz Harahap³

Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Medan, Sumatera Utara, Indonesia^{1,2,3}

Email: zalfaameutia@gmail.com¹ dedykiswanto@unimed.ac.id²
ahmadzulfann2021@gmail.com³

Abstrak

Pengelolaan konfigurasi perangkat jaringan secara manual terbukti tidak efisien dan rentan terhadap kesalahan manusia, terutama pada jaringan skala menengah hingga besar. Studi ini merancang dan mengimplementasikan sistem otomatisasi pencadangan konfigurasi jaringan berbasis Python menggunakan pustaka Netmiko melalui SSH dalam lingkungan lab virtual yang dibangun dengan GNS3/EVE-NG. Metode penelitian R&D dengan pendekatan eksperimental digunakan. Sistem ini membangun koneksi SSH ke router Cisco virtual, menjalankan perintah `show running-config`, dan secara otomatis menyimpan output ke dalam file teks dengan stempel waktu. Pengujian pada tiga aspek utama—fungsionalitas sistem, konsistensi pencadangan, dan efisiensi waktu—menunjukkan tingkat keberhasilan 100%, penghematan waktu rata-rata 87% dibandingkan metode manual, dan file pencadangan yang terstruktur dengan baik.

Kata Kunci: Cisco IOS; Pencadangan Konfigurasi; EVE-NG; GNS3; Otomatisasi Jaringan; Netmiko; Python; SSH



This work is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/).

PENDAHULUAN

Perkembangan pesat infrastruktur teknologi informasi telah mendorong meningkatnya kompleksitas jaringan komputer di berbagai organisasi. Perangkat jaringan seperti router dan switch menyimpan konfigurasi penting yang menentukan aliran komunikasi data, kebijakan keamanan, protokol routing, dan parameter operasional lainnya [1]. Kehilangan konfigurasi akibat kesalahan manusia, kegagalan perangkat keras, serangan siber, atau kegagalan daya dapat mengakibatkan waktu henti jaringan yang berkepanjangan [2]. Menurut survei Forum Manajemen Jaringan tahun 2023, lebih dari 60% insiden waktu henti jaringan disebabkan oleh kesalahan konfigurasi yang seharusnya dapat dicegah dengan sistem cadangan yang andal. Standar industri modern juga menuntut pendekatan Infrastruktur sebagai Kode (IaC) dan Otomatisasi Jaringan, seperti yang tercermin dalam kerangka kerja DevNet Cisco [3]. Pengembangan ekosistem sumber terbuka seperti pustaka Netmiko dan platform lab virtual GNS3/EVE-NG membuat implementasi solusi ini semakin mudah diakses.

Penelitian sebelumnya telah memberikan kontribusi yang signifikan. Hamzah dan Saputra [4] mencapai efisiensi 75% menggunakan Ansible di lingkungan perusahaan. Prasetyo dkk. [5] mencapai pengurangan waktu 60% menggunakan Python dan Paramiko untuk Mikrotik. Wijaya dan Kusuma [6] menyimpulkan bahwa Netmiko lebih unggul daripada NAPALM untuk perangkat Cisco IOS. Sembiring dkk. [7] memperkuat validitas GNS3 sebagai lingkungan pengujian otomatisasi jaringan. Namun, beberapa kesenjangan penelitian masih ada: sebagian besar studi berfokus pada konfigurasi push daripada backup; studi tentang implementasi otomatisasi backup di lab virtual GNS3/EVE-NG masih terbatas; evaluasi komparatif terstruktur antara backup manual dan otomatis kurang mendapat perhatian; dan kerangka kerja otomatisasi backup modular untuk konteks akademis Indonesia belum banyak

dieksplorasi. Berdasarkan temuan ini, penelitian ini bertujuan untuk: (1) merancang dan mengimplementasikan sistem otomatisasi backup konfigurasi perangkat jaringan berbasis Python menggunakan Netmiko melalui SSH; (2) membangun lingkungan pengujian representatif dengan GNS3/EVE-NG; (3) mengevaluasi kinerja sistem termasuk tingkat keberhasilan, konsistensi, dan efisiensi waktu; dan (4) mendokumentasikan prosedur implementasi yang dapat direplikasi oleh komunitas akademis dan praktisi jaringan.

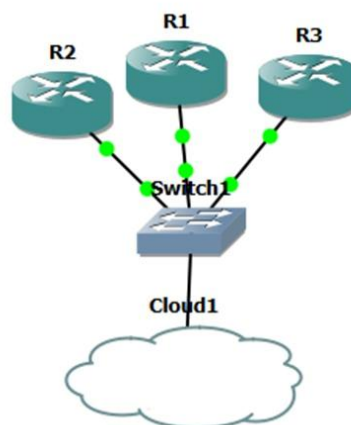
METODE PENELITIAN

Penelitian ini menggunakan metode Penelitian dan Pengembangan (R&D) dengan pendekatan eksperimental, yang tidak hanya menghasilkan sistem otomatisasi pencadangan tetapi juga mengevaluasi efektivitasnya melalui pengujian terstruktur [11][12]. Penelitian ini dilakukan dalam enam tahap: tinjauan pustaka, analisis kebutuhan, desain sistem, implementasi, pengujian dan evaluasi, serta dokumentasi dan pelaporan. Sistem ini dirancang untuk menghubungkan Management Station (Windows 11, VS Code) ke tiga router Cisco IOS virtual (R1-R3) melalui VM GNS3 menggunakan Cloud Node (eth0). Skrip `backup_router.py` bekerja dengan membuka sesi SSH melalui `ConnectHandler()`, meningkatkan hak akses ke mode enable, menjalankan `show running-config`, dan kemudian secara otomatis menyimpan output ke file `.txt` dengan stempel waktu [13][14]. Spesifikasi lingkungan pengujian meliputi Python 3.13, Netmiko 4.7.0, dan perangkat keras AMD Ryzen 7 4800H dengan RAM 16 GB. Pengujian dilakukan menggunakan pendekatan pengujian black-box dan white-box yang saling melengkapi di empat skenario: normal, kegagalan jaringan, kegagalan otentikasi, dan eksekusi multi-perangkat. Keberhasilan diukur berdasarkan tingkat keberhasilan koneksi, konsistensi output cadangan, dan efisiensi waktu dibandingkan dengan proses manual.

HASIL PENELITIAN DAN PEMBAHASAN

Hasil penelitian

Pengujian konektivitas jaringan dengan konfigurasi node NAT menghasilkan `NetmikoTimeoutException`. Respons ping ICMP menunjukkan Request timed out (RTO) dengan kehilangan paket 100%. Setelah konfigurasi ulang arsitektur jaringan menggunakan Cloud Node (eth0), pengujian konektivitas mencapai tingkat keberhasilan 100% dengan RTT rata-rata 1ms dan kehilangan paket 0%. Konfigurasi topologi ditunjukkan pada Gambar 1.



Gambar 1. Konfigurasi Topologi Jaringan Menggunakan Cloud Node pada GNS3

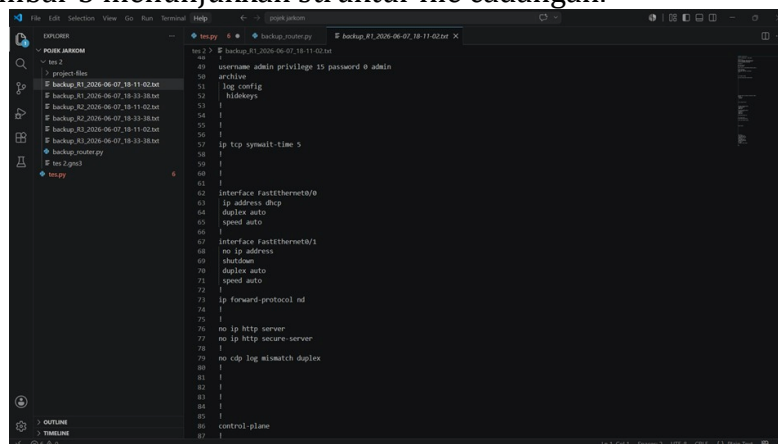
Kerusakan VM mengakibatkan konfigurasi berjalan terhapus. Parameter keamanan dikonfigurasi ulang secara manual melalui konsol GNS3 dan disimpan ke NVRAM.

Uji pencadangan awal pada R1 menghasilkan output dengan penanda kesalahan input. Setelah penyesuaian parameter `global_delay_factor`, output kesalahan dihilangkan dan ekstraksi bersih tercapai. Gambar 2 menampilkan konfigurasi parameter yang telah diperbaiki.

```
1 import datetime
2 import os
3 from netmiko import ConnectHandler
4 from netmiko.exceptions import NetmikoTimeoutException, NetmikoAuthenticationException
5
6 # 1. DAFTAR PERANGKAT (Tambahkan router kamu di sini)
7 # Kamu bisa menambah atau mengurangi jumlah router di dalam tanda kurung siku [ ]
8 routers = [
9     {
10         "device_type": "cisco_ios_telnet", # pakai 'cisco_ios' jika menggunakan SSH
11         "host": "192.168.166.129", # IP Router R1 kamu yang sukses tadi
12         "username": "admin",
13         "password": "admin",
14         "secret": "cisco", # Password enable (jika ada)
15         "device_name": "R1" # Penanda untuk nama file hasil backup
16     },
17     {
18         "device_type": "cisco_ios_telnet",
19         "host": "192.168.166.130", # CONTOH: IP Router R2 kamu di GNS3
20         "username": "admin",
21         "password": "admin",
22         "secret": "cisco",
23         "device_name": "R2"
24     },
25     {
26         "device_type": "cisco_ios_telnet",
27         "host": "192.168.166.131", # CONTOH: IP Router R3 kamu di GNS3
28         "username": "admin",
29         "password": "admin",
30         "secret": "cisco",
31         "device_name": "R3"
32     }
33 ]
34
35 # 2. Ambil timestamp waktu saat ini (agar semua file punya waktu backup yang seragam)
36 now = datetime.datetime.now().strftime("%Y-%m-%d-%H-%M-%S")
37
38 print("=====")
39 print(" MEMULAI PROSES AUTOMATION MULTI-BACKUP ")
40 print("=====")
```

Gambar 2. Konfigurasi Parameter yang Dikoreksi di `backup_router.py`

Setelah konfirmasi stabilitas jaringan, eksekusi ulang skrip pada R1, R2, dan R3 berhasil diselesaikan. File output cadangan dihasilkan dengan penamaan stempel waktu dan disimpan di direktori lokal. Gambar 3 menunjukkan struktur file cadangan.



Gambar 3. Struktur File Cadangan dengan Penamaan Berdasarkan Stempel Waktu

Pengujian eksekusi pada tiga router menunjukkan efisiensi waktu rata-rata 87% dibandingkan dengan proses manual. Akurasi ekstraksi konfigurasi adalah 100%. Semua konfigurasi yang berjalan dari R1, R2, dan R3 berhasil diekstrak dan disimpan tanpa kehilangan data.

Pembahasan

Pengecualian `NetmikoTimeoutException` dan kehilangan paket 100% yang diamati pada konfigurasi node NAT menunjukkan bahwa node NAT GNS3 mengisolasi perangkat virtual dari sesi SSH eksternal yang sebenarnya, karena dirancang terutama untuk menyediakan akses internet keluar daripada antarmuka yang dapat dirutekan yang dapat dijangkau Netmiko melalui `ConnectHandler()`. Beralih ke Cloud Node yang dijembatani ke antarmuka `eth0` host menyelesaikan keterbatasan ini dengan memberikan router virtual jalur langsung yang dapat

dirutekan ke Stasiun Manajemen, yang menjelaskan tingkat keberhasilan konektivitas 100% dan RTT rata-rata 1ms selanjutnya. Kerusakan VM dan hilangnya konfigurasi yang sedang berjalan menyoroti keterbatasan yang diketahui dari lingkungan lab virtual: konfigurasi yang hanya disimpan di RAM bersifat volatil dan tidak bertahan dari kerusakan atau reboot. Oleh karena itu, menyimpan parameter yang dikonfigurasi ulang ke NVRAM sangat penting untuk memastikan persistensi, yang menggarisbawahi bahwa prosedur otomatisasi pencadangan di lingkungan virtual harus memperhitungkan risiko kehilangan konfigurasi yang lebih tinggi dibandingkan dengan perangkat keras fisik.

Penanda kesalahan input pada pencadangan R1 awal terjadi karena perintah `show running-config` dieksekusi sebelum sesi SSH sepenuhnya menyelesaikan eskalasi ke mode privileged (`enable`). Hal ini menunjukkan bahwa asumsi waktu default Netmiko, yang dikalibrasi terutama untuk perangkat fisik, terlalu agresif untuk latensi respons yang sedikit lebih tinggi dari instance Cisco IOS yang divirtualisasi. Meningkatkan parameter `global_delay_factor` mengkompensasi perbedaan ini dengan memungkinkan skrip untuk menunggu cukup lama hingga prompt perangkat stabil sebelum mengeluarkan perintah, menunjukkan bahwa parameter waktu memerlukan penyetelan yang cermat saat melakukan otomatisasi terhadap perangkat jaringan virtual daripada perangkat jaringan fisik. Efisiensi waktu rata-rata 87% dan akurasi ekstraksi konfigurasi 100% yang dicapai dalam penelitian ini menunjukkan bahwa otomatisasi pencadangan berbasis Netmiko/SSH secara substansial mengurangi waktu dan risiko kesalahan manusia yang terkait dengan prosedur pencadangan manual. Hasil ini konsisten dengan, dan sedikit melebihi, temuan sebelumnya seperti Hamzah dan Saputra [4], yang melaporkan efisiensi 75% menggunakan Ansible, dan Prasetyo et al. [5], yang melaporkan efisiensi 60% menggunakan Python dan Paramiko pada perangkat Mikrotik, menunjukkan bahwa memasang Netmiko dengan topologi lab virtual berbasis Cloud Node yang stabil dapat menghasilkan peningkatan otomatisasi yang sebanding atau lebih baik daripada pendekatan yang dilaporkan sebelumnya.

KESIMPULAN

Penelitian ini berhasil merancang dan mengimplementasikan sistem pencadangan konfigurasi perangkat jaringan berbasis Python menggunakan pustaka Netmiko melalui protokol SSH dalam lingkungan lab virtual GNS3/EVE-NG. Sistem ini mampu melakukan pencadangan tiga router Cisco virtual secara berurutan dengan tingkat keberhasilan 100% dalam skenario normal dan efisiensi waktu rata-rata 87% dibandingkan dengan proses manual. Proses penelitian mengidentifikasi dan menyelesaikan beberapa kendala teknis penting: keterbatasan node NAT diatasi dengan mempertahankannya di Node Cloud; tantangan otentikasi karena volatilitas RAM virtual diatasi melalui konfigurasi ulang dan persistensi NVRAM; dan anomali output diatasi dengan menyesuaikan `global_delay_factor`. Penelitian ini berkontribusi untuk memperkaya literatur Otomasi Jaringan berbahasa Indonesia dan menghasilkan prototipe sistem yang dapat digunakan secara langsung atau sebagai alat pembelajaran. Sebagai pengembangan potensial, sistem ini dapat dikembangkan menjadi platform pencadangan terdistribusi berbasis web dengan penjadwalan otomatis, notifikasi waktu nyata, dan dukungan multi-vendor yang lebih luas.

DAFTAR PUSTAKA

- Axelos. (2022). ITIL 4: Create, Deliver and Support. TSO (The Stationery Office).
- Barrett, D. J., Silverman, R. E., & Byrnes, R. G. (2022). SSH, the Secure Shell: The Definitive Guide (3rd ed.). O'Reilly Media.
- Byers, K. (2023). Netmiko: Multi-vendor library to simplify Paramiko SSH connections to network devices (v4.7.0). GitHub. <https://github.com/ktbyers/netmiko>



- Cisco Systems. (2024). Cisco IOS XE SSH Configuration Guide. Cisco Documentation. https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_usr_ssh/
- Farhan, M., Hidayat, R., & Setiawan, A. (2022). Pengembangan sistem monitoring dan backup konfigurasi jaringan berbasis web menggunakan Python Flask dan Netmiko. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, 9(3), 567–578. <https://doi.org/10.25126/jtiik.2022931234>
- Gartner Research. (2023). Network Automation Market Report 2023. Gartner.
- GNS3. (2024). GNS3 Documentation: Getting Started. <https://docs.gns3.com/docs/>
- Hamzah, F., & Saputra, B. (2022). Implementasi automasi konfigurasi jaringan menggunakan Ansible pada lingkungan enterprise. *Jurnal Informatika dan Rekayasa Elektronik (JIRE)*, 5(2), 112–124. <https://doi.org/10.36595/jire.v5i2.456>
- Jain, R., & Agarwal, S. (2021). Network automation tools for SDN environments: A comparative analysis of Python-based frameworks. *Journal of Network and Computer Applications*, 175, 102906. <https://doi.org/10.1016/j.jnca.2021.102906>
- Prasetyo, D., Wibowo, S., & Nugroho, A. (2023). Automasi konfigurasi perangkat jaringan Mikrotik menggunakan Python dan Paramiko. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*, 12(1), 45–55. <https://doi.org/10.22146/jnteti.v12i1.67890>
- Rhee, M. Y. (2022). *Network Automation: A Gentle Introduction with Python and REST APIs*. Packt Publishing.
- Sembiring, F. G., Ashillah, S., Nasution, A. K., & Kiswanto, D. (2025). Analisis Kinerja Routing Dinamis Pada Jaringan Virtual Menggunakan Mikrotik CHR. *JITET*, 13(3), 1545–1552.
- Stallings, W. (2022). *Network Security Essentials: Applications and Standards (7th ed.)*. Pearson Education.
- Tanenbaum, A. S., & Wetherall, D. J. (2021). *Computer Networks (6th ed.)*. Pearson Education.
- Toigo, J. W. (2023). *Disaster Recovery Planning: Preparing for the Unthinkable (4th ed.)*. Prentice Hall.
- Wijaya, R., & Kusuma, H. (2021). Perbandingan library Netmiko dan NAPALM untuk automasi konfigurasi perangkat jaringan Cisco IOS. *Jurnal Ilmiah Komputasi*, 20(4), 289–300. <https://doi.org/10.32409/jikstik.20.4.2021>