E-ISSN : 3025-1311
https://journal.cib.institute/index.php/kohesi

Kohesi: Jurnal Multidisiplin Saintek
Volume 11 No 5 Tahun 2026

OTOMATISASI JARINGAN MULTIVENDOR BERBASIS SCRIPTING UNTUK EFISIENSI PENYEBARAN KONFIGURASI JARINGAN DI KORLANTAS POLRI

***Wirayuga Rizkia Suwahyo[1], *Kussigit Santosa[2]**
[1],[2] Program Studi Teknik Informatika
Fakultas Ilmu Komputer, Universitas Pamulang
Jl. Raya Puspitek No.11, Buaran, Serpong, Kota Tangerang Selatan, Banten 15310

**E-mail : rizkiawirayuga@gmail.com[1], kussigit@gmail.com [2]*

ABSTRACT

Managing network devices from multiple vendors (multivendor) presents challenges because each device has different characteristics and configuration mechanisms. In the Information and Communication Technology (ICT) Division of Korlantas POLRI, the configuration and backup processes for MikroTik and Cisco devices were previously performed manually through the Command Line Interface (CLI). This condition required considerable time and accuracy and created the potential for configuration errors. This study aims to develop a scripting-based multivendor network automation system to support efficient network configuration deployment. The Waterfall model was used for system development, while data were collected through literature review, interviews, and observation. The system was developed as a web-based application using PHP, MySQL, and Python, with the Netmiko library utilized to establish SSH connections to network devices. The system provides user and device management, configuration deployment, device status monitoring, activity logging, deployment history, configuration backup, and reporting features. System testing was conducted using Black Box and White Box methods. The test results indicate that the main system functions operated according to the specified requirements. The system is capable of deploying configurations through SSH connections, displaying configuration results, recording activities, and supporting more structured management of multivendor network devices.

***Keywords**: Network Automation, Multivendor, Scripting, Python, Netmiko.*

ABSTRAK

Pengelolaan perangkat jaringan dari berbagai vendor (*multivendor*) menghadirkan tantangan karena setiap perangkat

Article History
Received: Oktober 2026
Reviewed:Oktober 2026
Published: Oktober 2026

Plagiarism Checker No 237.u873247.fh8374

Copyright : Author
Publish by : Kohesi

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License

memiliki karakteristik dan mekanisme konfigurasi yang berbeda. Pada Bagian Teknologi Informasi dan Komunikasi (TIK) Korlantas POLRI, proses konfigurasi dan backup perangkat MikroTik dan Cisco sebelumnya dilakukan secara manual melalui *Command Line Interface* (CLI). Kondisi tersebut menyebabkan proses konfigurasi membutuhkan waktu dan ketelitian serta berpotensi menimbulkan kesalahan konfigurasi. Penelitian ini bertujuan mengembangkan sistem otomatisasi jaringan *multivendor* berbasis scripting untuk mendukung efisiensi penyebaran konfigurasi jaringan. Metode pengembangan sistem yang digunakan adalah *Waterfall*, sedangkan pengumpulan data dilakukan melalui studi pustaka, wawancara, dan observasi. Sistem dikembangkan sebagai aplikasi berbasis web menggunakan PHP, MySQL, dan Python dengan memanfaatkan *library* Netmiko untuk membangun koneksi SSH ke perangkat jaringan. Sistem menyediakan fungsi pengelolaan pengguna dan perangkat, *deployment* konfigurasi, *monitoring* status perangkat, pencatatan aktivitas, *history deployment*, *backup* konfigurasi, serta laporan. Pengujian dilakukan menggunakan metode *Black Box* dan *White Box*. Hasil pengujian menunjukkan bahwa fungsi utama sistem dapat berjalan sesuai dengan kebutuhan yang telah dirancang. Sistem mampu menjalankan penyebaran konfigurasi melalui koneksi SSH, menampilkan hasil konfigurasi, melakukan pencatatan aktivitas, serta membantu proses pengelolaan perangkat multivendor secara lebih terstruktur.

Kata Kunci: Otomatisasi Jaringan, Multivendor, Scripting, Python, Netmiko.

1. PENDAHULUAN

Jaringan komputer merupakan salah satu komponen penting dalam mendukung operasional organisasi karena menyediakan sarana pertukaran data dan informasi. Seiring meningkatnya kebutuhan terhadap infrastruktur jaringan, penggunaan perangkat dari berbagai vendor (*multivendor*) menjadi salah satu kondisi yang perlu dikelola oleh administrator jaringan. Perbedaan sintaks, standar, dan mekanisme konfigurasi antarperangkat dapat meningkatkan kompleksitas pengelolaan jaringan dan memperbesar kemungkinan terjadinya kesalahan ketika konfigurasi dilakukan secara manual [1].

Kondisi tersebut juga ditemukan pada Bagian Teknologi Informasi dan Komunikasi (TIK) Korlantas POLRI. Berdasarkan hasil observasi dan wawancara yang dilakukan dalam penelitian, infrastruktur jaringan yang digunakan melibatkan perangkat MikroTik dan Cisco. Proses konfigurasi serta backup perangkat sebelumnya dilakukan secara manual melalui CLI. Permasalahan yang ditemukan meliputi proses

konfigurasi yang membutuhkan waktu, adanya ketidaksesuaian pengaturan pada perangkat router, serta belum efektifnya proses konfigurasi dan backup perangkat MikroTik maupun Cisco. Infrastruktur jaringan yang tersebar juga menjadi salah satu faktor yang meningkatkan kebutuhan terhadap pengelolaan konfigurasi secara lebih terstruktur.

Salah satu pendekatan yang dapat digunakan untuk mengatasi pekerjaan konfigurasi yang bersifat berulang adalah *network automation*. Otomatisasi jaringan memungkinkan proses konfigurasi dan pengelolaan perangkat dilakukan melalui perangkat lunak atau skrip sehingga administrator tidak harus memasukkan perintah secara manual pada setiap perangkat. Penerapan otomasi pada lingkungan multivendor dapat membantu menjaga konsistensi proses konfigurasi serta mengurangi ketergantungan terhadap pekerjaan manual [2], [3].

Penelitian sebelumnya menunjukkan bahwa otomasi jaringan telah diterapkan menggunakan berbagai pendekatan. Ramadhan mengembangkan sistem network automation menggunakan Python Django pada perangkat Cisco dan MikroTik [4]. Firmansyah menerapkan network automation menggunakan Python dan GNS3 untuk mendukung proses otomasi konfigurasi [5]. Marzuki et al. menerapkan Ansible untuk mengotomasi konfigurasi VLAN, Inter-VLAN, dan DHCP sehingga proses konfigurasi dapat dilakukan secara lebih efisien dibandingkan pendekatan manual [6]. Penelitian lain juga menunjukkan penerapan otomasi pada pengelolaan perangkat jaringan *multivendor* serta penggunaan Python sebagai pendukung otomatisasi [7], [8].

Berdasarkan penelitian terdahulu tersebut, penelitian ini berfokus pada pengembangan sistem otomatisasi jaringan *multivendor* berbasis scripting yang mengintegrasikan aplikasi berbasis web dengan Python dan Netmiko. Sistem tidak hanya digunakan untuk melakukan deployment konfigurasi, tetapi juga menyediakan pengelolaan perangkat, monitoring status, pencatatan aktivitas, *history deployment, backup* konfigurasi, dan laporan. Kebaruan penelitian terletak pada pengintegrasian fungsi-fungsi tersebut dalam satu sistem berbasis web yang ditujukan untuk kebutuhan pengelolaan perangkat MikroTik dan Cisco pada lingkungan penelitian Korlantas POLRI.

Tujuan penelitian ini adalah mengembangkan sistem otomatisasi jaringan multivendor berbasis scripting yang dapat membantu proses konfigurasi dan backup perangkat MikroTik dan Cisco, menerapkan Python dalam proses otomasi, serta mendukung pengelolaan konfigurasi jaringan secara lebih terstruktur. Tujuan tersebut sesuai dengan batasan penelitian yang berfokus pada perangkat router MikroTik dan Cisco serta tidak mencakup pengamanan jaringan secara mendalam maupun monitoring performa jaringan secara menyeluruh.

2. TINJAUAN PUSTAKA

2.1 Network Automation

Network automation merupakan pendekatan yang memanfaatkan perangkat lunak, skrip, atau alat bantu untuk melakukan konfigurasi dan pengelolaan perangkat jaringan secara otomatis. Penggunaan otomasi ditujukan untuk mengurangi pekerjaan yang berulang dan membantu administrator menjalankan konfigurasi secara lebih konsisten. Dalam penelitian ini, otomasi digunakan untuk mendukung proses deployment konfigurasi dan backup pada perangkat jaringan multivendor.

2.2 Jaringan Multivendor

Jaringan *multivendor* merupakan lingkungan jaringan yang menggunakan perangkat dari lebih dari satu vendor. Perbedaan sistem operasi jaringan, sintaks perintah, serta mekanisme konfigurasi menyebabkan pengelolaan perangkat memerlukan pendekatan yang dapat menyesuaikan karakteristik masing-masing vendor. Penelitian ini membatasi perangkat yang digunakan pada router MikroTik dan Cisco.

2.3 Scripting dan Python

Scripting digunakan untuk menyusun serangkaian instruksi yang dapat dijalankan secara otomatis oleh komputer. Dalam konteks penelitian ini, Python digunakan sebagai bahasa pemrograman untuk menjalankan proses otomasi konfigurasi perangkat jaringan. Skrip menerima parameter perangkat dan konfigurasi dari aplikasi, kemudian menjalankan proses komunikasi dengan perangkat jaringan.

2.4 Netmiko

Netmiko digunakan sebagai *library* Python untuk membantu membangun koneksi SSH dengan perangkat jaringan. Dalam sistem yang dikembangkan, modul Python menerima data perangkat dari aplikasi web kemudian menggunakan Netmiko untuk melakukan komunikasi dengan perangkat sebelum menjalankan konfigurasi yang diberikan.

2.5 Penelitian Terdahulu

Ramadhan mengembangkan sistem *network automation* menggunakan Python Django dan Paramiko pada perangkat Cisco dan MikroTik. Penelitian tersebut menunjukkan bahwa otomasi dapat digunakan untuk melakukan konfigurasi perangkat *multivendor* secara terpusat [4].

Firmansyah mengembangkan penerapan *network* automation menggunakan GNS3 dan Python untuk otomasi konfigurasi pada lingkungan PT. Semut Merah Squad [5]. Sementara itu, Marzuki et al. menggunakan Ansible untuk mengotomasi konfigurasi VLAN, Inter-VLAN, dan DHCP pada perangkat jaringan [6].

Penelitian lain mengenai pengelolaan jaringan *multivendor* juga membahas penggunaan SSH dan Python untuk mendukung otomasi konfigurasi [7], [9]. Perbedaan penelitian ini dengan penelitian terdahulu terletak pada pengembangan sistem berbasis web yang mengintegrasikan proses deployment, monitoring, pencatatan aktivitas, backup, dan laporan dalam satu sistem untuk perangkat MikroTik dan Cisco.

3. METODE

3.1 Metode Penelitian

Penelitian ini menggunakan pendekatan studi lapangan dengan melakukan observasi langsung pada lingkungan Bagian TIK Korlantas POLRI. Data penelitian diperoleh melalui tiga teknik, yaitu studi pustaka, wawancara, dan observasi. Studi pustaka digunakan untuk memperoleh referensi yang berkaitan dengan otomasi jaringan, scripting, Python, dan pengelolaan jaringan multivendor. Wawancara dilakukan dengan staf atau teknisi TIK untuk memperoleh informasi mengenai kondisi infrastruktur serta proses konfigurasi yang berjalan. Observasi dilakukan dengan mengamati proses konfigurasi, pengelolaan, dan backup perangkat jaringan.

3.2 Metode Pengembangan Sistem

Pengembangan sistem menggunakan model Waterfall. Model ini dipilih karena proses pengembangan dilakukan secara bertahap dan terstruktur. Tahapan pengembangan terdiri atas:

1. *Requirement Analysis*, yaitu mengidentifikasi kebutuhan otomasi,

permasalahan konfigurasi manual, kebutuhan backup, dan kebutuhan efisiensi.

2. *System Design*, yaitu merancang sistem otomasi berbasis web dan struktur script Python.
3. *Implementation*, yaitu membangun aplikasi serta menerapkan script otomasi pada perangkat uji.
4. *Testing/Verification*, yaitu melakukan pengujian terhadap fungsi sistem dan script otomasi.
5. *Maintenance*, yaitu melakukan perbaikan atau pengembangan apabila ditemukan kesalahan maupun kebutuhan baru.

Tahapan *Waterfall* tersebut digunakan sebagai dasar dalam pengembangan sistem otomatisasi jaringan yang dibangun dalam penelitian.

3.3 Analisis Sistem Berjalan

Pada sistem berjalan, proses konfigurasi perangkat dilakukan secara manual melalui CLI. Administrator harus mengakses perangkat satu per satu dan memasukkan perintah konfigurasi sesuai dengan vendor perangkat. Kondisi tersebut menjadi lebih kompleks ketika konfigurasi harus diterapkan pada beberapa perangkat sekaligus.

Selain konfigurasi, proses backup juga memerlukan pengelolaan secara tersendiri. Berdasarkan permasalahan yang ditemukan, sistem berjalan memiliki beberapa kendala berupa kebutuhan waktu yang relatif tinggi, risiko kesalahan konfigurasi, serta kesulitan menjaga konsistensi konfigurasi pada perangkat multivendor.

3.4 Sistem yang Diusulkan

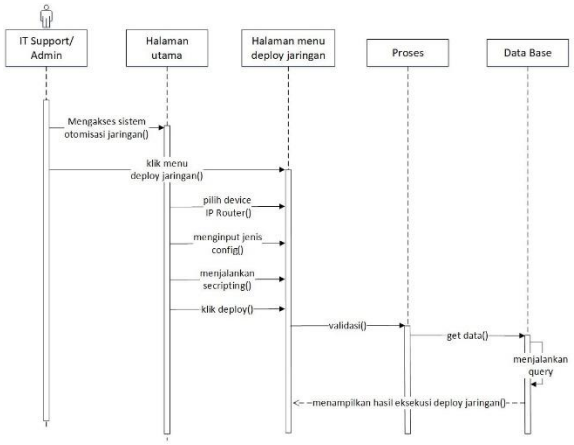
Sistem yang diusulkan berbentuk aplikasi berbasis web yang terhubung dengan modul Python. Pengguna sistem terdiri atas Administrator dan IT Support. Sistem menyediakan fungsi:

a. *Login*;
b. pengelolaan data pengguna;
c. pengelolaan data perangkat;
d. *deployment* konfigurasi;
e. monitoring status perangkat;
f. *history deployment*;
g. *backup* konfigurasi;
h. pencatatan aktivitas;
i. laporan konfigurasi.

Pada proses deployment, pengguna memilih perangkat, menentukan jenis konfigurasi, kemudian memasukkan konfigurasi yang akan dikirim. Data tersebut diteruskan ke modul Python. Selanjutnya, Python menggunakan Netmiko untuk membangun koneksi SSH dan menjalankan konfigurasi pada perangkat yang dipilih. Hasil proses dikembalikan ke aplikasi dan disimpan sebagai riwayat.

3.5 Arsitektur Implementasi

Sistem dikembangkan menggunakan PHP, Python, HTML, CSS, JavaScript, dan Bootstrap. Database yang digunakan adalah MySQL, sedangkan XAMPP digunakan sebagai *web server*. Python dan Netmiko digunakan untuk mendukung proses otomasi jaringan. Lingkungan virtualisasi jaringan menggunakan MikroTik CHR dan VirtualBox.



Gambar 1. Arsitektur Sistem Otomatisasi Jaringan Multivendor *(diambil/diadaptasi dari Gambar 3.18*

Sequence Diagram Deploy Jaringan pada skripsi)

Alur utama sistem dimulai dari proses autentikasi pengguna. Setelah berhasil masuk, pengguna dapat mengelola data perangkat atau menjalankan deployment. Pada proses deployment, sistem mengambil informasi perangkat dari database, meneruskan parameter kepada modul Python, membangun koneksi SSH, menjalankan konfigurasi, kemudian mengembalikan hasil proses kepada aplikasi.

3.6 Pengujian Sistem

Pengujian dilakukan menggunakan metode *Black Box* dan *White Box*. *Black Box Testing* digunakan untuk memeriksa fungsi sistem berdasarkan masukan dan keluaran tanpa memperhatikan struktur kode internal. Pengujian mencakup login, logout, data perangkat, *deployment* konfigurasi, dan *monitoring* status jaringan.

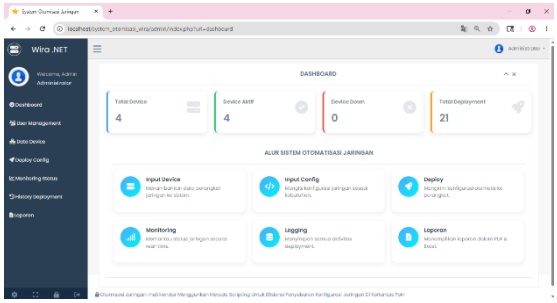
Sementara itu, *White Box Testing* digunakan untuk memeriksa struktur internal, logika, dan jalur eksekusi program. Pengujian ini digunakan untuk menganalisis beberapa fungsi utama, termasuk proses *login* dan *deployment* konfigurasi.

4. HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

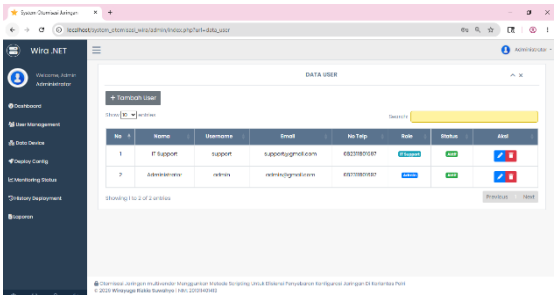
Hasil implementasi menghasilkan aplikasi otomatisasi jaringan *multivendor* berbasis web. Antarmuka sistem menyediakan halaman *login*, *dashboard*, data pengguna, data perangkat, *deployment* konfigurasi, monitoring, *history deployment*, dan laporan.

Halaman *dashboard* menampilkan informasi utama berupa jumlah perangkat, perangkat aktif, perangkat *down*, dan jumlah *deployment* konfigurasi. Menu yang tersedia digunakan untuk mendukung pengelolaan perangkat dan proses otomatisasi jaringan.



Gambar 2. Halaman Dashboard Sistem

Halaman data perangkat digunakan untuk menyimpan dan mengelola informasi perangkat, termasuk nama perangkat, IP Address, vendor, lokasi, dan status perangkat. Pengguna dapat melakukan penambahan, perubahan, pencarian, serta penghapusan data perangkat.



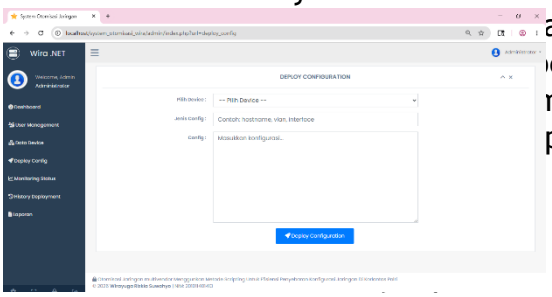
Gambar 3. Halaman Data *Device*

4.2 Implementasi Deployment Konfigurasi

Fitur *deployment* merupakan bagian utama sistem yang digunakan untuk mengirim konfigurasi ke perangkat jaringan. Pengguna memilih perangkat, menentukan jenis konfigurasi, memasukkan script konfigurasi, kemudian menjalankan proses *deploy*.

Sistem selanjutnya meneruskan data tersebut ke modul Python. Modul tersebut menggunakan Netmiko untuk

membangun koneksi SSH dengan perangkat. Apabila proses koneksi dan konfigurasi berhasil, sistem menampilkan hasil konfigurasi dan menyimpan informasi proses ke dalam *database*.

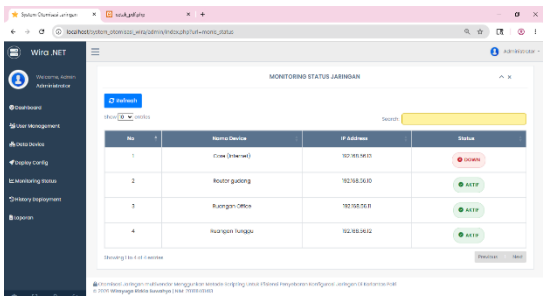


Gambar 4. Halaman *Deploy* Konfigurasi Jaringan

Implementasi ini menunjukkan bahwa *scripting* tidak berdiri sendiri, tetapi terintegrasi dengan aplikasi web dan database. Aplikasi berfungsi sebagai antarmuka serta pengelola data, sedangkan Python berperan dalam pelaksanaan otomasi konfigurasi pada perangkat jaringan.

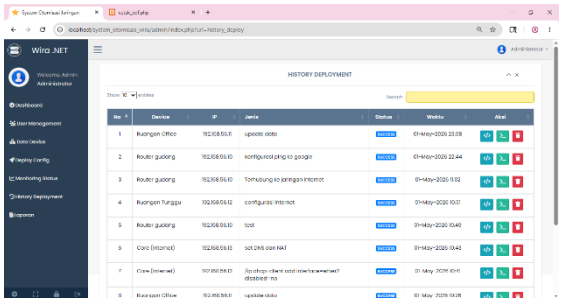
4.3 Implementasi *Monitoring*

Sistem juga menyediakan halaman monitoring yang digunakan untuk memeriksa kondisi perangkat berdasarkan *IP Address*. Hasil pemeriksaan ditampilkan dalam bentuk status perangkat, seperti *online* atau *down*. Selain itu, sistem melakukan pengecekan terhadap koneksi SSH perangkat.



Gambar 5. Halaman *Monitoring* Status Jaringan

Perlu diperhatikan bahwa monitoring pada penelitian ini tidak dimaksudkan sebagai *monitoring* performa jaringan secara menyeluruh atau *real-time* arameter enelitian n tidak performa riwayat deployment yang telah dilakukan. Informasi yang ditampilkan meliputi nama perangkat, IP Address, jenis konfigurasi, status deployment, waktu proses, dan detail konfigurasi.



Gambar 6. Halaman *History Deployment*

Sistem juga menyediakan halaman laporan yang dapat digunakan untuk menampilkan data *deployment* berdasarkan tanggal. Laporan dapat dicari dan diekspor, termasuk dalam bentuk Excel dan PDF. Informasi laporan mencakup perangkat, *IP Address*, jenis konfigurasi, pengguna, status *deployment*, waktu proses, konfigurasi yang dikirim, dan hasil keluaran perangkat.

4.5 Hasil Pengujian *Black Box*

Hasil pengujian *Black Box* menunjukkan bahwa fungsi utama sistem dapat

merespons masukan sesuai dengan skenario yang telah ditentukan. Pengujian mencakup autentikasi, pengelolaan perangkat, *deployment* konfigurasi, dan *monitoring*.

Tabel 1. Ringkasan Pengujian *Black Box*

No.	Fitur	Skenario Pengujian	Hasil Pengujian
1	Login	Data autentikasi valid	Berhasil masuk ke *dashboard*
2	Login	Data autentikasi tidak valid	Sistem menolak autentikasi dan memberikan pesan kesalahan
3	Logout	Pengguna mengakhiri sesi	Sistem mengakhiri sesi dan kembali ke halaman *login*
4	Data Device	Menambah data perangkat	Data berhasil disimpan
5	Data Device	Data tidak lengkap	Sistem memberikan validasi
6	Data Device	Mengubah data	Data berhasil diperbarui
7	Data Device	Menghapus data	Data berhasil dihapus
8	Deployment	Konfigurasi benar	Konfigurasi berhasil dikirim
9	Deployment	Konfigurasi tidak diisi	Sistem memberikan validasi
10	Deployment	Perangkat *offline*	Sistem menampilkan kegagalan koneksi
11	Monitoring	Perangkat aktif	Status *online* ditampilkan
12	Monitoring	Perangkat tidak aktif	Status *offline* ditampilkan
13	Monitoring	IP Address kosong	Sistem memberikan validasi
14	Monitoring	*Refresh* status	Status perangkat diperbarui
15	Monitoring	Pemeriksaan SSH	Status koneksi SSH ditampilkan

Data pengujian tersebut berasal dari skenario pengujian *Black Box* dalam skripsi. Pada pengujian deployment, sistem berhasil mengirim konfigurasi pada skenario yang benar, memberikan validasi ketika konfigurasi tidak diisi, dan menolak proses ketika perangkat berada dalam kondisi *offline*.

Pada pengujian *monitoring*, sistem berhasil menampilkan status *online* dan *offline*, memberikan validasi ketika IP Address tidak tersedia, memperbarui status melalui *refresh*, serta menampilkan status koneksi SSH.

4.6 Hasil Pengujian White Box

Pengujian *White Box* digunakan untuk menelusuri jalur logika internal program. Salah satu pengujian dilakukan terhadap proses login dengan nilai *cyclomatic complexity* sebesar 3 berdasarkan perhitungan:

$$V(G) = E - N + 2$$

$$V(G) = 7 - 6 + 2 = 3$$

Hasil tersebut menunjukkan terdapat tiga jalur independen yang dianalisis pada fungsi login. Pengujian juga dilakukan terhadap fungsi *deployment* konfigurasi jaringan. Fungsi tersebut memiliki nilai *cyclomatic complexity* sebesar 3 dengan tiga jalur independen.

Pada fungsi *deployment*, jalur proses meliputi kondisi ketika tombol *deploy* tidak dijalankan, kondisi ketika data perangkat atau konfigurasi tidak memenuhi persyaratan, serta kondisi ketika data sesuai sehingga sistem menjalankan proses *deployment* melalui koneksi SSH dan menyimpan hasilnya ke *database*.

4.7 Pembahasan

Hasil implementasi menunjukkan bahwa pendekatan *scripting* dapat digunakan untuk mengotomatisasi proses konfigurasi pada perangkat *multivendor*. Integrasi antara aplikasi *web*, Python, Netmiko, dan *database* membentuk alur kerja yang memungkinkan pengguna melakukan konfigurasi melalui satu antarmuka tanpa harus memasukkan perintah secara manual pada setiap perangkat.

Hasil tersebut sejalan dengan penelitian Ramadhan yang menerapkan Python untuk mengotomatisasi konfigurasi perangkat Cisco dan MikroTik [4]. Penelitian Firmansyah juga menunjukkan pemanfaatan Python dalam *network automation* pada lingkungan simulasi GNS3 [5]. Sementara itu, penelitian Marzuki et al. menunjukkan bahwa otomasi dapat digunakan untuk mengurangi pekerjaan konfigurasi berulang pada perangkat jaringan [6].

Perbedaan utama penelitian ini terletak pada integrasi beberapa fungsi dalam satu aplikasi berbasis *web*. Selain *deployment*, sistem menyediakan pengelolaan perangkat, *monitoring* status, *history deployment*, *backup*, pencatatan aktivitas, dan laporan. Dengan demikian, sistem tidak hanya berfungsi sebagai eksekutor konfigurasi, tetapi juga sebagai sarana pengelolaan informasi terkait proses konfigurasi jaringan.

Meskipun demikian, hasil penelitian belum dapat digunakan untuk menyatakan besaran peningkatan efisiensi secara kuantitatif karena penelitian tidak menyajikan pengukuran waktu konfigurasi sebelum dan sesudah penerapan sistem dalam bentuk data statistik atau persentase. Oleh sebab itu, efisiensi dalam penelitian ini dipahami berdasarkan kemampuan sistem menyederhanakan dan mengotomatisasi proses konfigurasi yang sebelumnya dilakukan secara manual.

Selain itu, penerapan sistem masih memiliki batasan. Penelitian hanya mencakup router MikroTik dan Cisco, sedangkan perangkat jaringan lain seperti switch layer 3, firewall khusus, dan perangkat *wireless* tidak menjadi objek penelitian. Pengujian juga dilakukan pada lingkungan simulasi/lab yang mewakili kondisi jaringan penelitian dan tidak diterapkan pada seluruh perangkat produksi.

5. KESIMPULAN

Berdasarkan hasil penelitian dan implementasi sistem, dapat disimpulkan bahwa sistem otomatisasi jaringan *multivendor* berbasis *scripting* berhasil dikembangkan untuk mendukung proses penyebaran konfigurasi jaringan pada lingkungan penelitian Korlantas POLRI. Sistem mengintegrasikan aplikasi berbasis *web* dengan Python dan Netmiko untuk menjalankan komunikasi SSH serta proses *deployment* konfigurasi pada perangkat MikroTik dan Cisco. Sistem yang dikembangkan menyediakan fungsi pengelolaan pengguna dan perangkat, *deployment* konfigurasi, *monitoring* status perangkat, *history deployment*, *backup*, pencatatan aktivitas, serta laporan. Hasil pengujian menggunakan *Black Box* dan *White Box* menunjukkan bahwa fungsi-fungsi utama sistem dapat berjalan sesuai dengan skenario dan rancangan yang ditetapkan.

Penerapan *scripting* membantu mengurangi ketergantungan terhadap proses konfigurasi manual dan menyediakan mekanisme konfigurasi yang lebih terstruktur. Namun, penelitian ini belum menghasilkan pengukuran kuantitatif mengenai persentase atau besaran peningkatan efisiensi waktu, sehingga klaim efisiensi dibatasi pada hasil implementasi dan pengujian fungsional sistem. Pengembangan selanjutnya dapat diarahkan pada penambahan monitoring performa jaringan secara *real-time*, notifikasi otomatis ketika terjadi gangguan atau kegagalan konfigurasi, mekanisme *backup* dan *restore* yang lebih terintegrasi, serta fitur *scheduler* untuk menjalankan *deployment* dan *backup* secara terjadwal.

DAFTAR PUSTAKA

[1] M. Anwar *et al.*, "Implementasi Network Automation Menggunakan Python pada Infrastruktur Jaringan Enterprise," *Jurnal Teknologi Informasi dan Komunikasi*, vol. 8, no. 1, pp. 12–20, 2022.

[2] I. Ardana *et al.*, "Manajemen Jaringan Komputer pada Lingkungan Multivendor," *Jurnal Teknologi Informasi dan Komunikasi Indonesia*, vol. 7, no. 2, pp. 45–53, 2022.

[3] D. Cahyono and R. Saputra, "Perancangan Sistem Otomasi Konfigurasi Router Menggunakan Metode Scripting," *Jurnal Sistem Komputer dan Informatika*, vol. 5, no. 2, pp. 77–85, 2023.

[4] M. Dwiyaniti and N. Nadhiroh, "Implementasi SCADA pada Sistem PLC Multivendor Berbasis Modbus TCP/IP," *Seminar Nasional Teknik Elektro*, vol. 11, no. 1, 2025.

[5] M. A. A. Fachruddin, "Implementasi Network Automation pada Aplikasi Administrasi Jaringan Multi-Vendor Berbasis Web," Institut Teknologi Indonesia, 2023.

[6] I. Firmansyah, "The Implementasi Network Automation pada GNS3 untuk Otomasi Konfigurasi di PT. Semut Merah Squad," *Buletin Ilmiah Informatika Teknologi*, vol. 2, no. 2, pp. 43–48, 2024.

[7] A. Gunawan *et al.*, "Implementasi Manajemen Jaringan Berbasis SSH pada Lingkungan Multivendor," *Jurnal Informatika Terapan*, vol. 6, no. 1, pp. 33–41, 2022.

[8] A. Hidayat, "Implementasi Secure Shell (SSH) dalam Pengelolaan Perangkat Jaringan," *Jurnal Informatika Nusantara*, vol. 5, no. 1, pp. 21–28, 2022.

[9] R. Manalu *et al.*, "Analisis Efisiensi Konfigurasi Jaringan pada Infrastruktur Multivendor," *Jurnal Sistem Informasi dan Teknologi Komputer*, vol. 8, no. 1, pp. 11–20, 2024.

[10] K. Marzuki *et al.*, "Otomasisasi Manajemen VLAN InterVLAN dan DHCP Server Menggunakan Ansible," *Jurnal*

Informatika & Rekayasa Elektronika, vol. 4, no. 2, pp. 171–180, 2021.

[11] F. Maulana *et al.*, "Analisis Efektivitas Network Automation pada Pengelolaan Perangkat Cisco dan MikroTik," *Jurnal Sistem Informasi dan Teknologi*, vol. 9, no. 1, pp. 21–29, 2024.

[12] F. Nisa and S. Ramadona, "Sistem Pencegahan Serangan Distributed Denial of Service pada Jaringan SDN," *Jurnal Sistim Informasi dan Teknologi*, pp. 22–30, 2023.

[13] Y. Nugraha and A. Prasetyo, "Implementasi Backup Konfigurasi Otomatis pada Perangkat Jaringan Menggunakan Python," *Jurnal Rekayasa Perangkat Lunak*, vol. 4, no. 3, pp. 101–109, 2023.

[14] B. Prasetyo *et al.*, "Pengembangan Sistem Monitoring dan Otomasi Jaringan Berbasis Web," *Jurnal Teknologi Informasi Modern*, vol. 8, no. 2, pp. 45–53, 2022.

[15] F. Rahman, "Implementasi SFTP dan SCP untuk Backup Konfigurasi Perangkat Jaringan," *Jurnal Teknologi Komputer Indonesia*, vol. 6, no. 3, pp. 88–96, 2023.

[16] T. Ramadhan, "Rancang Bangun Sistem Network Automation Menggunakan Python Django pada Perangkat Jaringan Multivendor," Politeknik Negeri Jakarta, Jakarta, 2023.

[17] H. Saputra, "Penerapan Library Netmiko untuk Otomasi Konfigurasi Perangkat Jaringan," *Jurnal Ilmu Komputer dan Teknologi Informasi*, vol. 5, no. 1, pp. 14–22, 2023.

[18] A. Sitepu *et al.*, "Implementasi Otomasi Jaringan pada Infrastruktur Teknologi Informasi Pemerintahan," *Jurnal Teknologi Informasi dan Komunikasi*, vol. 9, no. 2, pp. 55–64, 2023.

[19] E. Utomo *et al.*, "Implementasi Network Automation Berbasis Web pada Infrastruktur Jaringan Komputer," *Jurnal Informatika dan Sistem Informasi*, vol. 10, no. 1, pp. 61–69, 2024.

[20] M. Yusuf and F. Ramadhan, "Analisis Kinerja Otomasi Jaringan Menggunakan Python dan Netmiko pada Lingkungan Multivendor," *Jurnal Teknologi Informasi dan Komunikasi Indonesia*, vol. 9, no. 2, pp. 88–96, 2024.