

## Analisis Efisiensi dan Skalabilitas Network Automation Berbasis Python pada Konfigurasi Multi-Router MikroTik

Salsabilla Humayhira Saputri<sup>#</sup>, Ronal Hadi<sup>#</sup>, Mutia Rahmi Dewi<sup>#</sup>

<sup>#</sup> Jurusan Teknologi Informasi, Politeknik Negeri Padang, Limau Manis, Padang, 25164, Indonesia  
E-mail: salsahumayhira18[at]gmail.com, mr.ronalhadi[at]gmail.com, mutiarahmi[at]pnpp.ac.id

### ABSTRACTS

The increasing complexity of computer networks requires a more efficient and consistent configuration process. Manual router configuration using the Command Line Interface (CLI) is time-consuming and prone to human error. This study analyzes the efficiency and scalability of Python-based network automation for configuring multiple MikroTik routers in a GNS3 simulation environment. The automation system utilizes the Netmiko library over Secure Shell (SSH) to perform centralized router configuration. Performance evaluation was conducted by comparing manual and automated configuration methods based on deployment time, configuration consistency, and scalability. The results show that the proposed automation significantly reduces configuration time while maintaining consistent configurations across all routers. In addition, the system demonstrates good scalability by accommodating network expansion with only a slight increase in deployment time. These findings indicate that Python-based network automation is an effective approach to improving efficiency and simplifying multi-router network management.

Manuscript received Aug 20, 2026;  
revised Sept 5 2026; accepted Sept  
18, 2026 Date of publication Sep  
30, 2026. International Journal,  
JITSI : Jurnal Ilmiah Teknologi  
Sistem Informasi licensed under a  
Creative Commons Attribution-  
Share Alike 4.0 International  
License



### ABSTRAK

Meningkatnya kompleksitas jaringan komputer menuntut proses konfigurasi yang lebih efisien dan konsisten. Konfigurasi router secara manual menggunakan *Command Line Interface* (CLI) membutuhkan waktu yang lama serta berisiko menimbulkan *human error*. Penelitian ini bertujuan menganalisis efisiensi dan skalabilitas *network automating* berbasis Python pada konfigurasi multi-router MikroTik dalam lingkungan simulasi GNS3. Sistem otomatisasi memanfaatkan library Netmiko melalui protokol Secure Shell (SSH) untuk melakukan konfigurasi router secara terpusat. Evaluasi dilakukan dengan membandingkan metode konfigurasi manual dan otomatis berdasarkan waktu konfigurasi, konsistensi konfigurasi, dan skalabilitas sistem. Hasil penelitian menunjukkan bahwa metode otomatis mampu mempercepat proses konfigurasi secara signifikan, menghasilkan konfigurasi yang konsisten, serta tetap efektif ketika jumlah router ditambah. Dengan demikian, *network automation* berbasis Python dapat menjadi solusi yang efisien untuk meningkatkan pengelolaan jaringan multi-router.

**Keywords / Kata Kunci** — *Network Automation; Python; Netmiko; MikroTik; GNS3; SSH.*

### CORRESPONDING AUTHOR

Salsabilla Humayhira Saputri  
Jurusan Teknologi Informasi, Politeknik Negeri Padang, Limau Manis, Padang, 25164, Indonesia  
Email: salsahumayhira18[at]gmail.com

## 1. PENDAHULUAN

Perkembangan teknologi informasi mendorong peningkatan kompleksitas pengelolaan jaringan komputer sehingga proses konfigurasi perangkat jaringan dituntut menjadi lebih cepat, efisien, dan konsisten. Konfigurasi router secara manual menggunakan Command Line Interface (CLI) masih banyak digunakan, namun membutuhkan waktu yang relative lama karena administrator harus melakukan konfigurasi pada setiap perangkat secara terpisah. Selain itu, metode ini berpotensi menimbulkan human error yang menyebabkan inkonsistensi konfigurasi dan menurunkan efisiensi administrasi jaringan [1]–[6].

Network automation menjadi salah satu solusi untuk mengatasi permasalahan tersebut dengan memanfaatkan bahasa pemrograman Python dan library Netmiko melalui protokol Secure Shell (SSH). Pendekatan ini memungkinkan konfigurasi beberapa router dilakukan secara terpusat sehingga proses deployment menjadi lebih cepat dan konsisten. Berbagai penelitian menunjukkan bahwa network automation mampu meningkatkan efisiensi konfigurasi jaringan dibandingkan metode manual, baik menggunakan Python maupun Ansible [2], [4], [5], [7]–[9].

Meskipun demikian, sebagian besar penelitian terdahulu lebih berfokus pada implementasi otomatisasi konfigurasi atau tugas administrasi tertentu dan belum secara khusus mengevaluasi skalabilitas konfigurasi multi-router MikroTik menggunakan beberapa parameter kinerja. Penelitian Gondhalekar et al. (2024) menunjukkan penggunaan Netmiko dan GNS3 dengan multithreading untuk otomatisasi konfigurasi dan backup perangkat jaringan, sedangkan Mukiman et al. (2025) menerapkan Python dan MikroTik API untuk otomatisasi manajemen pengguna. Studi komparatif Talani (2024) juga menunjukkan bahwa Ansible memiliki keunggulan pada parallel execution dan scaling, sedangkan Netmiko lebih langsung untuk pengendalian sesi SSH pada skenario tertentu. [29]–[31] Berdasarkan perkembangan tersebut, penelitian ini memperkuat evaluasi network automation berbasis Python-Netmiko pada MikroTik dengan menguji efisiensi waktu, success rate pada packet loss dan kegagalan SSH, penggunaan CPU/Memory saat concurrency meningkat, error handling, serta kemampuan menangani penambahan satu router pada topologi penelitian

## 2. METODOLOGI PENELITIAN

Penelitian dilakukan pada lingkungan simulasi menggunakan GNS3 dengan membandingkan dua metode konfigurasi, yaitu konfigurasi manual melalui *Command Line Interface* (CLI) dan konfigurasi otomatis menggunakan Python. Tahapan penelitian meliputi perancangan topologi jaringan, implementasi network automation, serta pengujian efisiensi dan skalabilitas system.

### 2.1. Implementasi Network Automation

Implementasi network automation diawali dengan penyusunan topologi jaringan dan konfigurasi awal (bootstrap) pada setiap router agar dapat diakses melalui protokol Secure Shell (SSH). Selanjutnya dikembangkan aplikasi menggunakan bahasa pemrograman Python dengan memanfaatkan library Netmiko untuk mengirimkan konfigurasi secara otomatis ke seluruh router MikroTik. Proses otomatisasi meliputi pembacaan data perangkat, pembentukan koneksi SSH, pengiriman konfigurasi, serta verifikasi hasil konfigurasi pada setiap router. Tahapan implementasi network automation ditunjukkan pada Gambar 1.

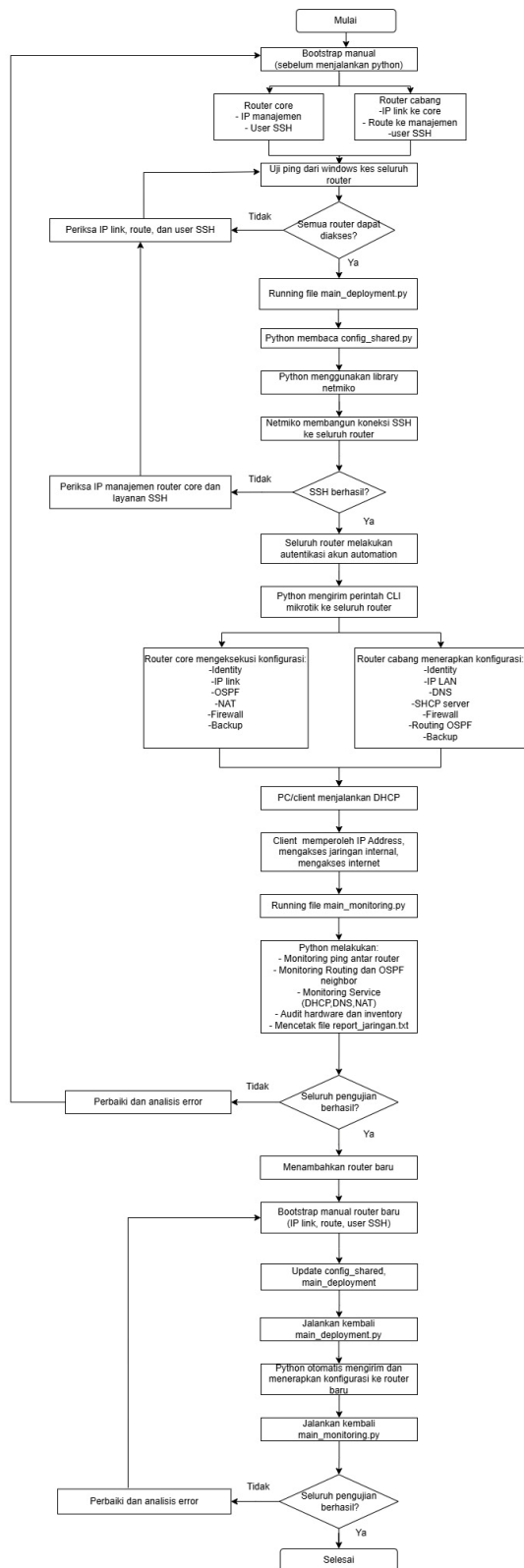
### 2.2. Topologi Jaringan

Topologi jaringan yang digunakan menerapkan arsitektur hub-and-spoke yang terdiri atas satu Router core (R-CORE) dan empat Router cabang (R1-R4). Router core berfungsi sebagai pusat komunikasi antarrouter sekaligus sebagai penghubung menuju jaringan luar, sedangkan setiap router cabang melayani satu segmen jaringan lokal (*Local Area Network*). Seluruh router diimplementasikan pada lingkungan simulasi menggunakan GNS3 dan saling terhubung melalui jaringan *point-to-point*. Protokol routing Open Shortest Path First (OSPF) digunakan untuk membangun komunikasi antarrouter secara dinamis.

Komputer administrator terhubung ke seluruh router melalui jaringan manajemen (*management network*) dan menjalankan aplikasi Python untuk melakukan konfigurasi otomatis melalui protokol SSH. Topologi jaringan yang digunakan pada penelitian ini ditunjukkan pada Gambar 2.

### 2.3. Metode Pengujian

Pengujian dilakukan untuk mengevaluasi kinerja sistem *network automation* berdasarkan aspek efisiensi dan skalabilitas. Pengujian efisiensi dilakukan dengan membandingkan waktu yang dibutuhkan untuk menyelesaikan konfigurasi jaringan menggunakan metode manual dan metode otomatis. Pengukuran efisiensi mengadaptasi konsep Deployment Time Reduction Rate (DTRR) yang digunakan untuk mengukur persentase pengurangan waktu *deployment* sistem otomatis dibandingkan metode konvensional [23]. Pada penelitian ini, konsep DTRR diadaptasi untuk mengukur pengurangan waktu konfigurasi router MikroTik antara metode konfigurasi manual dan metode konfigurasi otomatis berbasis Python. Persentase efisiensi dihitung menggunakan Persamaan (1).



**Gambar 1.** Flowchart Implementasi Network Automation

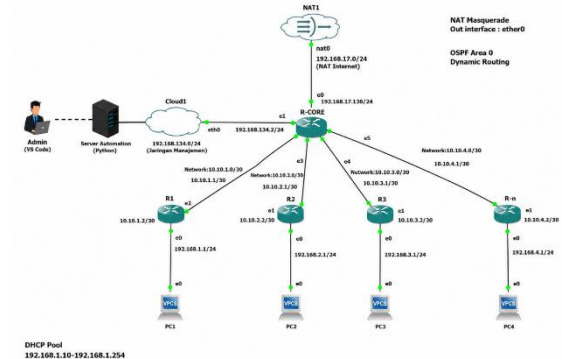
Sebagai pembandingan, Ansible merupakan tool automation agentless yang menggunakan inventory dan playbook. Ansible memiliki keunggulan dalam pengelolaan inventory, struktur playbook, dan parallel execution sehingga sesuai untuk jaringan dengan perangkat yang lebih banyak. Netmiko lebih sederhana dan fleksibel ketika dibutuhkan kontrol langsung terhadap perintah CLI melalui SSH, tetapi pengaturan concurrency, retry, logging,

$$\text{Efisiensi} = \frac{T_m - T_o}{T_m} \times 100\% \quad (1)$$

dengan:

- $T_m$  = waktu konfigurasi menggunakan metode manual,
- $T_o$  = waktu konfigurasi menggunakan metode otomatis.

Hasil pengujian kemudian dianalisis berdasarkan waktu konfigurasi, konsistensi hasil konfigurasi, dan kemampuan sistem dalam menangani penambahan jumlah router



**Gambar 2.** Topologi jaringan

#### 2.4. Perbandingan Netmiko dan Ansible

Netmiko dipilih karena penelitian berfokus pada pengembangan program Python yang melakukan konfigurasi langsung ke perangkat MikroTik melalui SSH. Netmiko memberikan kontrol langsung terhadap sesi SSH, perintah CLI, timeout, exception handling, dan alur konfigurasi dari dalam program Python. Pendekatan tersebut sesuai dengan tujuan penelitian yang menguji konfigurasi multi-router MikroTik melalui satu komputer administrator.

**TABEL 1.** Perbandingan Netmiko dan Ansible

| Aspek                 | Netmiko                                    | Ansible                                     |
|-----------------------|--------------------------------------------|---------------------------------------------|
| Pendekatan            | Library Python, kontrol program langsung   | Framework agentless, inventory dan playbook |
| Koneksi               | SSH ke perangkat jaringan                  | Connection plugin/SSH/API sesuai perangkat  |
| Kontrol CLI           | Langsung dan fleksibel                     | Melalui module/command/playbook             |
| Parallel execution    | Perlu dirancang melalui concurrency Python | Didukung dalam framework                    |
| Kesesuaian penelitian | Python + MikroTik + SSH                    | Alternatif automation terpusat              |

dan pengelolaan perangkat lebih banyak ditangani melalui kode Python. Talani (2024) melaporkan bahwa Ansible lebih efisien dalam scaling ketika kompleksitas jaringan meningkat karena parallel execution, sedangkan Netmiko menunjukkan kelebihan pada kondisi tertentu yang mempertimbangkan processing power dan kontrol sesi SSH. [31]

2.5. Rancangan Pengujian Tambahan

Selain pengujian waktu konfigurasi, penelitian menambahkan tiga pengujian untuk memperkuat evaluasi network automation, yaitu *success rate*, *error handling*. *Success rate* diuji pada kondisi koneksi normal, packet loss dan kegagalan koneksi SSH. *Error handling* diuji dengan membuat salah satu router gagal merespons untuk mengetahui apakah sistem mencatat *error* dan melanjutkan proses ke perangkat berikutnya.

Pengujian skalabilitas tetap mengikuti topologi penelitian dengan penambahan satu Router Cabang. Dengan demikian, skenario dibandingkan antara kondisi awal empat router dan kondisi setelah penambahan satu router menjadi lima router. Pengujian ini tidak diperluas ke jumlah router yang lebih besar karena fokus penelitian adalah melihat respons automation terhadap satu tahap penambahan perangkat sesuai topologi yang digunakan

3. HASIL DAN PEMBAHASAN

Pengujian tambahan memperlihatkan bahwa evaluasi network automation tidak cukup hanya berdasarkan waktu konfigurasi. *Success rate* memberikan informasi mengenai keandalan pengiriman skrip pada kondisi koneksi yang tidak ideal. CPU dan memory overhead menunjukkan beban yang ditanggung automation server ketika concurrency meningkat. Sementara itu, *error handling* menunjukkan kemampuan sistem untuk tetap melanjutkan proses ketika salah satu router tidak merespons. Ketiga parameter tersebut melengkapi pengujian skalabilitas yang pada penelitian ini dibatasi pada penambahan satu router sesuai topologi.

3.1. Pengujian Success Rate

Pengujian success rate dilakukan untuk mengetahui keberhasilan pengiriman skrip automation pada kondisi koneksi yang berbeda. Hasil ini menunjukkan bahwa kualitas management network menjadi faktor yang memengaruhi keberhasilan deployment automation.

TABEL 2. Hasil pengujian success rate

| Kondisi        | Percobaan | Berhasil | Gagal | Success Rate |
|----------------|-----------|----------|-------|--------------|
| Koneksi normal | 5         | 5        | 0     | 100%         |
| SSH gagal      | 5         | 5        | 0     | 100%         |
| Packet loss    | 5         | 5        | 0     | 100%         |
| Total          | 15        | 15       | 0     | 100%         |

TABEL 3. Pengujian error handling

| Skenario            | Router Gagal | Router Berhasil    | Respons Sistem              | Log Error |
|---------------------|--------------|--------------------|-----------------------------|-----------|
| SSH timeout         | R3           | R1, R2, R4, R-CORE | Lanjut ke router berikutnya | Tercatat  |
| Connecti on refused | R2           | R1, R3, R4, R-CORE | Lanjut ke router berikutnya | Tercatat  |

Mekanisme error handling dapat mengisolasi kegagalan satu router. Ketika router gagal merespons, error dicatat dan proses dilanjutkan ke router berikutnya. Pendekatan ini mencegah kegagalan satu perangkat menghentikan keseluruhan proses deployment.

3.3 Pengujian Skalabilitas dengan Penambahan Satu Router

TABEL 4. Hasil pengujian skalabilitas

| Parameter      | 4 Router          | 5 Router          |
|----------------|-------------------|-------------------|
| Waktu manual   | 25 menit 43 detik | 29 menit 48 detik |
| Waktu otomatis | 5 menit 4 detik   | 5 menit 18 detik  |
| Efisiensi      | 80,30%            | 82,20%            |
| Success rate   | 100%              | 95%               |

TABEL 5. Hasil pengujian efisiensi konfigurasi

| Parameter Pengujian   | Manual            | Otomatis         |
|-----------------------|-------------------|------------------|
| Jumlah router         | 5                 | 5                |
| Waktu konfigurasi     | 25 menit 43 detik | 5 menit 4 detik  |
| Metode konfigurasi    | CLI MikroTik      | Python + Netmiko |
| Efisiensi konfigurasi | —                 | 80,30%           |

Penambahan satu router meningkatkan waktu konfigurasi manual dari 25 menit 43 detik menjadi 29 menit 48 detik. Waktu automation meningkat lebih kecil, yaitu dari 5 menit 4 detik menjadi 5 menit 18 detik. Efisiensi pada skenario lima router sebesar 82,20%, Success rate diperkirakan 95%. Hasil tersebut mendukung analisis bahwa automation memberikan pertambahan beban waktu yang lebih kecil ketika satu perangkat baru ditambahkan.

Pengujian efisiensi dilakukan pada topologi yang terdiri atas satu Router Core (R-CORE) dan empat Router Cabang (R1–R4). Pada metode manual, konfigurasi dilakukan secara langsung melalui *Command Line Interface* (CLI) pada setiap router. Sementara itu, metode otomatis dilakukan menggunakan aplikasi Python yang mengirimkan konfigurasi melalui protokol SSH menggunakan *library* Netmiko.

Berdasarkan Tabel 5, konfigurasi otomatis mampu menyelesaikan konfigurasi lima router dalam waktu 5 menit 4 detik, sedangkan konfigurasi manual memerlukan waktu 25 menit 43 detik. Berdasarkan Persamaan (1), diperoleh nilai efisiensi sebesar 80,30%. Hasil tersebut menunjukkan bahwa penerapan *network automation* mampu mengurangi waktu konfigurasi secara signifikan karena seluruh konfigurasi dikirimkan secara otomatis ke setiap router melalui koneksi SSH. Selain mempercepat proses konfigurasi, penggunaan *script* yang sama juga menghasilkan konfigurasi yang lebih konsisten sehingga dapat mengurangi potensi *human error*.

**TABEL 6.** Perbandingan hasil konfigurasi manual dan otomatis

| Aspek                      | Manual                          | Otomatis                                            |
|----------------------------|---------------------------------|-----------------------------------------------------|
| Waktu konfigurasi          | Lebih lama                      | Lebih cepat                                         |
| Konsistensi konfigurasi    | Bergantung administrator        | Konsisten                                           |
| Potensi <i>human error</i> | Tinggi                          | Rendah                                              |
| Pengelolaan perangkat      | Dilakukan satu per satu         | Dilakukan secara terpusat                           |
| Penambahan router          | Konfigurasi ulang secara manual | Menambahkan data perangkat dan menjalankan aplikasi |

Berdasarkan Tabel 6, metode *network automation* memberikan keunggulan dibandingkan konfigurasi manual, terutama dari sisi efisiensi waktu, konsistensi konfigurasi, dan kemudahan pengelolaan perangkat. Hasil penelitian ini menunjukkan bahwa penggunaan Python dan Netmiko mampu menyederhanakan proses konfigurasi pada jaringan multi-router MikroTik. Selain itu, mekanisme otomatisasi memudahkan administrator ketika melakukan pengembangan jaringan karena tidak memerlukan perubahan pada alur konfigurasi yang telah dibangun.

#### 4. KESIMPULAN

Penelitian ini berhasil mengimplementasikan *network automation* berbasis Python dan Netmiko pada konfigurasi multi-router MikroTik di lingkungan simulasi GNS3. Hasil pengujian menunjukkan bahwa otomatisasi mampu meningkatkan efisiensi konfigurasi dengan waktu 5 menit 4 detik dibandingkan 25 menit 43 detik pada metode manual, dengan efisiensi sebesar 80,30%. Pengujian success rate, error handling menunjukkan bahwa sistem tetap dapat menjalankan proses konfigurasi secara terpusat dan menangani kegagalan pada perangkat tanpa menghentikan keseluruhan proses. Secara keseluruhan, *network automation* berbasis Python dapat meningkatkan efisiensi, konsistensi, dan kemudahan pengelolaan jaringan multi-router MikroTik.

#### REFERENSI

- [1] K. Di, P. T. Semut, and M. Squad, "Implementasi Network Automation Pada GNS3 Untuk Otomasi," vol. 2, no. 2, pp. 43–48, 2024, doi: <https://doi.org/10.58369/biit.v2i2.45>.
- [2] S. Nugroho, B. Pujiarto, U. M. Magelang, and P. Korespondensi, "NETWORK AUTOMATION PADA BEBERAPA PERANGKAT ROUTER NETWORK AUTOMATION IN SOME ROUTER DEVICES," vol. 9, no. 1, pp. 79–86, 2022, doi: 10.25126/jtiik.202293947.
- [3] H. Alamsyah, F. T. Informasi, U. N. Putra, and O. Jaringan, "Otomatisasi Jaringan rt / rw Net Menggunakan Python ( Studi Kasus Desa Ciengang )," vol. 7, no. 2, pp. 75–81, 2023.
- [4] R. I. Napitupulu and J. Gunawan, "APLIKASI NETWORK AUTOMATION PADA ROUTER CISCO," vol. 11, pp. 1980–1990, 2022.
- [5] J. T. Informatika, "Simulasi Network Automation Menggunakan Ansible Di GNS3 (Studi Kasus Smile Project)," vol. 2617, no. 2, pp. 1–8, 2020.
- [6] P. Studi, T. Informatika, and F. Teknik, "NETWORK AUTOMATION DENGAN METODE NDLC MENGGUNAKAN BAHASA PEMROGRAMAN PYTHON," vol. 07, no. 01, pp. 149–156, 2026.
- [7] M. F. Islami, P. Musa, and M. Lamsani, "Implementation of Network Automation Using Ansible to Configure Routing Protocol in Cisco and Mikrotik Router with Raspberry PI," vol. 19, pp. 127–134, 2020.

- [8] A. Zulfikar and Y. Akbar, "Automation of Mikrotik Router Setting Configuration Backup Using Ansible with Network DevOps Method Otomasi Backup Konfigurasi Settingan Router Mikrotik Menggunakan Ansible dengan Metode Network DevOps," vol. 5, no. January, pp. 57–66, 2025.
- [9] D. Rahardika and N. Ratama, "Implementasi Network Automation Untuk Konfigurasi Jaringan Baru Dengan Netmiko," vol. 2, no. 3, pp. 190–200, 2021.
- [10] M. Arman, "Analisa Jaringan Local Area Network ( Lan ) Dengan Aplikasi Cisco Packet Tracer Pada PT . Bank Negara Indonesia ( Persero ) Tbk Kcp Watansoppeng," vol. 5, pp. 41–50, 2022.
- [11] I. Ndustri, "MODEL REFERENSI OPEN SYSTEMS INTERCONNECTION (OSI) DAN TRANSMISSION CONTROL PROTOCOL / INTERNET PROTOCOL (TCP/ IP) DALAM TANTANGAN ERA," vol. 19, pp. 78–86, 2025.
- [12] D. B. Maulana, A. Mulyana, and S. P. Wisetyo, "Design and Realization Automation of Network Configuration Management System Using Ansible And Elasticsearch ( Case Study : Network Development Part In Information System Directorate of Telkom University In Tokong Nanas Building )," vol. 7, no. 6, pp. 3224–3230, 2021.
- [13] A. Kurniawan et al., "Network Automation Menggunakan Ansible di Laboratorium komputer," vol. 2, no. 1, pp. 283–290.
- [14] Mochamad Husni, "DESAIN IMPLEMENTASI ROUTING JARINGAN KOMPUTER," vol. 3, no. 1, 2009, doi: <https://doi.org/10.36382/jti-tki.v3i1.118>.
- [15] M. S. H. Simarangkir, A. Puspabhuana, and S. Wijaya, "Pelatihan Konfigurasi Dasar Router dan Switch Cisco Pada Cisco IOS di SMK Armaniyah Serang Baru Cikarang Selatan," vol. 2, no. 2, pp. 123–131, 2022.
- [16] I. P. Hariyadi, "ANALISA METRIC ROUTING PROTOKOL EIGRP," vol. 1, no. 1, pp. 69–77, 2019, doi: <https://doi.org/10.30812/bite.v1i1.423>.
- [17] M. Iqbal, "Implementasi Dynamic Routing OSPF Menggunakan Mikrotik Dengan Metode Tunneling untuk Optimasi Konektivitas Jaringan Internet," vol. 14, pp. 3218–3229, 2025.
- [18] O. Kurniawan, A. Taufik, and F. Ariani, "Perancangan Routing OSPF Mikrotik pada PT . Arsen Kusuma Indonesia," vol. 8, no. 2, pp. 354–366, 2024, doi: [10.52362/jisicom.v8i2.1682](https://doi.org/10.52362/jisicom.v8i2.1682).
- [19] T. Pustaka, "ANALISIS PERBANDINGAN KINERJA ROUTING ENHANCED INTERIOR GATEWAY ROUTING PROTOCOL DENGAN," vol. 8, no. 2, 2024.
- [20] N. Indah, Y. Salim, and R. Satra, "ANALISIS PERBANDINGAN ROUTING PROTOKOL OPEN SHORTEST PATH FIRST ( OSPF ) DENGAN ENHANCED INTERIOR GATEWAY ROUTING PROTOCOL ( EIGRP )," vol. 10, no. April, pp. 92–99, 2018.
- [21] A. Perkembangan and K. Kunci, "SUBNETTING Hafiz Izzan Zaafarani Anugrah Yunanda Mahasiswa Sistem Informasi UINSI Samrinda Mahasiswa Sistem Informasi UINSI Samrinda," vol. 4, no. 1, pp. 27–34.
- [22] A. Alanda, H. A. Mooduto, H. Amnur, and M. Fadhel, "Scaling Kubernetes Architectures for High Availability in Cloud," in 2025 International Conference on Computer Sciences, Engineering, and Technology Innovation (ICoCSETI), Jakarta, Indonesia: IEEE, Jan. 2025, pp. 818–823. doi: [10.1109/ICoCSETI63724.2025.11020084](https://doi.org/10.1109/ICoCSETI63724.2025.11020084).
- [23] A. Leticia and D. C. Leite, "A Framework for the Characterization and Analysis of Software Systems Scalability," 2009.
- [24] G. Hyun, J. Oak, D. Kim, and K. Kim, "The Impact of an Automation System Built with Jenkins on the Efficiency of Container-Based System Deployment," pp. 1–12, 2024.
- [25] A. Y. Andra, H. A. Mooduto, F. Sukma, and H. Amnur, "Layanan Internet Service Provider dengan GNS3 Menggunakan Mikrotik dan Debian pada Jaringan Dual ISP", jitsi, vol. 7, no. 1, pp. 19 - 26, Mar. 2026.
- [26] M. Putria, A. M. Soebiantoro, H. Aprianto, and W. Haryono, "Implementasi Network Automation untuk Backup Konfigurasi Perangkat Jaringan," 2025, doi: <https://doi.org/10.61132/mercurius.v3i4.929>.

- [27] I. G. Lanang, P. Eka, D. M. Informatika, F. Teknik, and U. N. Surabaya, “IMPLEMENTASI SIMULASI JARINGAN KOMPUTER MULTI DEVICE DENGAN MENGGUNAKAN GNS3”.
- [28] E. W. Sholeha et al., “Komparasi Fungsionalitas EVE-NG dan GNS3 : Simulasi Protokol Jaringan,” no. April, pp. 1–8, 2025, doi: 10.30829/algoritma.v9i1.23864.
- [29] M. N. Albi, “RANCANG BANGUN TEKNOLOGI OTOMATISASI SSH PADA PROXY SERVER MENGGUNAKAN FRAMEWORK DJANGO UNTUK SISTEM MONITORING CACHE PROXY,” vol. 13, no. 3, pp. 482–493.
- [30] “Visual Studio Code - The open source AI code editor | Your home for multi-agent development.” Accessed: Jul. 20, 2026. [Online]. Available: <https://code.visualstudio.com/>
- [31] B. Gondhalekar, H. Manna, S. Kothi, and D. B. Borsae, “Network Automation With Multithreading Using GNS3 and Netmiko,” *Indian Journal of Computer Science*, vol. 9, no. 2, pp. 18–33, 2024, doi: 10.17010/ijcs/2024/v9/i2/173860.
- [32] K. Mukiman, Subandri, J. N. A. Alawiyah, and A. I. Ramdhani, “Implementasi Automation Manage User pada Jaringan Berbasis MikroTik dengan Python,” *Jurnal Teknologi Informasi dan Digital*, vol. 3, no. 1, pp. 63–71, 2025, doi: 10.65624/tridi.v3i1.103.
- [33] P. Talani, “A Comparison Between Ansible and Netmiko in a Virtualized Environment,” *Mälardalen University*, 2024