

Network Automation Ansible Deploy Email Server Running Playbook

**Sabar Hanadwiputra^{*1}, Adhitya Ilham Ramdhani², Subandri³,
Zaenal Mutaqin Subekti⁴, Muhammad Arfan Aziz⁵**

^{1,3}Universitas Bani Saleh, Bekasi, Indonesia

Penulis Korespondensi: sabar@ubs.ac.id

²adhityair@gmail.com, ³andri.subandri@ymail.com, ⁴zaenalms@ubs.ac.id,

⁵andri.subandri@ymail.com, ⁶m.arfan@gmail.com

Article Information:

Received: 28 Agustus 2026; Accepted: 25 September 2026; Published: 01 Oktober 2026

Abstract

With the rapid development of the digital era and information technology, manual network management is becoming increasingly inefficient and time-consuming. Traditional network configuration, which requires administrators to physically access servers, creates a risk of human error that can disrupt services and increase operational costs. To address these challenges, Network Automation provides an effective solution for improving the efficiency and reliability of network infrastructure management, particularly for organizations with servers located in different locations. Ansible is one of the Network Automation tools that uses Playbooks to automate network configuration processes. Through protocols such as SSH, Ansible enables network engineers to remotely monitor and manage network infrastructure without requiring direct physical interaction with network devices. The structured use of Playbooks facilitates documentation, maintenance, and the consistent and rapid execution of administrative tasks. This study focuses on implementing Network Automation using Ansible Playbooks to improve time and resource efficiency in network management. Through this approach, network monitoring and configuration processes become more practical, secure, and reliable in supporting organizational operational needs in the current digital era.

Keywords: Ansible, Email, Playbook

1. Pendahuluan

Perkembangan teknologi informasi yang pesat mendorong organisasi untuk mengelola infrastruktur jaringan secara lebih efektif, cepat, dan andal. Pemanfaatan jaringan komputer tidak hanya berfungsi sebagai sarana komunikasi, tetapi juga menjadi fondasi utama dalam mendukung sistem operasional perusahaan. Namun, pada praktiknya, pengelolaan jaringan

Published by:

yang masih dilakukan secara manual sering kali menimbulkan berbagai permasalahan, seperti lamanya waktu konfigurasi, tingginya risiko kesalahan manusia, serta inkonsistensi sistem antarserver [1]. Pengelolaan server secara manual mengharuskan administrator melakukan konfigurasi satu per satu pada setiap perangkat. Kondisi ini menjadi tidak efisien ketika jumlah server semakin bertambah atau berada pada lokasi yang berbeda [2]. Selain itu, perbedaan konfigurasi antarserver dapat menyebabkan gangguan layanan, kesulitan dalam proses pemeliharaan, serta meningkatnya biaya operasional. Untuk mengatasi permasalahan tersebut, konsep network automation berkembang sebagai solusi dalam pengelolaan infrastruktur jaringan modern. Network automation memungkinkan proses konfigurasi, monitoring, dan pemeliharaan sistem dilakukan secara otomatis melalui skrip atau perangkat lunak khusus. Salah satu perangkat yang banyak digunakan dalam otomatisasi jaringan adalah Ansible, yang memanfaatkan playbook berbasis YAML untuk mengelola sistem secara terpusat [3].

Ansible menggunakan protokol *Secure Shell* (SSH) sebagai media komunikasi antara control node dan managed node, sehingga administrator dapat melakukan pengelolaan jaringan tanpa harus mengakses perangkat secara fisik [4]. Struktur playbook yang sistematis juga berfungsi sebagai dokumentasi teknis yang memudahkan proses pemeliharaan dan replikasi system. Berdasarkan hasil penelitian sebelumnya, penerapan Ansible dalam deployment email server mampu mengurangi waktu konfigurasi secara signifikan, dari rata-rata 20 menit secara manual menjadi sekitar 5 menit menggunakan otomatisasi [5]. Selain meningkatkan efisiensi waktu, penggunaan playbook juga terbukti mampu menekan tingkat kesalahan konfigurasi serta meningkatkan konsistensi sistem. Meskipun demikian, implementasi network automation di lingkungan pendidikan dan organisasi lokal masih relatif terbatas. Banyak institusi yang belum memanfaatkan teknologi ini secara optimal, khususnya dalam pengelolaan layanan email server. Oleh karena itu, diperlukan penelitian yang mengkaji secara sistematis penerapan Ansible dalam otomatisasi deployment email server guna meningkatkan efektivitas pengelolaan jaringan.

Automasi jaringan merupakan tren utama dalam pengelolaan infrastruktur TI modern untuk meningkatkan efisiensi, mengurangi kesalahan manusia (*human error*), serta mempercepat proses konfigurasi dan deployment layanan jaringan [6]. Dimulai dari konfigurasi manual via antarmuka baris perintah (CLI), otomatisasi kini menjadi kebutuhan dasar, terutama di lingkungan dengan banyak perangkat dan layanan. Dalam dekade terakhir, konsep Infrastructure as Code (IaC) yang memungkinkan konfigurasi tersimpan sebagai kode telah mengubah paradigma pengelolaan jaringan dengan menawarkan repeatability, konsistensi, dan auditabilitas [7]. Salah satu alat otomatisasi yang banyak digunakan adalah Ansible, sebuah alat manajemen konfigurasi agentless yang mengandalkan protokol SSH dan playbook berbasis YAML untuk mendeskripsikan langkah-langkah otomatisasi secara deklaratif. Penerapan Ansible telah meluas dari sekadar manajemen konfigurasi server menuju otomatisasi layanan jaringan serta sistem keamanan dan virtualisasi [8]. Playbook Ansible membuat proses deployment layanan seperti konfigurasi routing, VLAN, DHCP, atau update sistem menjadi lebih cepat, konsisten, dan terukur dibandingkan konfigurasi manual yang memakan waktu dan rentan kesalahan [9].

Penelitian terbaru menunjukkan bahwa penggunaan *Ansible* mampu mempercepat deployment server dan konfigurasi jaringan secara signifikan, serta memperbaiki keamanan dan efisiensi sumber daya, bila dibandingkan dengan metode tradisional. Skala penerapan dapat mencakup integrasi dengan virtualization dan kebijakan *firewall* yang konsisten di

seluruh node jaringan [8]. Selain itu, studi komparatif terhadap berbagai alat otomasi jaringan juga menunjukkan posisi Ansible sebagai salah satu alat yang efektif, meskipun dengan kekuatan dan keterbatasan tertentu jika dibandingkan dengan alat lain seperti Netmiko, Paramiko, atau Puppet [10]. Metode ini juga selaras dengan pendekatan DevOps/NetDevOps yang menggabungkan prinsip DevOps dalam manajemen jaringan untuk mempercepat integration, delivery, dan pengujian otomatis[7]. Beberapa studi juga menyoroti tantangan penggunaan Ansible dalam otomasi jaringan, termasuk isu pemeliharaan modul khusus vendor dan kebutuhan untuk menyesuaikan skrip saat skala jaringan bertambah besar. Tantangan ini mencakup dukungan modul yang kadang tidak konsisten serta evolusi tool yang cepat, sehingga konfigurasi perlu diatur pada versi tertentu untuk menghindari gangguan fungsionalitas [11]. Secara keseluruhan, *state of the art* otomasi jaringan menekankan integrasi alat otomasi seperti *Ansible*, prinsip *Infrastructure as Code*, serta pendekatan DevOps/NetDevOps dalam mengelola layanan jaringan skala besar dengan efisiensi tinggi.

Permasalahan utama dalam penelitian ini meliputi pengelolaan email server yang masih dilakukan secara manual sehingga kurang efisien dan rawan kesalahan, sulitnya menjaga konsistensi konfigurasi pada beberapa server, keterbatasan dokumentasi teknis yang terintegrasi dalam proses deployment, dan belum optimalnya pemanfaatan Ansible sebagai solusi otomasi jaringan. Dari permasalahan ini, penelitian ini memiliki tujuan khusus yaitu merancang sistem *network automation* berbasis *Ansible Playbook* untuk *deployment email server*, mengimplementasikan mekanisme otomatisasi instalasi dan konfigurasi layanan email server menggunakan Ansible, menguji efektivitas penerapan playbook dalam meningkatkan efisiensi waktu dan konsistensi konfigurasi, menganalisis tingkat keandalan sistem hasil otomasi dibandingkan dengan metode manual, serta menyusun dokumentasi teknis sebagai panduan implementasi dan pengembangan sistem di masa depan.

Urgensi penelitian ini didasarkan pada kebutuhan akan sistem pengelolaan jaringan yang efisien, terstandar, dan mudah dikembangkan. Di era transformasi digital, organisasi dituntut untuk menyediakan layanan teknologi informasi yang stabil dan responsif. Keterlambatan dalam konfigurasi atau kesalahan teknis dapat berdampak langsung terhadap kualitas layanan dan produktivitas kerja. Penerapan Ansible sebagai alat otomasi jaringan memberikan peluang untuk meningkatkan profesionalisme pengelolaan infrastruktur teknologi informasi, khususnya pada layanan email server. Email server masih menjadi media komunikasi utama dalam berbagai institusi, sehingga kestabilan dan keamanannya harus terjaga dengan baik.

Selain itu, penelitian ini mendukung pengembangan kompetensi sumber daya manusia di bidang teknologi jaringan, khususnya dalam penerapan konsep DevOps dan otomatisasi sistem. Hasil penelitian diharapkan dapat menjadi referensi bagi mahasiswa, praktisi, dan institusi dalam mengembangkan sistem jaringan berbasis otomasi. Dari sisi skema penelitian, penelitian ini termasuk dalam skema penelitian terapan, dengan fokus pada implementasi langsung teknologi Ansible dalam lingkungan operasional. Spesifikasi khusus penelitian ini terletak pada penggunaan Ansible sebagai platform utama otomasi jaringan, deployment layanan email server berbasis Postfix dan Dovecot, pemanfaatan SSH sebagai media komunikasi utama, pengujian sistem pada lingkungan server nyata, penyusunan playbook modular yang dapat digunakan kembali. Dengan karakteristik tersebut, penelitian ini diharapkan mampu menghasilkan model implementasi *network automation* yang praktis, teruji, dan mudah direplikasi.

2. Metode Penelitian

Published by:

Penelitian ini terdiri dari lima tahapan, yaitu Studi Awal dan Literatur, Perancangan Arsitektur Otomasi, Implementasi & Integrasi DevOps/NetDevOps, Evaluasi Kinerja & Uji Sistem, dan Penyusunan Dokumentasi & Optimalisasi.

a. Studi Awal dan Literatur:

Penelitian literatur ilmiah untuk memahami konsep otomasi jaringan, Infrastructure as Code, prinsip DevOps, dan alat otomasi populer termasuk Ansible, Netmiko, Paramiko, Puppet, serta praktik terbaiknya dalam konteks jaringan kompleks.

b. Perancangan Arsitektur Otomasi

Mendesain infrastruktur control node dan managed node, serta merumuskan playbook yang mencakup kebutuhan otomasi seperti deployment email server, konfigurasi VLAN, routing, atau policy security.

c. Implementasi & Integrasi DevOps/NetDevOps:

Melibatkan integrasi otomatisasi dengan proses CI/CD, pengujian otomatis, serta manajemen konfigurasi yang dapat memenuhi kebutuhan update dan rollback.

d. Evaluasi Kinerja & Uji Sistem

Pengujian sistem otomasi terhadap metrik efisiensi waktu, konsistensi konfigurasi, kestabilan layanan, dan reduksi kesalahan dibandingkan metode manual.

e. Penyusunan Dokumentasi & Optimalisasi

Menyusun dokumentasi teknis, template playbook, dan panduan implementasi untuk memfasilitasi replikasi serta perbaikan di masa mendatang.

3. Hasil Dan Pembahasan

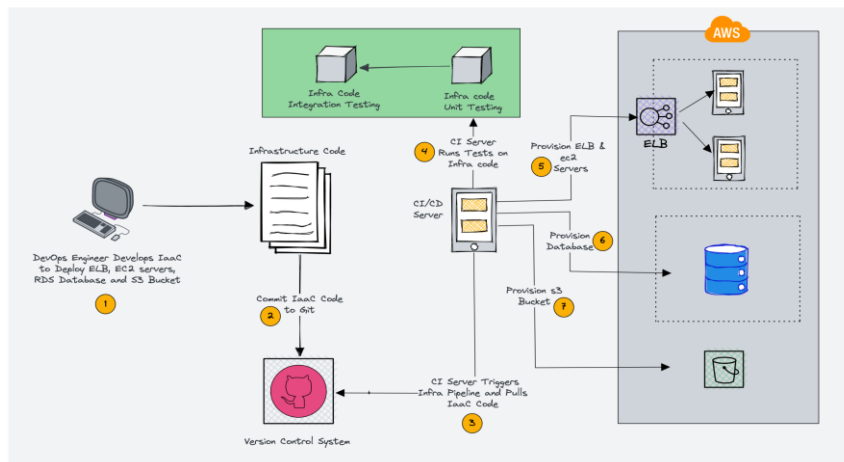
3.1 Hasil

Pelaksanaan penelitian menghasilkan rancangan dan implementasi sistem otomasi deployment email server menggunakan Ansible Playbook. Sistem dibangun dengan pendekatan control node dan managed node yang berkomunikasi melalui SSH. Tahapan penelitian yang direncanakan telah diarahkan pada analisis kebutuhan, perancangan arsitektur, simulasi prototipe, implementasi, pengujian, monitoring, evaluasi, dan dokumentasi. Hasil implementasi menunjukkan bahwa proses deployment dapat dituangkan ke dalam playbook sehingga tahapan konfigurasi tidak lagi bergantung sepenuhnya pada instruksi manual administrator. Playbook menjadi dokumentasi teknis sekaligus instruksi otomasi yang dapat dijalankan kembali

Perancangan Sistem

Perancangan sistem merupakan tahap yang dilakukan untuk menggambarkan alur kerja dan hubungan antar-komponen dalam proses pengembangan serta penerapan sistem. Pada penelitian ini, perancangan sistem mengadopsi konsep *Continuous Integration/Continuous Deployment* (CI/CD) untuk mengotomatisasi proses pengujian, pembangunan (*build*), dan penyebaran (*deployment*) aplikasi ke lingkungan *cloud*. Proses diawali dari pengembang (*developer*) yang melakukan perubahan kode sumber aplikasi. Kode yang telah diperbarui kemudian disimpan pada *version control system* menggunakan GitHub. Selanjutnya, perubahan kode akan memicu proses *Continuous Integration* yang mencakup pengujian kode (*unit testing*) dan *integration testing* untuk memastikan bahwa perubahan yang dilakukan

tidak menyebabkan kesalahan pada sistem.



Gambar 1: Topologi Jaringan

Setelah proses pengujian berhasil, sistem melanjutkan ke tahap *Continuous Deployment*. Pada tahap ini, kode aplikasi dibangun menjadi *container image* dan selanjutnya diterapkan pada infrastruktur *cloud*. Komponen infrastruktur yang digunakan meliputi *Elastic Load Balancer* (ELB), *server*, *database*, dan penyimpanan objek. ELB berfungsi mendistribusikan permintaan pengguna ke server aplikasi, sedangkan database digunakan untuk menyimpan data yang diperlukan oleh aplikasi. Dengan rancangan tersebut, proses pengembangan dan penerapan aplikasi dapat dilakukan secara lebih terstruktur dan otomatis. Penerapan CI/CD juga memungkinkan setiap perubahan kode melalui tahapan pengujian sebelum diterapkan ke lingkungan produksi sehingga dapat meningkatkan konsistensi, efisiensi, dan keandalan proses *deployment* sistem.

Pengujian dilakukan dengan menjalankan *playbook* secara berurutan mulai dari pemeriksaan konektivitas hingga validasi layanan email.

Tabel 1. Hasil Pengujian Otomasi

No.	Skenario Pengujian	Parameter	Hasil yang Diharapkan
1	Koneksi SSH	Koneksi berhasil	Managed node dapat diakses
2	Instalasi paket	Status task	Paket terpasang
3	Instalasi Postfix	Status service	Postfix aktif
4	Instalasi Dovecot	Status service	Dovecot aktif
5	Konfigurasi layanan	Validasi konfigurasi	Konfigurasi sesuai
6	Restart/reload	Status service	Service kembali aktif
7	Pengujian email	Kirim/terima email	Email berhasil diproses
8	Eksekusi ulang	Perubahan konfigurasi	Tidak terjadi perubahan tidak diperlukan

Hasil tersebut menunjukkan bahwa pendekatan playbook dapat mengintegrasikan beberapa aktivitas administrasi server ke dalam satu alur otomatis. Hal ini merupakan salah satu keunggulan Ansible karena playbook dapat mendeskripsikan konfigurasi dan deployment secara terstruktur serta dapat digunakan kembali [1,2].

Catatan pengisian laporan akhir: nilai waktu deployment manual, waktu deployment otomatis, jumlah percobaan, jumlah task berhasil/gagal, serta persentase keberhasilan perlu dimasukkan berdasarkan log aktual penelitian. Angka tersebut tidak terdapat dalam dokumen yang tersedia sehingga tidak dicantumkan secara fiktif.

4. Kesimpulan

Berdasarkan pelaksanaan penelitian, dapat disimpulkan bahwa:

1. Penelitian telah menghasilkan rancangan implementasi Network Automation menggunakan Ansible Playbook untuk deployment email server dengan pendekatan control node dan managed node.
2. Ansible dapat digunakan untuk mengotomatisasi tahapan konfigurasi server yang sebelumnya dilakukan secara manual, termasuk proses instalasi dan konfigurasi layanan Postfix dan Dovecot.
3. Penggunaan playbook memberikan struktur konfigurasi yang lebih sistematis, terdokumentasi, dapat digunakan kembali, dan mendukung konsistensi konfigurasi pada server target.
4. Penggunaan SSH sebagai mekanisme komunikasi memungkinkan proses pengelolaan server dilakukan secara remote, sesuai dengan karakteristik arsitektur agentless Ansible [1,4].
5. Hasil penelitian mendukung tujuan penerapan Infrastructure as Code dalam pengelolaan infrastruktur TI, karena konfigurasi deployment dapat direpresentasikan dalam bentuk kode dan dijalankan secara berulang [8].
6. Untuk menyatakan peningkatan efisiensi secara kuantitatif, laporan akhir perlu menggunakan data hasil pengukuran aktual berupa waktu deployment, jumlah percobaan, tingkat keberhasilan task, dan hasil pengujian layanan.

Berdasarkan hasil penelitian, beberapa pengembangan yang disarankan adalah:

1. Penelitian berikutnya perlu melakukan pengujian pada jumlah server yang lebih banyak untuk mengukur skalabilitas Ansible.
2. Playbook dapat dikembangkan menggunakan struktur Ansible Roles agar lebih modular dan mudah dipelihara.
3. Pengamanan kredensial perlu ditingkatkan menggunakan mekanisme seperti Ansible Vault atau pengelolaan secret yang sesuai.
4. Penelitian selanjutnya dapat mengintegrasikan Ansible dengan Git dan pipeline CI/CD sehingga perubahan konfigurasi dapat melalui proses version control, testing, approval, dan deployment secara otomatis.
5. Pengujian dapat diperluas dengan parameter reliability, availability, recovery time, resource utilization, serta keamanan layanan email.

Daftar Pustaka

- [1] Ansible Community. Introduction to Ansible [Internet]. Ansible Community Documentation; 2026.
- [2] Ansible Community. Working with playbooks [Internet]. Ansible Community Documentation; 2026.
- [3] Fadhudin A, Ramayanti D. Implementasi sistem otomatisasi konfigurasi server berbasis Ansible dan Semaphore pada infrastruktur TI perusahaan ritel. *Riau Jurnal Teknik Informatika*. 2026;5(1).
- [4] Ansible Community. Connection plugins [Internet]. Ansible Community Documentation; 2026.
- [5] Sembiring A, et al. Continuous delivery implementation with zero-downtime rolling update using Ansible. *Jurnal Komputer Terapan*. 2022.
- [6] Putra YR, et al. Automation of Open vSwitch-based virtual network configuration using Ansible on Proxmox Virtual Environment. *International Journal of Electronics and Communications Systems*. 2023.
- [7] Putra WRA, Nurwa ARA, Priambodo DF, Hasbi M. Infrastructure as Code for security automation and network infrastructure monitoring. *MATRIK: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*. 2022;22(1).
- [8] Ansible Community. Run your first command and playbook [Internet]. Ansible Community Documentation; 2026.
- [9] Ansible Community. Installing Ansible [Internet]. Ansible Community Documentation; 2026.
- [10] Infrastructure-as-Code in Open-Networking: Git, Ansible, and Cumulus-Linux case study. *Proceedings of the IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC)*. 2023. doi:10.1109/CCWC57344.2023.10099084.
- [11] Ansible Community. Module maintenance and support [Internet]. Ansible Core Documentation; 2026.

Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>)

