

Perancangan Sistem Network Automation Berbasis MikroTik API untuk Otomatisasi Manajemen Router Menggunakan Metode Waterfall

I Gusti Agung Gede Punar Nata Kepakisan, Gede Arna Jude Saskara*, I Made Gede Sunarya

Fakultas Teknik dan Kejuruan, Pendidikan Teknik Informaika, Universitas Pendidikan Ganesha, Singaraja, Indonesia

Email: ¹agung.punar@undiksha.ac.id, ^{2,*}jude.saskara@undiksha.ac.id, ³sunarya@undiksha.ac.id

Email Penulis Korespondensi: jude.saskara@undiksha.ac.id

Abstrak—Perkembangan teknologi informasi menuntut pengelolaan jaringan komputer yang lebih efektif, khususnya pada lingkungan pemerintahan yang memiliki banyak perangkat jaringan. Pengelolaan router MikroTik di seluruh Satuan Kerja Perangkat Daerah (SKPD) Kabupaten Buleleng masih dilakukan secara manual sehingga memerlukan waktu yang lama serta berpotensi menimbulkan kesalahan konfigurasi. Penelitian ini bertujuan mengembangkan sistem Network Automation berbasis PHP yang terintegrasi dengan MikroTik API untuk membantu administrator jaringan dalam melakukan konfigurasi, monitoring, pengelolaan perangkat, serta administrasi jaringan secara terpusat. Metode penelitian yang digunakan adalah Research and Development (R&D) dengan model pengembangan perangkat lunak Waterfall yang meliputi analisis kebutuhan, perancangan, implementasi, dan pengujian sistem. Sistem dikembangkan menggunakan PHP, MySQL, serta MikroTik API dan dilengkapi fitur manajemen perangkat, monitoring status jaringan, konfigurasi otomatis, pencatatan log aktivitas, serta notifikasi Telegram secara real-time. Pengujian dilakukan menggunakan metode Black Box Testing, White Box Testing, dan uji performa. Hasil pengujian Black Box menunjukkan seluruh 18 skenario pengujian berhasil dijalankan sesuai dengan kebutuhan fungsional. White Box Testing menghasilkan nilai Cyclomatic Complexity sebesar 8 dengan seluruh jalur independen tervalidasi. Pengujian performa menunjukkan sistem mampu memonitor hingga 10 perangkat dengan penggunaan sumber daya yang efisien dan waktu respons yang stabil. Hasil penelitian menunjukkan bahwa sistem yang dikembangkan mampu meningkatkan efisiensi administrasi jaringan, mengurangi risiko human error, serta mendukung pengelolaan perangkat MikroTik secara terpusat pada seluruh SKPD Kabupaten Buleleng.

Kata Kunci: Network Automation; MikroTik; Monitoring; PHP; Waterfall

Abstract—Advances of information technology has increased the demand for efficient computer network management, particularly in government institutions that operate numerous network devices. Router management across all Regional Government Work Units (SKPD) in Buleleng Regency is still performed manually, resulting in lengthy configuration processes and a higher risk of configuration errors. This study aims to develop a PHP-based Network Automation system integrated with the MikroTik API to assist network administrators in centralized device configuration, monitoring, network administration, and management. The research employed the Research and Development (R&D) method using the Waterfall software development model, which consists of requirement analysis, system design, implementation, and testing. The developed system utilizes PHP, MySQL, and the MikroTik API and provides features including device management, real-time network monitoring, automated configuration, activity logging, and Telegram notifications. System evaluation was conducted using Black Box Testing, White Box Testing, and performance testing. The Black Box Testing results indicated that all 18 functional test scenarios were successfully executed as expected. White Box Testing produced a Cyclomatic Complexity value of 8, with all independent execution paths validated successfully. Performance testing demonstrated that the system could efficiently monitor up to 10 MikroTik devices while maintaining stable response times and low resource utilization. The results indicate that the proposed system improves network administration efficiency, reduces the risk of human error, and supports centralized management of MikroTik devices across all SKPD in Buleleng Regency.

Keywords: Network Automation; MikroTik; Monitoring; PHP; Waterfall

1. PENDAHULUAN

Perkembangan teknologi informasi telah menjadikan jaringan komputer sebagai infrastruktur utama dalam mendukung operasional organisasi, termasuk instansi pemerintahan. Di lingkungan Satuan Kerja Perangkat Daerah (SKPD) Kabupaten Buleleng, jaringan komputer berperan penting dalam mendukung layanan pemerintahan berbasis digital sehingga diperlukan pengelolaan jaringan yang andal, aman, dan efisien [1]. Seiring meningkatnya jumlah perangkat, pengguna, dan layanan yang terhubung, kompleksitas pengelolaan jaringan juga semakin tinggi. Kondisi tersebut menuntut adanya sistem administrasi jaringan yang mampu mengurangi kesalahan konfigurasi sekaligus meningkatkan efisiensi operasional [2].

Transformasi digital yang berlangsung pada sektor pemerintahan telah mendorong hampir seluruh layanan administrasi memanfaatkan teknologi informasi sebagai media utama dalam penyampaian layanan kepada masyarakat. Berbagai aplikasi pemerintahan, layanan administrasi kependudukan, sistem informasi keuangan daerah, layanan surat-menyerut elektronik, hingga sistem berbasis cloud membutuhkan konektivitas jaringan yang stabil dan memiliki tingkat ketersediaan yang tinggi. Oleh karena itu, kualitas pengelolaan infrastruktur jaringan menjadi salah satu faktor penting dalam menjaga kontinuitas layanan publik. Gangguan jaringan yang terjadi akibat kesalahan konfigurasi atau keterlambatan penanganan dapat berdampak langsung terhadap proses pelayanan kepada masyarakat serta menurunkan efektivitas kerja perangkat daerah.

Pengelolaan perangkat jaringan, khususnya router MikroTik, pada umumnya masih dilakukan secara manual melalui Command Line Interface (CLI) maupun aplikasi administrasi seperti Winbox. Metode ini mengharuskan administrator melakukan konfigurasi pada setiap perangkat secara satu per satu sehingga membutuhkan waktu yang lama dan berpotensi menimbulkan human error, terutama ketika jumlah perangkat yang dikelola semakin banyak [3]. Kesalahan konfigurasi tidak hanya mengganggu ketersediaan layanan jaringan, tetapi juga dapat menimbulkan risiko

keamanan yang berdampak pada operasional instansi [4]. Selain itu, proses administrasi manual menyebabkan pekerjaan yang bersifat berulang, seperti pembaruan konfigurasi, pencadangan konfigurasi, pemeriksaan status perangkat, maupun proses pemulihan layanan menjadi kurang efisien. Kondisi tersebut semakin sulit apabila administrator harus mengelola banyak perangkat yang tersebar pada lokasi berbeda dalam waktu yang bersamaan.

Permasalahan tersebut juga dihadapi oleh administrator jaringan di Dinas Komunikasi, Informatika, Persandian, dan Statistik Kabupaten Buleleng yang bertanggung jawab mengelola jaringan pada seluruh SKPD. Tugas administrator meliputi konfigurasi router, pengelolaan firewall, monitoring jaringan, pemeliharaan perangkat, pencadangan konfigurasi, hingga penyusunan laporan. Pelaksanaan seluruh aktivitas tersebut secara manual menjadi kurang efektif apabila jumlah perangkat yang dikelola terus bertambah. Selain membutuhkan waktu yang lebih lama, metode manual juga menyulitkan administrator dalam menjaga konsistensi konfigurasi antarperangkat serta mempercepat proses penanganan ketika terjadi gangguan jaringan. Akibatnya, proses pemeliharaan jaringan menjadi kurang optimal dan berpotensi memengaruhi kualitas layanan teknologi informasi yang digunakan oleh masing-masing SKPD.

Salah satu solusi yang banyak dikembangkan adalah penerapan *Network Automation*, yaitu proses otomatisasi konfigurasi, pengelolaan, pemantauan, dan pemeliharaan perangkat jaringan menggunakan perangkat lunak. Penerapan *network automation* terbukti mampu meningkatkan efisiensi operasional, mempercepat proses konfigurasi, menjaga konsistensi konfigurasi antarperangkat, serta mengurangi risiko kesalahan manusia [5]. Selain itu, otomatisasi jaringan memungkinkan administrator melakukan pengelolaan banyak perangkat secara terpusat sehingga waktu administrasi dapat dikurangi secara signifikan [6]. Melalui mekanisme otomatisasi, berbagai pekerjaan rutin dapat dijalankan secara otomatis berdasarkan aturan yang telah ditentukan sehingga administrator dapat lebih fokus pada proses perencanaan, pengembangan, dan peningkatan kualitas jaringan.

Implementasi *network automation* juga memberikan manfaat dari sisi keamanan dan keandalan sistem. Proses monitoring dapat dilakukan secara berkala untuk mendeteksi kondisi perangkat maupun gangguan jaringan secara lebih cepat. Apabila ditemukan kondisi tertentu, sistem dapat menjalankan tindakan otomatis sesuai kebijakan yang telah ditentukan, seperti melakukan reboot perangkat, memperbarui konfigurasi, atau memberikan notifikasi kepada administrator. Pendekatan tersebut mampu mengurangi waktu respons terhadap gangguan sekaligus meningkatkan ketersediaan layanan jaringan.

Berbagai penelitian telah membuktikan bahwa *network automation* mampu meningkatkan efisiensi pengelolaan jaringan komputer. Penelitian oleh Zakaria *et al.* [7] pada tahun 2019 berhasil mengembangkan sistem berbasis PHP, MikroTik API, dan MySQL untuk mengotomatisasi proses registrasi serta aktivasi pengguna hotspot sehingga pelayanan menjadi lebih cepat dan efisien. Islami *et al.* [8] pada tahun 2020 menerapkan Ansible untuk mengotomatisasi konfigurasi routing pada router Cisco dan MikroTik secara bersamaan dengan tingkat konsistensi konfigurasi yang tinggi. Santyadiputra *et al.* [9] pada tahun 2021 mengembangkan *Automatic Network Administration* (ANA) yang mampu mengotomatisasi administrasi perangkat jaringan sehingga proses konfigurasi menjadi lebih efisien. Penelitian Fahmi *et al.* [10] pada tahun 2021 juga menunjukkan bahwa otomatisasi menggunakan bahasa pemrograman Python dapat mempercepat penyediaan konfigurasi internet serta pengelolaan perangkat MikroTik. Selain itu, Prasetyo *et al.* [11] pada tahun 2024 membuktikan bahwa sistem otomatisasi *backup* konfigurasi mampu meningkatkan efisiensi pengelolaan perangkat jaringan, sedangkan Paksi dan Widiyari [12] pada tahun 2022 mengembangkan aplikasi *network automation* berbasis web menggunakan framework Django yang berhasil mengotomatisasi administrasi jaringan secara terpusat. Berdasarkan penelitian-penelitian tersebut dapat disimpulkan bahwa *network automation* mampu meningkatkan efisiensi, konsistensi konfigurasi, serta mempercepat proses administrasi jaringan. Namun, pengembangan sistem *network automation* berbasis PHP yang mengintegrasikan konfigurasi, monitoring, pengelolaan keamanan, dan administrasi router MikroTik dalam satu aplikasi terpusat masih relatif terbatas sehingga penelitian ini dilakukan untuk mengisi kesenjangan tersebut.

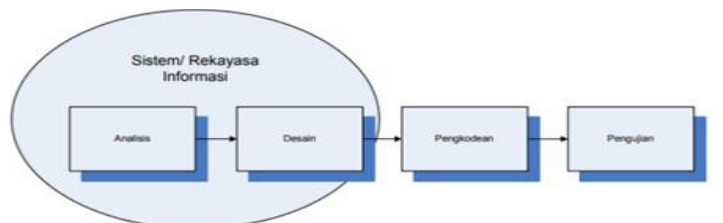
Meskipun demikian, sebagian besar penelitian sebelumnya masih berfokus pada otomatisasi konfigurasi dasar, manajemen hotspot, atau pengelolaan perangkat dalam skala terbatas. Belum banyak penelitian yang mengembangkan sistem *network automation* berbasis PHP yang mengintegrasikan fungsi konfigurasi perangkat, monitoring jaringan, pengelolaan keamanan, serta administrasi router MikroTik secara terpusat dalam lingkungan pemerintahan yang memiliki banyak lokasi kerja. Padahal, karakteristik jaringan pemerintahan membutuhkan sistem yang mampu mengelola banyak perangkat secara terintegrasi dengan tetap menjaga konsistensi konfigurasi, keamanan, dan kemudahan pengelolaan.

Berdasarkan berbagai penelitian tersebut, dapat disimpulkan bahwa *network automation* mampu meningkatkan efisiensi, kecepatan, konsistensi konfigurasi, serta keandalan pengelolaan jaringan. Namun, penelitian yang mengembangkan sistem *network automation* berbasis PHP untuk pengelolaan router MikroTik pada lingkungan pemerintahan, khususnya yang mengelola banyak SKPD secara terpusat, masih relatif terbatas. Oleh karena itu, penelitian ini bertujuan mengembangkan sistem *network automation* berbasis PHP yang memanfaatkan MikroTik API untuk membantu administrator jaringan dalam melakukan konfigurasi, monitoring, pengelolaan keamanan, serta administrasi perangkat MikroTik secara terpusat pada seluruh SKPD Kabupaten Buleleng. Sistem yang dikembangkan diharapkan mampu meningkatkan efisiensi operasional, mengurangi risiko *human error*, mempercepat proses administrasi jaringan, meningkatkan konsistensi konfigurasi antarperangkat, serta mendukung terselenggaranya layanan pemerintahan berbasis digital yang lebih andal, aman, dan berkelanjutan.

2. METODOLOGI PENELITIAN

2.1 Jenis Penelitian

Dalam penelitian ini metode yang digunakan adalah pengembangan (Research and Development). Dalam bukunya Sugiyono mengemukakan bahwa metode Research and development adalah metode penelitian yang digunakan untuk merancang, memproduksi, mengembangkan, menciptakan suatu produk dan menguji validitas produk tersebut. Dalam menunjang penyusunan penelitian ini, maka penulis menggunakan metode pengembangan perangkat lunak dan pengumpulan data-data yang dibutuhkan. Metode yang digunakan pada pengembangan perangkat lunak perancangan sistem informasi ujian berbasis web menggunakan model waterfall, alasan menggunakan metode waterfall adalah Karena Metode ini tahapan dan juga urutan dari metode yang dilakukan berurutan dan berkelanjutan, seperti layaknya sebuah air terjun [13].



Gambar 1. Tahapan model waterfall

2.2 Objek Penelitian

Objek penelitian dalam penelitian ini adalah sistem *Network Automation* untuk manajemen router MikroTik yang dikembangkan berbasis web menggunakan bahasa pemrograman PHP, basis data MySQL, dan MikroTik API sebagai media komunikasi dengan perangkat jaringan. Sistem ini dirancang untuk membantu administrator jaringan dalam melakukan pengelolaan perangkat secara terpusat sehingga proses konfigurasi, monitoring, dan administrasi jaringan dapat dilakukan secara lebih cepat, konsisten, dan efisien. Objek yang dianalisis meliputi perangkat router MikroTik, administrator jaringan sebagai pengguna sistem, serta data yang digunakan dalam proses pengelolaan jaringan. Data tersebut terdiri atas data perangkat, konfigurasi router, status konektivitas, data monitoring, konfigurasi firewall, serta log aktivitas administrator yang tersimpan pada basis data. Seluruh proses pengujian dilakukan menggunakan lingkungan simulasi router MikroTik untuk memastikan setiap fungsi sistem dapat berjalan sesuai dengan kebutuhan tanpa memengaruhi jaringan operasional.

2.3 Tahapan Penelitian

Tahapan penelitian merupakan rangkaian proses yang dilakukan secara sistematis untuk mencapai tujuan penelitian, yaitu merancang dan mengembangkan sistem *Network Automation* berbasis PHP untuk manajemen perangkat MikroTik di seluruh SKPD Kabupaten Buleleng. Tahapan penelitian dimulai dari menganalisis kebutuhan hingga diperoleh sistem yang telah diuji dan siap digunakan. Setiap tahapan saling berkaitan sehingga hasil dari satu tahap menjadi dasar pelaksanaan tahap berikutnya.

2.2.1 Analisis Kebutuhan

Dalam analisa kebutuhan ini bertujuan untuk menganalisis kebutuhan yang dibutuhkan dalam perancangan baik berupa dokumen maupun sumber lain yang dapat membantu dalam menentukan solusi permasalahan yang ada [14]. Pada tahap analisis kebutuhan, dilakukan identifikasi terhadap fitur yang diperlukan agar sistem *Network Automation* untuk Manajemen MikroTik SKPD Kabupaten Buleleng dapat berjalan sesuai dengan tujuan.

2.2.2 Desain

Pada tahapan ini akan dilakukan perancangan desain alur sistem, berupa fitur-fitur yang diperlukan, perancangan Use Case, perancangan Diagram Hubungan Entitas atau entity relationship diagram (ERD), dan juga desain User Interface atau antar muka pengguna [15].

2.2.3 Pengkodean

Pada tahap implementasi, sistem *Network Automation* untuk Manajemen MikroTik SKPD Kabupaten Buleleng mulai dikembangkan berdasarkan hasil analisis kebutuhan dan perancangan yang telah dilakukan sebelumnya. Implementasi ini mencakup pengembangan backend menggunakan PHP dengan framework Laravel, pengembangan frontend berbasis Vue.js atau React, serta integrasi dengan API MikroTik untuk mengotomatisasi konfigurasi dan monitoring jaringan.

Proses implementasi diawali dengan instalasi dan konfigurasi lingkungan pengembangan, termasuk server aplikasi, database, serta pustaka pendukung. Backend dikembangkan dengan arsitektur REST API dan SSH atau secure shell adalah seperangkat aturan, protokol, dan alat yang memungkinkan dua aplikasi perangkat lunak untuk saling berkomunikasi. API berfungsi sebagai jembatan penghubung antara dua sistem yang berbeda, sehingga memungkinkan akses ke data atau layanan tertentu tanpa harus mengetahui detail internal dari sistem yang digunakan. Dalam konteks

otomasi jaringan, API menjadi sarana penting untuk melakukan kontrol dan konfigurasi perangkat SSH (secure shell) adalah protokol jaringan yang berada di lapisan aplikasi pada protokol TCP/IP, memfasilitasi sistem komunikasi yang aman diantara dua sistem yang menggunakan arsitektur klien server dengan menyediakan kerahasiaan dan integritas data melalui teknik enkripsi dan dekripsi yang dilakukan secara otomatis didalam koneksinya, untuk menggunakan SSH dibutuhkan otentifikasi user berupa kunci umum dan password yang terenkripsi[16]. Agar dapat berkomunikasi dengan frontend secara dinamis. Sementara itu, frontend dirancang untuk menampilkan data secara real-time dan memberikan pengalaman pengguna yang interaktif.

Selain itu, sistem juga mengimplementasikan mekanisme autentikasi dan otorisasi untuk memastikan keamanan akses pengguna. Seluruh perubahan konfigurasi yang dilakukan melalui sistem akan dikirim ke perangkat MikroTik melalui API, serta dicatat dalam log aktivitas untuk memudahkan audit dan troubleshooting.

2.2.4 Pengujian

Metode pengujian yang digunakan dalam sistem Network Automation untuk Manajemen MikroTik SKPD Kabupaten Buleleng adalah:

a. Blackbox Testing

Blackbox testing adalah metode pengujian perangkat lunak yang berfokus pada evaluasi fungsionalitas sistem tanpa memerlukan pengetahuan tentang desain dan kode program di baliknya. Metode ini hanya mempertimbangkan batas bawah dan batas atas dari data yang diharapkan sebagai parameter pengujian. Tujuan utama dari blackbox testing adalah untuk mengidentifikasi kelemahan dalam sistem, sehingga hasil yang diperoleh dari pengujian dapat sesuai dengan data yang dimasukkan oleh pengguna. Dengan demikian, metode ini membantu menghindari kekurangan dan kesalahan dalam aplikasi sebelum digunakan secara luas oleh pengguna[17].

b. Whitebox

White Box Testing adalah metode pengujian perangkat lunak yang berfokus pada pemeriksaan struktural alur kode program secara menyeluruh, dengan tujuan memastikan bahwa input yang diterima dan output yang dihasilkan sesuai dengan spesifikasi yang dibutuhkan. Pengujian ini dilakukan dengan memeriksa desain program, struktur kode, dan logika internal yang mendasari aplikasi[18]

c. Uji Performance.

Uji performa dilakukan untuk mengetahui tingkat kemampuan sistem network automation dalam menjalankan proses otomasi konfigurasi pada perangkat MikroTik. Pengujian ini bertujuan untuk mengukur kinerja sistem berdasarkan beberapa indikator teknis yang telah ditentukan.[19].

3. HASIL DAN PEMBAHASAN

3.1 Penjelasan Perancangan

Sistem *Network Automation* untuk manajemen router MikroTik dirancang menggunakan arsitektur *client-server* berbasis web untuk memudahkan administrator jaringan dalam melakukan pengelolaan perangkat secara terpusat. Sistem menerapkan arsitektur tiga lapis (*three-tier architecture*) yang terdiri atas lapisan presentasi (*frontend*), lapisan logika bisnis (*backend*), dan lapisan data (*database*). Lapisan *frontend* dibangun menggunakan HTML, CSS, JavaScript, dan Bootstrap sehingga menyediakan antarmuka yang responsif untuk mengakses dashboard, manajemen perangkat, monitoring jaringan, pengelolaan keamanan, serta riwayat aktivitas sistem. Antarmuka dirancang dengan mempertimbangkan kemudahan penggunaan (*usability*) sehingga administrator dapat mengelola banyak perangkat MikroTik secara lebih efektif.

Lapisan *backend* dikembangkan menggunakan bahasa pemrograman PHP yang bertugas menangani autentikasi pengguna, pengelolaan data perangkat, komunikasi dengan router melalui MikroTik API, proses monitoring jaringan, eksekusi konfigurasi otomatis, pengelolaan firewall, pengiriman notifikasi, serta pencatatan aktivitas sistem. Selain itu, *backend* mengimplementasikan mekanisme *role-based access control* sehingga hanya pengguna yang memiliki hak akses tertentu yang dapat melakukan konfigurasi maupun administrasi perangkat jaringan.

Lapisan *database* menggunakan MySQL sebagai basis data relasional untuk menyimpan seluruh informasi yang digunakan oleh sistem. Basis data dirancang secara terstruktur untuk menyimpan data pengguna, perangkat router, konfigurasi perangkat, status monitoring, data firewall, log aktivitas, serta hasil proses otomatisasi jaringan. Struktur basis data dioptimalkan agar mampu menangani proses pencarian, pembaruan, dan penyimpanan data secara efisien sehingga mendukung pengelolaan banyak perangkat secara terpusat. Komunikasi antara aplikasi web dan router MikroTik dilakukan menggunakan MikroTik API melalui protokol TCP sehingga proses konfigurasi, monitoring, dan administrasi perangkat dapat dilakukan secara otomatis tanpa harus mengakses setiap router secara langsung melalui Winbox maupun Command Line Interface (CLI). Dengan arsitektur tersebut, sistem mampu memberikan pengelolaan jaringan yang lebih terintegrasi, konsisten, dan efisien serta mengurangi risiko *human error* dalam proses administrasi perangkat jaringan.

3.2 Hasil Penelitian

Hasil penelitian merupakan keluaran yang diperoleh dari proses perancangan, implementasi, dan pengujian sistem *Network Automation* berbasis PHP untuk manajemen perangkat MikroTik di seluruh SKPD Kabupaten Buleleng. Penelitian ini menghasilkan sebuah aplikasi berbasis web yang dirancang untuk membantu administrator jaringan dalam

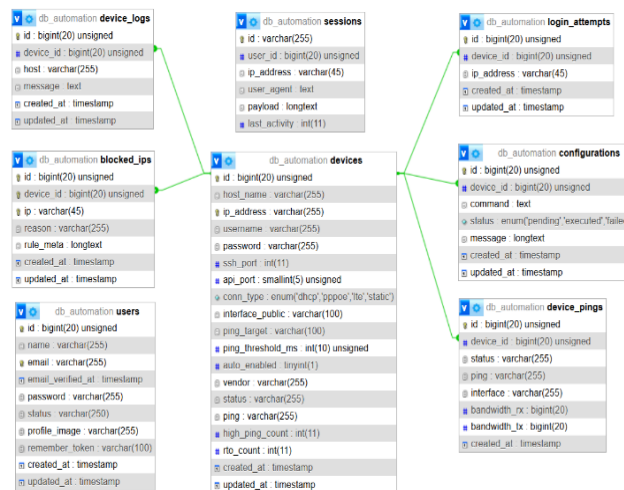
melakukan pengelolaan perangkat MikroTik secara terpusat sehingga proses administrasi jaringan menjadi lebih efektif, efisien, dan terstruktur.)

3.2.1 Implementasi sistem

Implementasi sistem merupakan tahapan penerapan hasil perancangan ke dalam bentuk aplikasi yang dapat digunakan sesuai dengan kebutuhan pengguna. Pada tahap ini, seluruh rancangan sistem yang telah dibuat pada tahap sebelumnya diimplementasikan menggunakan bahasa pemrograman PHP dengan database MySQL serta diintegrasikan dengan MikroTik melalui MikroTik API. Implementasi dilakukan untuk menghasilkan sistem network automation yang mampu membantu administrator jaringan dalam mengelola perangkat MikroTik secara terpusat pada seluruh SKPD Kabupaten Buleleng.

a. Implementasi Database

Hasil Implementasi Database Sistem dapat dilihat pada Gambar 2 implementasi database. Implementasi database merupakan tahap penerapan rancangan basis data ke dalam sistem menggunakan **MySQL** sebagai media penyimpanan data. Database dirancang sesuai dengan kebutuhan fungsional sistem, meliputi data pengguna, perangkat MikroTik, konfigurasi, aktivitas, dan data pendukung lainnya yang saling berelasi sehingga pengelolaan data dapat dilakukan secara terstruktur serta menjaga konsistensi dan integritas data selama sistem beroperasi.

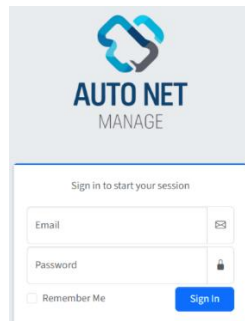


Gambar 2. Implementasi Database

Implementasi Database pada Gambar 2 implementasi database mencakup sembilan tabel utama, yaitu tabel devices, device_logs, blocked_ips, Users, sessions, login_attempts, configurations, dan device_pings.

b. Implementasi Halaman Login

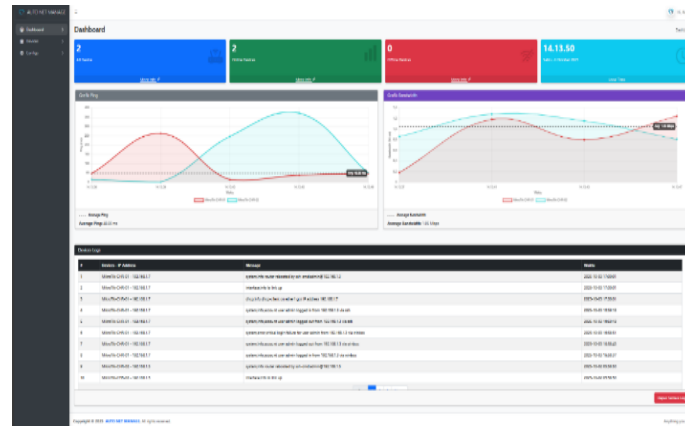
Hasil Implementasi halaman Login dapat dilihat pada Gambar 3. Implementasi Halaman Login. Pada Implementasi Halaman Login ini dibuat untuk autentikasi User sebelum masuk ke halaman dashboard, dengan memasukkan sebuah Input Email dan password yang sudah terdaftar pada sistem.



Gambar 3. Implementasi Halaman Login

c. Implementasi Halaman Dashboard

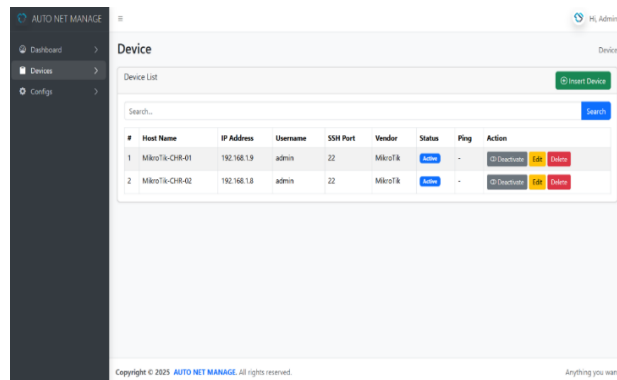
Hasil Implementasi halaman Dashboard dapat dilihat pada Gambar 4 implementasi halaman dashboard. Implementasi Halaman Dashboard ini dibuat untuk User memantau informasi dari setiap perangkat mikrotik, informasi yang ditampilkan antara lain jumlah perangkat yang ada pada sistem, jumlah perangkat yang sedang aktif/online, jumlah perangkat yang tidak aktif/offline, waktu system, grafik informasi ping pada setiap perangkat, grafik penggunaan bandwidth pada setiap perangkat, dan tabel informasi log aktivitas yang masuk pada setiap perangkat mikrotik.



Gambar 4. Implementasi Halaman Dashboard

d. Implementasi Halaman Devices

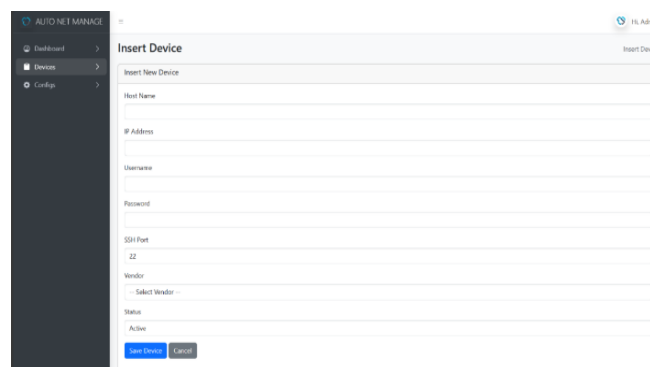
Hasil Implementasi halaman Devices dapat dilihat pada Gambar 5. Implementasi Halaman Devices. Pada Implementasi Halaman Devices ini dibuat untuk User memantau informasi daftar perangkat mikrotik secara detail, dimana terdapat sebuah tabel Device yang berisikan informasi Host Name, IP Address, Username, SSH Port, Vendor, Status, Ping, dan Action untuk melakukan Tindakan mengaktifkan atau menonaktifkan, edit dan hapus perangkat mikrotik yang terdaftar pada sistem. Pengguna juga dapat menambahkan perangkat baru pada sistem dengan menggunakan tombol Insert Device, selain itu terdapat formulir pencarian untuk mempermudah mencari perangkat secara spesifik.



Gambar 5. Implementasi Halaman Devices

e. Implementasi Halaman Insert Device

Hasil Implementasi halaman Insert Device dapat dilihat pada Gambar 6. Implementasi Halaman Insert Device. Pada Implementasi Halaman Insert Device ini dibuat untuk User menambahkan perangkat mikrotik baru kedalam sistem, dimana ada beberapa formulir yang harus dilengkapi User sebelum menambahkan perangkat baru seperti Host Name, IP Address, Username, Password, SSH Port, Vendor, dan Status.

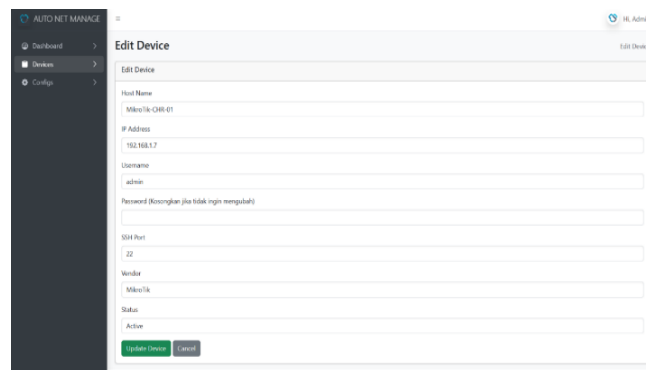


Gambar 6. Implementasi Halaman Insert Device

f. Implementasi Halaman Edit Device

Hasil Implementasi halaman Edit Device dapat dilihat pada Gambar 7. Implementasi Halaman Edit Device. Pada Implementasi Halaman Edit Device ini dibuat untuk User agar bisa melakukan perubahan data perangkat pada mikrotik yang sudah terdaftar pada system, dimana pada halaman tampil sebuah formulir yang sudah terisi oleh data

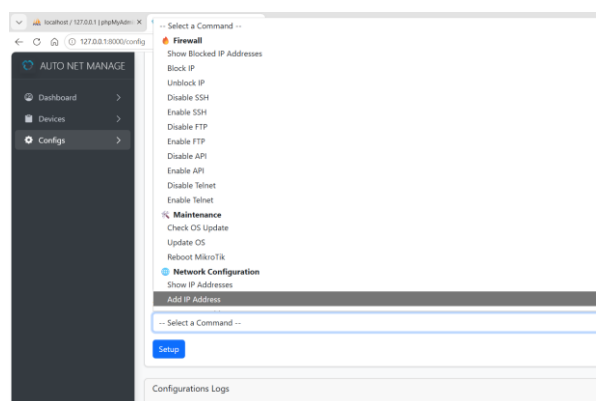
perangkat yang dipilih seperti informasi Host Name, IP Address, Username, Password, SSH Port, Vendor, dan Status, namun pada bagian Password tidak ditampilkan dengan alasan keamanan.



Gambar 7. Implementasi Halaman Edit Device

g. Implementasi Halaman Configs

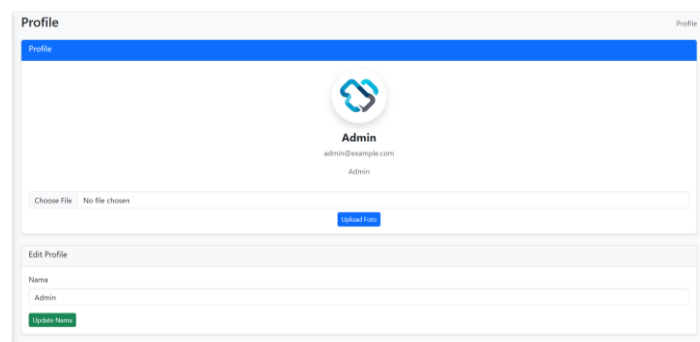
Hasil Implementasi halaman Configs dapat dilihat pada Gambar 8. Implementasi Halaman Configs. Pada Implementasi Halaman Configs ini dibuat untuk User dapat melakukan konfigurasi kepada setiap perangkat mikrotik yang terdaftar pada system, dengan milih konfigurasi yang tersedia maupun memasukan kode tersendiri secara manual, terdapat juga sebuah tabel yang menampilkan informasi hasil eksekusi dan respon perangkat mikrotik terhadap konfigurasi yang dilakukan oleh User, informasi yang ditampilkan berupa Device, Command, Status, Message, Created At, dan Action, User juga dapat menghapus riwayat konfigurasi yang dilakukan dengan cara menekan tombol hapus yang tersedia.



Gambar 8. Implementasi Halaman Configs

h. Implementasi Halaman Profile

Hasil Implementasi halaman Profile dapat dilihat pada Gambar 9. Implementasi Halaman Profile. Pada Implementasi Halaman Profile ini dibuat untuk User dapat melakukan perubahan data User tersebut, dimana perubahan data itu mencakup perubahan Nama, Foto Profile, Email, dan Password.

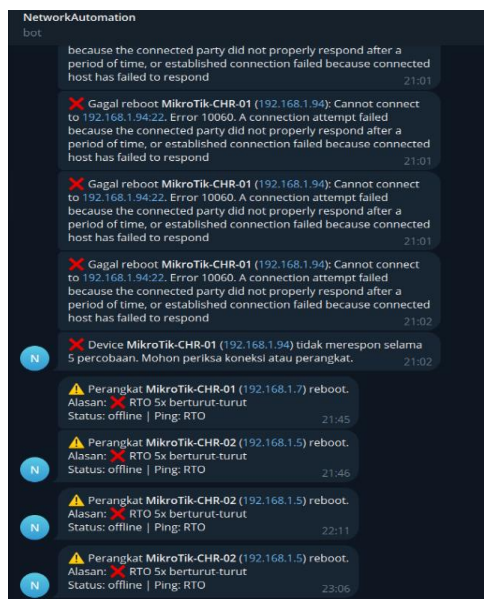


Gambar 9. Implementasi Halaman Profile

i. Implementasi Notifikasi Pesan Otomatis Telegram

Hasil Implementasi Notifikasi Pesan Otomatis pada aplikasi Telegram dapat dilihat pada Gambar 10. Implementasi Notifikasi Pesan Otomatis Telegram. Pada Implementasi Notifikasi Pesan Otomatis Telegram ini dibuat

untuk User dapat memantau perangkat mikrotik melalui aplikasi telegram dimana sistem mengirimkan notifikasi secara Realtime kepada User. Notifikasi yang dikirimkan sistem terkait informasi perangkat yang sedang Reboot maupun perangkat yang tidak merespon.



Gambar 10. Implementasi Notifikasi Pesan Otomatis Telegram

3.2.2 Pengujian sistem

Hasil pengujian sistem yang telah dilakukan terhadap Sistem menggunakan beberapa metode seperti Black Box Test, White Box Test, Uji Performa Sistem.

a. Black Box Test

Pengujian Black Box Test dilakukan dengan beberapa skenario atau prosedur, Prosedur Pengujian Black Box Test. Prosedur Pengujian Black Box Test dilakukan oleh Admin utama yang berwenang pada Kantor Kominfo Kabupaten Buleleng yaitu Bapak Dewa Putu Deni Hermawan dan Bapak Pande Gede Setya Permana, hasil pengujian dari dua responden tersebut dapat dilihat pada Tabel 1 hasil dari pengujian Black Box Test.

Tabel 1. Hasil Pengujian Black Box Test

ID	Hasil Yang Didapat	Status		Penguji	Tanggal Pengujian
		Berhasil/ Gagal	Kesimpulan		
01	Sesuai dengan Output	Berhasil	Valid	Dewa Putu Deni Hermawan Pande Gede Setya Permana	30 Oktober 2025
02	Sesuai dengan Output	Berhasil	Valid	Dewa Putu Deni Hermawan Pande Gede Setya Permana	30 Oktober 2025
03	Sesuai dengan Output	Berhasil	Valid	Dewa Putu Deni Hermawan Pande Gede Setya Permana	30 Oktober 2025
04	Sesuai dengan Output	Berhasil	Valid	Dewa Putu Deni Hermawan Pande Gede Setya Permana	30 Oktober 2025
05	Sesuai dengan Output	Berhasil	Valid	Dewa Putu Deni Hermawan Pande Gede Setya Permana	30 Oktober 2025
06	Sesuai dengan Output	Berhasil	Valid	Dewa Putu Deni Hermawan Pande Gede Setya Permana	30 Oktober 2025
07	Sesuai dengan Output	Berhasil	Valid	Dewa Putu Deni Hermawan Pande Gede Setya Permana	30 Oktober 2025
08	Sesuai dengan Output	Berhasil	Valid	Dewa Putu Deni Hermawan Pande Gede Setya Permana	30 Oktober 2025
09	Sesuai dengan Output	Berhasil	Valid	Dewa Putu Deni Hermawan Pande Gede Setya Permana	30 Oktober 2025
10	Sesuai dengan Output	Berhasil	Valid	Dewa Putu Deni Hermawan Pande Gede Setya Permana	30 Oktober 2025
11	Sesuai dengan Output	Berhasil	Valid	Dewa Putu Deni Hermawan Pande Gede Setya Permana	30 Oktober 2025

ID	Hasil Yang Didapat	Status		Penguji	Tanggal Pengujian
		Berhasil/ Gagal	Kesimpulan		
12	Sesuai dengan Output	Berhasil	Valid	Dewa Putu Deni Hermawan Pande Gede Setya Permana	30 Oktober 2025
13	Sesuai dengan Output	Berhasil	Valid	Dewa Putu Deni Hermawan Pande Gede Setya Permana	30 Oktober 2025
14	Sesuai dengan Output	Berhasil	Valid	Dewa Putu Deni Hermawan Pande Gede Setya Permana	30 Oktober 2025
15	Sesuai dengan Output	Berhasil	Valid	Dewa Putu Deni Hermawan Pande Gede Setya Permana	30 Oktober 2025
16	Sesuai dengan Output	Berhasil	Valid	Dewa Putu Deni Hermawan Pande Gede Setya Permana	30 Oktober 2025
17	Sesuai dengan Output	Berhasil	Valid	Dewa Putu Deni Hermawan Pande Gede Setya Permana	30 Oktober 2025
18	Sesuai dengan Output	Berhasil	Valid	Dewa Putu Deni Hermawan Pande Gede Setya Permana	30 Oktober 2025

Berdasarkan Tabel 1. Hasil Pengujian Black Box Test, hasil yang didapatkan sesuai dengan Output yang diharapkan.

b. White Box Test

Pengujian White Box Test dilakukan pada fungsi utama system yaitu pada fungsi monitoring perangkat dengan otomatis. Pengujian difokuskan pada file pingDevice.php yang dibuat dengan bahasa pemrograman PHP untuk diuji fungsinya. Berdasarkan Flowgraph tersebut dapat ditentukan hasil perhitungan sebagai berikut:

1. Parameter graph

a. Jumlah node (N) = 21

(node bernomor 1 - 21 dari diagram)

b. Jumlah edge (E) = 27

(daftar edge dihitung dari semua possible control-flow di diagram)

c. Jumlah predicate/decision nodes (P) = 7

(node keputusan: 3, 4, 8, 10, 12, 14, 17)

2. Perhitungan *Cyclomatic Complexity*

Menggunakan rumus persamaan sebagai berikut:

$$V(G) = E - N + P \quad (1)$$

Pada persamaan tersebut, **V(G)** merupakan nilai Cyclomatic Complexity, **E** adalah jumlah *edges* (garis penghubung antar node), **N** adalah jumlah *nodes* (simpul pada flowgraph), dan **P** merupakan jumlah komponen terhubung (*connected component*). Pada penelitian ini digunakan satu flowgraph sehingga nilai **P** = 1. Sehingga berdasarkan persamaan persamaan tersebut didapatkan hasil sebagai Cyclomatic Complexity berikut

$$V(G) = 27 - 21 + 1 = 8$$

Jadi $V(G) = 8 \rightarrow$ diperlukan 8 jalur independen.

3. Jalur Independen

Berdasarkan hasil perhitungan Cyclomatic Complexity didapatkan 8 jalur independent yang menutupi semua cabang keputusan utama sebagai berikut:

Path 1 — Cooldown true (skip) $\rightarrow$ kembali & keluar loop.

Yaitu: 1 - 2 - 3 - 4 - 3 - 21

Path 2 — RebootAttempt $\geq$ Max (langsung set INACTIVE)

Yaitu: 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 3 - 21

Path 3 — Butuh reboot $\rightarrow$ SSH gagal $\rightarrow$ failcount $\geq$ max $\rightarrow$ set INACTIVE

Yaitu: 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 11 - 12 - 13 - 14 - 15 - 3 - 21

Path 4 — Butuh reboot $\rightarrow$ SSH gagal $\rightarrow$ failcount $<$ max $\rightarrow$ kembali (tidak INACTIVE)

Yaitu: 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 11 - 12 - 13 - 14 - 3 - 21

Path 5 — Butuh reboot $\rightarrow$ SSH ok $\rightarrow$ kirim reboot & reset counter

Yaitu: 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 11 - 12 - 16 - 3 - 21

Path 6 — Tidak butuh reboot → Device offline after reboot? = Yes → cek gateway & internet → report

Yaitu: 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 17 - 18 - 19 - 3 - 21

Path 7 — Tidak butuh reboot → Device offline after reboot? = No → simpan log DevicePing

Yaitu: 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 17 - 20 - 3 - 21

Path 8 — Cabang umum tanpa reboot/diagnosis, lewat semua update & simpan log lalu lanjut

Yaitu: 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 17 - 20 - 3 - 21

(Path 7 dan 8 mirip tapi dianggap berbeda karena variasi kondisi internal — pada praktik Anda bisa memilih variasi lain yang men-trigger cabang yang berbeda seperti ping tinggi vs RTO untuk memenuhi 8 jalur independen)

4. Test Case Dan Skenario

Berdasarkan 8 jalur independen tersebut dapat dirancang sebuah Test Case dan skenario pengujian yang dapat dilihat pada Tabel 2. Test Case dan Skenario White Box Test

Tabel 2. Test Case dan Skenario White Box Test

TC	Deskripsi	Input	Langkah (Node Sequence)	Hasil yang diharapkan	Path	Kesimpulan
TC1	Cooldown aktif → skip ping	lastReboot[device] baru saja di-set (time - lastReboot < rebootCooldown)	1-2-3-4-3-21	Device dilewati; tidak ada ping/reboot; loop lanjut	1	Valid
TC2	RebootAttempt >= max → set inactive	rebootAttemptCounters[device] >= maxRebootAttempts	1-2-3-4-5-6-7-8-9-3-21	Device status → inactive; kirim Telegram peringatan	2	Valid
TC3	Butuh reboot → SSH gagal berkali-kali → INACTIVE	device.auto_enabled=true; rtoCounters >= maxFailCount; SSH login selalu gagal; failedRebootCounters naik sampai >= maxFailCount	1-2-3-4-5-6-7-8-10-11-12-13-14-15-3-21	Device set inactive; Telegram gagal reboot & saran cek perangkat	3	Valid
TC4	Butuh reboot → SSH gagal sekali (fail < max) → coba ulang nanti	sama namun failedRebootCounters < maxFailCount	1-2-3-4-5-6-7-8-10-11-12-13-14-3-21	failedRebootCounters++; device tetap aktif (not INACTIVE)	4	Valid
TC5	Butuh reboot → SSH sukses → reboot terkirim	device.auto_enabled=true; rtoCounters >= maxFailCount; SSH login sukses	1-2-3-4-5-6-7-8-10-11-12-16-3-21	Kirim perintah reboot, counters reset, lastReboot set, kirim Telegram reboot	5	Valid
TC6	Setelah reboot masih offline → gateway/internet down	rebootAttemptCounters >=1; status offline; gateway & internet unreachable	1-2-3-4-5-6-7-8-10-17-18-19-3-21	Telegram: Gateway & Internet tidak terjangkau (diagnosis)	6	Valid
TC7	Setelah reboot masih offline → gateway ok, internet ok (problem internal)	rebootAttemptCounters >=1; status offline; gatewayOk=true; internetOk=true	1-2-3-4-5-6-7-8-10-17-18-19-3-21	Telegram: Gateway & Internet OK → kemungkinan masalah perangkat	6	Valid
TC8	Normal online / log tersimpan	device online, ping normal, tidak butuh reboot	1-2-3-4-5-6-7-8-10-17-20-3-21	DevicePing entry disimpan; no reboot; console info ditampilkan	7 / 8	Valid

Hasil pengujian sistem menggunakan metode White Box Test pada Sistem Dapat dilihat pada Tabel 2. Test Case dan Skenario White Box Test, Dimana berdasarkan jalur independen yang didapat dari hasil perhitungan Cyclomatic Complexity dengan melakukan delapan pengujian, didapatkan hasil kedelapannya valid, artinya tidak ditemukan kesalahan fungsi.

5. Uji Performa

Pengujian performa ini dilakukan dengan menjalankan 1 sampai 10 Device untuk sampel mengukur performa sistem dalam memonitoring Devices. Hasil Uji Performa dapat dilihat pada Tabel 3. Hasil Uji Performa Sistem

Tabel 3. Hasil Uji Performa Sistem

Jumlah Device	CPU	RAM	Waktu Eksekusi	Kondisi
1–4	8–14%	180–198 MB	Sangat cepat	Sangat ringan
5–7	17–23%	207–226 MB	Stabil	Ringan
8–10	26–32%	237–252 MB	Cukup Stabil	Stabil

Berdasarkan hasil pengujian Performa sistem pada Tabel 3. Hasil Uji Performa Sistem. Sistem dapat memonitoring sangat efisien dan dapat menangani hingga 10 perangkat tanpa mengalami masalah performa.

3.3 Pembahasan

Hasil penelitian menunjukkan bahwa sistem *network automation* yang dikembangkan mampu mengotomatisasi pengelolaan perangkat MikroTik secara terpusat melalui antarmuka berbasis web. Seluruh fitur utama, meliputi manajemen perangkat, monitoring konektivitas, reboot perangkat, serta pencatatan aktivitas, berhasil diimplementasikan dan berfungsi sesuai dengan kebutuhan sistem. Berdasarkan hasil pengujian fungsional (*black box testing*), setiap fitur dapat dijalankan tanpa ditemukan kesalahan fungsi. Administrator dapat menambahkan, mengubah, menghapus, dan mengelola data perangkat MikroTik, melakukan reboot perangkat dari sistem, memantau status koneksi perangkat. Hal ini menunjukkan bahwa sistem telah memenuhi kebutuhan fungsional yang dirancang pada tahap analisis.

Dari sisi kinerja, sistem mampu mengelola beberapa perangkat MikroTik secara efisien dengan memanfaatkan RouterOS API sebagai media komunikasi antara aplikasi dan router. Proses otomatisasi mengurangi waktu konfigurasi dibandingkan dengan metode manual menggunakan Winbox atau SSH, sehingga pekerjaan administrator menjadi lebih cepat, konsisten, dan meminimalkan kesalahan konfigurasi. Implementasi database MySQL memungkinkan seluruh data perangkat, aktivitas pengguna, serta hasil proses otomatisasi tersimpan secara terstruktur. Selain memudahkan pengelolaan data, pencatatan aktivitas juga meningkatkan kemudahan dalam proses monitoring dan evaluasi apabila terjadi gangguan jaringan.

Secara keseluruhan, hasil penelitian membuktikan bahwa sistem *network automation* yang dikembangkan mampu meningkatkan efisiensi pengelolaan jaringan MikroTik. Dengan adanya proses otomatisasi dan pengelolaan terpusat, administrator dapat melakukan konfigurasi serta pemantauan perangkat dengan lebih cepat, efektif, dan akurat dibandingkan metode pengelolaan secara manual.

4. KESIMPULAN

Berdasarkan hasil perancangan, implementasi, dan pengujian, sistem *Network Automation* untuk manajemen router MikroTik berhasil dikembangkan sebagai solusi pengelolaan jaringan secara terpusat. Sistem menyediakan berbagai fitur, seperti manajemen perangkat, otomatisasi konfigurasi, monitoring status perangkat, notifikasi, dan pengelolaan keamanan jaringan yang membantu administrator dalam mengelola banyak router secara lebih efisien. Hasil pengujian menggunakan *Black Box Testing* dan *White Box Testing* menunjukkan bahwa seluruh fungsi dan logika sistem telah berjalan sesuai dengan kebutuhan, sedangkan pengujian performa membuktikan sistem mampu memonitor hingga 10 perangkat secara stabil dengan penggunaan sumber daya yang efisien. Secara keseluruhan, sistem yang dikembangkan mampu meningkatkan efisiensi operasional, mengurangi risiko *human error*, mempercepat proses administrasi, serta mendukung monitoring dan pengelolaan jaringan secara terpusat dan otomatis. Untuk pengembangan selanjutnya, sistem dapat diperluas agar mendukung perangkat jaringan dari berbagai vendor selain MikroTik, seperti Cisco, Juniper, dan Aruba, sehingga memiliki interoperabilitas yang lebih baik. Selain itu, fitur otomatisasi dapat dikembangkan dengan menambahkan mekanisme *backup* konfigurasi otomatis, otomatisasi VLAN, manajemen *Quality of Service* (QoS), *provisioning* perangkat secara massal, serta integrasi dengan sistem notifikasi dan pemantauan yang lebih komprehensif guna meningkatkan keandalan, fleksibilitas, dan efisiensi pengelolaan jaringan.

REFERENCES

- [1] L. G. Mauboy and T. Wellem, “Studi Perbandingan Library Untuk Implementasi Network Automation Menggunakan Paramiko Dan Netmiko Pada Router Mikrotik,” *JURIKOM (Jurnal Riset Komputer)*, vol. 9, no. 4, p. 790, 2022, doi: 10.30865/jurikom.v9i4.4420.
- [2] E. S. Ginting, S. Suroso, and I. Hadi, “Pengujian Konfigurasi Otomatis Penambahan Gateway pada Virtual Router Menggunakan Aplikasi Otomatisasi Jaringan Berbasis Web,” *Jurnal Media Informatika Budidarma*, vol. 4, no. 4, pp. 1126–1131, 2020, doi: 10.30865/mib.v4i4.2485.
- [3] S. Nugroho and B. Pujiarto, “Network Automation pada Beberapa Perangkat Router Menggunakan Pemrograman Python,” *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 9, no. 1, p. 79, 2022, doi: 10.25126/jtiik.2022913947.

- [4] S. Hanadwiputra, S. Subandri, and D. Prawinarko, “Implementasi Konsep Software Defined Networking (SDN) Wide Area Network (WAN) Pada Mikrotik Dengan Python 3,” *JUPITER : Journal of Computer & Information Technology*, vol. 4, no. 2, pp. 66–79, 2023, doi: 10.53990/jupiter.v4i2.231.
- [5] P. Mihăilă, T. Balan, R. Curpen, and F. Sandu, “Network Automation and Abstraction using Python Programming Methods,” *MACRo 2015*, vol. 2, Oct. 2017, doi: 10.1515/macro-2017-0011.
- [6] N. A. Santoso, Z. Zakaria, and R. D. Kurniawan, “Analisis Jaringan Komputer Menggunakan Teknologi Virtualisasi,” *Jurnal Minfo Polgan*, vol. 11, no. 2, pp. 52–58, 2022, doi: 10.33395/jmp.v11i2.11652.
- [7] A. Zakaria, A. Prihantara, and A. A. Hartono, “Integrasi Application Programming Interface, PHP, dan MySQL untuk Otomatisasi Verifikasi dan Aktivasi Pengguna Layanan Hotspot MikroTik,” *JUITA : Jurnal Informatika*, vol. 7, no. 2, p. 63, 2019, doi: 10.30595/juita.v7i2.4361.
- [8] 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,” *Jurnal Ilmiah Komputasi*, vol. 19, no. 2, pp. 127–134, 2020, doi: 10.32409/jikstik.19.2.80.
- [9] G. S. Santyadiputra, I. M. E. Listartha, and G. A. J. Saskara, “The effectiveness of Automatic Network Administration (ANA) in network automation simulation at Universitas Pendidikan Ganesha,” *J. Phys. Conf. Ser.*, vol. 1810, no. 1, 2021, doi: 10.1088/1742-6596/1810/1/012028.
- [10] M. Fahmi, M. Maisyaroh, I. Komarudin, S. Faizah, and I. Fadhillah, “Otomatisasi Jaringan Menggunakan Script Python Untuk Penyediaan Konfigurasi Internet Dan Manajemen Mikrotik,” *Bina Insani Ict Journal*, vol. 8, no. 1, p. 53, 2021, doi: 10.51211/biict.v8i1.1517.
- [11] F. D. J. Prasetyo, H. Sutarjo, D. Triantoro, and D. V. S. Y. Sakti, “Otomatisasi Backup Konfigurasi Perangkat Jaringan Komputer Cisco,” *Journal MIND Journal | ISSN*, vol. 9, no. 1, pp. 99–112, 2024, [Online]. Available: <https://ejurnal.itenas.ac.id/index.php/mindjournal/article/view/11204/3665>
- [12] Y. C. Eka Paksi and I. R. Widiyari, “Website Network Automation Design and Implementation in Rt Rw Net Senden Dusun Magelang With Django Framework,” *Jurnal Teknik Informatika (Jutif)*, vol. 3, no. 5, pp. 1313–1322, 2022, doi: 10.20884/1.jutif.2022.3.5.350.
- [13] D. S. Purnia, A. Rifai, and S. Rahmatullah, “Penerapan Metode Waterfall dalam Perancangan Sistem Informasi Aplikasi Bantuan Sosial Berbasis Android,” *Seminar Nasional Sains dan Teknologi 2019*, pp. 1–7, 2019.
- [14] R. A. Wiryawan and N. R. Rosyid, “Pengembangan Aplikasi Otomatisasi Administrasi Jaringan Berbasis Website Menggunakan Bahasa Pemrograman Python,” *Simetris: Jurnal Teknik Mesin, Elektro dan Ilmu Komputer*, vol. 10, no. 2, pp. 741–752, 2019.
- [15] Muchamad Mashudi, “Network Automation Menggunakan Bahasa Pemrograman Python,” *Jurnal Teknik industri, Sistem informasi dan Teknik informatika*, no. December, pp. 120–122, 2022.
- [16] H. Jusuf, “Penggunaan Secure Shell (SSH) Sebagai Sistem Komunikasi Aman Pada Web Ujian Online,” *Bina Insani Ict Journal*, vol. 2, no. 2, pp. 75–84, 2015.
- [17] S. C. Alfara, “Pengembangan dan Implementasi Sistem Informasi Manajemen Zakat Berbasis Web Menggunakan Metode Waterfall dan Blackbox Testing,” *Jurnal Pendidikan dan Teknologi Indonesia*, vol. 3, no. 9, pp. 391–412, 2023, doi: 10.52436/1.jpti.327.
- [18] C. T. Pratala, E. M. Asyer, I. Prayudi, and A. Saifudin, “Pengujian White Box pada Aplikasi Cash Flow Berbasis Android Menggunakan Teknik Basis Path,” *Jurnal Informatika Universitas Pamulang*, vol. 5, no. 2, p. 111, 2020, doi: 10.32493/informatika.v5i2.4713.
- [19] A. Ramli, S. Sriyono, and H. Ramza, “Analisa Kecepatan Lalu Lintas Data Jaringan Local Area Network Menggunakan Graphical Network Simulator 3 (GNS-3),” *Electrical Engineering Acta*, vol. 1, no. 1, pp. 13–19, 2021, doi: 10.22236/ate.v1i1.6946.