

Perancangan Aplikasi Manajemen Perangkat Mikrotik Dengan Memanfaatkan *Application Programming Interface RouterOS* Menggunakan Metode *Waterfall*

Rudi Hartono*, Dzulfiqar Rafid Adhitya Pratama, Fendi Aji Purnomo
Prodi Diploma III Teknik Informatika, Sekolah Vokasi, Universitas Sebelas Maret
*Email: rudi.hartono@staff.uns.ac.id

Info Artikel

Kata Kunci :

Codelgniter, RouterOS API, Manajemen Tool, Web, Metode Waterfall

Keywords :

Codelgniter, RouterOS API, Tool Management, Web, Waterfall Method

Tanggal Artikel :

Dikirim : 11 April 2023

Direvisi : 3 Mei 2023

Diterima : 30 Mei 2023

Abstrak

Penelitian ini mengembangkan aplikasi manajemen perangkat MikroTik menggunakan *Application Programming Interface (API) RouterOS* dengan metode *waterfall*. Aplikasi ini bertujuan untuk memudahkan pengelolaan dan konfigurasi perangkat jaringan MikroTik, dengan fitur-fitur utama seperti konfigurasi perangkat, pemantauan perangkat, dan manajemen pengguna, serta fitur tambahan seperti pengelolaan log, manajemen firewall, dan manajemen hotspot. Proses pengembangan dimulai dengan analisis kebutuhan yang melibatkan identifikasi kebutuhan pengguna dan fitur yang diperlukan. Desain sistem mencakup perancangan arsitektur sistem, *database*, dan antarmuka pengguna. Implementasi melibatkan pengembangan kode program, integrasi dengan API RouterOS, dan pengujian menyeluruh menggunakan metode *black box* untuk memastikan semua fungsi berjalan sesuai kebutuhan fungsional. Hasil pengujian menunjukkan bahwa aplikasi ini berfungsi dengan baik dan dapat memenuhi kebutuhan pengguna. Dengan demikian, aplikasi ini diharapkan dapat membantu organisasi dan bisnis dalam mengelola perangkat MikroTik secara lebih efisien dan efektif.

Abstract

This study presents the development of a MikroTik device management application using the RouterOS Application Programming Interface (API) and the waterfall methodology. The primary goal of the application is to facilitate the management and configuration of MikroTik network devices. It features essential functionalities such as device configuration, device monitoring, and user management, along with additional features like log management, firewall management, and hotspot management. The development process commenced with a needs analysis, identifying user requirements and necessary features. The system design phase included the architectural design, database design, and user interface design. Implementation involved coding, integration with the RouterOS API, and comprehensive testing using the black box method to ensure all functions met the specified requirements. The testing results demonstrated that the application functions effectively and meets user needs. Consequently, this application is expected to assist organizations and businesses in managing MikroTik devices more efficiently and effectively.

1. PENDAHULUAN

Dalam era digital saat ini, penggunaan jaringan menjadi sangat penting dalam menjalankan bisnis dan organisasi. Jaringan yang kompleks membutuhkan manajemen yang efisien dan terstruktur agar jaringan dapat berjalan dengan baik. Salah satu perangkat jaringan yang populer digunakan adalah MikroTik, yang dapat diakses melalui sistem operasi RouterOS. Untuk membantu pengguna dalam mengelola perangkat MikroTik secara efisien, aplikasi manajemen perangkat terstruktur sangat penting. Aplikasi semacam itu dapat memanfaatkan metode *Network Development Life Cycle* (NDLC) untuk merancang dan mengimplementasikan sistem yang kuat yang memfasilitasi manajemen dan pemantauan pengguna, serta alokasi *bandwidth* untuk pengguna yang terhubung, seperti yang ditunjukkan dalam implementasi STMIK Kampus Denpasar [1]. Aplikasi ini dapat memperoleh manfaat dari memanfaatkan file model YANG yang diperluas dengan *tag* yang cukup jelas untuk pelabelan operasi jaringan dan data pemeliharaan, yang meningkatkan akurasi pengumpulan data dan efisiensi manajemen [2]. Selain itu, menggabungkan metode manajemen perangkat di mana perangkat jaringan pertama mengirim permintaan RPC ke perangkat jaringan kedua untuk beralih ke model YANG yang kompatibel dapat mengurangi ketergantungan dan memperluas jangkauan aplikasi, sehingga meningkatkan efisiensi manajemen [3]. Sistem ini juga dapat mengintegrasikan tag identifikasi frekuensi radio (RFID) dan terminal seluler untuk membaca, memodifikasi, dan mengirimkan informasi terkait perangkat, memastikan pembaruan dan kontrol waktu nyata [4]. Selanjutnya, arsitektur multi-server di mana terminal berkomunikasi dengan beberapa server dan perangkat dapat merampingkan proses manajemen dengan mengirimkan status koneksi dan instruksi pengaturan parameter, sehingga mengatasi masalah efisiensi rendah yang lazim dalam sistem manajemen perangkat tradisional [5]. Dengan menggabungkan metodologi dan teknologi canggih ini, aplikasi dapat memberikan solusi komprehensif untuk mengelola perangkat MikroTik, memastikan efisiensi tinggi, akurasi, dan kemudahan penggunaan bagi pengguna..

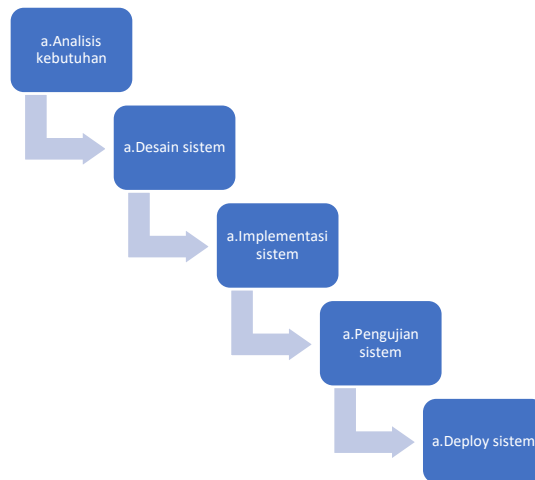
Application Programming Interface (API) RouterOS dapat digunakan untuk mengakses dan mengelola perangkat MikroTik. Dengan memanfaatkan API RouterOS, pengguna dapat melakukan konfigurasi, pemantauan, dan manajemen perangkat MikroTik dengan lebih mudah dan efisien. Selain itu, API RouterOS juga dapat digunakan untuk mengembangkan aplikasi manajemen perangkat MikroTik yang lebih terstruktur dan efisien. RouterOS API memang dapat digunakan untuk mengembangkan aplikasi manajemen perangkat MikroTik yang lebih terstruktur dan efisien. Pengembangan aplikasi tersebut dapat ditingkatkan secara signifikan dengan memanfaatkan RouterOS API, yang memungkinkan pembuatan aplikasi seluler yang memungkinkan administrator jaringan untuk mengelola router MikroTik dari jarak jauh, sehingga meningkatkan efisiensi dan fleksibilitas dalam manajemen jaringan [6]. Ini sangat bermanfaat dalam lingkungan jaringan yang kompleks, seperti yang ditemukan dalam aplikasi industri 4.0, di mana jaringan internet berkinerja tinggi sangat penting. Dengan mengkonfigurasi berbagai parameter jaringan seperti rute IP, filter *firewall*, NAT, dan manajemen *bandwidth* melalui RouterOS API, administrator dapat memastikan kinerja jaringan yang optimal dan menerapkan kebijakan secara efektif [7]. Selanjutnya, penggunaan kerangka kerja aplikasi dan pola desain, seperti yang ditunjukkan oleh kerangka kerja Layla untuk antarmuka manajemen jaringan (NMI), dapat merampingkan proses pengembangan dan meningkatkan kualitas aplikasi yang dihasilkan [8]. Selain itu, mengintegrasikan mekanisme manajemen otomatis ke dalam aplikasi ini dapat lebih meningkatkan fungsinya dengan mendeteksi dan melaporkan kesalahan tanpa memerlukan intervensi manual, sehingga mengurangi upaya pengembangan dan membuat manajemen sistem terdistribusi lebih mudah [9]. Pendekatan hibrida menggabungkan sumber energi yang berbeda, seperti yang terlihat dalam sistem hibrida FC/UC/BT, juga dapat diterapkan pada manajemen jaringan dengan menyeimbangkan berbagai sumber daya jaringan untuk memenuhi permintaan dinamis secara efisien [10]. Secara keseluruhan, RouterOS API memberikan dasar yang kuat untuk mengembangkan aplikasi manajemen MikroTik canggih yang efisien dan mampu menangani kompleksitas lingkungan jaringan modern.

Penelitian ini bertujuan untuk merancang aplikasi manajemen perangkat MikroTik dengan memanfaatkan API RouterOS menggunakan metode *waterfall*. Metode *waterfall* dipilih karena dapat memastikan tahapan perancangan dan pengembangan sistem yang sistematis dan terstruktur. Tahapan metode *waterfall* meliputi analisis kebutuhan, desain sistem, implementasi sistem, pengujian sistem, dan *deploy* sistem

2. METODE PENELITIAN

Dalam penelitian ini menggunakan metode *waterfall*. Metode *waterfall* banyak dipakai dalam pengembangan produk digital [11], [12]. Metode *waterfall* adalah metode yang diperkenalkan oleh Roice dan mengalami perbaikan dengan versi terbaru diperkenalkan oleh Sommerville pada tahun 2011. Tahapan metode

waterfall ditunjukkan pada gambar 1. Perancangan aplikasi manajemen perangkat MikroTik dengan memanfaatkan *Application Programming Interface* (API) RouterOS menggunakan metode *waterfall* dapat dilakukan dengan langkah-langkah pada gambar. 1:



Gambar. 1 Tahapan metode *waterfall*

Berdasar gambar 1, Langkah pertama dalam perancangan aplikasi adalah melakukan analisis kebutuhan. Analisis ini bertujuan untuk memastikan bahwa aplikasi yang dibuat dapat memenuhi kebutuhan pengguna dengan baik. Dalam tahap ini, perlu dipertimbangkan berbagai fitur yang diperlukan, jenis data yang akan digunakan, dan kebutuhan pengguna lainnya untuk memastikan aplikasi yang dirancang benar-benar sesuai dengan harapan pengguna. Setelah analisis kebutuhan selesai, langkah selanjutnya adalah merancang sistem secara keseluruhan. Pada tahap desain sistem, dibuatlah rancangan yang mencakup desain *database*, antarmuka pengguna, dan fungsionalitas aplikasi. Selain itu, desain sistem juga harus mempertimbangkan arsitektur sistem, pemilihan teknologi, dan skema pengembangan aplikasi yang tepat.

Setelah desain sistem selesai, tahap berikutnya adalah implementasi. Pada tahap ini, pengembangan kode program dilakukan sesuai dengan rancangan yang telah dibuat. Proses ini juga melibatkan integrasi dengan API RouterOS dan pengujian sistem secara menyeluruh untuk memastikan semua bagian berfungsi dengan baik. Selama implementasi, penting untuk memperhatikan standar kualitas kode, tata letak antarmuka pengguna yang baik, dan integrasi yang tepat dengan API RouterOS.

Setelah sistem diimplementasikan, perlu dilakukan pengujian sistem secara menyeluruh untuk memastikan bahwa sistem bekerja dengan baik dan sesuai dengan kebutuhan pengguna. Pengujian ini mencakup pengujian fungsionalitas, kinerja, dan keamanan untuk memastikan bahwa sistem dapat diandalkan dan aman digunakan. Setelah pengujian selesai dan sistem dinyatakan siap, langkah terakhir adalah men-*deploy* sistem ke lingkungan produksi. Pada tahap ini, faktor-faktor seperti keamanan, skalabilitas, dan ketersediaan sistem harus diperhatikan. Setelah sistem di-*deploy*, perlu dilakukan pemantauan dan pemeliharaan secara berkala untuk memastikan sistem tetap berjalan dengan baik dan terus memenuhi kebutuhan pengguna. Dengan mengikuti langkah-langkah tersebut, aplikasi manajemen perangkat MikroTik dengan memanfaatkan API RouterOS dapat dirancang dan diimplementasikan dengan baik menggunakan metode *waterfall*.

3. HASIL DAN PEMBAHASAN

API RouterOS merupakan sebuah antarmuka pemrograman aplikasi yang digunakan untuk mengakses dan mengelola perangkat MikroTik. Dalam penelitian ini, penggunaan API RouterOS sangat penting untuk mengakses data pada perangkat MikroTik dan melakukan konfigurasi pada perangkat tersebut. Selain itu, penggunaan API RouterOS juga memungkinkan aplikasi untuk mengakses data pada perangkat MikroTik dari jarak jauh. Hasil aplikasi manajemen perangkat MikroTik yang dilengkapi dengan fitur-fitur dasar seperti konfigurasi perangkat, pemantauan perangkat, dan manajemen pengguna. Selain itu, aplikasi ini juga dilengkapi dengan fitur tambahan seperti

pengelolaan log, manajemen *firewall*, dan manajemen *hotspot*. Aplikasi ini dapat membantu organisasi dan bisnis dalam mengelola perangkat MikroTik dengan lebih mudah dan efisien.

3.1 Kebutuhan Fungsional

Kebutuhan Fungsional yang terdapat pada Aplikasi Konfigurasi MikroTik Dengan RouterOS API (*Application Programming Interface*) PHP Berbasis Web adalah sebagai berikut pada Tabel 1:

Tabel 1. Kebutuhan fungsional game

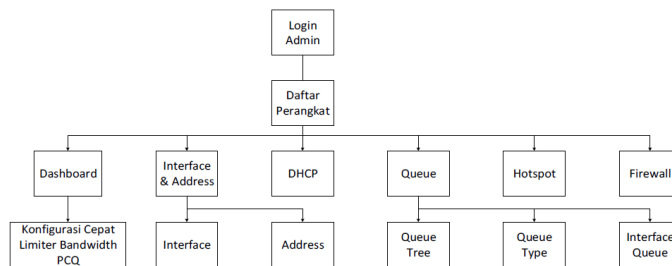
Kode FR	Deskripsi FR	Aktor	Dependensi
FR01	Sistem mampu menerima <i>login</i>	Admin	
FR02	Sistem dapat menambahkan admin	Admin	
FR03	Sistem dapat menghapus admin	Admin	
FR04	Sistem dapat menampilkan IP perangkat	Admin	
FR05	Sistem dapat menambahkan IP Perangkat	Admin	
FR06	Sistem dapat menghapus IP perangkat	Admin	
FR07	Sistem dapat mengubah IP perangkat	Admin	
FR08	Sistem dapat <i>login</i> atau menyambungkan ke IP perangkat	Admin	
FR09	Sistem dapat <i>logout</i> atau memutuskan koneksi dari IP perangkat	Admin	
FR10	Sistem dapat menampilkan informasi Resources dari perangkat router	Admin	
FR11	Sistem dapat menampilkan informasi <i>Interface</i> dari perangkat router	Admin	
FR12	Sistem dapat menampilkan informasi <i>IP Address</i> dari perangkat router	Admin	
FR13	Sistem dapat menambahkan <i>IP Address</i> ke perangkat router	Admin	
FR14	Sistem dapat menghapus <i>IP Address</i> di perangkat router	Admin	
FR15	Sistem dapat mengedit <i>IP Address</i> di perangkat router	Admin	
FR16	Sistem dapat menampilkan <i>firewall mangle</i> di perangkat router	Admin	
FR17	Sistem dapat menghapus <i>firewall mangle</i> di perangkat router	Admin	
FR18	Sistem dapat mengedit <i>firewall mangle</i> di perangkat router	Admin	
FR19	Sistem dapat mengedit <i>firewall mangle</i> di perangkat router	Admin	
FR20	Sistem dapat menghapus <i>firewall NAT</i> di perangkat router	Admin	
FR21	Sistem dapat mengedit <i>firewall NAT</i> di perangkat router	Admin	
FR22	Sistem dapat mengedit <i>firewall NAT</i> di perangkat router	Admin	
FR23	Sistem dapat menghapus <i>Queue Tree</i> di perangkat router	Admin	
FR24	Sistem dapat mengedit <i>Queue Tree</i> di perangkat router	Admin	
FR25	Sistem dapat mengedit <i>Queue Tree</i> di perangkat router	Admin	

FR26	Sistem dapat menghapus <i>Queue Type</i> di perangkat router	Admin
FR27	Sistem dapat mengedit <i>Queue Type</i> di perangkat router	Admin
FR28	Sistem dapat membuat konfigurasi manajemen <i>bandwidth</i> PCQ	Admin

Aplikasi berkomunikasi dengan perangkat Mikrotik menggunakan RouterOS API (*Application Programming Interface*). Aktor yang ada adalah yaitu admin, admin dapat mengolah semua data yang ada pada aplikasi, melihat kondisi perangkat router mikrotik, mengkonfigurasi perangkat router mikrotik. Pengolahan yang dilakukan admin adalah pengolahan data perangkat, pengolahan konfigurasi perangkat. Admin juga dapat membuat konfigurasi perangkat seperti *limiter bandwidth* PCQ, membuat konfigurasi *IP Address*, membuat konfigurasi *firewall* NAT dan Mangle.

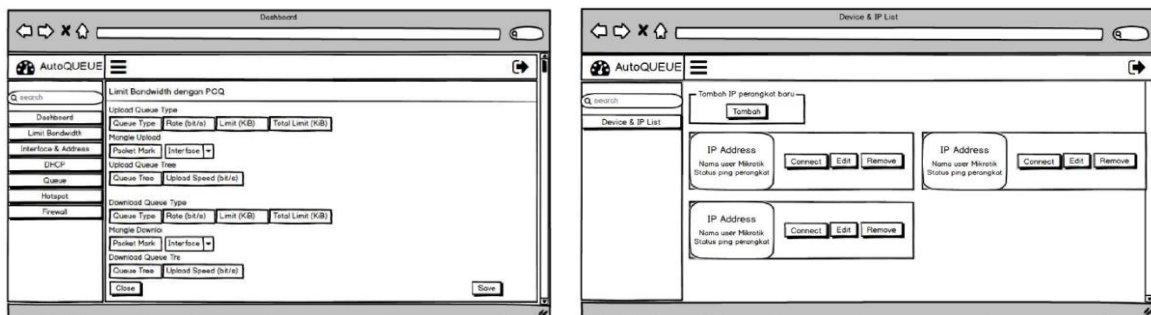
3.2 Struktur dan Desain layout

Berikut ini perancangan struktur dan desain layout dari aplikasi yang digunakan untuk melakukan manajemen perangkat mikrotik, ditunjukkan pada gambar. 2.



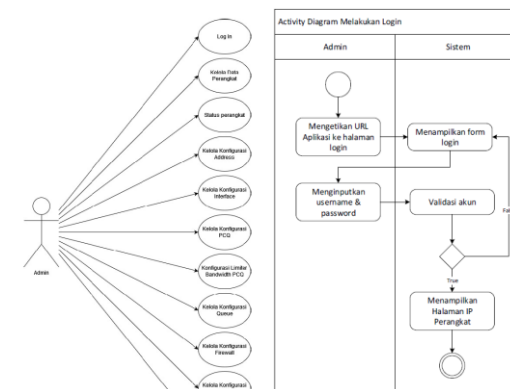
Gambar. 2 Struktur dan Desain layout

Berikut ini adalah mockup halaman *device* dan *IP* yang berisikan daftar *IP* perangkat dan statusnya, dalam *mockup* tersebut juga terdapat tombol untuk menambahkan *IP address* yang ditunjukkan pada gambar. 3



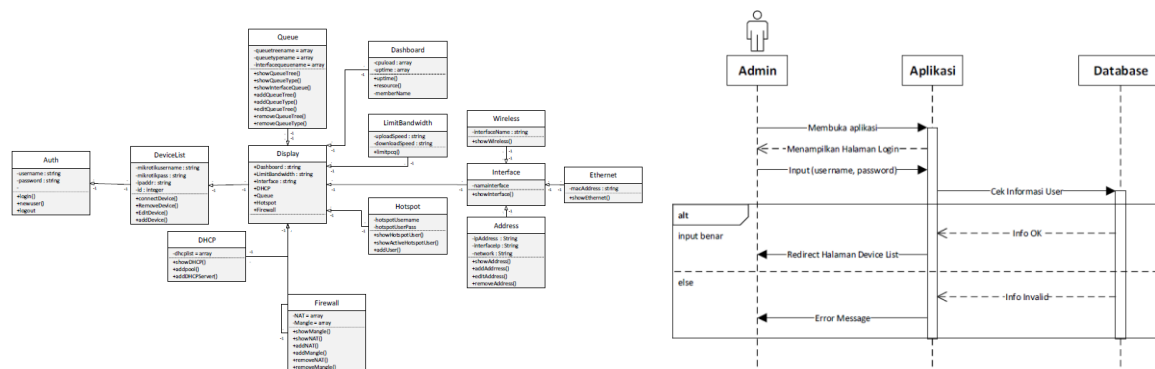
Gambar. 3 Mockup utama dan detail aplikasi

Use case diagram dan *Activity login* yang digunakan dalam pengembangan aplikasi ini mengacu pada *use case* yang ditunjukkan pada gambar. 4



Gambar. 4 Use case dan activity login pengembangan aplikasi

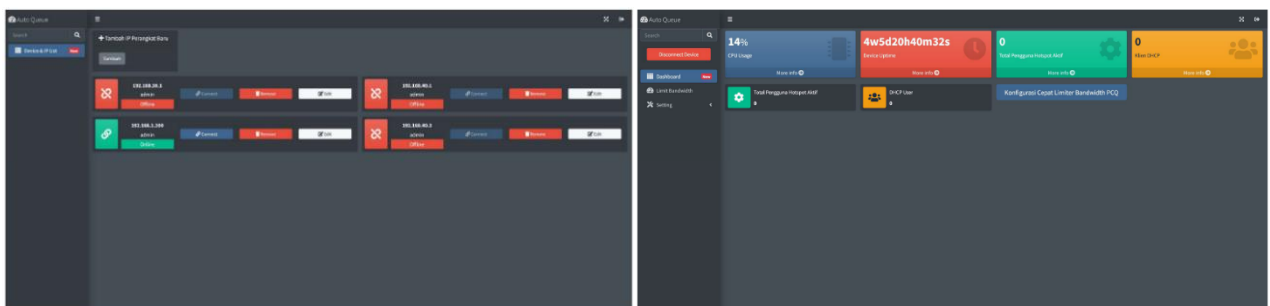
Class diagram dan sequence diagram login yang digunakan dalam pengembangan aplikasi dapat dilihat detail pada gambar. 5

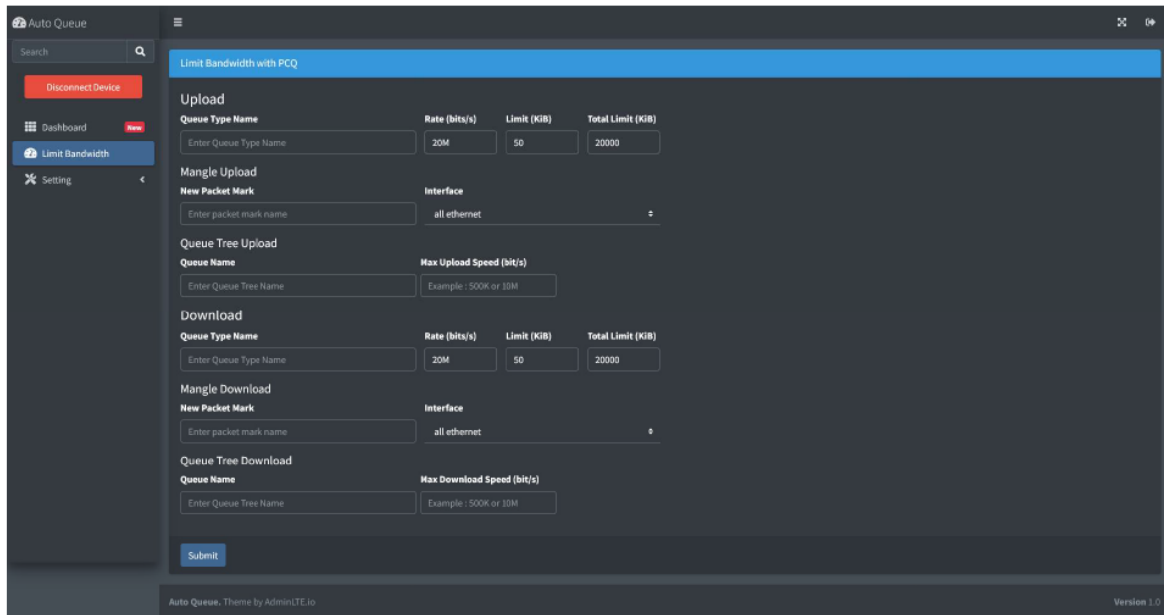


Gambar. 5 Class diagram dan sequence diagram login yang digunakan dalam pengembangan aplikasi

3.3 Pembahasan dan implementasi

Pengujian aplikasi ini menggunakan metode *black box*. Pengujian *black box* merupakan pengujian program berdasarkan fungsi dari program. Tujuan dari metode *black box* ini adalah untuk menemukan kesalahan fungsi pada program. Pengujian dengan metode *black box* dilakukan dengan cara memberikan sejumlah input pada program aplikasi yang kemudian diproses sesuai dengan kebutuhan fungsionalnya untuk melihat apakah program aplikasi menghasilkan output yang diinginkan dan sesuai dengan fungsi dari program tersebut. Apabila dari input yang diberikan proses menghasilkan *output* yang sesuai dengan kebutuhan fungsionalnya, maka program aplikasi yang bersangkutan telah benar. Tetapi jika *output* yang dihasilkan tidak sesuai dengan kebutuhan fungsionalnya, maka masih terdapat kesalahan pada program aplikasi. Tampilan dari aplikasi yang telah dikembangkan ditunjukkan pada gambar.





Gambar. 6 Tampilan hasil aplikasi yang telah dibuat

Pengujian Aplikasi ini menggunakan data uji berdasarkan data dari input data yang dimasukan. Pengujian selengkapnya terlihat pada tabel 2 dibawah ini :

Tabel 2. Hasil pengujian fungsionalitas aplikasi

ID	Object	Butir Uji	Pengujian
T1	Login ke aplikasi	Input <i>username</i> dan <i>password</i> akun aplikasi	Berhasil
T2	Logout dari aplikasi	Menekan button <i>logout</i> pada aplikasi	Berhasil
T3	Input data perangkat ke aplikasi	Memasukan data perangkat mikrotik (IP, <i>Username</i> & <i>Password</i>) ke <i>database</i> dengan aplikasi	Berhasil
T4	Edit data perangkat di aplikasi	Memasukan input data perangkat mikrotik (IP, <i>Username</i> & <i>Password</i>) dan mengubah data table <i>database</i> dengan aplikasi	Berhasil
T5	Hapus data perangkat dari aplikasi	Menekan <i>button remove</i> pada salah satu data perangkat pada aplikasi	Berhasil
T6	Connect ke perangkat Mikrotik dari aplikasi dengan API Mikrotik	Menekan <i>button connect</i> pada salah satu data perangkat yang aktif pada aplikasi	Berhasil
T7	Disconnect dari perangkat Mikrotik dari aplikasi dengan API Mikrotik	Menekan button <i>Disconnect</i> pada aplikasi	Berhasil
T8	Fungsi Konfigurasi <i>Limiter Bandwidth</i> PCQ Mikrotik	Memasukan data Form di Halaman Konfigurasi <i>Limiter Bandwidth</i> PCQ	Berhasil
T9	Menampilkan Halaman DHCP	Menekan menu <i>sidebar</i> DHCP	Berhasil

T10	Menampilkan Halaman <i>Queue / Queue Tree</i>	Menekan menu <i>Sidebar Queue</i> atau menu <i>Queue Tree</i> di halaman <i>Queue</i>	Berhasil
-----	---	---	----------

4. KESIMPULAN

Berdasarkan penelitian telah berhasil merancang dan mengimplementasikan aplikasi manajemen perangkat MikroTik dengan memanfaatkan API RouterOS menggunakan metode *waterfall*. Aplikasi ini dilengkapi dengan fitur-fitur dasar seperti konfigurasi perangkat, pemantauan perangkat, dan manajemen pengguna, serta fitur tambahan seperti pengelolaan log, manajemen *firewall*, dan manajemen *hotspot*. Proses pengembangan dimulai dari analisis kebutuhan pengguna, dilanjutkan dengan desain sistem, implementasi kode, integrasi dengan API RouterOS, dan pengujian menyeluruh menggunakan metode *black box*. Hasil pengujian menunjukkan bahwa aplikasi berfungsi dengan baik dan sesuai dengan kebutuhan fungsional yang telah ditentukan. Aplikasi ini diharapkan dapat membantu organisasi dan bisnis dalam mengelola perangkat MikroTik secara lebih efisien dan efektif..

DAFTAR PUSTAKA

- [1] I. K. J. Arta and N. B. S. Nugraha, "Implementasi aplikasi user management hotspot mikrotik berbasis php dengan application programming interface (api) dan framework bootstrap," vol. 3, no. 1, pp. 66–71, 2020.
- [2] Z. Pufeng, W. Qin, and S. Wei, "Device management method, apparatus, system, device, and storage medium," 2021.
- [3] D. Xinning, C. Rui, L. Chuxi, and W. Qin, "Device management method, device, system and device, and storage medium," 2021.
- [4] D. Lixian *et al.*, "Device management system and method," 2019.
- [5] Z. Qingtao, Z. Zhenyu, and W. Chunsheng, "Device management system," 2017.
- [6] Y. H. T. Assakur, M. S. Fahrudin, and F. Ferdiansyah, "Implementasi API Mikrotik untuk Management Router Berbasis Android (Studi Kasus: PT Sigma Adi Perkasa)," *J. Syst. Integr.*, vol. 6, no. 1, pp. 92–101, 2020.
- [7] T. Rahman, Sumarna, and H. Nurdin, "Analisis Performa RouterOS MikroTik pada Jaringan Internet," pp. 178–192, 2020.
- [8] R. K. Keller, J. Tessier, and G. von Bochmann, "A pattern system for network management interfaces," *Commun. ACM*, vol. 41, no. 9, pp. 86–93, 1998.
- [9] P.-J. Yeh, R.-S. Wu, and S.-M. Yuan, "Automatic injection of management mechanism to applications," 2006, pp. 1033–1038.
- [10] E. H. Majlan *et al.*, "ENERGY MANAGEMENT STRATEGY FOR A FUEL CELL/ULTRACAPASITOR/BATTERY HYBRID SYSTEM FOR PORTABLE APPLICATIONS (Strategi Pengurusan Tenaga bagi Sistem Hibrid Sel Bahan Api/ Ultrakapasitor/Bateri untuk Aplikasi Mudah Alih)," 2016.
- [11] D. Saputra, H. Haryani, A. Surniadari, M. Martias, and F. Akbar, "Sistem Informasi Bimbingan Tugas Akhir Mahasiswa Berbasis Website Menggunakan Metode Waterfall," *MATRIK: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 21, no. 2, pp. 403–416, Mar. 2022, doi: 10.30812/matrik.v21i2.1591.
- [12] H. Hermansyah, S. Wahyuni, and A. Akbar, "Perancangan Sarana Media Informasi Berbasis Web Desa Klambir Lima Menggunakan Metode Waterfall," *JURIKOM (Jurnal Riset Komputer)*, vol. 9, no. 2, p. 515, Apr. 2022, doi: 10.30865/jurikom.v9i2.3803.