JTERA (Jurnal Teknologi Rekayasa), Vol. 7, No. 2, Desember 2022, Hal. 223-232
Terakreditasi "**Peringkat 3**" oleh Kemenristek/BRIN, Nomor SK: 85/M/KPT/2020
DOI: 10.31544/jtera.v7.i2.2022.223-232
p-ISSN 2548-737X
e-ISSN 2548-8678

Implementasi *Application Programmable Interface* pada Sistem Manajemen Jaringan Berbasis *Website*

Erick Andika[1#], Lani Nurlani[2], Nila Natalia[3], Syayidul Muwahhid[4]
[1,3,4]Program Studi Teknik Komputer, Politeknik Sukabumi
Jl. Babakan Sirna 25, Kota Sukabumi, Jawa Barat, Indonesia
[2]Jurusan Manajemen Informatika, Politeknik Negeri Subang
Jl. Brigjen Katamso No. 37, Kabupaten Subang, Jawa Barat 41211, Indonesia
[#]erickandika@polteksmi.ac.id

Abstrak

Berkembangnya aplikasi dan media informasi digital mengharuskan tersedianya jaringan komputer yang baik. Beberapa administrator jaringan tidak memperhatikan dan mengatur besar *bandwitdh* yang digunakan oleh setiap *user*. Beragamnya aktifitas *user* seperti *browsing*, *streaming* ataupun bermain *video games* sangat membutuhkan jaringan internet yang stabil. Masalah yang dihadapi administrator jaringan dalam melakukan limitasi lalu lintas jaringan adalah kompleksitasnya konfigurasi *bandwidth* dan ketidaktahuannya akan pentingnya limitasi lalu lintas. Solusi masalah tersebut adalah dibuatkannya aplikasi *network automation* untuk manajemen *bandwidth* pada *router* MikroTik. Tujuan penelitian ini mengembangkan aplikasi yang dapat memfasilitasi administrator jaringan untuk melakukan konfigurasi dasar, *monitoring*, dan *controllin*g jaringan menggunakan *Application Programmable Interface* (API) MikroTik. Metode pengembangan sistem yang digunakan adalah *Rapid Aplication Development* (RAD). Sedangkan untuk pengujiannya menggunakan metode *Black-box*. Hasil dari pembuatan aplikasi ini adalah terciptanya sistem *network automation* yang dapat melakukan konfigurasi bandwidth secara otomatis dan mudah dengan pilihan metode yang diinginkan pengguna yaitu *Simple Queue*, *Queue Tree*, dan PCQ.

Kata kunci: *network automation*, MikroTik, *bandwidth*, *traffic monitoring*, MikroTik API

Abstract

The development of digital information media and applications requires the availability of a good computer network. Some network administrators do not pay attention and adjust the amount of bandwidth used by each user. Various user activities such as browsing, streaming or playing video games really need a stable internet network. The problem faced by network administrators in limiting network traffic is the complexity of bandwidth configuration and their ignorance of the importance of limiting traffic. The solution to this problem is the creation of a network automation application for bandwidth management on MikroTik routers. The aim of this research is to develop an application that can facilitate network administrators to perform basic configuration, monitoring, and controlling the network using the MikroTik Application Programmable Interface (API). The system development method used is Rapid Application Development (RAD). As for testing using the Black-box method. The result of making this application is the creation of a network automation system that can automatically and easily configure bandwidth by selecting the method the user wants, namely Simple Queue, Queue Tree, and PCQ.

Keywords: *network automation, MikroTik, bandwidth, traffic monitoring, MikroTik API*

I. PENDAHULUAN

Berkembangnya aplikasi dan media informasi digital mengharuskan tersedianya jaringan komputer yang baik. Hasil survei Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) 2018, jumlah pengguna internet di Indonesia sebesar 171,1 juta naik sebesar 27,9 juta dari tahun 2017 yang berjumlah 143,2 Juta. Sedangkan survei pada tahun 2019 menunjukan pengguna internet sebesar 196,7 juta [1]. Penggunaan internet dalam kegiatan perkantoran, pemerintahan maupun pendidikan

Diterima: 30 September 2022; Direvisi: 19 November 2022; Disetujui: 12 Desember 2022
JTERA, Vol. 7, No. 2, Desember 2022

© 2022 JTERA (Jurnal Teknologi Rekayasa), Politeknik Sukabumi

diperuntukan memudahkan pekerjaan. Namun dengan konfigurasi jaringan yang tidak tepat membuat pengguna internet melakukan beragam aktifitas seperti *browsing*, *streaming* ataupun bermain *video games* dalam waktu tidak seharusnya. Sehingga aktifitas ini dapat mengganggu kecepatan akses layanan jaringan dan dapat mengganggu kepentingan *user* lain.

Quality of Service (QoS) didefinisikan sebagai suatu pengukuran tentang seberapa baik jaringan dan merupakan suatu usaha untuk mendefinisikan karakteristik dan sifat dari suatu layanan. QoS mengacu pada kemampuan jaringan untuk menyediakan layanan yang lebih baik pada trafik jaringan tertentu melalui teknologi yang berbeda-beda. QoS merupakan suatu tantangan yang besar dalam jaringan berbasis IP dan internet secara keseluruhan. Tujuan dari QoS adalah untuk memenuhi kebutuhan-kebutuhan layanan yang berbeda, yang menggunakan infrastruktur yang sama. QoS menawarkan kemampuan untuk mendefinisikan atribut-atribut layanan yang disediakan, baik secara kualitatif maupun kuantitatif. [2]. Beberapa metode untuk mengatur hal tersebut, beberapa cara untuk mengatur bandwitdh dengan sebuah router adalah dengan metode *Simple Queue*, *Queue Tree* dan *Perconnection Queue* (PCQ). MikroTik yang merupakan salah satu produk *router*, saat ini telah menyediakan *Application Programming Interface* (API) untuk memfasilitasi developer membangun sistem yang dapat memudahkan administrator jaringan dalam mengelola jaringan.

Beberapa penelitian sebelumnya telah membahas mengenai bagaimana memaksimalkan kebutuhan internet seperti manajemen *traffic* menggunakan HTB [3],[4], menggunakan Simple Quee, que tree [5], [6], dan manajemen *bandwidth* pada MikroTik berbasis *website* [7], [8]. Selain teknik mengatur *bandwidth*, ada pula penerapan pada VLAN [9], ataupun monitoring jaringan yang diterapkan berbasis Android [10], [11]. Beberapa penelitian yang telah dilaksanakan terkait dengan konfigurasi *bandwidth*, sebagian lain terkait monitoring jaringan. Karena teknologi API pada jaringan masih baru, sehingga memerlukan penelitian yang seksama agar dapat dimanfaatkan untuk mengembangkan aplikasi yang bisa memastikan QoS dari sebuah jaringan sudah sesuai kebutuhan. Selain itu pada penelitian ini menunjukan bahwa pengembangan aplikasi berbasis *website* tidak selalu terkait dengan sistem informasi, namun dapat juga dimanfaatkan sebagai media *monitoring* dan *controlling* jaringan. Aplikasi berbasis *website* ini dapat dimanfaatkan mulai dari melakukan konfigurasi dasar pada jaringan, manajemen *bandwidth*, membatasi akses *user*, *monitoring traffic*, dan memprioritaskan *traffic* tertentu pada protokol jaringan.

II. METODE PENELITIAN

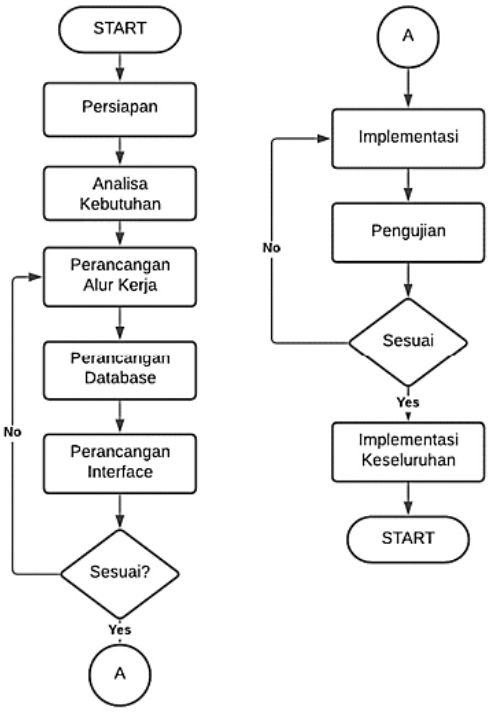
Langkah-langkah penelitian yang akan dilakukan dapat dilihat pada Gambar 1. Metode yang digunakan untuk pengembangan sistem adalah metode *Rapid Aplication Development* (RAD). Metode RAD adalah metode yang berfokus pada pengembangan aplikasi secara cepat [11]. Beberapa tahap yang perlu dilakukan pada metode ini adalah sebagai berikut.

A. Requirement Planning

Pada tahap perencanaan kebutuhan dilakukan pengumpulan dan analisis alat, bahan dan sumber daya terkait dengan sistem yang akan dibangun. Tahap ini berfokus pada penyelesaianpenyelesaian masalah yang disesuaikan dengan informasi kebutuhan dan kendala yang dihadapi pengguna.

B. Design

Fase ini adalah tahap untuk merancang sistem yang digambarkan melalui diagram-diagram yang digunakan sebagai dasar untuk membuat prototype, kemudian tim peneliti pada tahap ini akan menunjukan representasi visual kepada pengguna dan memperbaiki prototipe yang dibuat sesuai dengan respon pengguna. Tahap ini secara keseluruhan dapat digambarkan sebagai *workshop*.



Gambar 1. *Flowchart* tahap kegiatan

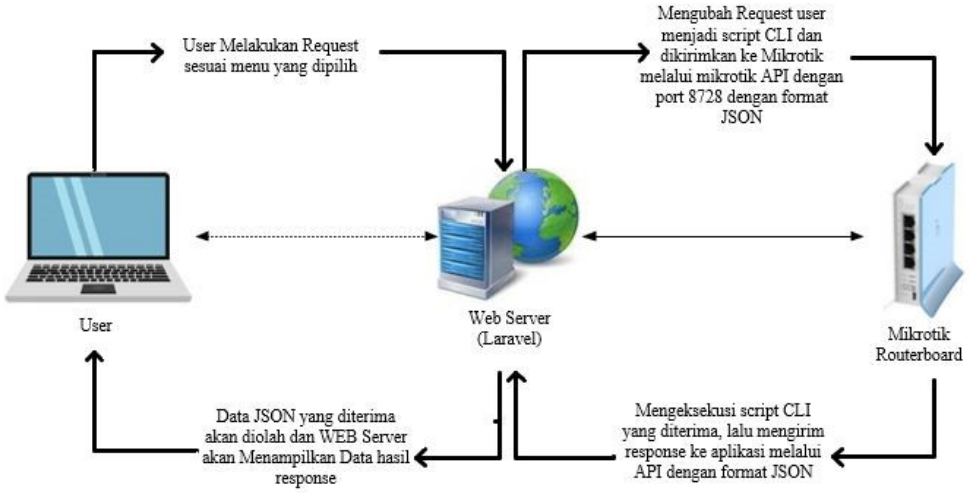
C. Implementation

Pada tahap ini dilakukan implementasi ke dalam bentuk coding secara keseluruhan dari desain yang sudah diperbaiki berdasarkan respon pengguna. Modul-modul yang telah dibuat dalam tahap design sudah dapat diganbungkan menjadi aplikasi yang utuh. Pada tahap ini juga dilakukan pengujian untuk memastikan bahwa aplikasi sudah sesuai kebutuhan.

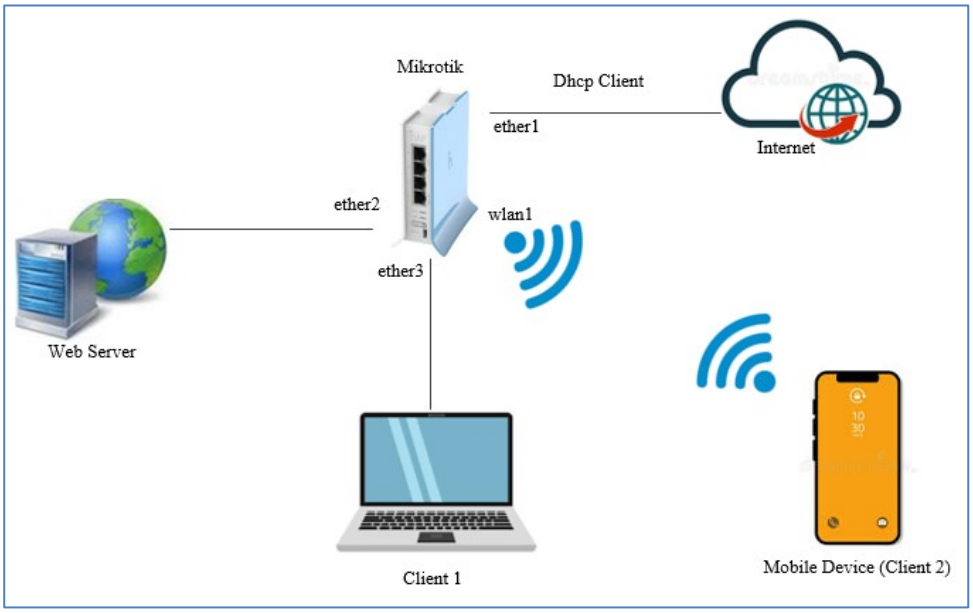
D. Mekanisme API

Secara umum gambaran sistem yang dibangun dan mekanisme API terlihat pada Gambar 2 dan rancangan topologi yang dibangun sesuai dengan Gambar 3.

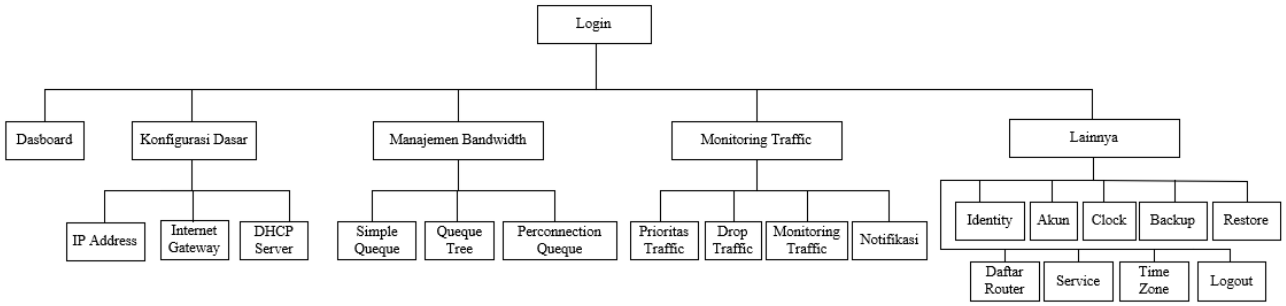
Berdasarkan gambaran umum sistem, diketahui bahwa *user* melakukan konfigurasi pada *router* melalui aplikasi sesuai menu yang dipilih, kemudian sistem akan merubah data-data yang didapat menjadi sebuah *script* perintah MikroTik misalkan "ip dhcp-client add interface=ether1 disable=no" kemudian dikirim ke *router* melalui *port* 8728 pada MikroTik API. *Router* menjalankan seluruh *script* perintah yang dikirim oleh sistem dan memberikan sebuah respon data berupa notifikasi dan data melalui *port* API 8728 pada MikroTik dengan format JSON. Respon data dengan format JSON tersebut akan diolah kembali sistem dan akan ditampilkan pada antarmuka aplikasi. Aplikasi terinstal pada sebuah PC/*server* yang terhubung dengan MikroTik *routerboard. Router* dapat terhubung ke internet dan dikonfigurasi sesuai keinginan. Untuk melakukan konfigurasi dapat langsung pada PC dimana aplikasi diinstal ataupun di-*remote* oleh PC *client*.



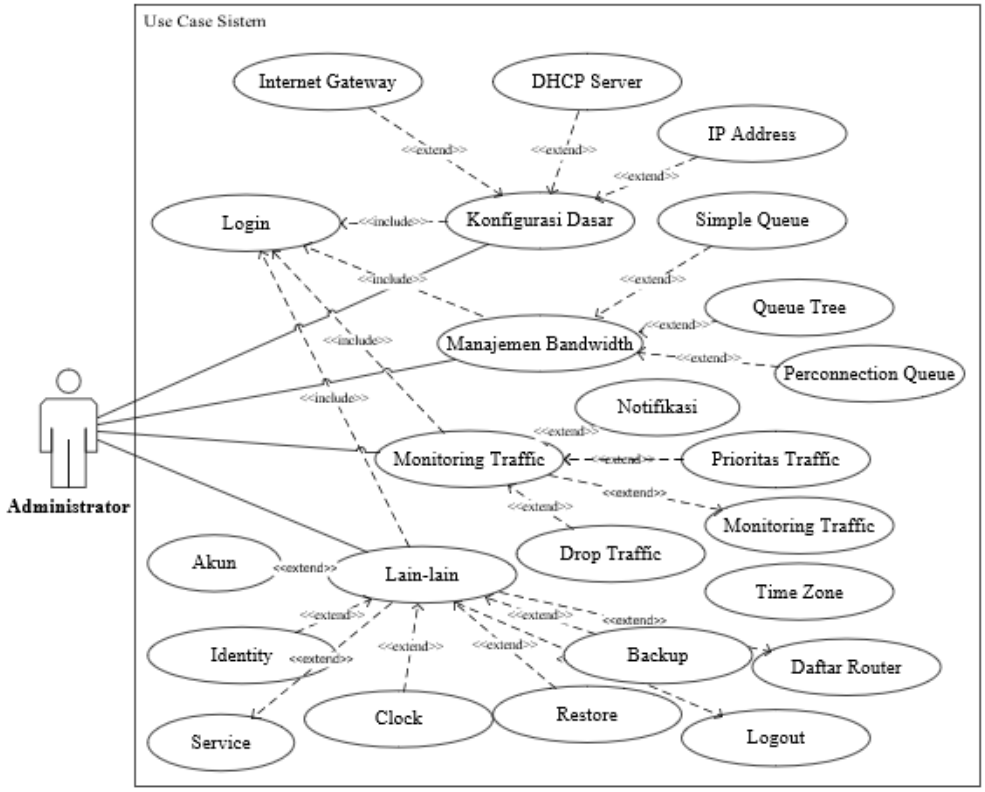
Gambar 2. Gambaran umum sistem



Gambar 3. Rancangan topologi jaringan



Gambar 4. Hirarki menu aplikasi



Gambar 5. *Use case diagram*

E. Rancangan Aplikasi

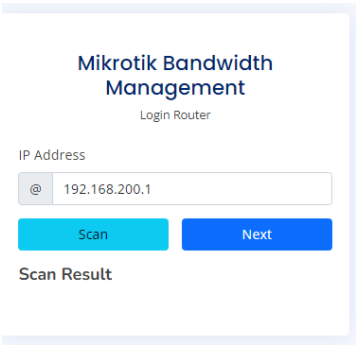
Gambar 4 menunjukan hirarki menu aplikasi yang akan dirancang. Terdapat menu dan submenu yang dimiliki sistem ini, yaitu menu Login, Dashboard, Konfigurasi Dasar, Management Bandwidth, Monitoring Traffic, dan Lainnya.

Berdasarkan *use case* pada Gambar 5, administrator dapat melakukan konfigurasi dasar yang didalamnya dimuat DHCP *server*, IP *address*, dan *internet gateway*. Lalu manajemen *bandwidth* yaitu *simple queue*, *queue tree*, dan *perconnection queue*. Kemudian memonitoring *traffic* jaringan dengan memprioritaskan *traffic*, melakukan *drop traffic*, dan notifikasi Telegram, serta menu lain-lain sebagai pengaturan tambahan untuk merubah *identity*, waktu, melakukan *backup*, dan *restore* pengaturan.

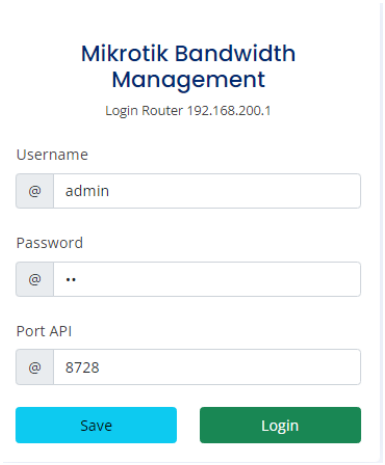
III. HASIL DAN PEMBAHASAN

A. Implementasi Sistem

Pada bagian ini akan dijelaskan hasil implementasi dari aplikasi yang telah dibuat.

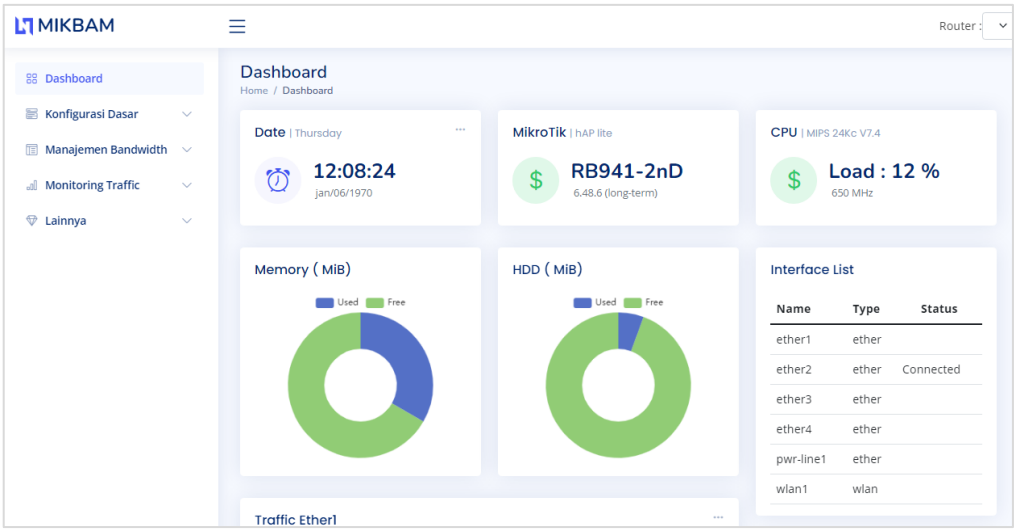


Gambar 6. Halaman *login* 1

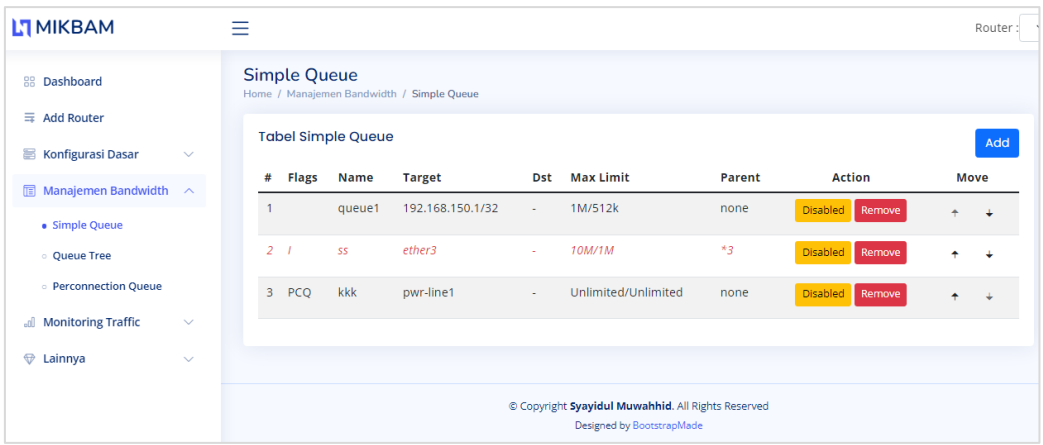


Gambar 7. Halaman *login* 2

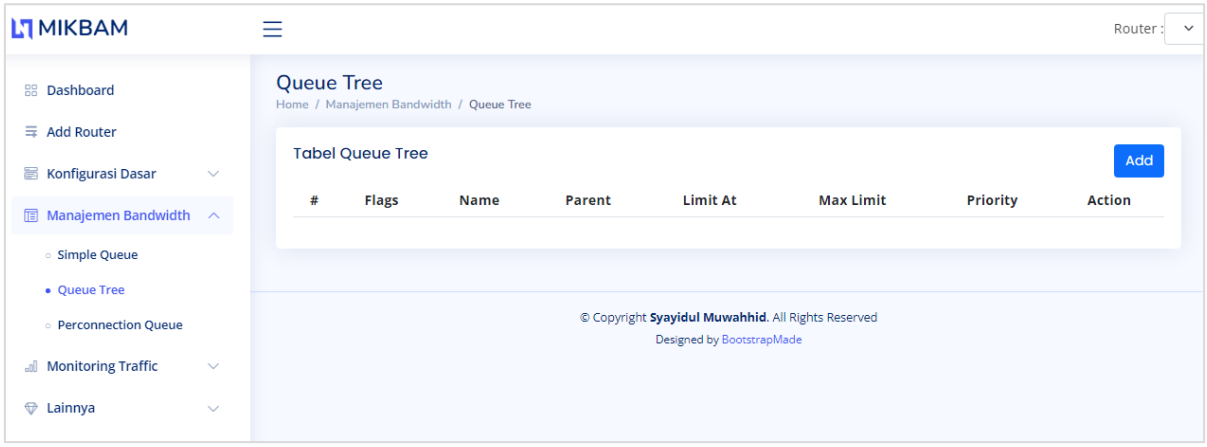
Gambar 6 merupakan halaman *login user* diharuskan menginputkan alamat IP dari sebuah *router*. Lalu sistem akan mencoba mengetes koneksi jika terhubung maka akan diminta mengisikan *username*, *password*, dan *port* yang digunakan sesuai Gambar 7. Gambar 8 merupakan tampilan *dashboard* dari aplikasi yang dibuat. Pada halaman ini memuat semua informasi sumber daya dari *router*, mulai dari waktu, status memori, CPU, *interface* serta *traffic* internet pada *interface.* Gambar 9 menampilkan halaman *simple queue*, dimana *user* dapat membuat *simple queue* yang digunakan untuk limitasi sebuah *traffic* menggunakan *interface* dan alamat IP. Pada halaman ini, *user* dapat melakukan limitasi dengan metode *simple queue*, hanya perlu memasukan target dan besar limitasinya saja. Pada halaman ini juga, user dapat memodifikasi setiap *queue*, seperti menonaktifkan, menghapus, dan merubah susunannya. Gambar 10 menampilkan halaman konfigurasi *queue tree*, pada halaman ini *user* dapat melakukan limitasi yang lebih kompleks, menggunakan *mangle*, user dapat memilih tipe limitasinya berdasarkan alamat IP dan *service* yang disediakan. Gambar 11 menampilkan konfigurasi *perconnection queue*. pada halaman ini *user* dapat melakukan limitasi dengan *simple queue* menggunakan tipe *perconnection queue* (PCQ).



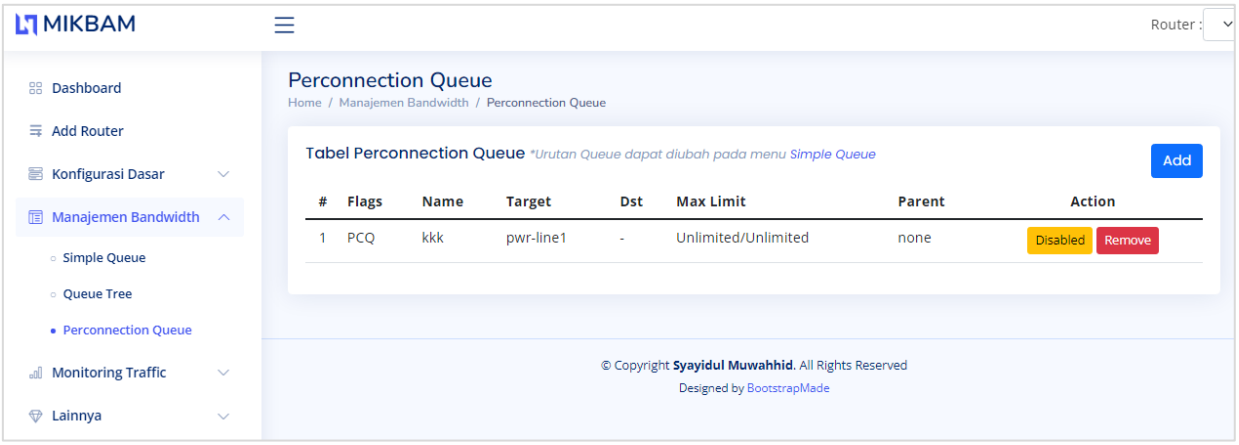
Gambar 8. Halaman *dashboard*



Gambar 9. Halaman konfigurasi *simple queue*



Gambar 10. Halaman konfigurasi *queue tree*



Gambar 11. Halaman konfigurasi *perconnection queue*

Tabel 1. Hasil *black box testing*

No	Pengujian	Skenario	Hasil yang diharapkan	Hasil
1	Melakukan Login Tahap I	Pengguna mengisikan alamat IP dengan benar	Pengguna dapat melanjutkan untuk autentikasi tahap selanjutnya	Sesuai
		Pengguna mengisikan alamat IP dengan salah	Pengguna tidak dapat melanjutkan ke tahap berikutnya.	Sesuai
2	Melakukan Login Tahap II	Pengguna memasukan username, password dan port dengan benar	Pengguna berhasil login dan diarahkan ke halaman dasbor	Sesuai
		Pengguna memasukan username, password dan port dengan benar	Pengguna tidak dapat melanjutkan dan harus mengisi kembali form login	Sesuai
3	Melakukan Konfigurasi IP Address	Pengguna menambah alamat IP pada interface yang belum memiliki IP	Data tersimpan	Sesuai
		Pengguna menambah alamat IP pada interface yang sudah memiliki IP	Data Tidak tersimpan dan tampil notifikasi error	Sesuai
		Pengguna memasukan alamat IP yang tidak sesuai penulisan yang tepat	Tampil notifikasi eror.	Sesuai
		Pengguna masukan alamat IP dengan benar tetapi tidak diakhiri dengan penambahan prefix	Tampil notifikasi error	Sesuai
4	Konfigurasi DHCP Server	Pengguna menambahkan harga pada nama barang yang ada	Pengguna dapat menambahkan harga barang sesuai nama barang dan kategorinya	Sesuai
		Pengguna melakukan konfigurasi dhcp server dengan memilih	Berhasil menyimpan data	Sesuai

No	Pengujian	Skenario	Hasil yang diharapkan	Hasil
		interface dan mengisi dns server		
		Pengguna melakukan konfigurasi dhcp server dengan memilih interface tanpa mengisi dns server	Tampil notifikasi eror	Sesuai
		Pengguna melakukan konfigurasi DHCP Server pada interface yang sama	Tampil notifikasi eror	Sesuai
5	Konfigurasi Simple Queue	Pengguna melakukan konfigurasi simple queue dengan target interface	Berhasil konfiguasi	Sesuai
		Pengguna melakukan konfigurasi simple queue dengan target alamat IP	Berhasil konfigurasi	Sesuai
		Pengguna melakukan konfigurasi simple queue tanpa mengisi max limit	Berhasil menyimpan dan max limit otomatis unlimited	Sesuai
		Pengguna melakukan konfigurasi simple queue dengan menambahkan max limit	Berhasil menyimpan	Sesuai
		Pengguna melakukan konfigurasi simple queue tanpa mengisi dst address	Berhasil menyimpan dan dst kosong	Sesuai
6	Konfigurasi Queue Tree	Pengguna melakukan konfigurasi queue tree tanpa memilih tipe limitasi	Muncul pesan error	Sesuai
		Pengguna melakukan konfigurasi dengan tipe limitasinya ip address	Berhasil menyimpan konfigurasi	Sesuai
		Pengguna melakukan konfigurasi dengan tipe limitasinya service	Berhasil menyimpan	Sesuai
7	Konfigurasi Perconnection Queue	Pengguna melakukan konfigurasi PCQ dengan target sebuah interface	Berhasil konfigurasi	Sesuai
		Pengguna melakukan konfigurasi PCQ dengan target alamat IP	Berhasil konfigurasi	Sesuai
8	Konfigurasi Prioritas Traffic	Pengguna menambah konfigurasi prioritas traffic berdasarkan service yang dipilih	Berhasil konfigurasi	Sesuai
9	Konfigurasi Drop Traffic	Pengguna memblokir sebuah traffic dengan memilih nama service	Berhasil konfigurasi	Sesuai
10	Monitoring Traffic	Pengguna memilih tipe queue simple dan memilih queuenya	Tampil grafik dari queue yang dipilih	Sesuai
		Pengguna memilih tipe queue tree dan memilih queuenya	Tampil grafik dari queue yang dipilih	Sesuai
11	Konfigurasi Notifikasi	Pengguna melakukan konfigurasi dengan lengkap	Berhasil Konfigurasi	Sesuai
		Pengguna melakukan konfigurasi tanpa mengisi “message when up”	Berhasil Konfigurasi	Sesuai
		Pengguna melakukan konfigurasi tanpa mengisi “message when down”	Berhasil Konfigurasi	Sesuai
12	Konfigurasi Service	Pengguna menambah sebuah data service.	Berhasil meyimpan data	Sesuai
		Pengguna menghapus sebuah data service	Berhasil menghapus data	Sesuai
13	Menu Change Identity	Pengguna merubah identity	Berhasil Konfigurasi	Sesuai
14	Menu Clock	Pengguna mengklik tombol “automatically” dan memilih nama negara	Sistem melakukan otomatisasi konfigurasi tanggal dan waktu dengan memanfaatkan fitur SNTP Client	Sesuai
15	Menu Akun	Pengguna menambah user baru sesuai dengan grup dan tidak	Berhasil konfigurasi	Sesuai

No	Pengujian	Skenario	Hasil yang diharapkan	Hasil
		mengisi allowed address		
		Pengguna menambah user baru sesuai dengan grup dan mengisi allowed address	Berhasil konfigurasi	Sesuai
		Pengguna menghapus data user	Berhasil konfigurasi	Sesuai
16	Menu Backup	Pengguna melakukan backup tanpa menambahkan enkripsi	Berhasil melakukan backup dan mengunduh file	Sesuai
		Pengguna melakukan backup dengan memilih enkripsi dan memasukan password	Berhasil melakukan backup dan menunduh file	Sesuai
17	Menu Retore	Pengguna mengunggah file backup dan tidak mengisi password	Mengecek apakah file backup dienkripsi atau tidak, jika tidak maka router akan mereset dan mengembalikan konfigurasi sesuai file backup.	Sesuai
		Pengguna mengunggah file backup dan mengisi password	Mengecek apakah password sesuai lalu melakukan restore.	Sesuai
18	Konfigurasi Daftar Router	Pengguna menambah daftar router	Berhasil menyimpan	Sesuai
		Pengguna menghapus daftar router	Berhasil menghapus	Sesuai
19	Konfigurasi Time Zone	Menambah data time zone	Berhasil menyimpan	Sesuai
		Menghapus data time zone	Berhasil menghapus	Sesuai

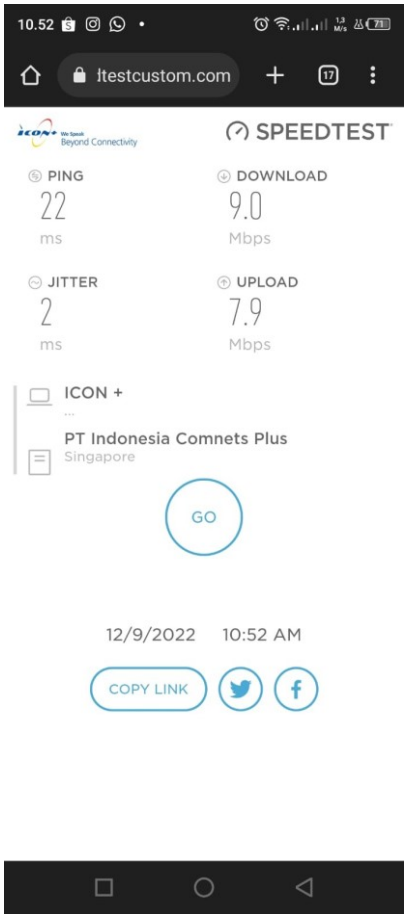
B. Pengujian Sistem

Pengujian yang dilakukan untuk aplikasi ini adalah *black-box* testing yaitu menguji cara kerja aplikasi. Pengujian dilakukan oleh pembuat aplikasi. Hasil pengujian secara lengkap dijelaskan pada Tabel 1. Berdasarkan hasil pengujian, dapat dilihat bahwa 19 jenis pengujian mendapatkan hasil yang sesuai dengan yang diharapkan. Aplikasi yang dibuat telah bekerja dengan berhasil.

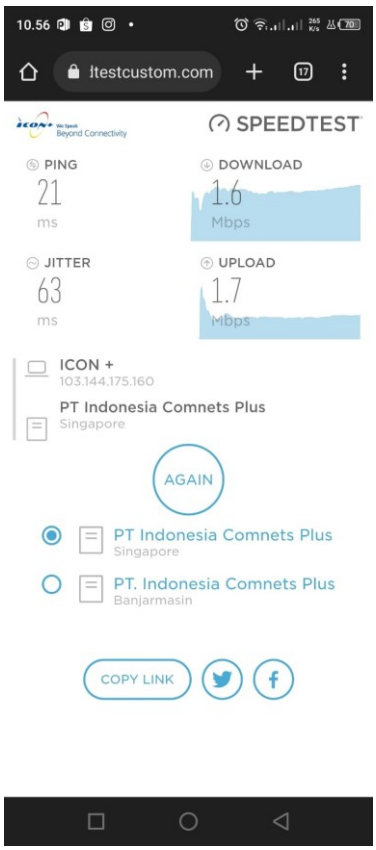
C. Pengujian QoS Simple Queue

Dalam pengembangan aplikasi ini ada tiga QoS yang di terapkan yaitu *simple queue*, *queue tree*, dan *perconnection queue*. Pengujian *simple queue*, dilakukan pada target *client* yang akan di batasi kecepatan internetnya. Gambar 12 menunjukan kecepatan internet sebelum di batasi. Dengan menu *simple queue* yang ada pada aplikasi yang dikembangkan, dilakukan pembatasan kecepatan *client* yang ditargetkan menjadi *upload* 2 Mbps dan *download* 2 Mbps. Gambar 13 menunjukan kecepatan internet di *client* telah sesuai dengan yang konfigurasi pembatasan *download* dan *upload*.

Berdasarkan pengujian dapat dilihat kecepatan internet *user* yang dibatasi di *simple queue* telah sesuai. Hasil pengujian menunjukan aplikasi yang dibangun telah dapat menerapkan perintah konfigurasi pembatasan kecepatan *upload* dan *download*.



Gambar 12. Kecepatan internet sebelum dibatasi



Gambar 13. Monitoring kecepatan setelah dibatasi

IV. KESIMPULAN

Berdasarkan pengujian yang telah dilakukan, aplikasi dapat melakukan otomatisasi konfigurasi bandwidth yang meliputi konfigurasi *simple queue*, *queue tree*, dan *perconnection queue*. Aplikasi juga sudah berhasil melakukan monitoring *traffic* antara lain menggunakan *netwatch* dengan Telegram dan memonitoring grafik lalu lintas data pada *queue* yang dibuat. Selain itu juga aplikasi sudah dilengkapi dengan menu-menu dasar yang biasa digunakan oleh administrator jaringan seperti *setting indentity*, *clock*, *backup*, dan *restore*. Dengan kelengkapan menu tersebut aplikasi akan mempermudah administrator jaringan dalam melakukan manajemen dan konfigurasi jaringan komputer.

UCAPAN TERIMA KASIH

Ucapan terimakasih penulis sampaikan kepada Kementerian Riset Teknologi dan Pendidikan Tinggi melalui hibah Penelitian Dosen Pemula dengan perjanjian kontrak nomor 763b/POLSMI/PN/IX/2022.

REFERENSI

[1] Indonesia Survey Center, "Laporan Survei Internet APJII 2019 - 2020 (Q2)"

[2] S. Dani Ahmad Nur, A. Tri Arsanto "Analisa QOS (Quality Of Service) Menggunakan Simple Queque dan Queque Tree Menggunakan Mikrotik," *J. Explor. IT*, vol. 11, no. 2, pp. 38–44, 2019.

[3] J. R. Dwi, "Rancang Bangun Sistem Monitoring Jaringan Menggunakan Mikrotik Router OS," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 3, no. 4, pp. 391–395, 2019.

[4] R. Taufiqur, E. K Nurnawati, and E. Susanti, "Analisis, Perancangan Dan Implementasi Manajemen Bandwidth Menggunakan Queue Tree Pada Hotspot Mikrotik Di Wisma Muslim," *Jarkom*, vol. 7, no. 1, p. 64, 2019.

[5] W. Bayu, "Manajemen Trafik Menggunakan HTB Untuk Meningkatkan Kualitas Layanan IP Network," *J. Sains Terap.*, vol. 11, no. 1, pp. 88–103, 2021.

[6] R. R. Trimantaraningsih and M. Istiqomatul, "Implementasi Mikrotik Sebagai Manajemen Bandwith," *Tek. Inform. IST AKPRIND Yogyakarta*, pp. 283–295, 2012.

[7] M. F. Asnawi, "Aplikasi Konfigurasi Mikrotik Sebagai Manajemen Bandwidth dan Internet Gateway Berbasis WEB," *Ppkm I*, vol. 42–48, no. 1, pp. 42–48, 2018.

[8] S. Yusran, A. H. Jatmika, and I. W. A. Arimbawa, "Sistem Pendaftaran Hotspot Online Berbasis Web Menggunakan Mikrotik API, PHP, MySql Pada SMK Plus Nurul Hakim Kediri," *J. Teknol. Informasi, Komputer, dan Apl. (JTIKA)*, vol. 1, no. 2, pp. 141–148, 2019.

[9] L. Nora, W. Agustin "Aplikasi Network Traffic Monitoring Menggunakan Simple Network Management Protocol (SNMP) pada Jaringan Virtual Private Network", *Sains dan Teknologi Informasi (SATIN)*, vol. 1, no. 1, Juni 2015.

[10] H. Muhammad and M. Kamisutara, "Sistem Monitoring Trafik Pada Mikrotik Berbasis App Mobile Dengan Notifikasi Telegram," *Netw. Eng. Res. Oper.*, vol. 6, no. 1, p. 1, 2021.

[11] A. Safrian, dkk, "Studi Analisis Model Rapid Application Development dalam Pengembangan Sistem Informasi", *Matrik*, vol. 16, no. 2, Mei 2017.