*AUTO*MASI *WEBSITE BROWSER* UNTUK MELAKUKAN *AUTO*LOGIN KE DALAM *CAPTIVE PORTAL*

I Made Edy Listartha
Fakultas Teknik dan Kejuruan Universitas Pendidikan Ganesha
Jl. Udayana No.11, Banyuasri, Kec. Buleleng, Kabupaten Buleleng, Bali 81116
listartha@undiksha.ac.id

Abstrak

Penggunaan captive portal dalam sekuriti jaringan komputer menuntut pengguna untuk selalu melakukan registasi pada halaman hotspot saat diperlukan. Penelitian ini menanfaatkan website browser lynx berbasis teks sebagai sarana untuk melakukan registrasi pada captive portal yang di automasi dengan script yang berisi username dan password pengguna. Perangkat penelitiannya sendiri menggunakan software simulasi yang menjalankan RouterOS mikrotik dan linux sebagai pengguna. Mikrotik dibangun sebagai captive portal dengan menggunaan setup wizard dengan konfigurasi IP 192.168.200.1/24, menjalankan fungsi DHCP Server pada sebuah interface1 yang terhubung langsung dengan OS Linux pengguna sehingga mendapatkan IP melalui DHCP Client. Halaman captive portal terbentuk pada alamat http://test.mikrotik.local dengan tampilan standar. Perekaman proses login melalui browser lynx dilakukan untuk mendapatkan script proses login dan digunakan untuk proses automasi login. Pengujian dilakukan dengan menjalankan perintah cmd_script berisi rekaman proses login dan menghapus sesi login pada mikrotik setelah berhasil untuk pengujian selanjutnya. Pengujian dilakukan sebanyak 20 kali dengan hasil tanpa adanya kegagalan jika tidak terdapat gangguan oleh pengguna maupun sistem saat proses automasi berjalan.

***Kata Kunci**: Automasi Komputer, Captive Portal, Mikrotik Hotspot, Hotspot Autologin*

Abstract

The use of a captive portal in computer network security requires users to always register on the hotspot page when needed. This study utilizes a text-based lynx browser website as a means of registering on a captive portal which is automated with a script containing the user's username and password. The research tool itself uses simulation software that runs Mikrotik and Linux RouterOS as users. Mikrotik is built as a captive portal by using a setup wizard with IP configuration of 192.168.200.1/24, running the DHCP Server function on an interface1 that is directly connected to the user's Linux OS so that it gets IP via the DHCP Client. A captive portal page is formed at the address http: //test.mikrotik.local with a standard view. Recording of the login process through the lynx browser is done to get a login process script and is used for the login automation process. Testing is done by running the cmd_script command containing the login process record and deleting the login session on the proxy after successful for further testing. The test was carried out 20 times with the results without failure if there was no interference by the user or the system during the automation process..

***Keywords**: Computer Automation, Captive Portal, Mikrotik Hotspot, Hotspot Autologin.*

PENDAHULUAN

Dewasa ini penerapan *captive portal* sangat umum dilihat pada area WIFI yang dapat digunakan untuk pengguna umum maupun khusus dengan jumlah pengguna yang banyak. *Captive portal* merupakan sebuah halaman *website* yang diakses oleh

pengguna untuk dapat terhubung dengan WIFI atau jaringan kabel [1]. Teknik verifikasi ini pada dasarnya merupakan sebuah perangkat *router* atau *gateway* yang melakukan proteksi terhadap jaringan internal menuju jaringan eksternal saat pengguna melakukan pengiriman *traffic* data.

Teknik ini memaksa pengguna untuk melakukan registrasi dalam sebuah tampilan *website*. Registrasi ini sendiri dapat disesuaikan dengan *policy* dari jaringan tersebut, seperti perlunya melakukan klik terhadap sebuah tombol, memasukkan email, memasukkan *username* dan *password* yang telah diberikan oleh pengelola jaringan dan lain sebagainya [2]. Penerapan dalam area pengguna umum, sebuah *captive portal* dirancang untuk melakukan sarana promosi atau informasi melalui layanan koneksi internet melalui WIFI dengan gratis [3]. Namun untuk pengguna khusus yang menjadi bagian dari sebuah organisasi, *captive portal* memberikan kemudahan dalam hal pengaturan pengguna karena setiap *user* atau pengguna dapat diberikan *username* dan *password* masing-masing sehingga lalu lintas data dalam jaringan dapat dimonitor dan dikenali dengan baik [4].

Dalam organisasi, pengaturan pengguna dapat dilakukan dengan mengatur penggunaan *bandwith* internet, banyaknya *username* yang bias digunakan oleh perangkat dan lain sebagainya. *Monitoring* sendiri bisa mengintegrasikan *firewall* untuk membatasi pengguna mengakses situs dan aplikasi tertentu [5]. Penerapan *captive portal* ini dalam pelayanan umum maupun organisasi memiliki perbedaan dalam proses registrasi perangkat, dimana dalam penerapan teknik verifikasi dengan *passkey* pengguna hanya perlu memasukkannya sekali [6] dan selanjutnya perangkat akan mengingat dan menggunakannya secara otomatis menjadi sebuah *profile* dan menggunakannya saat SSID WIFI tersebut masuk ke dalam jangkauan perangkat dan pengguna dapat langsung terhubung dengan jaringan.

Implementasi *captive portal* ini dapat dilakukan dengan beberapa cara seperti HTTP *redirect*, metode ini akan mengarahkan semua *traffic website* ke *captive portal*. Pengguna yang melakukan permintaan HTTP ke URL akan mengharapkan mendapatkan kode status HTTP 200, jika perangkat mendapatkannya maka dianggap koneksi berjalan baik. Di sini kode 200 dimanipulasi untuk memberikan kode HTTP 302 (*redirect*) ke halaman *captive portal* sehingga pengguna harus registrasi [7]. Kemudian ada teknik ICMP *redirect*, protokol ICMP yang diarahkan ke alamat *captive portal* pada layer 3. Serta yang terakhir teknik DNS *redirect*, ketika pengguna mengakses *website*, maka alamat IP dari domain *website* tersebut dialihkan melalui DNS. Dalam jaringan yang mengimplementasikan *captive portal*, *firewall* akan memastikan hanya DNS lokal jaringan yang akan digunakan, sehingga DNS lokal ini akan memberikan domain *captive portal* sebagai pengganti sementara domain *website* yang diminta oleh pengguna.

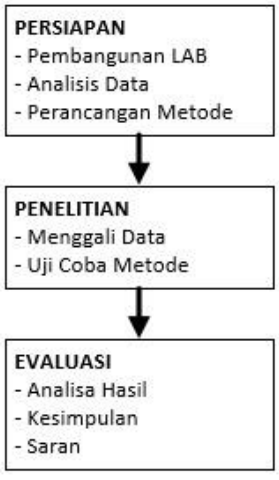
Pengguna captive *port*al perlu melakukan registrasi berulang-ulang melalui halaman *website login* sesuai dengan waktu yang dirancang dalam *policy* bahkan saat *user* tidak melakukan aktivitas apa pun, sehingga pengguna secara otomatis terputus. Masalah perlunya registrasi berulang-ulang ini tentunya menjadi kendala bagi pengguna yang memerlukan koneksi terus menerus, maupun penerapan koneksi ini untuk *availability host* khusus tanpa adanya hak akses pada *administrator* jaringan *captive portal*.

Penelitian ini akan membahas bagaimana meng*auto*masi *website browser* untuk melakukan proses memasukkan *username* dan *password* ke dalam sebuah halaman *captive portal* saat diperlukan. Jenis registrasi ini dipilih karena mewakili jenis registrasi pada *captive portal* lainnya dan tidak menyalahi prosedur proses *login* yang diminta. Eksperimen yang akan dilakukan nantinya akan menggunakan mikrotik *Router*OS x86 yang berjalan pada VirtualBox dan sistem operasi linux ubuntu sebagai pengguna dan terpasangnya browser lynx yang berbasis teks untuk menjalankan *auto*masi nantinya.

METODE PENELITIAN

Penelitian *auto*masi *login* ini menggunakan tiga tahap utama, yaitu Persiapan, Penelitian dan Evaluasi sesuai yang ditampilkan dalam Gambar 1. Runtutan proses mengikuti ilustrasi gambar.

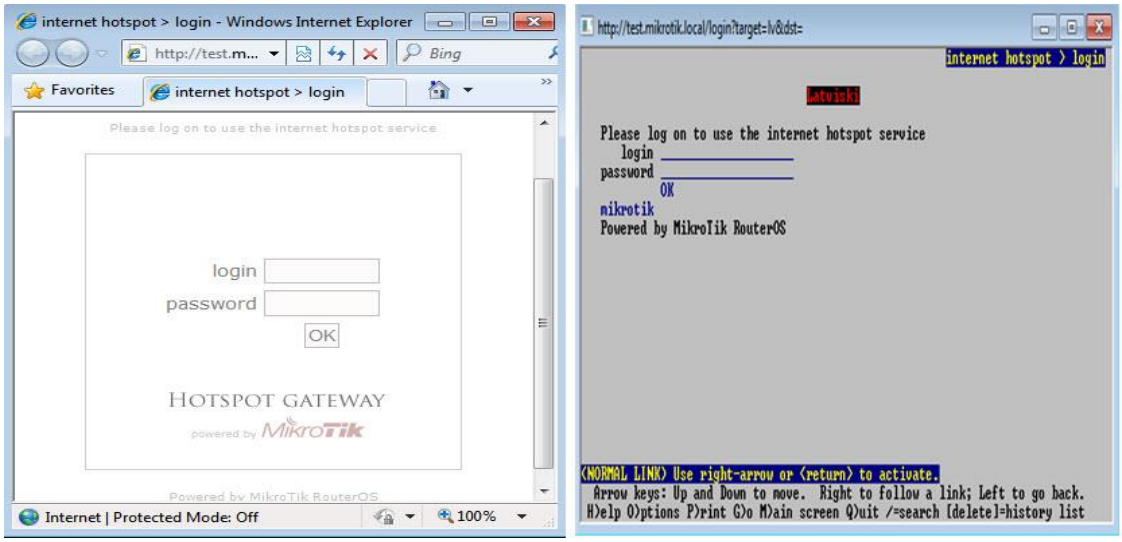
Tahap pertama yaitu persiapan dilakukan dengan pembangunan LAB dalam VirtualBox, yang menggunakan sebuah *router* Mikrotik *Router*OS v.5.20 sebagai perangkat penyedia *captive portal* dan sebuah *end device* komputer dengan sistem operasi linux dimana komputer terhubung langsung ke Mikrotik melalui *port* ether1 dengan IP 192.168.200.1/24 dan komputer menggu-nakan mode DHCP *Client*.



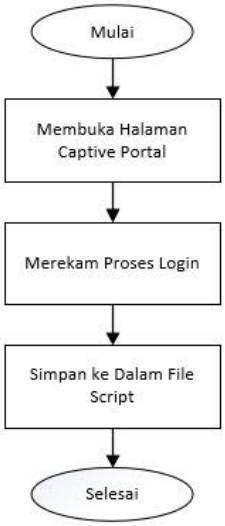
Gambar 1. Tahapan Metode Penelitian

Konfigurasi *captive portal* dilakukan dengan melalui menu IP>*Hotspot*>*Hotspot Setup* dalam Mikrotik sesuai [8]. Penggunaan *setup* ini akan mengaktifkan fitur DHCP *Server* pada *port* ether1 dan diberikan rentangan *pool* DHCP secara otomatis. DNS *Server* menggunakan IP ether1 Mikrotik dan DNS *Name* test.mikrotik.local.

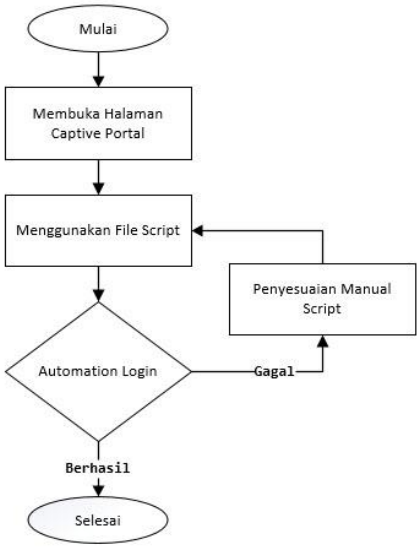
Analisis data dilakukan dengan melakukan *login* ke halaman *captive portal* Mikrotik pada alamat http://test.mikrotik.local menggunakan *website browser* Lynx dan IE dengan hasil perbandingan ditunjukan melalui Gambar 2, dimana fungsi utamanya untuk memasukkan *username* dan *password* tetap ada, namun didalamnya menggunakan cara navigasi yang berbeda. Navigasi dalam mode grafis memerlukan *pointer* untuk digerakkan ke koordinat posisi lokasi *textbox username* dan *password*. Nilai koordinat ini dapat berubah-ubah tergantung dari mode *window*, resolusi layar, posisi *window* pada layer dan *template skin* dari halaman *login*. Berbeda dengan mode teks dari *browser* lynx, navigasi dapat dilakukan dengan menggunakan *keyboard* dan menghilangnya gambar membuat posisi *textbox* menjadi tidak berubah-ubah. Perancangan metode *login* dilakukan dengan memanfaatkan rekaman proses *login* yang dilakukan dan menciptakan sebuah file yang saat dijalankan akan mengulangi proses *login*, sehingga mengurangi waktu proses memasukkan data secara manual. Pola-pola yang dihasilkan, dipelajari untuk mendapatkan model *auto*masi yang tepat. Tahap kedua yaitu Penelitian, dimulai dengan menggali data tampilan *login* dalam mode teks. Proses ini untuk mencari banyaknya kombinasi/runtutan proses penekanan tombol *keyboard* yang diperlukan sesuai ilustrasi Gambar 3, hingga proses *login* berhasil dilakukan.



Gambar 2. *Website Browser* Grafis (Kiri) dan Teks (Kanan)



Gambar 3. Proses Penggalian Data Proses *Login*



Gambar 4. Alur Proses Uji Coba Metode

Merekam proses *login* dengan *browser* lynx dilakukan dengan memberikan parameter -cmd_log dan memberikan alamat *portal login*. Perintah ini membuat *browser* lynx untuk membuat rekaman navigasi yang dilakukan dan *log* atau rekaman itu dapat disimpan dalam sebuah file teks. Keseluruhan runtutan proses ini yang disebut *script file* akan digunakan pada proses selanjutnya. Proses *login* manual pada halaman default sesuai dengan Gambar 2 dalam *browser* lynx adalah melakukan tab/arah turun untuk pindah ke *textbox username*. Memasukkan *username*. Melakukan tab/arah turun untuk pindah ke *textbox password*. Memasukkan *password*. Melakukan tab/arah turun untuk

pindah ke link OK. Menekan enter/arah kanan untuk menyelesaikan proses *login*. Proses uji coba metode diilustrasikan dalam Gambar 4. Dalam proses tersebut *script file* digunakan sebagai parameter *input* untuk *browser* lynx dengan menggunakan parameter cmd_*script*.

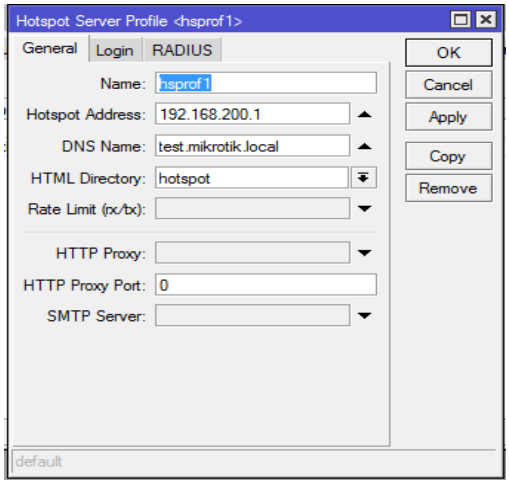
Dalam proses ini, ditambahkan penyesuaian manual *script* untuk mengoreksi runtutan pola navigasi yang dihasilkan untuk mengoptimalkan *auto*masi *login* jika diperlukan. Tahap terakhir yaitu evaluasi dengan menganalisis semua data yang didapatkan dari uji coba metode, semua data ini akan digunakan untuk menentukan keberhasilan dari penelitian dan aspek tambahan yang diperlukan untuk meningkatkan hasil penelitian.

HASIL DAN PEMBAHASAN

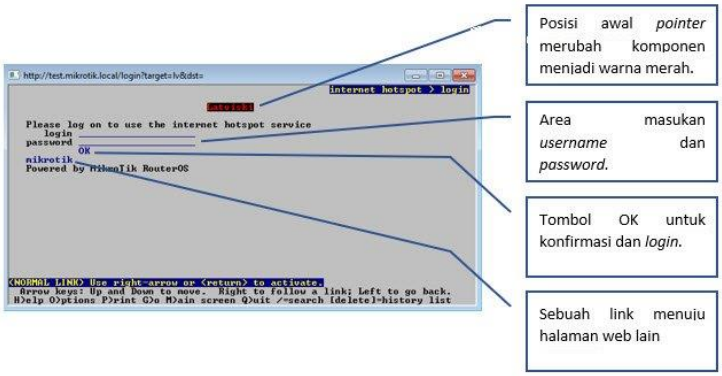
Konfigurasi *router* mikrotik dilakukan melalui CLI dengan menjalankan fungsi *setup* dan memberikan IP awal 192.168.200.1/24 pada *interface*1. Untuk melakukan konfigurasi melalui sistem operasi linux calon pengguna, dilakukan *setup* IP 192.168.200.10/24 pada *interface* yang terhubung langsung pada mikrotik. Konfigurasi perangkat mikrotik dan linux yang sudah dalam satu *network* dilanjutkan dengan konfigurasi *captive portal* dengan mode *wizard* dengan hasil sesuai Gambar 5.

Penelitian ini menggunakan *skin* atau tampilan standar yang disediakan, area analisis data ditunjukkan oleh Gambar 6. Posisi *pointer input* saat pertama kali dibuka berada pada sebuah *link* pada area paling atas halaman *website*.

Pada tahap penggalian data proses *login*, proses *login* memerlukan navigasi dari *user* untuk menekan tombol bawah/tab satu kali lalu memasukkan *username*. Kemudian menekan tombol bawah/tab satu kali untuk masuk ke *textbox*t *password* lalu memasukkan sandinya. Selanjutnya menekan tombol bawah/tab satu kali untuk memilih tombol OK dan menekan tombol *enter* atau tombol kanan.



Gambar 5. Profil *Hotspot*



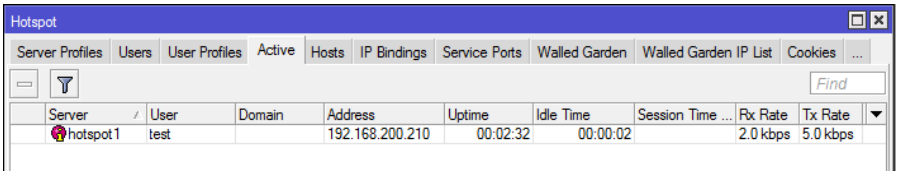
Gambar 6. Analisis Area Penelitian

```
# Command logfile created by Lynx 2.9.0dev.5
# Arg0 = lynx
# Arg1 = -cmd_log=login.txt
# Arg2 = http://test.mikrotik.local/
key <tab>
key t
key e
key s
key t
key <tab>
key t
key e
key s
key t
key <tab>
key ^J
key q
key y
```

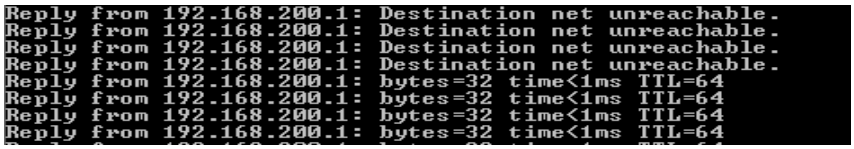
Gambar 7. Hasil Rekaman untuk *Script*

Setelah proses *login* berhasil, selanjutnya melakukan *exit* dari *browser* lynx dengan menekan tombol q dan mengkonfirmasinya dengan tombol y. Perekaman proses ini tidak memperhatikan waktu, karena jeda atau *delay* yang terjadi saat memberikan nilai tidak dihitung. Saat semua proses diatas direkam dengan parameter khusus, *browser* lynx akan menyimpannya dalam sebuah file, didapatkan data seperti pada Gambar 7. Gambar ini menampilkan urutan proses dalam *script* proses *login* dengan *username* test dan *password* test. Proses *login* ini akan sangat sesuai dengan semua tipe *captive portal redirect* lakukan, karena menggunakan *browser* sebagai media registrasi seperti ada proses manual yang biasanya dilakukan. *Website browser* lynx memiliki fitur yang sama seperti pada *website browser* lainnya hanya saja menampilkannya dengan tampilan yang lebih sederhana.

Perekaman data mengharuskan tidak adanya kesalahan dalam proses *login,* karena apa pun yang terjadi akan tercatat dan akan diperlukan penyesuaian kembali namun hanya jika diperlukan. Proses akhir untuk mencoba *script* dilakukan dengan memonitor proses *ping* dan melihat *user* yang aktif pada menu *hotspot* mikrotik sesuai pada Gambar 8 dan proses transisi *ping* terlihat pada Gambar 9.



Gambar 8. Hasil *Login* dengan *Script*



Gambar 9. Transisi *Ping* Saat Verifikasi Berhasil

Terlihat proses *login* berjalan dengan baik dan *username* test dapat *login* ke dalam *captive portal* mikrotik dilihat adanya respons dari *host* yang di *ping* sesuai Gambar 9. Pengujian *autologin* ini dilakukkan sebanyak 20 kali dengan mengahapus sesi *login* yang terdaftar pada tab *active* dan menjalankan *autologin* kembali. Jika tidak menghapus sesi ini, maka pengguna masih akan dianggap terhubung hingga *timeout* terlewati [9]. Sehingga untuk mempercepat proses *timeout* ini lebih mudah dengan cara menghapus sesi yang tercatat pada tab *active* sesuai pada Gambar 8. Proses *autologin* ini dijalankan dengan mengeksekusi perintah "lynx -cmd_script=login.txt" sesuai dengan isi rekaman pada Gambar 7.

KESIMPULAN DAN SARAN

Proses *autologin* yang dilakukan oleh *browser* lynx mengharuskan tidak ada gangguan dalam proses eksekusi *script*. Jika dalam proses ini terganggu oleh adanya koneksi yang terjeda, *pop-pup system*/aplikasi dan *user* sedang aktif menggunakan *keyboard* pada saat bersamaan, maka proses *autologin* akan gagal. Penyempurnaan penelitian ini dapat dilakukan dengan memperbaiki faktor diatas dan menambah kemampuan deteksi pada koneksi apakah *user* sudah *login* atau belum. Hal ini dikarenakan umumnya *Client* dari *captive portal* menggunakan DHCP *Client*, yang membuat *user* akan mendapatkan IP baru dalam waktu tertentu [10] dan membuat *user* untuk *login* kembali sehingga perlu adanya deteksi apakah sesi *login user* sudah habis atau tidak. Dari sisi sekuriti teknik ini sangat rendah karena *username* dan *password* tersimpan secara *plaintext* tanpa *enkripsi* dalam sebuah file, kepekaannya teknik ini dapat disempurnakan jika terdapat sebuah *Application Programming Interface* yang mampu mengontrol secara langsung sebuah *website* browser secara latar belakang sehingga seluruh *password* dan *username* dapat disimpan salam sebuah aplikasi eksekusi/binari.

Pengembangan *auto*masi berbasis *website* ini dapat digunakan untuk melakukan

network automation pada perangakat yang konfigurasinya hanya dapat dilakukan melalui *interface website*, seperti melakukan perubahan *passkey* secara keseluruhan dalam perangkat yang banyak maupun konfigurasi lainnya.

DAFTAR PUSTAKA

[1] H. Yutanto, "Penerapan model promosi berbasis website captive portal hotspot dengan manajemen terpusat," Jurnal Sistem Informasi Bisnis, vol. 8, no. 1, Apr., hal. 49-56, 2018.

[2] C. Nainggolan dan S. D. Putra, "Penggunaan teknologi router mikrotik dalam menunjang jaringan hotspot dan voucher hotspot pada Warnet BNET," Journal of Information System, Informatics and Computing, vol. 2, no. 1, hal. 57-67, 2018.

[3] W. Adhiwibowo dan W. Mindatama, "Implementasi sistem voucher dengan router mikrotik," Jurnal Pengembangan Rekayasa dan Teknologi, vol. 15, no. 2, hal. 118–123, 2019.

[4] I. K. J. Arta dan N. B. S. Nugraha, "Implementasi aplikasi user management hotspot mikrotik PHP dengan Application Programming Interface (API) dan Framework Bootstrap," Jurnal Rekayasa Sistem Komputer (Resistor), vol. 3, no. 1, Apr., hal. 66-71, 2020.

[5] M. D. L. Siahan, M. S. Panjaitan, dan A. P. U. Siahaan, "MikroTik bandwidth management to gain the users prosperity prevalent," International Journal of Engineering Trends and Technology, vol. 42, no. 5, Des., hal. 218-222, 2016.

[6] F. Nugraha, "Analisis keamanan wireless LAN pada jaringan dengan autentikasi captive portal," Jurnal Buffer Informatika, vol. 5, no. 1, Apr., hal. 16-22, 2019.

[7] A. Dabrowski, G. Merzdovnik, N. Kommenda, dan E. Weippl, "Browser history stealing with captive Wi-Fi portals," Dalam Prosiding 2016 IEEE Security and Privacy Workshops, 2016, hal. 234-240.

[8] I. Sofana, Jaringan komputer berbasis MikroTik, Bandung: Informatika, 2017.

[9] E. Wahyudi dan M. M. Efendi, "Wireless penetration testing method to analyze WPA2-PSK system security and captive portal," Jurnal Explore STMIK Mataram, vol. 9, no. 1, hal. 1-7, 2019.

[10] M. Khadilkar, N. Feamster, M. Sanders, dan R. Clark, "Usage-based DHCP lease time optimization," Dalam Proceedings of the ACM SIGCOMM Internet Measurement Conference, 2007, hal. 71-76.