IMPLEMENTASI KEAMANAN JALUR INTERNET MENGGUNAKAN IP TUNNELING PADA OPENVPN *ACCESS SERVER* DENGAN PROTOKOL OPENVPN DAN PROTOKOL DNS OVER HTTPS

Yhudi Winawang
Universitas Mercu Buana, Jakarta, Indonesia
Email: 55417120025@student.mercubuana.ac.id

INFO ARTIKEL
Diterima
5 April 2021
Direvisi
10 April 2021
Disetujui
15 April 2021

Keywords:
internet, security, openvpn, dns, server, https, qos

ABSTRACT

OpenVPN Access server is one of the VPN solutions that is very suitable for corporate and personal because it is easy to configure and can be installed on-premise server. Although users already use the OpenVPN protocol, there are still security gaps on the DNS side if they still use ISP's DNS that applies a lot of filtering. Therefore the OpenVPN protocol can be combined with the DNS Over HTTPS (DOH) protocol so that DNS traffic is also difficult to manipulate or block. The security and stability of access using OpenVPN has a Quality of Service (QoS) that is close to direct access, which has a difference in packets sent compared to packets received around 1.7%. When added OpenVPN protocol and DOH protocol simultaneously then the average difference of loss package will increase to 1,8%. Access to DNS with OpenVPN+DOH has a greater response time compared to direct DNS access to ISPs. However, this DNS response time does not have much effect on access speed because DNS serves to translate domain addresses into IP addresses. The purpose of this research is to analyze the security of internet access by means of a VPN dial using the OpenVPN protocol that succumbs to the DNS Over HTTPS protocol.

ABSTRAK

OpenVPN *access server* merupakan salah satu solusi VPN yang sangat cocok diterapkan untuk perusahaan maupun personal karena mudah dikonfigurasi dan bisa dipasang pada *server* secara *on-premise*. Walaupun pengguna sudah menggunakan protokol OpenVPN namun masih ada celah keamanan pada sisi DNS apabila masih menggunakan DNS milik ISP yang menerapkan banyak *filtering*. Untuk itu protokol OpenVPN dapat digabungkan dengan protokol DNS Over HTTPS (DOH)

How to cite: Winawang, Yhudi (2021) Implementasi Keamanan Jalur Internet Menggunakan Ip Tunneling Pada Openvpn Access Server Dengan Protokol Openvpn Dan Protokol Dns Over Https. *Jurnal Syntax Admiration* 2(4). https://doi.org/10.46799/jsa.v2i4.207
E-ISSN: 2722-5356
Published by: Ridwan Institute

agar *traffic* DNS juga sulit dimanipulasi atau diblokir. Keamanan dan kestabilan akses menggunakan OpenVPN mempunyai Quality of Service (QoS) yang mendekati akses langsung, yang mempunyai nilai selisih paket dikirim dibanding paket diterima sekitar 1,7%. Ketika ditambahkan protokol OpenVPN dan protokol DOH secara bersamaan maka selisih rata-rata paket loss akan naik menjadi 1,8%. Akses ke DNS dengan OpenVPN+DOH mempunyai response time yang lebih besar dibandingkan dengan akses DNS ke ISP secara langsung. Namun *response time* DNS ini tidak terlalu berpengaruh pada kecepatan akses karena DNS berfungsi untuk menerjemahkan alamat domain menjadi IP *address*. Tujuan dari penelitian ini adalah untuk melakukan analisis keamanan akses internet dengan cara melakukan dial VPN menggunakan protocol OpenVPN yang digabungkan dengan protokol DNS Over HTTPS.

Kata Kunci:
Internet; keamanan; openvpn; dns; server; https; qos

Pendahuluan

Saat ini perusahaan memberikan kesempatan kepada karyawan untuk bekerja dari luar kantor dan di rumah. Ketika sebuah perusahaan memungkinkan karyawan mereka untuk mendapatkan akses ke jaringan internal kantor maka sangat penting dipastikan keamanannya (Iqbal & Riadi, 2019). Salah satu upaya yang dilakukan adalah dengan membangun jaringan pribadi pada layanan jaringan publik atau sering disebut dengan Virtual Private Networks (VPN). Sistem VPN yang terintegrasi ke dalam sistem komunikasi mampu mewujudkan keamanan yang sangat tinggi, sehingga dapat menjamin keamanan VPN dengan penggunaan enkripsi dan deskripsi (Zhou & Luo, 2013). Sebuah *tunnel* IP adalah saluran komunikasi jaringan Internet Protocol (IP) di antara dua jaringan. Hal tersebut digunakan untuk mengangkut protokol jaringan lain melalui enkapsulasi dan dengan merangkum protokol jaringannya dalam paket TCP/IP yang dibawa melalui internet (Jahan et al., 2017). Tunnel IP digunakan untuk menghubungkan dua jaringan IP terpisah yang tidak terhubung langsung satu sama lain (Zhou & Luo, 2013).

VPN dapat dibagi menjadi beberapa jenis, masing-masing jenis mempunyai pendekatan tertentu untuk keamanan, serta mempunyai kelebihan dan kelemahan yang tergantung pada kombinasi protokol dan standar yang digunakan. Saat ini solusi VPN yang sering dipakai ada beberapa jenis yaitu GRE, IPSec, PPTP dan TLS (Jahan et al., 2017). VPN menjadi kompleks karena mempunyai berbagai variasi dan faktanya ada berbagai implementasi yang berbeda, sehingga beberapa implementasi VPN memiliki kerentanan keamanan yang belum ditemukan (Hauser et al., 2020). Pada penelitian ini akan digunakan aplikasi penunjang dalam pembuatan VPN yang bersifat *open source*. Aplikasi penunjang tersebut yaitu OpenVPN *Access server*. OpenVPN *Access server* adalah solusi VPN yang dirancang khusus untuk bisnis, dibangun di atas proyek *open*

source. *Access server* hadir dalam satu paket untuk menyederhanakan implementasi solusi *remote-access* VPN (Qu et al., 2012).

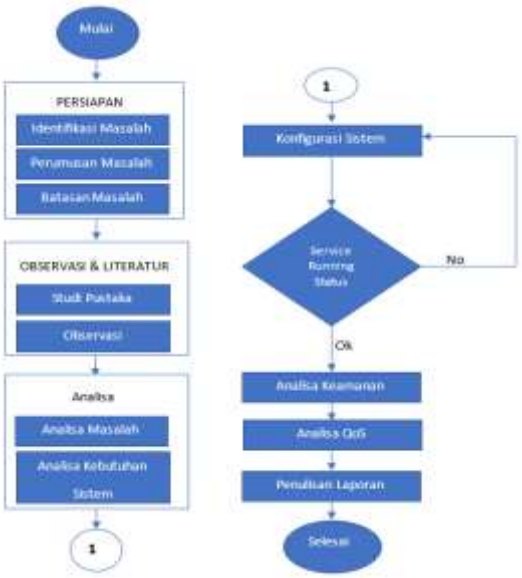
Walaupun sudah menggunakan VPN, namun apabila DNS yang digunakan masih menggunakan DNS dari ISP berarti masih ada celah keamanan dalam *resolving domain*. Sehingga selain VPN masih ada satu hal yang perlu diperhatikan, yaitu mengenai Domain Name System (DNS). DNS adalah protokol lama yang tidak memiliki semua bentuk keamanan (protokol UDP *port* 53) (Zhu et al., 2015). Walaupun begitu DNS adalah salah satu protokol paling mendasar dari Internet. Untuk menambahkan enkripsi ke DNS, ada beberapa metode keamanan, salah satunya adalah DNS over HTTPS (DOH). Dalam karya lain juga telah diusulkan perubahan protokol DNS agar menggunakan koneksi dan enkripsi, yang mendiskusikan masalah transportasi DNS tradisional berbasis UDP yang sebagian besar dapat diatasi dengan menggunakan DNS Over HTTPS sebagai gantinya (Zhu et al., 2015).

Pada penelitian ini akan berfokus pada implementasi OpenVPN Access Server yang menggunakan metode keamanan berbasis TLS untuk membantu meningkatkan keamanan dalam akses internet ketika menggunakan jaringan internet public. Jalur VPN yang sudah terbentuk akan ditambahkan protokol DNS over HTTPS agar *traffic* DNS dari VPN server sulit difilter oleh ISP karena *traffic* DNS akan menggunakan protokol HTTPS, sehingga *traffic* DNS tersebut akan dianggap sebagai traffic browsing.

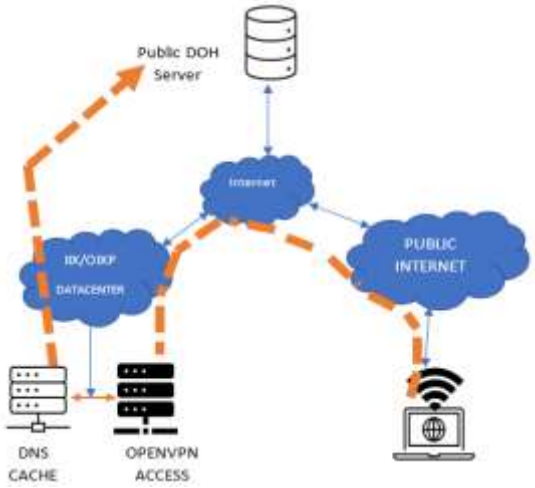
Tujuan dari penelitian ini adalah untuk melakukan analisis keamanan akses internet dengan cara melakukan dial VPN menggunakan *protocol* OpenVPN yang digabungkan dengan protokol DNS Over HTTPS.

Metode Penelitian

Untuk langkah kerja dalam penelitian ini akan menggunakan tahapan seperti Gambar 1



Gambar 1
Kerangka Penelitian



Gambar 2
Topologi Jaringan

Pada penelitian ini secara topologi dapat digambarkan sesuai gambar 2 diatas. Komputer klien terkoneksi ke jaringan internet umum. *Server* VPN dan DNS berada di data senter IIX/OIXP sehingga masih cepat diakses karena masih berada di Indonesia, tepatnya masih di Kota Jakarta. Untuk tujuan DOH menggunakan layanan DNS *public* yang mendukung *protocol* DOH. Dalam penelitian ini menggunakan Next DNS yang mempunyai layanan bersifat gratis.

Untuk DNS *server* menggunakan hardware *routerboard* seri RB450G yang menggunakan sistem operasi Mikrotik RouterOS. Mikrotik pada perangkat keras berbasis Personal Computer (PC) dikenal karena stabil, kontrol kualitas, dan fleksibilitasnya untuk *handling* berbagai jenis paket data dan penanganan proses (*routing*). Untuk *software* yang dipasang disisi *client* menggunakan OpenVPN *Client Connect* versi 3, yang saat ini merupakan versi paling baru. Ini adalah program resmi yang direkomendasikan mendukung OpenVPN *Access*.

Pada tahap konfigurasi yang dilakukan adalah melakukan instalasi OpenVPN *Access Server*, kemudian konfigurasi DNS *cache server* ke layanan NextDNS, dan yang terakhir mengkonfigurasi server OpenVPN agar diarahan menggunakan DNS cache yang telah menggunakan NextDNS tadi. Untuk mengukur *packet loss*, *latency* dan *jitter* ke internet digunakan aplikasi Speedtest-cli. Dalam aplikasi ini terdapat parameter-parameter Ping (*latency*), *packet loss* dan *jitter*.

Untuk mengukur throughput digunakan aplikasi iPerf3, yang merupakan aplikasi untuk pengukuran dan penyetelan kinerja jaringan. iPerf3 adalah alat lintas *platform* yang dapat menghasilkan pengukuran kinerja standar untuk jaringan apapun. iPerf3 memiliki fungsionalitas klien, *server*, dan dapat membuat aliran data untuk mengukur *throughput* antara keduanya berakhir dalam satu atau kedua arah. Untuk melakukan uji *sniffing* digunakan aplikasi Wireshark yang merupakan penganalisis protokol jaringan terkemuka dan banyak digunakan di dunia. Perangkat lunak ini bisa digunakan untuk melihat apa yang terjadi di jaringan kita pada tingkat mikroskopis dan merupakan standar *de facto* dan sering *de jure*.

Untuk melakukan *benchmark* DNS digunaan aplikasi Namebench. Namebench adalah alat pengukuran Domain Name System (DNS) yang bersifat *opensource* dan dimiliki oleh Google. Alat ini dilisensikan di bawah lisensi Apache, versi 2.0. Namebench berjalan pada Windows, OS X, dan Unix, dan tersedia dengan antarmuka pengguna grafis serta antarmuka baris perintah. Tujuannya adalah untuk menemukan server DNS tercepat yang dapat digunakan.

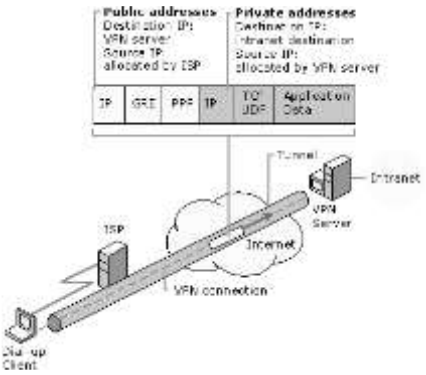
Hasil dan Pembahasan

A. VPN

Virtual Private Network (VPN) bersifat virtual, karena tidak ada koneksi jaringan langsung yang nyata antara dua (atau lebih) mitra komunikasi, tetapi hanya koneksi virtual yang disediakan oleh *software* VPN, dan biasanya melalui koneksi internet atau jaringan publik (Skendzic & Kovacic, 2017). VPN dibedakan menjadi dua jenis, yaitu:

1. *Remote Access* VPN.

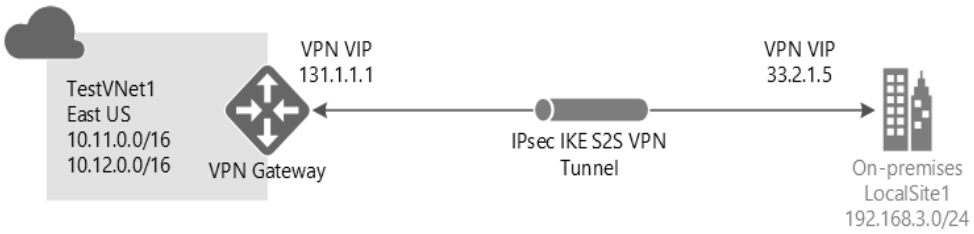
   Remote Access VPN menyediakan akses dengan *remote*, *mobile*, dan komunikasi karyawan dari sebuah organisasi ke jaringan sumber korporasi. Secara khusus, permintaan *remote* akses dibuat oleh pengguna yang selalu berkembang yang ingin mengakses jaringan Local Area Network (LAN) perusahaan. Dengan mengimplementasikan *Remote Access* VPN, pengguna *remote* dan cabang kantor hanya perlu melakukan *setting* koneksi lokal dialup ke ISP dan mengkoneksikan ke jaringan perusahaan melalui internet (Aung & Thein, 2020).



Gambar 3
Jaringan *Remote Access* VPN

2. *Site-to-Site VPN.*

   Tipe VPN ini membuat jalur aman dan tetap antar *site* dengan *site*, misalnya antar kantor pusat dan kantor-kantor cabang lewat internet. masing-masing *site* mempunyai *server* VPN untuk membuat jalur VPN yang dibutuhkan. Setelah jalur VPN terbentuk antara kantor cabang dengan kantor pusat, maka pemakai komputer yang berada pada LAN di kantor cabang dapat akses data yang berada pada LAN di kantor pusat. hanya tentu kecepatan akses terbatas dengan bandwidth jalur VPN yang digunakan (Aung & Thein, 2020).



Gambar 4
Jaringan *Site to Site* VPN

3. OpenVPN

Open Virtual Private Network (OpenVPN) (Daniel et al., 2018). Sebuah VPN memanfaatkan jaringan publik untuk menghubungkan beberapa lokasi jarak jauh. VPN memperluas jaringan *private* menggunakan jaringan publik, seperti Internet dengan membuat koneksi *point-to-point* dan protokol terowongan virtual. Ini memungkinkan komputer untuk berkomunikasi di seluruh jaringan publik seolah-olah dicolokkan langsung ke jaringan pribadi. *Tunneling* dapat menyembunyikan sifat lalu lintas yang dijalankan melalui *tunnel* yang sudah dibuat, menggunakan standar enkripsi untuk mengemas ulang data lalu lintas ke dalam bentuk yang berbeda. Protokol *tunneling* bekerja dengan menggunakan bagian data dari paket (muatan) untuk membawa paket yang menyediakan layanan, memanfaatkan model protokol berlapis seperti model Open Systems Interconnection (OSI) atau rangkaian protokol TCP/IP. *Tunneling* digunakan di semua VPN. Salah satu solusi lapisan aplikasi *open source* umum yang tersedia adalah OpenVPN (Meng, 2013).

Aplikasi OpenVPN harus terpasang disisi klien dan *server*, dan harus terkonfigurasi agar koneksi dapat terbuat. Apabila hal ini telah dilakukan maka dua sisi (*server* dan klien) akan dapat terhubung melalui jaringan virtual. Setiap data yang dilewatkan pada OpenVPN dienkripsi terlebih dahulu dan didekripsi sesudah transmisi. Enkripsi menjamin keamanan data seperti sebuah terowongan kereta api di gunung yang menjaga agar kereta api aman melewati gunung tersebut. Terowongan inilah yang lebih dikenal dengan nama *tunnel*. Sebuah koneksi OpenVPN biasanya dibuat diantara dua buah akses internet dengan *firewall* dan aplikasi OpenVPN (Meng, 2013).

4. TLS *Security*

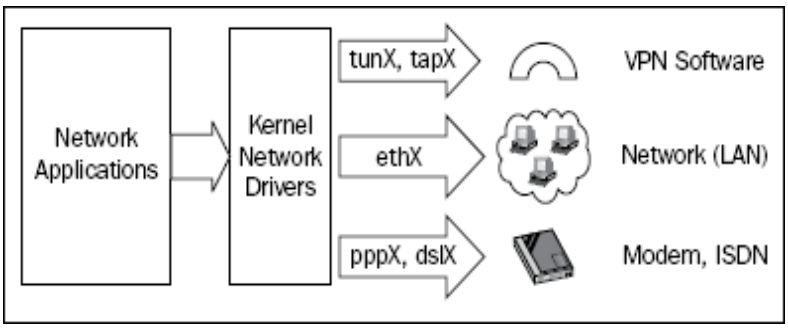
Library SSL/TLS dapat digunakan untuk melakukan autentikasi dan enkripsi. *Library* ini adalah bagian dari OpenSSL yang terpasang pada hampir semua sistem operasi modern. Sesi TLS dengan autentikasi dua arah dinegosiasikan antara klien dan *server* (yaitu keduanya harus menunjukkan sertifikat mereka sendiri) dan digunakan untuk membuat *session key* dengan aman. Ada dua metode pembentukan kunci sesi: dalam metode kunci 1, setiap rekan menghasilkan sandi dan kunci HMAC mereka sendiri dan mengirimkannya ke yang lain. Sementara dalam *key method* 2, kunci dihitung dengan mencampur

bahan acak dari kedua belah pihak menggunakan fungsi *pseudorandom* TLS (PRF). Setelah kedua rekan menerima kunci sesi, terowongan data dapat dimulai dengan data aktual (paket IP atau bingkai Ethernet) untuk mengirimkan dienkripsi, MAC-ed dan dienkapsulasi dalam pesan data (de la Cruz et al., 2020).

5. OpenVPN *Access Server*

OpenVPN *Access Server* adalah sebuah solusi *sofware* yang mendukung penuh fitur SSL yang mengintegrasikan kemampuan server OpenVPN, kemampuan managemen perusahaan dan paket *software* OpenVPN *client* yang mengakomodasi Windows, MAC, dan OS Linux. OpenVPN *Access Server* mendukung berbagai konfigurasi, termasuk akses *remote* yang aman ke jaringan internal dan atau sumber daya jaringan pribadi serta aplikasi dengan kontrol akses. OpenVPN *Access Server* mempunyai kelebihan dalam penggunaanya karena menggunakan antarmuka sistem berbasis *web*, karena itu OpenVPN *Access Server* relatif mudah di konfigurasi dan digunakan juga disisi klien jika dengan OpenVPN *Access Server* ini klien tidak perlu repot menyalin *file-key* dan *certificate* karena *client* hanya cukup menggunakan Browser memasukan alamat VPN *server* kemudan *login*, setelah login klien hanya perlu men-*download file* berbentuk *exe* yang di dalamnya sudah disertakan *file-key* dan *certificate* kemudain menjalankan *file exe* tersebut untuk menginstal dan mengkonfigurasi OpenVPN *client* secara otomatis (Skendzic & Kovacic, 2017).

Driver Universal TUN/TAP dikembangkan untuk dapat menyediakan dukungan pada Linux kernel untuk keperluan proses *tunneling*. *Driver* ini merupakan sebuah *virtual network interface* yang muncul sebagai otentik untuk semua aplikasi dan pengguna yang mencirikannya dari peralatan lainnya adalah dari penamaannya dengan tunX atau tapX. Setiap aplikasi yang memungkinkan penggunaan *network interface* dapat menggunakan *tunnel* ini. *Driver* ini merupakan salah satu faktor utama yang membuat OpenVPN mudah untuk dimengerti, mudah untuk dikonfigurasi dan tidak lupa keamanannya (Christov, 2020). Gambar 5 menunjukan *interface* sederhana yang digunakan oleh OpenVPN.



Gambar 5
OpenVPN *Standard Interface*

Sebuah TUN dapat digunakan seperti sebuah virtual interface untuk melakukan koneksi *point-to-point*, seperti sebuah modem atau DSL link. Ini disebut dengan *mode-routed*, karena *rute* antara pasangan VPN telah dikonfigurasi sebelumnya. Sebuah TAP dapat digunakan seperti sebuah virtual ethernet *adapter*. Hal ini memungkinkan daemon membaca *interface* untuk menangkap ethernet *frames* yang tidak mungkin dilakukan oleh TUN. Mode ini disebut dengan *bridging mode* karena jaringan-jaringan yang terhubung seolah-olah berada dalam satu *hardware* yang sama. Aplikasi-aplikasi dapat dibaca/ditulis pada *interface* ini, perangkat lunak (*tunnel driver*) akan mengambil semua data dan menggunakan *cryptographic libraries* dari SSL/TLS untuk mengenkripsi mereka. Data tersebut dibungkus dan dikirim kepada ujung lain dari *tunnel*. Pengemasan ini terselesaikan atas standarisasi UDP atau TCP (opsional). UDP merupakan pilihan pertama, tetapi TCP dapat sangat membantu dalam beberapa hal. Pemilihan protocol ini diserahkan kepada penggunanya (Cheng et al., 2007).

B. DNS

Domain Name System (DNS) secara elemental berfungsi untuk menerjemahkan teks yang dapat dibaca manusia ke alamat IP yang dapat dibaca oleh komputer ketika pengguna mengakses situs *web*. DNS *query* dibentuk dari maksimal 5 tipe data. Dua tipe data ini selalu ada dalam pesan DNS *Header*, yang berisi informasi tentang DNS *query*, dan pesan ke *server* DNS. Untuk menentukan alamat IP berdasarkan input dari manusia, Browser mengirim kueri ke DNS *resolver*, yang kemudian akan menemukan alamat IP yang menjawab pertanyaan dalam kueri.

1. HTTPS

HTTP over TLS atau Hypertext Transfer Protocol Secure (HTTPS) adalah protokol yang paling sering digunakan oleh pengguna internet, HTTPS adalah protokol yang mengamankan lalu lintas HTTP untuk mencegah pihak ketiga menyadap atau mengubah konten yang dikembalikan sebagai HTTP *response*. Sebelum HTTPS, ISP dapat menambahkan iklan mereka sendiri ke situs apa pun dengan mengubah data respons HTTP *response*. Mereka juga dapat melihat semua data situs *web* yang di kueri, dapat melihat dengan tepat apa yang klien mereka lakukan di Internet. HTTPS membuat praktik tersebut tidak mungkin dilakukan dengan menggunakan TLS untuk mengenkripsi konten HTTP yang dikirim antara pengguna dan situs *web*. Untuk melakukan ini, klien dan *server* terlebih dahulu harus melakukan handshake TLS seperti yang dijelaskan dalam RFC2818 (Durumeric et al., 2017).

2. DNS *over* HTTPS

Sebelum 2018, sebagian besar kueri DNS dilakukan dengan mengirim pesan *plain text* melalui protokol UDP atau TCP pada *port* 53, mengikuti pedoman RFC1035 pada tahun 1987. Tanpa enkripsi, Penyedia Layanan Internet (ISP) dapat mencatat kueri DNS ini dan pendengar lain di jaringan dapat menguping kueri yang dilakukan oleh pengguna internet. Selain hanya menguping, respons

DNS juga dapat dimanipulasi oleh pihak ketiga untuk mengembalikan jawaban yang salah atas pertanyaan, mungkin menyebabkan pengguna ke situs web berbahaya, seperti yang disebutkan dalam RFC 7626 (Borgolte et al., 2019).

3. *Quality of Service*

Karena pada penelitian ini adalah dibidang jaringan, sehingga akan digunakan parameter dari *Quality of Services* (QoS) (Wulandari, 2016). QOS merupakan suatu parameter ukur pada saat melakukan pengujian *bandwith*, tekniknya adalah untuk mengelola *delay time*, *packet loss* dan *jitter*. Tujuan dari mekanisme QoS adalah mempengaruhi setidaknya satu diantara empat parameter dasar QoS yang telah ditentukan. QoS sendiri didesain menyesuaikan kebutuhan para *end-customer* yang ingin memastikan bahwa aplikasi yang berbasis jaringan mendapatkan perfomansi yang handal dan maksimal pada saat pengguna menggunakannya. QoS mengacu pada kemampuan jaringan untuk menyediakan layanan yang lebih baik pada *traffic* 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 (Silitonga, 2014)

4. *Network Security*

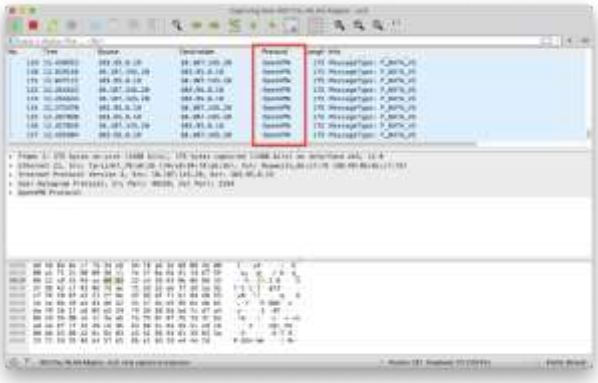
Selain parameter QoS, yang terakhir dalam penelitian ini akan mengukur dari segi parameter keamanan jaringan komputer. Keamanan jaringan komputer diartikan sebagai perlindungan sumber daya terhadap upaya perubahan dan perusakan yang disebabkan oleh seseorang yang tidak diperbolehkan, terdapat dua hal yang berkaitan dengan keamanan dan kerahasiaan data dalam jaringan komputer yaitu representasi data dan kompresi data, yang kemudian dikaitkan dengan masalah enkripsi.

C. Kondisi Awal

Ketika melakukan akses ke internet menggunakan jaringan publik maka tidak bisa dijamin keamanannya. Karena kita tidak mengetahui proses ditengah-tengah antara perangkat yang kita gunakan dan *server* yang kita tuju. Gambar 6 menunjukkan hasil *sniffing* pada komputer yang melakukan akses internet misalnya *browsing*.

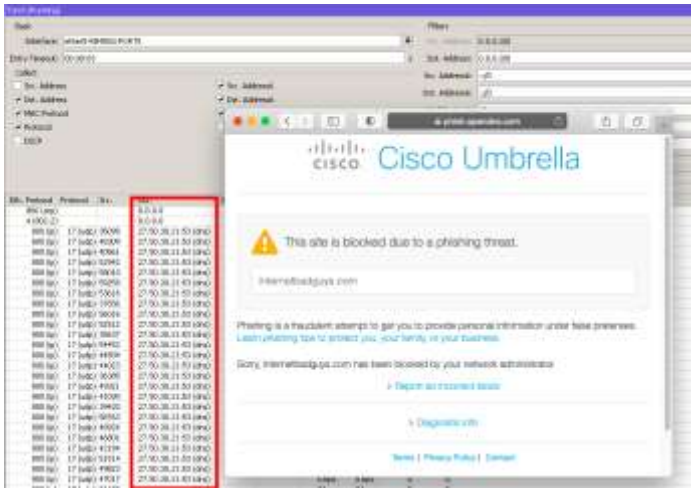
Gambar 6
***Sniffing* Komputer yang Akses Internet Secara Langsung**

Pada Gambar 6 diatas ditunjukkan bahwa lalu lintas di jaringan bisa disadap untuk tujuan alamat *web* yang dituju. Pada gambar tersebut diketahui bahwa komputer sedang menuju ke alamat news.detik.com. Apabila komputer sudah melakuan dial VPN, dalam kasus ini menggunakan protokol OpenVPN maka ketika dilakukan *sniffing* data akan susah disadap karena dengan VPN akan dibuat *tunnel* khusus antara perangkat pengguna dan *server* VPN di internet. Pada Gambar 7 ditunjukkan bahwa komputer yang sudah melakukan VPN maka ketika disadap yang terdeteksi hanyalah protokol VPN-nya. Hal ini terlihat dari protokol yang terdeteksi sebagian besar hanya protokol OpenVPN.



Gambar 7
***Sniffing* Komputer yang Melakukan Dial VPN Menggunakan Protokol OpenVPN**

Namun walaupun komputer sudah melalui terowongan VPN, masih ada celah dalam keamanan ini yaitu disisi VPN yang menggunakan DNS dari ISP. Contoh berikut adalah apabila DNS dari ISP menggunakan *filtering* dari Cisco Umbrella. ISP bisa mengarahkan DNS dari server VPN pengguna ke *rule* yang ada pada DNS Cisco Umbrella.



Gambar 8
Torch Protocol DNS dan Hasil *Filtering* DNS

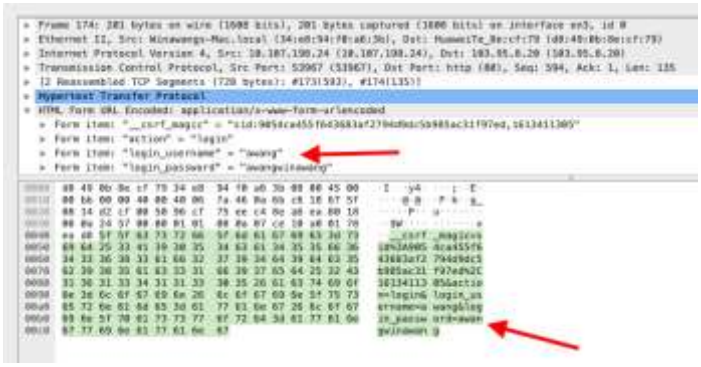
Gambar 8 diatas menunjukkan bahwa *protocol* DNS yang lama, yang masih menggunakan protokol UDP dan *port* 53 masih bisa dengan mudah dimanipulasi. Dalam percobaan ini domain tujuan tidak bisa dibuka karena termasuk dalam *filter* DNS Cisco Umbrella.

D. Kondisi yang Diinginkan

Setelah dipercobaan sebelumnya sudah berhasil menggunakan protokol OpenVPN dan paket data sudah sulit untuk di-s*niffing*. Sehingga pada tahap ini adalah melakukan penambahkan protokol DOH pada sisi VPN server sehingga dari ISP tidak bisa melakukan *filtering* atau modifikasi trafik DNS. Sehingga bisa menambahkan privasi dan keamanan untuk akses internet oleh pengguna yang sudah melakukan t*unnelling* ke server VPN. Berikut hasil pengujian yang dilakukan dengan menggunakan beberapa *tools*. Mikrotik *tool torch* untuk mengamati lalu lintas jaringan secara langsung (*realtime*) dari server VPN dan server DNS. Dan menggunakan Wireshark untuk melihat paket data yang berasal dari komputer *user*.

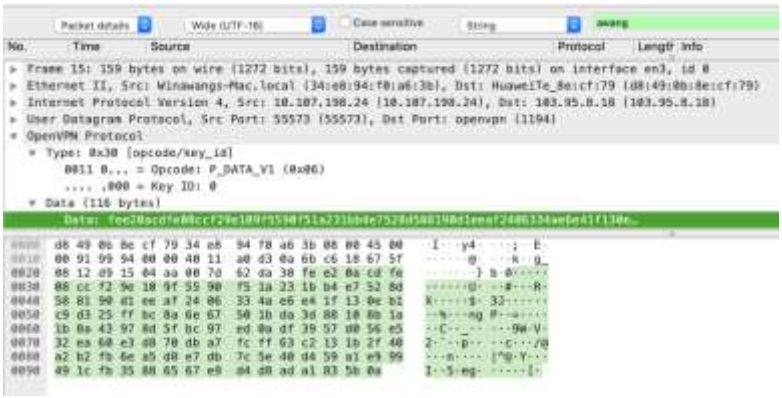
E. Pengujian Keamanan

Sniffing komputer tanpa OpenVPN yang melakukan *login* ke *web-server* dengan protokol HTTP.



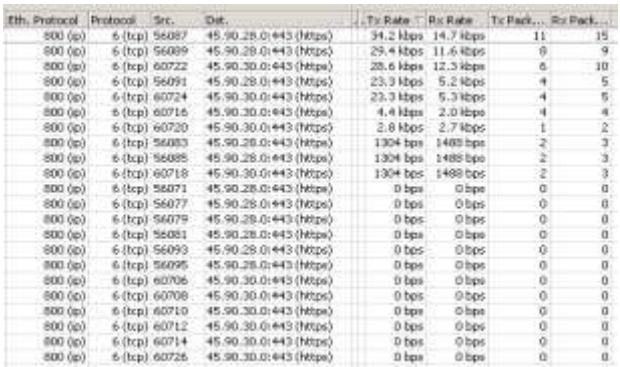
Gambar 9
***Sniffing* Protokol HTTP**

Pada gambar 9 dilakukan pengujian penyadapan isi dari lalu lintas data yang sedang mengirimkan *username* dan *password* ke *web-server* dengan protokol HTTP. *Username* dan *password* yang dikirimkan bisa terbaca karena tidak ada enkripsi.



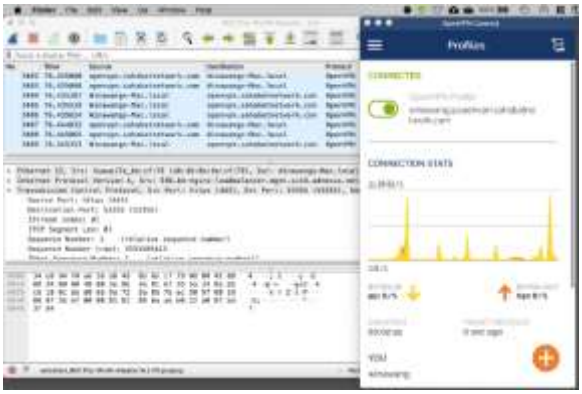
Gambar 10
***Sniffing* Protokol OpenVPN**

Pada gambar 10 menunjukkan hasil *sniffing* pada protokol OpenVPN dan menunjukkan data tidak bisa dilihat dengan mudah karena sudah dienkripsi dengan TLS yang merupakan keamanan bawaan pada aplikasi OpenVPN *Access Server*. Dengan *server* DNS yang menggunakan protokol DOH dan melakukan DOH ke penyedia layanan *public* DNS maka trafik DNS akan menggunakan protokol HTTPS dan apabila di-*torch* akan terlihat seperti trafik *browsing*. Sudah tidak terdeteksi bahwa protokol tersebut adalah protokol DNS.



Gambar 11
***Sniffing* Traffic DNS yang Sudah Menggunakan DOH**

Gambar 11 menunjukkan aksi *torch* dari lalu lintas server VPN untuk melakukan penyadapan protokol DNS. *Traffic* DNS terpantau menggunakan protokol TCP dengan *port* 443. Ketika dilakukan penyadapan lalu lintas jaringan dari komputer, juga hanya akan terdeteksi jalur VPN saja. Data-data yang dibungkus dalam *tunnel* akan terenkripsi dan sangat sulit untuk disadap. Gambar berikut adalah hasil *sniffing* menggunakan Wireshark.

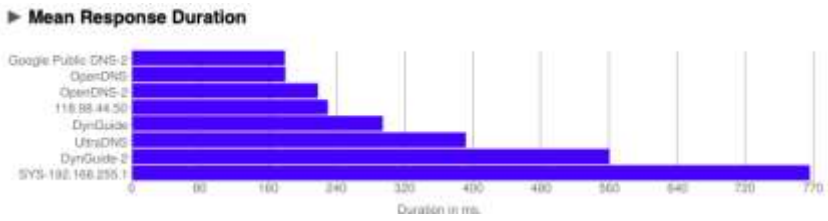


Gambar 12
***Sniffing* Komputer yang Sudah Menggunakan Openvpn+DOH**

Gambar 12 menunjukkan hasil *sniffing* lalu lintas dari komputer pengguna yang sudah melakukan dialing VPN dengan aplikasi OpenVPN client menuju ke OpenVPN Access Server yang terdeteksi hanya *source* dan *destination* yaitu komputer dan alamat *server* VPN.

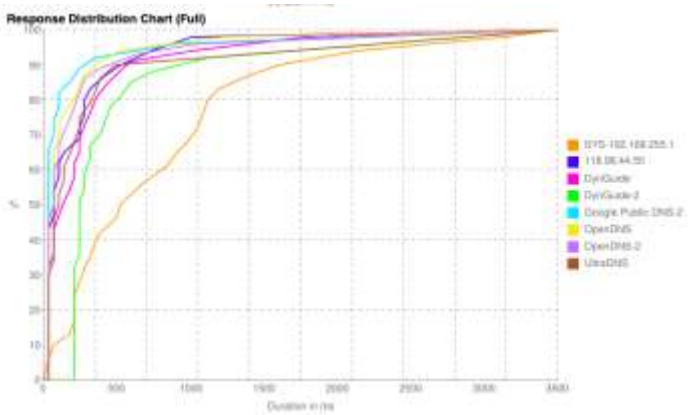
F. Pengujian *Quality of Services*

Setelah melakukan VPN+DOH maka DNS yang digunakan adalah DNS private 192.168.255.1 yang mengarah ke layanan public DOH server yaitu NextDNS. Berikut hasil perbandingan *DNS private* yang menggunakan NextDNS dengan beberapa *public* DNS. Hasil pengukuran durasi respon pada gambar 10 menunjukkan kecepatan DNS Google yang paling cepat, sedangkan DNS *private* yang dibuat memiliki *response time* yang paling besar.



Gambar 13
Mean Response Duration

Walaupun paling lambat namun ketika dilakukan pengujian secara terus menerus dalam waktu lama akan terlihat dalam grafik *name bench* bahwa *response time* DNS *private* yang telah dibuat (192.168.255.1) akan memiliki hasil yang sama dengan *public* DNS yang lain.



Gambar 14
Full Chart Response* dari *Name Bench

Gambar 14 menunjukkan DNS *private cache* 192.168.255. Pada awalnya memiliki *response time* yang paling lama, namun semakin lama memiliki *response time* yang mendekati DNS publik yang lain. Berikut tabel 1 adalah hasil *speedtest-cli* tanpa OpenVPN dan tanpa DOH:

Tabel 1
***Speedtest-Cli* Tanpa OpenVPN dan Tanpa DOH**

OPENVPN dan DOH					
	Packet Loss	***Latency***	***Jitter***	***Throughput***	
	(%)	**(ms)**	**(ms)**	**DL (Mbps)**	**UL (Mbps)**
CBN (12807)	0.3	5,56	1,35	16,3	17,68
BIZNET (797)	0,7	5,31	2,24	17,31	18,13
INDOSAT (13039)	0,4	5,6	1,19	16,93	17,3
MYREPUBLIC (11118)	0,7	5,1	1,68	18,17	19,09
BALIFIBER (12909)	0,3	8,54	1,1	15,92	16,23

Berikut tabel 2 adalah hasil *speedtest-cli* dengan OpenVPN namun tanpa DOH:

Tabel 2
***Speedtest-Cli* dengan OpenVPN dan Tanpa DOH**

+OPENVPN dan -DOH					
	Packet Loss	***Latency***	***Jitter***	***Throughput***	
	(%)	**(ms)**	**(ms)**	**DL (Mbps)**	**UL (Mbps)**
CBN (12807)	0,4	8,16	1,53	15,44	17,07
BIZNET (797)	0,9	9,68	2,27	16,77	18,77
INDOSAT (13039)	0,5	8,75	1,94	15,8	17,57
MYREPUBLIC (11118)	0,9	8,38	1,95	15,65	19,03
BALIFIBER (12909)	0,8	9,52	1,43	11,84	15,11

Berikut tabel 3 adalah hasil *speedtest-cli* dengan OpenVPN dan DOH:

Tabel 3
***Speedtest-Cli* Menggunakan OpenVPN+DOH**

+OPENVPN dan +DOH					
	Packet Loss	***Latency***	***Jitter***	***Throughput***	
	(%)	**(ms)**	**(ms)**	**DL (Mbps)**	**UL (Mbps)**
CBN (12807)	0,4	8,65	1,24	15,54	17,06
BIZNET (797)	0,8	9,96	2,79	15,3	16,97
INDOSAT (13039)	0,6	8,76	1,93	15,49	17,81
MYREPUBLIC (11118)	0,9	9,66	1,87	15,28	18,84
BALIFIBER (12909)	1	9,58	1,71	11,58	15,46

Dari hasil percobaan diatas, dari parameter *packet loss* dan *latency*, dapat dilihat bahwa *server speedtest* yang paling stabil pada waktu pengetesan adalah ke tujuan CBN. Kemudian dapat dilihat juga bahwa akses langsung (tanpa OpenVPN dan tanpa DOH) memiliki *packet loss* dan *latency* yang paling kecil. *Packet loss* dan *latency* akan bertambah ketika ditambahkan protokol OpenVPN dan akan bertambah ketika ditambahkan juga protokol DOH. Untuk parameter jitter (bagian dari parameter *latency* dalam *Speedtest cli*) yang dihasilkan dari pengujian *speedtest-cli* ditunjukkan bahwa angka *jitter* tidak selalu bertambah walaupun sudah ditambahkan *tunneling* menggunakan protokol OpenVPN dan DOH (*jitter* merupakan variasi setiap waktu dari *latency*).

Table 4
Hasil Pengujian Iperf Ke VPN Server Tanpa OpenVPN (Langsung) dan Tanpa Protokol DOH

iPerf OPENVPN dan DOH			
***TEST* ke-**	***SENDER* (Mbps)**	***RECEIVER* (Mbps)**	***Selish Loss* (%)**
1	10,9	10,7	1,8
2	11,1	11	0,9
3	8,55	8,45	1,2
4	11,7	11,6	0,9
5	9,74	9,67	0,7
6	10,3	10,2	1,0
7	9,61	9,47	1,5
8	10,42	9,87	5,3
9	10	9,71	2,9
10	11,2	11,1	0,9
AVERAGE			1,7

Tabel 5 merupakan hasil pengujian iPerf3 dengan melakukan OpenVPN namun masih tanpa protokol DOH.

Tabel 5
Pengujian dengan iPerf3 dengan OpenVPN dan Tanpa DOH

iPerf +OPENVPN dan -DOH			
***TEST* ke-**	***SENDER* (Mbps)**	***RECEIVER* (Mbps)**	**Selisih *Loss* (%)**
1	9,6	9,45	1,5
2	10,7	10,4	2,8
3	9,37	8,99	4,1
4	9,33	8,98	3,8
5	14,4	14,2	1,4
6	12,7	12,6	0,8
7	14,2	14,1	0,7
8	13,7	13,6	0,7
9	14,4	14,3	0,7
10	15,7	15,6	0,6
	AVERAGE		1,7

Tabel 6 merupakan hasil pengujian iPerf3 dengan melakukan OpenVPN dan sudah ditambahkan protokol DOH.

Tabel 6
Pengujian dengan iPerf3 dengan OpenVPN dan dengan DOH

iPerf +OPENVPN dan +DOH			
***TEST* ke-**	***SENDER***	***RECEIVER***	**Selisih *Loss* (%)**
	(Mbps)	**(Mbps)**	
1	9,5	9,3	2,1
2	12,6	12,4	1,6
3	13,7	13,1	4,4
4	15,9	15,7	1,2
5	17,3	17,1	1,1
6	14,7	14,5	1,4
7	14,6	14,4	1,3
8	13,3	12,9	3,0
9	10,6	10,5	0,9
10	18,3	18,1	1,1
	AVERAGE		1,8

Hasil percobaan menggunakan VPN dengan protokol OpenVPN menunjukan hasil yang relatif sama dengan akses langsung, yaitu mempunyai rata-rata selisih *loss* 1,7%. Hal ini menunjukkan bahwa protokol OpenVPN selain aman juga sangat stabil. Ketika OpenVPN ditambahkan juga dengan protokol DOH terdapat peningkatan rata-rata selisih *loss*, namun selisihnya tidak terlalu banyak juga yaitu 1,8 %.

Kesimpulan

Dengan menggunakan protokol OpenVPN dikatakan sudah cukup layak dalam hal peningkatan keamanan data dengan dibuktikan dari hasil pengujian *sniffing* data menggunakan aplikasi Wireshark, dengan mengirimkan *username* dan *password*,

setelah menggunakan *tunneling* dari OpenVPN, data *username* dan *password* tidak terdeteksi dan telah dienkripsi. P rotokol DNS Over HTTPS yang mengubah lalu lintas DNS yang sebelumnya menggunakan protokol UDP port 53 menjadi menggunakan protokol TCP port 443 membuat *traffic* DNS aman karena terenkripsi. Untuk pengukuran QOS dengan menggunakan OpenVPN dan DOH akan terjadi penurunan kualitas jaringan, yaitu di *throughput download. Packet loss* dan *latency* akan terjadi peningkatan juga. Hasil pengukuran *throughput* dengan iPerf3 setelah dilakukan beberapa kali pengujian, didapatkan hasil bahwa paket yang dikirim dan diterima antara *sender* dan *receiver*, akan terjadi selisih yang apabila dijumlahkan dan kemudian dibuat rata-rata, hasilnya adalah 1.7%. Ketika ditambahkan protokol OpenVPN dan protocol.

BIBLIOGRAFI

Aung, S. T., & Thein, T. (2020). Comparative Analysis Of Site-To-Site Layer 2 Virtual Private Networks. *2020 IEEE Conference On Computer Applications (ICCA)*, 1–5. Google Scholar

Borgolte, K., Chattopadhyay, T., Feamster, N., Kshirsagar, M., Holland, J., Hounsel, A., & Schmitt, P. (2019). How Dns Over Https Is Reshaping Privacy, Performance, And Policy In The Internet Ecosystem. *Performance, And Policy In The Internet Ecosystem (July 27, 2019)*. Google Scholar

Cheng, K., Yu, B., & Zhou, J. (2007). The Design And Implement Of Access System Based On OPENVPN. *J. Xiamen Univ*, *45*, S2. Google Scholar

Christov, Y. (2020). Building Personal Virtual Private Networks In Public Cloud Platforms. *Industry 4.0*, *5*(3), 112–113. Google Scholar

Daniel, L.-A., Poll, E., & De Ruiter, J. (2018). Inferring Openvpn State Machines Using Protocol State Fuzzing. *2018 IEEE European Symposium On Security And Privacy Workshops (Euros&PW)*, 11–19. Google Scholar

De La Cruz, J. E. C., Goyzueta, C. A. R., & Cahuana, C. D. (2020). Open Vproxy: Low Cost Squid Proxy Based Teleworking Environment With Openvpn Encrypted Tunnels To Provide Confidentiality, Integrity And Availability. *2020 IEEE Engineering International Research Conference (EIRCON)*, 1–4. Google Scholar

Durumeric, Z., Ma, Z., Springall, D., Barnes, R., Sullivan, N., Bursztein, E., Bailey, M., Halderman, J. A., & Paxson, V. (2017). The Security Impact Of HTTPS Interception. *NDSS*. Google Scholar

Hauser, F., Häberle, M., Schmidt, M., & Menth, M. (2020). P4-Ipsec: Site-To-Site And Host-To-Site VPN With Ipsec In P4-Based SDN. *IEEE Access*, *8*, 139567–139586. Google Scholar

Iqbal, M., & Riadi, I. (2019). Analysis Of Security Virtual Private Network (VPN) Using Openvpn. *Int. J. Cyber Secur. Digit. Forensics*, *8*, 58–65. Google Scholar

Jahan, S., Rahman, M. S., & Saha, S. (2017). Application Specific Tunneling Protocol Selection For Virtual Private Networks. *2017 International Conference On Networking, Systems And Security (Nsyss)*, 39–44. Google Scholar

Meng, D. (2013). Implementation Of A Host-To-Host Vpn Based On Udp Tunnel And Openvpn Tap Interface In Java And Its Performance Analysis. *2013 8th International Conference On Computer Science & Education*, 940–943. Google Scholar

Qu, J., Li, T., & Dang, F. (2012). Performance Evaluation And Analysis Of Openvpn On Android. *2012 Fourth International Conference On Computational And*

Information Sciences, 1088–1091. Google Scholar

Silitonga, P. (2014). Analisis Qos (Quality Of Service) Jaringan Kampus Dengan Menggunakan Microtic Routerboard. *Jurnal Times*, *3*(2), 19–24. Google Scholar

Skendzic, A., & Kovacic, B. (2017). Open Source System Openvpn In A Function Of Virtual Private Network. *IOP Conference Series: Materials Science And Engineering*, *200*(1), 12065. Google Scholar

Wulandari, R. (2016). Analisis Qos (Quality Of Service) Pada Jaringan Internet (Studi Kasus: Upt Loka Uji Teknik Penambangan Jampang Kulon Â€“LIPI). *Jurnal Teknik Informatika Dan Sistem Informasi*, *2*(2). Google Scholar

Zhou, S., & Luo, J. (2013). A Novel Ip Over Udp Tunneling Based Firewall Traversal For Peer-To-Peer Networks. *Proceedings Of 2013 IEEE International Conference On Service Operations And Logistics, And Informatics*, 382–386. Google Scholar

Zhu, L., Hu, Z., Heidemann, J., Wessels, D., Mankin, A., & Somaiya, N. (2015). Connection-Oriented DNS To Improve Privacy And Security. *2015 IEEE Symposium On Security And Privacy*, 171–186. Google Scholar