

IMPLEMENTASI *THREAT MITIGATION* DAN *TRAFFIC POLICY* MENGUNAKAN *UTM* PADA JARINGAN *TCP/IP*

M. Reza Hidayat^{*1}, Ruben Saragih,² Sofyan Basuki,³ Atik Charisma,⁴ dan Antrisha Daneraici Setiawan.⁵

^{1,2,3,4,5}Universitas Jenderal Achmad Yani, Cimahi

Email: ¹mreza.hidayat@unjani.ac.id, ²ruben.saragih@outlook.com, ³sofmae@gmail.com,

⁴atikcharisma@gmail.com, ⁵antrisha.setiawan@gmail.com

^{*}Penulis Korespondensi

(Naskah masuk: 25 Juli 2023, diterima untuk diterbitkan: 26 April 2024)

Abstrak

Penelitian bertujuan merancang *Unified Threat Management (UTM)* berbasis aplikasi *open-source* yang mampu melakukan *Threat Mitigation* dan menerapkan manajemen trafik pada jaringan *TCP/IP*. Metoda *Threat Mitigation* menggunakan *SNORT* sebagai *Intrusion Prevention System (IPS)* untuk melakukan tindakan terhadap ancaman serta melakukan monitoring trafik yang diintegrasikan dengan aplikasi *Splunk* sebagai *Security Information and Event Management (SIEM)*. Metoda *Traffic Policy* menggunakan *SQUID* sebagai *Proxy* untuk melakukan manajemen trafik. Pengujian perfomansi jaringan dilakukan dengan mengukur parameter *Quality of Service (QOS)* terlebih dahulu pada setiap perangkat akses untuk melihat performansi jaringan saat terjadi serangan sebelum dan sesudah implementasi *UTM*. Serangan *Distributed Denial of Service (DDOS)* berupa *Internet Control Message Protocol (ICMP) Flood* dan *SYN Flood*. Setelah melakukan simulasi serangan *DDOS* selama 5 menit, *Threat Mitigation* mampu melakukan *drop* terhadap paket yang berasal dari serangan *DDOS* sebanyak 232409 paket dengan nilai *throughput* maksimum 1,823 Mbps, lebih baik dari *throughput* yang dihasilkan serangan *DDOS* sebelum implementasi *UTM* yaitu 869 Mbps. Hasil indeks parameter *QOS* setiap perangkat akses jaringan memiliki nilai indeks 4, lebih baik dari indeks parameter *QOS* sebelum implementasi *UTM* yaitu 2,843. *Traffic Policy* pada *UTM* mampu melakukan efisiensi *bandwidth* sebesar 4,66% atau 943,6645 MB dari total *volume cache* 20,23 GB, dengan menerapkan *web cache* untuk akses *Hyper Text Transfer Proctocol (HTTP)* dan limitasi *throughput* sebesar 300 KB pada ekstensi *file image*, *audio*, *video* dan *executeable* berukuran diatas 20 MB.

Kata kunci: *DDOS*, *IPS*, *PROXY*, *QOS*, *UTM*

IMPLEMENTATION *THREAT MITIGATION* AND *TRAFFIC POLICY* WITH *UTM* IN *TCP/IP* NETWORK

Abstract

This final project aims to design *Unified Threat Management (UTM)* based on *open-source* application that capable to mitigate threat and implement traffic management on *TCP/IP* network. *Threat Mitigation* method uses *SNORT* as *Intrusion Prevention System (IPS)* and integrated with *Splunk* as *Security Information and Event Management (SIEM)*. *Traffic Policy* method use *SQUID* as *Proxy* to implement traffic management. Network performance testing will be carried out by measuring the *QOS* parameters on each access device to be able to see network performance when an attack occurs before and after *UTM* implementation. The *Distributed Denial of Service* attacks was simulated with *Internet Control Message Protocol (ICMP) Flood* and *SYN Flood*. After simulating *DDOS* attack for 5 minutes, *Threat Mitigation* was able to drop 232409 packet that originating from *DDOS* attack with a maximum *throughput* value 1.823 Mbps, was better before implementation of *UTM* which is 869 Mbps. The result of the *QOS* index parameters for each access device has an index value is 4, was better than before implementation of *UTM*, which is 2.843. *Traffic Policy* was able to perform *bandwidth* efficiency of 4.66% or 943.6645 MB from a total *cache volume* of 20.23 GB, by implementing *web cache* for *Hyper Text Transfer Protocol (HTTP)* access and limiting *throughput* of 300 KB of *image*, *audio*, *video* and *executable file* size above 20 MB.

Keywords: *DDOS*, *IPS*, *PROXY*, *QOS*, *UTM*

1. PENDAHULUAN

Perkembangan teknologi yang sangat pesat menuntut peningkatan kualitas keamanan infrastruktur jaringan. Reliabilitas pada sistem pendukung infrastruktur jaringan komputer dipengaruhi oleh 3 faktor utama yaitu *availability*, *performance* dan *security*. *Availability* dari sebuah layanan sistem harus dipertahankan 24 jam tanpa henti tidak terpengaruh oleh cuaca, jam kerja, pemadaman listrik dan lainnya, sehingga layanan dan sumber daya tersedia tanpa gangguan. Performansi sebuah sistem sangat dipengaruhi dari perangkat yang digunakan di pusat, distribusi maupun akses harus terjaga untuk tidak mengalami *downtime* seperti *crash*, *hang*, kurangnya pengetahuan tentang perangkat pendukung dan teknis pendukung. Kemudian *Security*, yang identik sebagai perangkat mahal dengan kegunaan terbatas yang menjamin ketersediaan dan performansi dari sistem jaringan, tetapi tidak sepenuhnya aman. (W. SUGENG, K. MUSTOFA, 2015) Ancaman yang dihadapi oleh keamanan jaringan sangatlah luas dan dapat dikategorikan sama seperti keilmuan lain, analisis keamanan jaringan harus mempunyai pandangan yang luas untuk melihat hal-hal yang sangat penting. Beberapa menganggap *resources attack* dan *logic attack* sebagai yang paling kritis. *Logic Attack* merupakan serangan yang mengeksploitasi celah *software* dan kekurangannya untuk memberikan akses kepada penyusup ke sistem yang dituju, menurunkan performansi jaringan atau merusak sistem secara menyeluruh. *Resource Attack* merupakan serangan yang ditujukan kepada sumber daya jaringan dengan maksud untuk membanjiri *Central Processing Unit (CPU)*, *memory*, trafik dan sumber daya lainnya dengan permintaan berkali lipat atau ukuran paket data dengan *volume* yang besar sehingga mampu menurunkan performansi dari perangkat dan jaringan. Penurunan kinerja jaringan pada sistem komputer merupakan hal yang paling tidak diinginkan terjadi. *Unified Threat Management (UTM)* adalah suatu sistem aplikasi yang mengintegrasikan berbagai fitur keamanan menjadi suatu *platform hardware* tunggal. *UTM* mampu mendeteksi dan mengurangi ancaman atau gangguan yang mempengaruhi performansi jaringan serta dapat mengatur alokasi trafik berdasarkan aplikasi, protokol atau *interface* pada jaringan. Kelebihan menggunakan *UTM* dibandingkan dengan menggunakan perangkat secara terpisah yang merupakan bagian dari *UTM* seperti *Intrusion Detection System (IDS)*, *Firewall*, *Antivirus* dan *Proxy Server* secara bersamaan ialah dari segi *cost* dan manajemennya yang dimana lebih efisien dan memiliki kompleksitas yang rendah dalam konfigurasinya. *UTM* pada umumnya merupakan produk *enterprise* dengan *cost* yang tinggi, sehingga diperlukan penyesuaian fitur dengan kondisi jaringan, agar *cost* yang dikeluarkan sebanding

dengan fitur yang akan digunakan. *UTM* dapat dirancang dengan mengkolaborasikan aplikasi *opensource* untuk mengurangi *cost* dan dapat lebih mudah dalam menyesuaikan fitur sesuai dengan kondisi jaringan.

Beberapa penelitian perancangan *UTM* dilakukan dengan metoda *spiral* dimana pengembangan aplikasi ini bersifat berkelanjutan. Penelitian pertama yang dilakukan dengan menggunakan *cross-platform firewall* (IPTABLES), *Intrusion Detection System (IDS)* (SNORT), *Proxy Server* (SQUID) dan Proteksi terhadap *e-mail* yang mengandung *virus* dan *spam* (ClamAV), mampu melakukan penyaringan *Spam* 92,50%, *email virus* 98,20% dan melakukan limitasi terhadap konten tertentu. (B. HERU & W. HENTO, 2002) Penelitian kedua dengan menggunakan *Intrusion Prevention System (IPS)* sebagai salah satu fitur *UTM* dalam melakukan *threat mitigation* pada jaringan *Wireless Fidelity (Wifi)* *small office* dan *home office* menggunakan SNORT dan Kismet sebagai *IPS/IDS* mampu menyadap dan memblokir paket dari *Internet Control Message Protocol (ICMP) Flood* sebanyak 8051 paket atau 99% dari total paket yang diterima dan menurunkan serangan pada trafik sebesar 95%. (M. KOR, J. LÁMER, & F. JAKAB, 2011) Penelitian ketiga terkait perbandingan performansi SNORT dan SURICATA sebagai *IPS/IDS* pada fitur *UTM*, SNORT unggul dalam melakukan akurasi deteksi, kecepatan deteksi dan efektivitas deteksi, sedangkan SURICATA lebih hemat dalam penggunaan sumber daya sistem. (E. RISYAD, M. DATA, & E. S. PRAMUKANTORO, 2009) Penelitian keempat terkait pengukuran performansi jaringan berdasarkan parameter QOS menggunakan protokol *ICMP* di jaringan AMIK DCC dengan hasil *throughput* 79%, *packet loss* dibawah 5%, *delay* di bawah 175 ms dan *jitter* dibawah 1%. (A. HAFIZ & D. SUSIANTO, 2019) Penelitian kelima terkait *Proxy Server* sebagai salah satu fitur dari *UTM* dalam menerapkan *traffic policy*, dalam beberapa penelitian merancang *Proxy Server* menggunakan SQUID dengan hasil mampu melakukan efisiensi *bandwidth* sebesar 20% dari total trafik. (S. S.KADAM & Y. C. KULKARNI, 2012)

Berdasarkan penelitian tersebut maka penulis akan merancang sebuah *UTM* berbasis aplikasi *opensource* yang mampu melakukan mitigasi ancaman jaringan (*Threat Mitigation*) dan menerapkan manajemen trafik. Metode *Threat Mitigation* yang dilakukan terdapat pada komponen *UTM* yaitu *IPS/IDS* menggunakan SNORT untuk melakukan *monitoring* trafik dan juga *Proxy* menggunakan SQUID untuk melakukan manajemen trafik dengan metoda *traffic policy*. Dengan kedua metoda tersebut diharapkan mampu menjaga jaringan dari berbagai ancaman yang dapat menurunkan performansi jaringan.

2. METODE PENELITIAN

2.1. Unified Threat Management (UTM)

UTM merupakan solusi pada industri keamanan jaringan yang juga merupakan kemajuan dari *firewall* tradisional yang dimana tidak hanya melindungi dari serangan saja tetapi dapat melakukan *filtering* konten, *filtering spam*, *intrusion detection / prevention*, dan sebagai *antivirus* pada sebuah perangkat. Tujuan UTM ialah menyederhanakan solusi keamanan jaringan secara menyeluruh meskipun meningkatnya cakupan dan kompleksitas dari permasalahan keamanan jaringan. Aspek yang paling terlihat pada hal ini ialah konsolidasi secara fisik dari beberapa produk menjadi sebuah teknologi. (terdefinisi menjadi *Unified Threat Management*).

UTM banyak digunakan dikarenakan kemudahan dalam melakukan instalasi dibandingkan dengan menggunakan berbagai produk / perangkat yang digunakan pada sebuah jaringan. Berikut merupakan gambaran cakupan dari perangkat UTM:

a) Bad Content

UTM menjadi sebuah pelindung untuk setiap *file* atau paket yang akan mengancam jaringan seperti *virus*, *malware*, *trojan*, *spam*, *spyware*.

b) Bad Activity

UTM menjadi sebuah pelindung untuk setiap aktivitas-aktivitas yang mengancam jaringan seperti *Phishing*, *Intrusion*, *DOS & DDOS Attack*.

c) Control Usage

UTM dapat melakukan *content filtering*, *application blocking* dan *management bandwidth* berdasarkan kondisi jaringan.

d) Enforce Policy

UTM dapat menentukan hirarki hak akses berdasarkan *control usage* yang telah ditetapkan serta melakukan pencatatan untuk setiap aktivitas *user*.

Fitur-fitur pada UTM diantaranya :

- *Firewall*
- *VPN*
- *Anti-Virus*
- *Anti-Spam*
- *Anti-Spyware*
- *Anti-Phishing*
- *IPS/IDS*
- *Banwidth Management*
- *Content Filtering*
- *Web Proxy* (J. M. SNYDER. Unified Threat Management Agenda: Unified Threat Management)

2.2. Quality Of Services (QOS)

Berdasarkan ITU-T E.800, *Quality of Service* merupakan mekanisme yang memungkinkan aplikasi atau layanan pada jaringan beroperasi seperti yang diharapkan. QOS dapat didefinisikan sebagai kemampuan untuk menyediakan jaminan

kinerja pada jaringan. Performansi atau kinerja (pada jaringan) merupakan kecepatan dan kehandalan pengiriman berbagai jenis data pada suatu sistem komunikasi. Parameter QOS ditetapkan oleh *Telecommunication and Internet Protocol Harmonization (TIPHON)* dalam mengukur performa jaringan komputer seperti *packet loss*, *delay*, *jitter* dan *throughput* yang dimana memberikan efek yang cukup besar bagi banyak aplikasi atau layanan. Dengan memiliki keterkaitan dari parameter tersebut, QOS dapat memperkirakan dan menyesuaikan kebutuhan perangkat dan aplikasi yang digunakan pada jaringan komputer. Berikut merupakan tabel karakteristik dari parameter QOS: (ALISYA ALIFAH & ANTRISHA DANERAICI SETIAWAN, 2020)

a) Throughput

Throughput merupakan *bandwidth* aktual pada suatu kanal dalam selang waktu tertentu untuk mentransmisikan data.

Tabel 1. Indeks *Throughput*

Kategori	Throughput	indeks
Sangat Bagus	100	4
Bagus	75	3
Sedang	50	2
Buruk	< 25	1

$$\text{Throughput} = \frac{\text{Paket data diterima}}{\text{Lama Pengamatan}} \quad (1)$$

b) Packet Loss

Packet Loss merupakan persentase dari paket yang hilang selama mentransmisikan data pada jaringan.

Tabel 2. Indeks *Packet Loss*

Kategori	Packet Loss	indeks
Sangat Bagus	0	4
Bagus	3	3
Sedang	15	2
Buruk	>25	1

$$\text{Packet Loss} = \frac{(\text{Paket data dikirim} - \text{Paket data diterima})}{\text{Paket data dikirim}} \times 100\% \quad (2)$$

c) Delay

Delay atau *latency* atau *round trip time delay* adalah waktu yang dibutuhkan sebuah paket yang dikirimkan melalui koneksi *end-to-end* pada jaringan

Tabel 3. Indeks *Delay*

Kategori	Delay (ms)	indeks
Sangat Bagus	<150	4
Bagus	150 s/d 300	3
Sedang	300 s/d 450	2
Buruk	> 450	1

$$\text{Delay} = \frac{\sum \text{delay per paket}}{\text{Total Paket}} \quad (3)$$

d) *Jitter*

Jitter atau variasi *delay* merupakan selisih dari setiap nilai *delay* pengiriman paket yang dikirimkan melalui koneksi *end-to-end* pada jaringan.

Tabel 4. Indeks *Jitter*

Kategori	<i>Jitter</i> (ms)	indeks
Sangat Bagus	0-5	4
Bagus	5-75	3
Sedang	75-125	2
Buruk	> 125	1

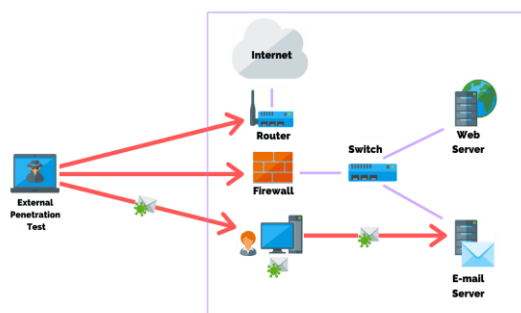
$$Jitter = \frac{\text{Total variasi delay}}{\text{Total Paket yang diterima}} \quad (4)$$

e) *Indeks QOS*

Indeks QOS sebagai indikator kualitas *link* pada jaringan.

Tabel 5. Indeks QOS

Kategori	Percentage (%)	indeks
Sangat Bagus	100	3.8 – 4
Bagus	75	3 – 3.79
Sedang	50	2 – 2.99
Buruk	< 25	1 – 1.99

2.3. *Penetration Testing*Gambar 1. Ilustrasi *penetration testing* [8]

Penetration Testing ialah satu praktis dimana kita melakukan *testing* sebuah sistem komputer, jaringan atau aplikasi *web* untuk menemukan kerentanan yang dapat dimanfaatkan oleh *attacker*. *Penetration Testing* dapat dilakukan secara otomatis dengan menggunakan aplikasi atau *software* atau dijalankan secara *manual*. *Penetration Testing* juga merupakan *testing* yang dilakukan untuk mencari kerentanan atau celah, termasuk potensi akses ilegal yang didapatkan ke dalam sistem dan data, yang dikuatkan atau bentuk dari *risk assessment*. (IMPERVA, “DDoS Attacks,” 2021.)

Definisi dari *National Cyber Security Center* menjelaskan *Penetration Testing* merupakan metoda untuk meningkatkan jaminan keamanan pada sistem jaringan dengan menempatkan beberapa pelanggaran pada keamanan sistem menggunakan *tools* dan teknik yang hampir serupa digunakan oleh penyerang. *Penetration Testing* merupakan simulasi *hacker* menjalankan serangan pada jaringan bisnis, aplikasi jaringan atau *website* bisnis. Tujuannya

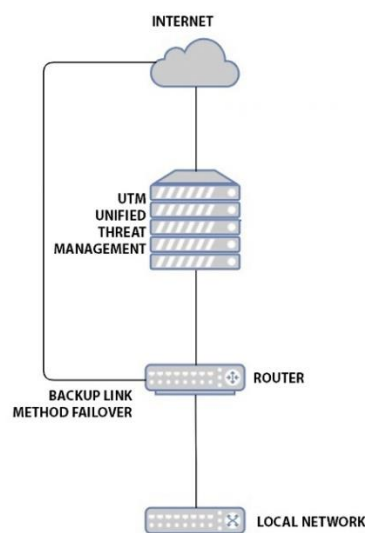
ialah simulasi untuk mengidentifikasi isu keamanan sebelum *hacker* dapat menemukan hal tersebut dan melakukan penyalahgunaan. (P. ENGBRETSON, 2013)

Keuntungan yang di dapatkan dari *Penetration Testing*:

- Membantu mengidentifikasi kerentanan yang belum diketahui sebelumnya.
- Membantu mencari potensi ancaman yang dapat dijalankan oleh *attacker* atau penyusup.
- Membantu untuk menentukan / mengidentifikasi secara *real time* celah pada sistem dan *web* aplikasi.
- Membantu untuk melakukan tes efektivitas dari sebuah aplikasi *firewall* berbasis *web*.
- Membantu melakukan tes kapabilitas pertahanan siber pada organisasi.
- Membantu mengidentifikasi dan menunjukan resiko *real* dan kerentanan resiko.
- Membantu melakukan *testing* seberapa efektif *control safeguard* dan *control counter measure* yang telah ditempatkan.

2.4. Diagram Blok Sistem

Penelitian ini dilakukan dengan mengumpulkan berbagai literatur yang berkaitan dengan topik yang akan di teliti kemudian mempelajarinya. Dalam praktiknya metode penelitian yang dilakukan meliputi pemasangan aplikasi *UTM* dan aplikasi *SIEM*. Selain itu pengukuran nilai QOS juga dilakukan untuk membandingkan kinerja jaringan sebelum dan sesudah pemasangan *UTM*. Data nilai QOS yang diukur antara lain *delay*, *packet loss*, *jitter* dan *throughput*.



Gambar 2. Diagram Blok Sistem

Gambar 2 menunjukan *UTM* akan di pasang pada jaringan publik sebagai *gateway* dari *router* jaringan lokal, dengan tujuan untuk menerima semua trafik masuk ataupun keluar dan dapat melakukan *filter* serta *cache* berdasarkan parameter konfigurasi pada

UTM. Untuk *Backuplink* pada *router* disediakan sebagai mitigasi jalur darurat apabila UTM mengalami *down time*.

2.5. Diagram Alir Penelitian

Pengambilan sampel QOS sebelum penerapan UTM ditujukan untuk mengetahui kondisi jaringan yang akan dianalisa dengan *Penetration Testing* yang dilakukan menggunakan *ICMP Flood* dan *SYN Flood*. Setelah itu perancangan UTM dengan melakukan instalasi sistem operasi, aplikasi UTM dan SIEM. Pengukuran kembali dilakukan saat UTM telah terpasang dan saat *Penetration Testing* dilakukan yang kemudian hasil dari pengambilan sampel akan di analisa untuk menentukan kinerja dari UTM. Gambar 3 menunjukan urutan dari penelitian yang akan dilakukan.



Gambar 3. Diagram Alir Penelitian

2.6. Skema Pengukuran Parameter QOS

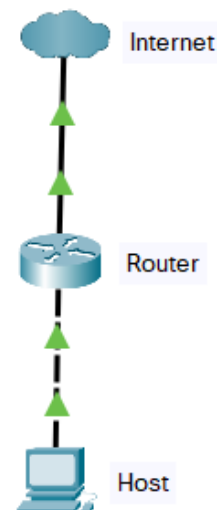
Penelitian Penelitian akan dimulai dari pengukuran parameter QOS pada jaringan akan di lakukan secara 3 tahap. Tahap pertama yaitu pengukuran parameter QOS tanpa *Penetration Testing* dan UTM, tahap kedua yaitu pengukuran parameter QOS pada saat *Penetration Testing* dan tanpa UTM, lalu tahap ketiga yaitu pengukuran parameter QOS pada saat *Penetration Testing* dengan UTM. Penerapan *Penetration Testing* akan dilakukan dengan teknik

yang sama dan target serangan yang sama. Berikut skema *Penetration Testing* yang dilakukan saat pengukuran parameter QOS dilakukan:

- *Penetration Testing* akan dilakukan menggunakan 3 perangkat terdiri dari 2 mesin *virtual* dan 1 *Personal Computer (PC)* yang menggunakan sistem operasi Kali Linux.
- *Penetration Testing* yang akan dilakukan ialah *ICMP Flood* dan *SYN Flood* dengan jenis serangan *DDOS volume based*.
- Tools yang digunakan pada *Penetration Testing* ialah HPING3 pada sistem operasi Kali Linux.
- Pengukuran parameter QOS dilakukan dengan menggunakan perintah *ping* terhadap akses internet untuk mengukur nilai *packet loss*, *delay* dan *jitter* serta aplikasi *wifiman* untuk mengukur nilai *throughput* pada setiap perangkat akses di jaringan.
- Pengukuran parameter QOS akan dilakukan dengan perintah *ping* selama 20 detik atau 20 urutan paket, hal ini disesuaikan dengan durasi dari aplikasi *wifiman* dalam melakukan pengukuran *throughput upload* dan *download* dengan durasi 20 detik. Pengukuran ini akan dilakukan pada setiap titik akses jaringan.
- Pengujian *Penetration Testing* dilakukan pada saat tidak ada trafik pada jaringan dan dilakukan bersamaan dengan pengukuran parameter QOS.

Berikut merupakan skema pengukuran parameter QOS yang dilakukan:

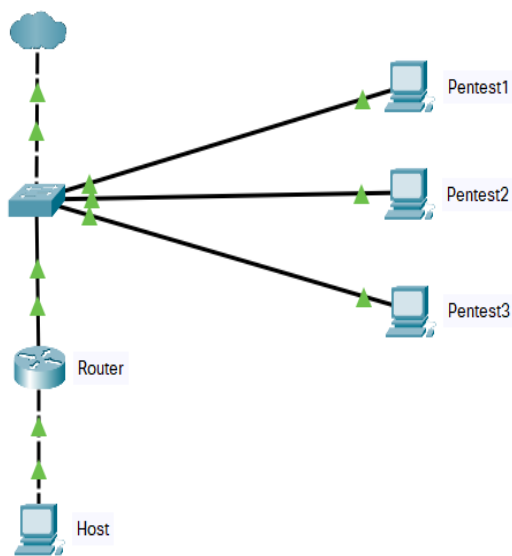
a) Pengukuran parameter QOS pada jaringan kondisi normal



Gambar 4. Skema pengukuran QOS pada jaringan saat kondisi normal

Gambar 4 menunjukan pengukuran parameter QOS akan dilakukan pada sebuah *host* yang telah terpasang aplikasi *wifiman* dan menggunakan perintah *ping* pada *command prompt* atau *terminal*.

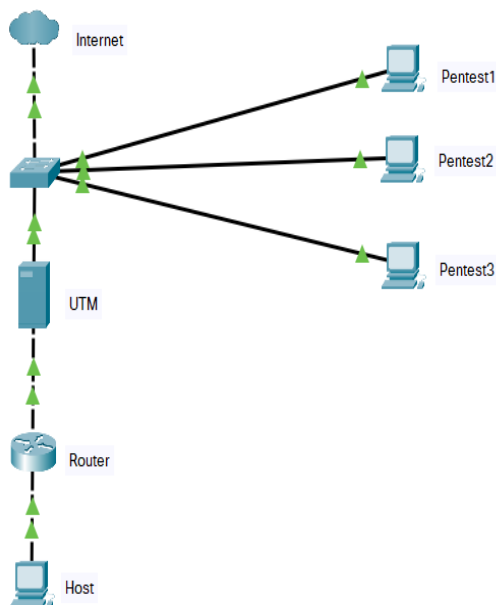
b) Pengukuran parameter QOS pada jaringan pada saat *penetration testing* dan UTM belum terpasang



Gambar 5. Skema pengukuran QOS pada jaringan saat *penetration testing* dan UTM belum terpasang

Gambar 5 menunjukkan skema pengukuran QOS pada saat *penetration testing* dilakukan terhadap router.

c) Pengukuran parameter QOS pada jaringan pada saat *penetration testing* dan UTM telah terpasang



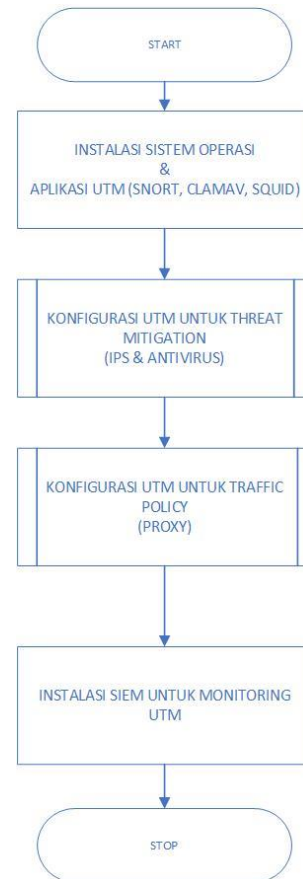
Gambar 6. Skema pengukuran QOS pada saat *penetration testing* & UTM telah terpasang

Gambar 6 menunjukkan skema pengukuran QOS pada saat *penetration testing* menggunakan ICMP Flood dan SYN Flood dilakukan terhadap UTM.

3. PERANCANGAN UTM

Gambar 7 menunjukkan diagram alir urutan instalasi UTM dari sistem operasi sampai aplikasi pendukung UTM beserta konfigurasi yang

kemudian akan diintegrasikan dengan *Security Information and Event Management (SIEM)* sebagai sumber informasi aktifitas trafik pada UTM. Konfigurasi *Intrusion Prevention System (IPS)* menggunakan aplikasi SNORT meliputi konfigurasi *interface*, *ruleset* yang digunakan, sinkronisasi *update ruleset* dan *startup script*.



Gambar 7. Diagram alir perancangan UTM

Konfigurasi *Proxy Server* menggunakan aplikasi SQUID meliputi konfigurasi *Access Control List (ACL)*, konfigurasi *Delay Pools*, *Delay Class* dan *Delay Parameter* untuk manajemen *bandwidth*. Perancangan UTM akan dilakukan dengan ketentuan sebagai berikut:

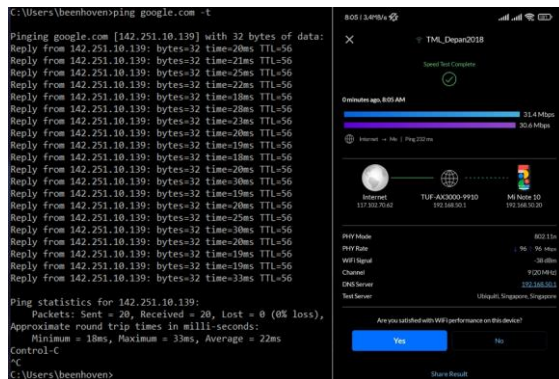
- Sistem Operasi yang digunakan berbasis Linux yang sifat aplikasinya *Open-source* dan memiliki kapabilitas sebagai sebuah server yaitu Ubuntu Server 20.04 LTS.
- Server yang digunakan untuk berfungsi sebagai UTM memiliki spesifikasi berikut:
 - HP proliant ml310e gen8 v2
 - Xeon E3-1220V3 3,1 GHz
 - RAM 8 GB DDR4
 - HDD 1 TB RAID 0
 - Dual Port NIC Gigabyte

4. HASIL DAN PEMBAHASAN

4.1. Pengukuran Parameter QOS

Kondisi pengujian dilakukan pada jaringan yang memiliki *bandwidth dedicated* 1:1 dengan *throughput* maksimum 30 Mbps dan dilakukan pada saat kondisi jaringan tidak ada aktivitas trafik. Berdasarkan poin pada skema pengukuran yaitu setiap parameter QOS akan dapat diperoleh dengan melakukan pengujian *throughput upload* dan *download* menggunakan aplikasi Wifiman untuk mendapatkan parameter *throughput*. Pengujian tes ping dilakukan selama 20 detik menyesuaikan durasi pengujian yang dilakukan oleh aplikasi Wifiman untuk mendapatkan parameter *packet loss*, *delay* dan *jitter*. Hasil dari pengukuran parameter QOS berdasarkan 3 metode pengujian.

1. Pengukuran parameter QOS pada jaringan kondisi normal.



Gambar 8. Hasil pengukuran ada jaringan kondisi normal

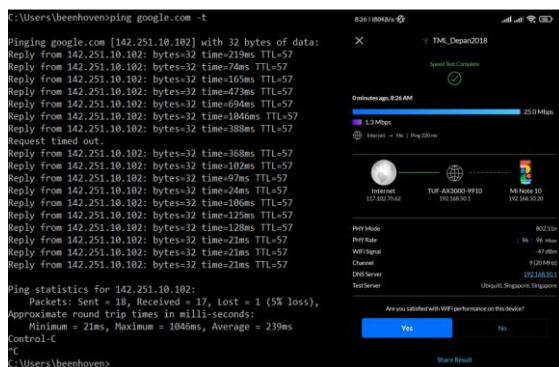
Packet Loss = 0%

Delay = $\sum \text{delay per paket} / \text{total paket}$
= $449 / 20 = 22.45 \text{ ms}$

Jitter = $\text{total variasi delay} / \text{total paket diterima}$
= $85/20 = 4.25 \text{ ms}$

Throughput = Down 31.4 Mbps & Up 30.6 Mbps

2. Pengukuran parameter QOS pada jaringan saat *penetration testing* dan UTM belum terpasang.



Gambar 9. Hasil pengukuran tanpa UTM saat *penetration testing*

Packet Loss = 5%

Delay = $\sum \text{delay per paket} / \text{total paket}$

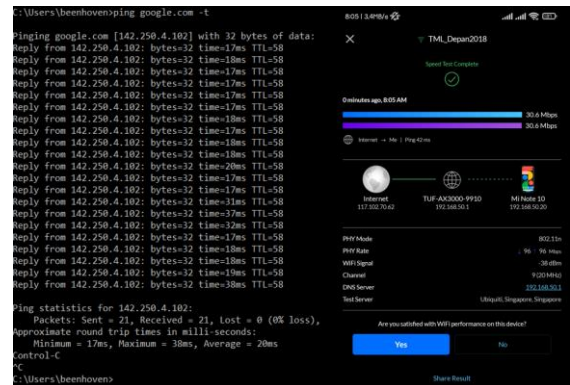
= $6072 / 20 = 303.6 \text{ ms}$

Jitter = $\text{total variasi delay} / \text{total paket diterima}$

= $2915 / 20 = 109.75 \text{ ms}$

Throughput = Down 25.0 Mbps & Up 1.3 Mbps

3. Pengukuran parameter QOS pada jaringan pada saat *penetration testing* dan UTM telah terpasang.



Gambar 10. Hasil pengukuran dengan UTM saat *penetration testing*

Packet Loss = 0%

Delay = $\sum \text{delay per paket} / \text{total paket}$
= $438 / 20 = 21.9 \text{ ms}$

Jitter = $\text{total variasi delay} / \text{total paket diterima}$
= $71/20 = 3.55 \text{ ms}$

Throughput = Down 30.6 Mbps & Up 30.6 Mbps

Hasil pengukuran parameter QOS sebelum dan sesudah konversi menjadi indeks QOS sesuai dengan standar *TIPHON* yang ditunjukkan pada Tabel 6 dan Tabel 7

Tabel 6. Hasil Pengujian

Index QOS	Before UTM		After UTM
Parameter	Non-Pentest	Pentest	Pentest
Packet Loss(%)	0	5	0
Jitter (ms)	4.25	109.75	3.55
Delay / Latency (ms)	22.45	303.6	21.9
Throughput down up (Mbps)	31.4 30.6	25.0 1.3	30.6 30.6

Tabel 7. Hasil Index QOS

Index QOS	Before UTM		After UTM
Parameter	Non-Pentest	Pentest	Pentest
Packet Loss	4	3	4
Jitter	4	2	4
Delay / Latency	4	2	4
Throughput	4	2	4
Rata-rata	4	2,25	4

Berikut merupakan analisa terkait dari tabel hasil pengujian:

a) *Packet Loss*

Berdasarkan hasil pengujian pengukuran *packet loss* pada menunjukkan kenaikan persentase pada saat

penetration testing dilakukan ke *router gateway*. Hal ini disebabkan *router* melakukan respon untuk setiap proses *request* yang diberikan oleh serangan *ICMP Flood* dan *SYN Flood* sehingga *resource* dari *router* baik itu memori ataupun *port gateway* mengalami *bottle neck* yang mengakibatkan *request* dari jaringan tidak seluruhnya terpenuhi. Paket *ICMP* yang tidak direspon oleh *router* memiliki *TTL (Time To Live)* dimana paket tersebut memiliki nilai waktu untuk merespon kembali ke *user* yang melakukan *request*. Jika paket tidak kembali melewati waktu tersebut maka paket akan dianggap hilang dan hasil dari perintah *ping* akan menunjukkan pesan *Request Time Out*. Pesan *Request Time Out* tersebut merupakan indikasi dari *packet loss* pada jaringan dari total 20 paket *ICMP* yang dikirimkan selama *interval* 20 detik untuk mengukur persentase dari *packet loss* pada setiap perangkat akses. Untuk hasil *packet loss* tertinggi dengan persentase 5% dan *packet loss* terendah dengan persentase 0%.

b) Delay / Latency

Berdasarkan hasil pengujian pengukuran menunjukkan kenaikan persentase pada saat *penetration testing* dilakukan ke *router gateway*. Hal ini disebabkan *router* melakukan respon untuk setiap proses *request* yang diberikan oleh serangan *ICMP Flood* dan *SYN Flood* sehingga *resource* dari *router* baik itu memori ataupun *port gateway* mengalami *bottle neck* yang mengakibatkan *request* dari jaringan tidak seluruhnya terpenuhi atau mengalami *delay*. Nilai *delay* yang diperoleh pada setiap perangkat akses merupakan penjumlahan nilai *processing delay*, *queueing delay*, *transmission delay* dan *propagation delay* untuk setiap paket yang dikirimkan. Untuk hasil rata-rata *delay* tertinggi dari pengiriman paket *ICMP* selama *interval* 20 detik dengan nilai 303.6 ms dan hasil terendah didapatkan dengan nilai 21.9 ms.

c) Jitter

Berdasarkan hasil pengujian pengukuran *jitter* menunjukkan kenaikan persentase pada saat *penetration testing* dilakukan ke *router gateway*. Hal ini disebabkan *router* melakukan respon untuk setiap proses *request* yang diberikan oleh serangan *ICMP Flood* dan *SYN Flood* sehingga *resource* dari *router* baik itu memori ataupun *port gateway* mengalami *bottle neck* yang mengakibatkan *request* dari jaringan tidak seluruhnya terpenuhi atau mengalami *delay*. Variasi dari nilai *delay* untuk setiap paket *ICMP* yang dikirimkan merupakan nilai *jitter* yang diperoleh pada perangkat akses. Untuk hasil *jitter* tertinggi pada perangkat akses dari pengiriman paket *ICMP* selama *interval* 20 detik dengan nilai 109.75 ms dan hasil terendah dengan nilai 3.55 ms.

d) Throughput

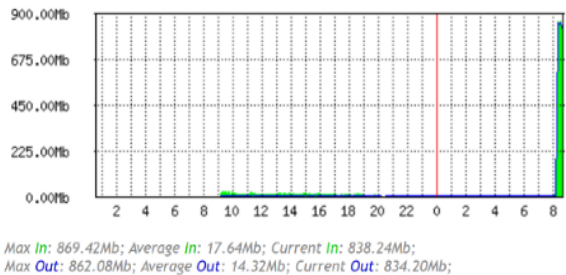
Berdasarkan hasil pengujian pengukuran *throughput* menunjukkan kenaikan persentase pada

saat *penetration testing* dilakukan ke *router gateway*. Hal ini disebabkan *router* melakukan respon untuk setiap proses *request* yang diberikan oleh serangan *ICMP Flood* dan *SYN Flood* sehingga *resource* dari *router* baik itu memori ataupun *port gateway* mengalami *bottle neck* yang mengakibatkan *request* dari jaringan tidak seluruhnya terpenuhi. *Throughput* yang diperoleh dari setiap perangkat akses merupakan maksimum *bandwidth* yang didapatkan dari *ISP* untuk digunakan oleh jaringan yaitu 30 Mbps. Untuk hasil *throughput* tertinggi pada perangkat akses dari pengujian *speedtest* selama *interval* 20 detik dengan nilai 30.6 Mbps *Downlink* dan 30,6 *Uplink* serta *throughput* terendah dengan nilai 25 Mbps *Uplink* dan 1,3 Mbps *Downlink*.

4.2. Threat Mitigation

a) Penetration Testing sebelum penerapan UTM

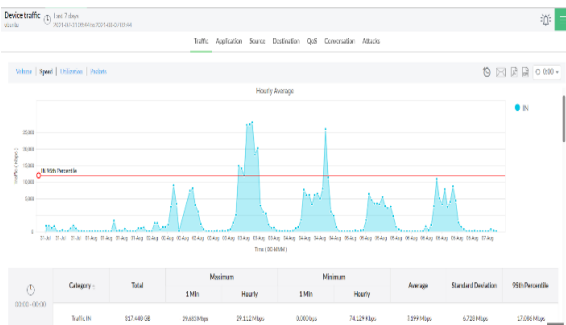
"Daily" Graph (5 Minute Average)



Gambar 11. Hasil Grafik throughput harian Penetration Testing ICMP Flood dan SYN Flood

Gambar 11 menunjukkan hasil *throughput* harian pada saat *penetration testing* dilakukan dengan nilai *throughput* yang dihasilkan dari serangan *ICMP Flood* dan *SYN Flood* maksimum sebesar 869,42 Mbps.

b) Penetration Testing setelah penerapan UTM



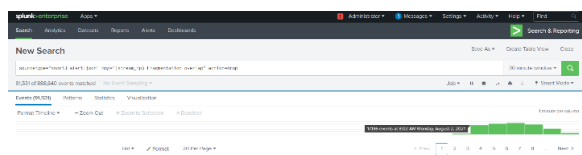
Gambar 12. Grafik throughput harian penetration testing pada UTM

Tabel 8. Throughput saat penetration testing setelah penerapan UTM

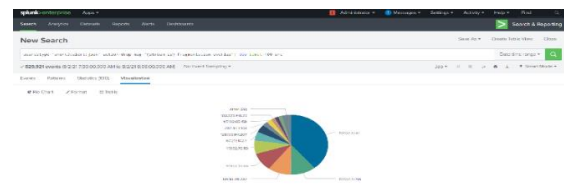
Time	Throughput (Mbps)
08:02	0,367
08:03	0,510
08:04	0,672
08:05	0,724
08:06	1,823

Pada Gambar 11 menunjukkan *monitoring* trafik yang dilakukan oleh *router* saat menerima *penetration testing ICMP Flood* dan *SYN Flood*. Serangan tersebut mampu menghasilkan *throughput* sebesar 896,42 Mbps selama 5 menit. Hasil yang berbeda ditunjukkan pada Gambar 12 dan Tabel 8 dimana setelah penerapan dari *UTM* sebagai *gateway* pada jaringan, *UTM* mampu melakukan *drop* terhadap paket dari *penetration testing ICMP Flood* dan *SYN Flood* dengan *throughput* maksimum yang diperoleh sebesar 1,823 Mbps. Hal tersebut menunjukkan bahwa *ruleset IPS* pada *UTM* mampu melakukan *drop* paket yang berasal dari serangan *ICMP Flood* dan *SYN Flood*.

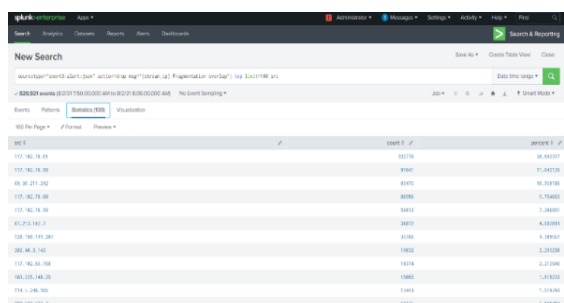
c) SIEM terhadap Penetration Testing pada UTM



Gambar 13. Pencarian indeks paket *penetration testing* yang di *drop* oleh *UTM*



Gambar 14. Grafis sumber paket *penetration testing* yang di *drop* oleh *UTM*

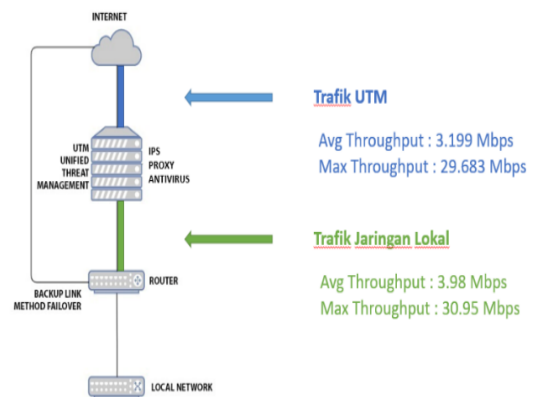


Gambar 15. Grafis sumber paket *penetration testing* yang di *drop* oleh *UTM*

Pada Gambar.13, Gambar. 14 dan Gambar. 15 menunjukan Splunk sebagai *SIEM* pada *UTM* mampu memberikan informasi secara *realtime* dari *penetration testing* yang dilakukan pada tanggal 2 Agustus 2021 Pukul 08.02 – 08.06 WIB. Persentase dari sumber paket dengan pesan “(stream_ip) fragmentation overlap” dapat diketahui baik total paket yang di *drop* dan juga sumber paket berasal. Total paket yang di *drop* selama *penetration testing* dengan *interval* 5 menit dari 3 perangkat sebanyak 232409 paket.

4.3. Traffic Policy

a) Throughput setelah penerapan Proxy Web Cache



Gambar 16. Trafik yang dianalisa untuk melihat efisiensi *bandwidth*

b) Efisiensi Bandwidth dilihat dari aktivitas Caching pada Proxy Web Server

Hasil *caching* pada Tabel 9 yang dilakukan oleh *UTM* terhadap *request* dari jaringan lokal. Terdapat *cache-in* yang merupakan *cache* yang tersimpan pada *UTM* untuk diakses oleh jaringan lokal, sedangkan *cache-out* merupakan *cache* yang dilakukan oleh *UTM* ke tujuan berdasarkan *request* jaringan lokal. Pengujian *caching* selama 5 hari kerja untuk mengakumulasi rata-rata *cache* harian. Dari total *cache* yang dilakukan selama 5 hari kerja sebesar 20,23 GB, *cache-in* yang dilakukan oleh *UTM* sebesar 943,6645 MB dan *cache-out* sebesar 19,26 GB. Hasil tersebut menunjukkan bahwa *cache* yang tersimpan pada *UTM* memiliki ukuran yang lebih kecil dibandingkan dengan permintaan dari jaringan lokal. Hal ini disebabkan *UTM* hanya melakukan *cache* terhadap setiap konten yang diakses melalui *port HTTP* dan memiliki waktu *default* penyimpanan *cache* selama 60 detik pada parameter konfigurasi *minimum_expiry_time*, sehingga jika *request* dilakukan terhadap *website* yang sama, dan lebih dari waktu tersebut, *UTM* akan menyatakan hal tersebut sebagai *cache-out*. Hal ini menunjukkan bahwa *UTM* melakukan respon pada *request* jaringan lokal yang terdapat pada *cache* dan berkaitan dengan efisiensi *bandwidth* yang dilakukan *UTM* sebesar 4,66% dari total trafik 20,23 GB.

5. KESIMPULAN

UTM sebagai *threat mitigation* telah berhasil mengidentifikasi paket pada trafik yang masuk ke dalam jaringan dan menghasilkan nilai indeks 4 dari perangkat akses pada jaringan. Hasil ini lebih baik sebelum implementasi dari *UTM* dimana serangan *DDOS* yang dilakukan mempengaruhi nilai dari indeks *QOS* pada perangkat akses yaitu 2,25.

UTM berhasil melakukan *Traffic Policy* menerapkan *Proxy Web Cache* dan *Access Control*

List beserta Delay Pools mampu mengendalikan throughput jaringan sehingga menghasilkan efisiensi bandwidth sebesar 4,66%.

Tabel 9. Caching Proxy Server

Hari Pengamatan	Cache Total	Cache In	Cache Out
Senin, 2 Agustus 2021	928,89 MB	46,4445 MB	882,4455 MB
Selasa, 3 Agustus 2021	9 GB	522,9 MB	8447,1 MB
Rabu, 4 Agustus 2021	3.92 GB	138,76 MB	3781,624 MB
Kamis, 5 Agustus 2021	5.60 GB	224,56 MB	5375,44 MB
Jumat, 6 Agustus 2021	790,97 MB	11 MB	779,97 MB
Total	20239,86 MB	943,6645 MB	19266,5795 MB
Rata-rata	4047,972 MB	188,7329 MB	3853,3159 MB

DAFTAR PUSTAKA

- A. HAFIZ & D. SUSIANTO, "Analysis of Internet Service Quality Using Internet Control Message Protocol," *J. Phys. Conf. Ser.*, vol. 1338, no. 1, 2019, doi: 10.1088/1742-6596/1338/1/012055.
- ALISYA ALIFAH & ANTRISHA DANERAICI SETIAWAN, "Performansi Jaringan VOIP Terhadap Peningkatan Pengguna Pada Variasi Bandwidth Menggunakan GNS3 dan Wireshark" *Journal of Electrical Engineering and Information Technology.*, vol. 18, no. 3, E-ISSN: 2745-5688, P-ISSN:1693-4989, Desember 2020.
- B. HERU & W. HENTO, "KEAMANAN JARINGAN MENGGUNAKAN UNIFIED THREAT MANAGEMENT PADA SERVER BERBASIS LAMP," pp. 48–59. BROUGHTON, J.M., 2002a. The Brettow Woods Proposal: a Brief Look. *Political Science Quarterly*, 42(6), p.564.
- E. RISYAD, M. DATA, & E. S. PRAMUKANTORO, "Perbandingan Performa Intrusion Detection System (IDS) Snort Dan Suricata Dalam Mendeteksi Serangan TCP SYN Flood," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 9, pp. 2615–2624, 2018.
- GOALIE, D. 2008. Remote Sensing Technology for Modern Soccer. *Popular science and Technology*, [online] Tersedia di: <<http://www.popsoci.com/b012378/soccer.html>> [Diakses 1 Juli 2009]
- IMPERVA, "DDoS Attacks," 2021. <https://www.imperva.com/learn/ddos/ddos-attacks/>.
- J. M. SNYDER, "Unified Threat Management Agenda : Unified Threat Management."
- M. KOR, J. LÁMER, & F. JAKAB, "INTRUSION PREVENTION / INTRUSION DETECTION SYSTEM (IPS / IDS) FOR WIFI NETWORKS," vol. 6, no. 4, pp. 77–89, 2014.
- CAKRANINGRAT, R., 2011. Sistem pendukung Keputusan untuk UMKM. [ebook]. UB Press. Tersedia melalui: Perpustakaan Universitas BX <<http://perpustakaan.ubx.ac.id>> [Diakses 1 Juli 2013]
- P. ENGBRETSON, *The basics of hacking and penetration testing: ethical hacking and penetration testing made easy*. Elsevier, 2013.
- S. S.KADAM & Y. C. KULKARNI, "Improving the Performance of Squid Proxy Server by using SCSI HDD and Blocking the Media Streaming," *Int. J. Comput. Appl.*, vol. 47, no. 25, pp. 38–41, 2012, doi: 10.5120/7540-0547.
- W. SUGENG, J. E. ISTIYANTO, K. MUSTOFA, & A. ASHARI, "The Impact of QoS Changes towards Network Performance," *Int. J. Comput. Networks Commun. Secur.*, vol. 3, no. 2, pp. 48–53, 2015, [Online]. Available: http://www.ijcnscs.org/published/volume3/issue2/p5_3-2.pdf.