Deteksi Konten Bittorrent Menggunakan Snort Pada Jaringan ITTelkom Surabaya

Agi Lobita Japtara Martadinata[*1)], Oktavia Ayu Permata[2)], dan Rizky Fenaldo Maulana[3)]
[1)] Fakultas Teknologi Informasi & Bisnis, Institut Teknologi Telkom Surabaya, Jl. Ketintang No.156, Surabaya, 60231, Indonesia
agitaraagi@student.ittelkom-sby.ac.id

Abstrak

Penelitian ini mengembangkan sistem untuk mencegah dan memberi peringatan penggunaan torrent di jaringan ITTelkom Surabaya. Sistem ini menggunakan snort sebagai alat deteksi torrent, barnyard2 sebagai alat pengolahan data, MySQL sebagai basis data, dan telegram sebagai media komunikasi. Penelitian ini menggunakan GNS3 dan virtualbox untuk mensimulasikan jaringan dan sistem. Hasil penelitian menunjukkan bahwa sistem dapat memonitoring trafik torrent dan mengirim notifikasi peringatan melalui telegram yang berisi informasi IP pengguna, IP tujuan, total data, dan durasi penggunaan. Hasil penelitian juga menunjukkan bahwa snort dapat mengurangi jumlah data yang digunakan oleh torrent dengan rata-rata 18025.4 bytes dalam 10 pengujian.

Kata kunci: *GNS3, Snort, Torrent, Monitoring*

1. Pendahuluan (Introduction)

Jaringan P2P merupakan jaringan komputer dimana setiap komputer yang terhubung dalam jaringan tersebut merupakan klien sekaligus juga server. Penyedia Torrent yang beredar saat ini berjalan pada jaringan internet karena membutuhkan pengguna yang cukup banyak agar bisa melakukan aktivitas berbagi konten (Saputra & Komputer, 2020). Pengguna yang berhasil mengunduh konten dari server Torrent disebut dengan *peer*. Seluruh *peer* akan saling berkontribusi untuk memberi pelayanan pada peer lainnya atau pengguna baru yang sedang mengunduh konten. Beberapa organisasi melarang penggunaan BitTorrent karena mencuri sumber daya jaringan dan membuat kerentanan kebocoran data yang berdampak negatif bagi organisasi maupun individu (Rahadiyan & Rezita Sari, 2019).

Penelitian terdahulu yang membahas hal yang sama mengenai deteksi pada jaringan kampus adalah *Interception of P2P Traffic in a Campus Network*. Penelitian tersebut menyebutkan penggunaan lalu lintas masuk dan keluar P2P mencapai 10% untuk mengunggah dan 34% untuk mengunduh menggunakan *bandwidth* kampus (MEHDI, 2019). Salah satu pengguna jaringan publik ITTelkom Surabaya berhasil mengunduh konten Torrent dan menjadi salah satu *peer* dalam *file sharing* BitTorrent. Hal ini diduga menyebabkan trafik torrent meningkat di jaringan kampus. Menurut data, trafik BitTorrent masuk lima besar penggunaan *bandwidth* di jaringan kampus. Hal ini berdampak negatif pada kegiatan akademis dan penelitian yang membutuhkan bandwidth tinggi.

App | App Group

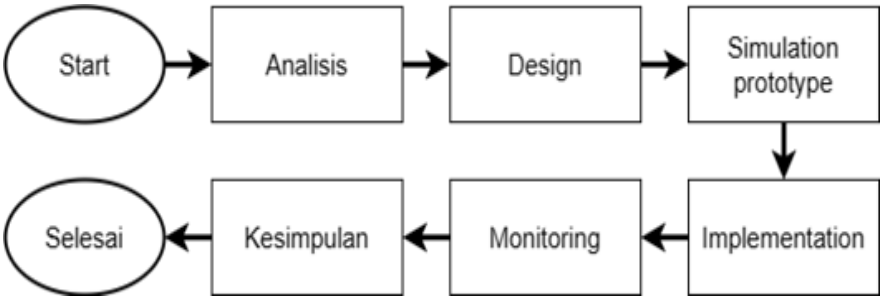
No.	Name
1	FileTransfer/HTTPS
2	IP-PROTOCOL-GROUP/UDP-TRANSFERS
3	P2P/P2P-DOWNLOAD-FLOW
4	HTTP/HTTP-BROWSE
5	HTTP/HTTP-Single-Threaded

Gambar 1. Lima Besar Penggunaan Trafik Pada Jaringan Publik ITTelkom

BitTorrent dapat mengganggu kinerja dan stabilitas jaringan kampus, serta membahayakan keamanan dan integritas data kampus karena kampus perlu menjamin keamanan jaringannya (Tampubolon, 2019). Lima besar penggunaan trafik pada jaringan publik ITTelkom Surabaya dapat dilihat pada Gambar 1. Data tersebut diperoleh oleh admin jaringan Institut Teknologi Telkom Surabaya pada tahun 2022 dengan menggunakan aplikasi pemantauan jaringan. Urutan penggunaan *peer-to-peer* yang berada pada peringkat ketiga diperoleh pada saat jam kerja atau jam perkuliahan. Manfaat dan pengaruh deteksi BitTorrent pada ITTelkom Surabaya adalah untuk mengoptimalkan penggunaan *bandwidth* jaringan, meningkatkan kualitas layanan jaringan, dan mencegah potensi ancaman keamanan. Dengan menggunakan Snort sebagai deteksi BitTorrent, jaringan ITTelkom Surabaya dapat mengidentifikasi dan memblokir lalu lintas BitTorrent yang tidak diinginkan (Fachri & Harahap, 2020). Selain itu, dengan menggunakan bot Telegram sebagai monitoring jaringan, admin jaringan dapat memantau kondisi jaringan secara *real-time* dan mendapatkan notifikasi jika ada aktivitas mencurigakan atau pelanggaran kebijakan jaringan (Christianto & Sulistyo, 2021).

2. Metode Penelitian (Methods)

Metode merupakan suatu tahapan untuk melakukan proses penelitian agar tercapainya hasil yang diharapkan. Metode yang digunakan pada penelitian ini adalah metode NDLC. NDLC adalah singkatan dari Network Development Life Cycle, yaitu sebuah metode yang digunakan untuk merencanakan, mendesain, mengimplementasikan, dan memantau sistem jaringan komputer. NDLC berasal dari SDLC (*System Development Life Cycle*), yang merupakan teknik analisis terstruktur yang digunakan untuk mengembangkan sistem informasi. NDLC memiliki beberapa tahap seperti analisis sistem, perancangan, simulasi prototipe, implementasi, monitoring, dan management (Sanjaya & Setiyadi, 2019). Model NDLC dapat dilihat pada Gambar 2.



Gambar 2. Diagram NDLC

2.1. Analisis Sistem

Analisis sistem merupakan tahap awal untuk pengumpulan data, identifikasi masalah, pemecahan masalah, dan analisis kebutuhan sistem (Mulyanto & Prakoso Budi, 2020). Hasil dari tahap ini akan menjadi dasar bagi tahap-tahap selanjutnya dalam proses pengembangan atau perancangan sistem jaringan. ini bertujuan untuk menguji kinerja snort dalam mendeteksi aktivitas torrent pada jaringan dan mengirimkan notifikasi ke telegram bot. Penelitian ini menggunakan 10 virtual komputer dan satu virtual komputer.

2.2. Perancangan Design

Perancangan merupakan tahapan perancangan guna memenuhi kebutuhan untuk memecahkan masalah. Tahap perancangan menggunakan logical design. Logical design merupakan desain abstrak yang berisi informasi dan design digambarkan tidak detail seperti implementasi nyata.

2.3. Simulation Prototype

Simulasi prototipe merupakan tahapan menerapkan hasil dari perancangan. Tahap ini dilakukan dengan simulasi menggunakan perangkat lunak Graphical Network Simulator 3(GNS3).

2.4. Implementation

Tahap implementasi merupakan tahap melakukan kegiatan pengembangan yang telah dirancang. Tahap ini tetap menggunakan perangkat lunak simulasi Graphical Network Simulator 3(GNS3) dan virtualbox (Bobo, 2021).

2.5. Monitoring

Tahap monitoring adalah tahap paling penting agar tetap menjaga jaringan komputer dan komunikasi agar sesuai dengan keinginan dan tujuan pengguna.

2.6. Management

Tahap manajemen merupakan tahapan mengelola jaringan agar dapat mendukung strategi bisnis. Membuat kebijakan terhadap layanan IT pada perusahaan bertujuan agar sistem yang telah dibangun dan berjalan dapat berlangsung lama dan unsur reliability terjaga. Tahap pengujian merupakan tahap uji coba sistem apakah telah berjalan sesuai yang telah direncanakan.

3. Hasil dan Pembahasan (Results and Discussions)

Penelitian ini bertujuan untuk menguji kinerja snort dalam mendeteksi aktivitas torrent pada jaringan dan mengirimkan notifikasi ke telegram bot. Penelitian ini menggunakan 10 virtual komputer dan satu virtual komputer untuk snort. 10 pc user ini akan melakukan beragam kegiatan dan sebagian akan melakukan torrenting. Contohnya jika pengujian ada pengguna yang melakukan *torrenting* sebanyak tiga pengguna maka tujuh pengguna sisanya akan melakukan kegiatan yang menggunakan ICMP. Komputer snort akan digunakan untuk mendeteksi dan menggunakan Barnyard2. Komputer snort akan mengirim ke bot telegram jika terdeteksi. Saat berjalan, komputer snort akan menjalankan tiga buah proses, yaitu snort, Barnyard2, dan bash untuk mendeteksi data baru dari MySQL lalu mengirimakan ke bot telegram menggunakan bash dengan jeda 5 detik, tetapi bash tidak dijalankan saat pengambilan data. Perangkat virtual komputer yang digunakan dapat dilihat pada Tabel 1.

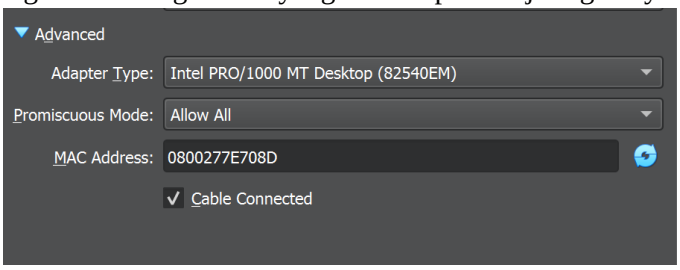
Tabel 1. Perangkat Virtual Komputer

Nama	Deskripsi
PC Master	Digunakan untuk klon virtual komputer klien
PC klien 1	Virtual komputer klien ke-1
PC klien 2	Virtual komputer klien ke-2
PC klien 3	Virtual komputer klien ke-3
PC klien 4	Virtual komputer klien ke-4
PC klien 5	Virtual komputer klien ke-5
PC klien 6	Virtual komputer klien ke-6
PC klien 7	Virtual komputer klien ke-7
PC klien 8	Virtual komputer klien ke-8
PC klien 9	Virtual komputer klien ke-9
PC klien 10	Virtual komputer klien ke-10
PC snort	Virtual komputer snort

3.1. Deteksi Konten Torrent

Torrent yang akan diunduh adalah torrent ubuntu *desktop* yang diunduh secara resmi dihalaman ubuntu. Selanjutnya melakukan konfigurasi seperti pembatasan kecepatan unduhan torrent 10kbps dan mencoba melakukan unduhan. Hal tersebut dilakukan agar torrent yang diunduh tidak memerlukan sumber daya internet yang besar dan mencegah unduhan selesai. Pengujian akan dilakukan

menggunakan dua protokol yang berbeda yaitu icmp dan tcp sebagai BitTorrent. Protokol icmp akan menggunakan perintah ping ke server snort dan protokol tcp akan dilakukan pengunduhan BitTorrent. Setelah bagian menguji unduhan BitTorrent berjalan dengan benar, menyalakan fitur mode promiscuous ke *allow all* pada konfigurasi jaringan di pc klien yang akan diklon, *mode promiscusous* dapat dilihat pada Gambar 3. Mode promiscuous mengizinkan virtual komputer bagian dari hub bukan switch, memungkinkan NIDS digunakan dengan cara yang sama seperti di jaringan nyata (Hänninen, 2019).



Gambar 3. *Mode Promiscusous*

Setelah bagian menguji unduhan BitTorrent berjalan dengan benar, menyalakan fitur mode promiscuous ke *allow all* pada konfigurasi jaringan di pc klien yang akan diklon, *mode promiscusous* dapat dilihat pada Gambar 3. Kemudian melakukan pemasangan snort menggunakan ubuntu repositori dengan perintah "sudo apt-get install snort". Setelah berhasil memasang, snort meminta konfigurasi dasar seperti *interface* yang digunakan yaitu enp0s3 dan ip yang ingin dilindungi. Kemudian membuat berkas baru didalam folder "/etc/snort/rules" dengan nama yang digunakan "bittorrent.rules". Aturan yang digunakan pada penelitian dapat dilihat pada Gambar 4. Snort tidak akan menghasilkan alert juga didalam perintahnya karena untuk menggunakan unified harus menggunakan quiet mode dengan perintah -q, jika tidak maka snort akan menghasil log file yang tidak dapat dibaca oleh Barnyard2.

```
alert tcp $EXTERNAL_NET any <> $HOME_NET any
(msg:"Content torrent"; flow:to_client,establ
ished; content:"torrent"; fast_pattern:only;
sid:1000001; rev:1;)
```

Gambar 4. Konfigurasi Aturan Torrent

Snort harus dapat membuat sebuah *output file* berupa u2 yang dapat dibaca oleh Barnyard2. Pengaturan untuk menghasil u2 dapat dilakukan di "etc/snort/snort.conf". Setelah ditambahkan konfigurasi u2 maka snort di *restard*. Gambar konfigurasi tambahan untuk snort agar menghasil u2 dapat dilihat pada Gambar 5.

```
output unified2: filename snort.u2, limit 128
```

Gambar 5. Konfigurasi Snort Untuk u2

Dalam pengembangan MySQL menggunakan user yang memiliki kata sandi dan akses penuh ke database. Barnyard memiliki struktur database dalam folder unduhannya. membuat sebuah user bernama snort dan diberi akses create, insert, select, delete, update terhadap database snort. Password yang digunakan adalah "MYPASSWORD". MySQL yang digunakan adalah versi terbaru dari milik ubuntu sedangkan Barnyard2 yang digunakan adalah versi stable yang dapat diunduh dalam repository github milik Barnyard2. Barnyard akan memantau file u2 yang terbaru dari snort, ketika terdapat pembaharuan dari file tersebut maka Barnyard2 akan menyimpannya di MySQL. Agar Barnyard2 dapat mengakses MySQL perlu dilakukan konfigurasi pada "/etc/snort/Barnyard2.conf" yang dapat dilihat pada Gambar 6.

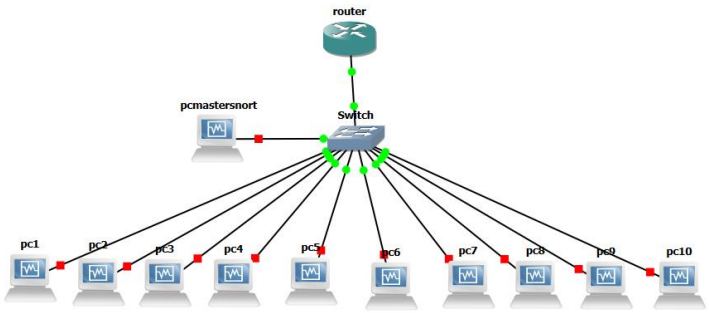
```
output database: log, mysql, user=snort password=MYPA
SSWORD dbname=snort host=localhost
```

Gambar 6 Konfigurasi Barnyard Untuk MySQL

3.2. Topologi Jaringan

Pengembangan komputer klien dan komputer snort, melakukan *import* 10 virtual komputer klien dan satu virtual komputer snort ke dalam GNS3 dengan menggunakan fitur preferensi untuk virtualbox VMs di GNS3 melalui menu Edit > Preferences > VirtualBox > VirtualBox VMs. Fitur ini memungkinkan untuk menambahkan virtual komputer yang sudah terinstal pada virtual box ke dalam GNS3 dengan mudah. Kemudian menambahkan node cloud ke dalam GNS3. Node cloud adalah perangkat yang dapat digunakan untuk menghubungkan jaringan GNS3 dengan jaringan luar, seperti internet. Setelah semuanya terhubung, tampilan node cloud menjadi router agar menggambarkan bahwa jaringan publik pada umumnya terhubung ke router.

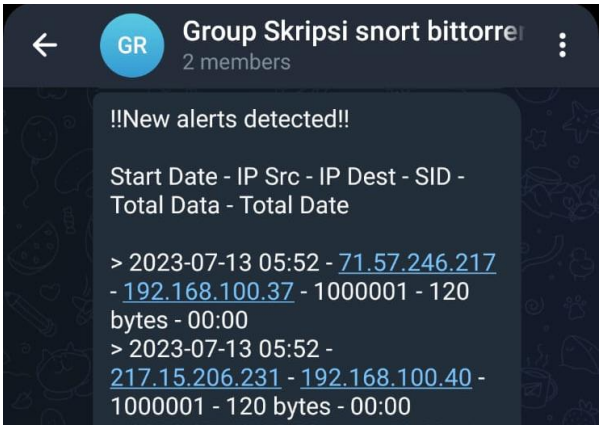
Topologi yang digunakan adalah topologi star. Topologi star adalah topologi jaringan yang menghubungkan setiap perangkat ke perangkat pusat, seperti switch atau hub. Implementasi topologi star di ITTelkom Surabaya atau kampus adalah ruangan laboratorium, ruangan kantor, dan ruangan server. Perancangan topologi pada GNS3 dapat dilihat pada Gambar 7.



Gambar 7 Penerapan Topologi Star di GNS3

3.3. Bot Telegram

Bot telegram menggunakan @BotFather. Untuk memulai membuat bot diperlukan perintah "/newbot" pada kolom chat. Setelah itu @BotFather akan menanyakan nama bot yang akan digunakan. Bot yang digunakan akan diberi nama dengan skripsi snort bittorrent bot. Bot yang telah berhasil dibuat dimasukan ke dalam channel Group Skripsi snort bittorrent. Kemudian membuat skrip bash untuk mengirimkan notifikasi ke channel telegram menggunakan bot telegram. Skrip bash akan mengirim ketika terdapat perbedaan jumlah data saat ini dengan data terbaru dari MySQL. Kemudian mengambil data tanggal dimulai, *ip source*, *ip destination*, sid atau aturan snort, total data, dan total waktu dengan format jam dan menit. Setiap IP akan menurun dan membuat baris baru agar daftar IP apa saja yang melakukan *torrrenting* pada jaringan lokal. Hasil tersebut dapat dilihat pada Gambar 8.



Gambar 8 Hasil Notifikasi Bot Telegram

3.3. Analisis Hasil Snort

Setiap pengujian yang akan dilakukan dengan menentukan virtual komputer klien apa saja yang akan melakukan torrenting atau tidak sebelum pengambilan data dan menjalan komputer snort bagian

terakhir. Pengambilan data tidak menggunakan GNS3, namun GNS3 dikembangkan setelah seluruh pengujian dan pengambilan data selesai. GNS3 digunakan sebagai penerapan topologi. Pengambilan data akan dilakukan selama 20 menit dengan menjalankan semua virtual komputer dan membatasi kecepatan unduhan menjadi 10kbps. Peneliti tidak melakukan *seeding* karena berfokus untuk mengatasi dalam torrent sedangkan kegiatan *seeding* sudah termasuk kegiatan *torrenting*. Dalam pengujian tidak menggunakan bash untuk notifikasi agar lebih efektif.

Tabel 2. Hasil Pengujian

No	Nama Pengujian	Jumlah Data (bytes)
1	10 koneksi BitTorrent	28075 bytes
2	9 koneksi BitTorrent	26184 bytes
3	8 koneksi BitTorrent	25656 bytes
4	7 koneksi BitTorrent	22652 bytes
5	6 koneksi BitTorrent	16228 bytes
6	5 koneksi BitTorrent	16134 bytes
7	4 koneksi BitTorrent	14914 bytes
8	3 koneksi BitTorrent	11715 bytes
9	2 koneksi BitTorrent	10416 bytes
10	1 koneksi BitTorrent	8280 bytes
Rata-rata	-	18025.4 bytes

Adapun grafik batang untuk memvisualisasikan hasil pengambilan data. Pada tabel diatas menggunakan sumbu x menunjukkan jumlah komputer yang menggunakan torrent dan sumbu y menunjukkan rentangan data dalam byte dari 0 byte hingga 30000 bytes. Gambar visualisasi data dapat dilihat pada Gambar 9.

Gambar 9 Diagram Batang Hasil Pengambilan Data

Berdasarkan diagram diatas, bahwa snort dapat mendeteksi aktivitas torrenting pada jaringan BitTorrent dengan cukup efektif, jumlah data yang didapatkan oleh snort cenderung menurun seiring dengan menurunnya jumlah pengguna yang melakukan koneksi BitTorrent atau *downloading*. Hal disebabkan bahwa semakin sedikit koneksi BitTorrent, semakin sedikit paket data yang ditransfer antara virtual komputer klien, sehingga snort dapat mendeteksi lebih banyak paket data tersebut. Pengujian 10, yaitu satu koneksi BitTorrent, adalah pengujian yang menggunakan data terkecil, yaitu 8280 bytes, karena hanya ada satu virtual komputer klien yang melakukan torrenting, sehingga snort dapat mendeteksi sedikit paket data yang ditransfer oleh virtual komputer klien tersebut. Rata-rata jumlah data yang digunakan oleh snort dalam 10 pengujian adalah 18025.4 bytes

3.1. Kesimpulan (Conclusion)

Bagian Berdasarkan hasil pengujian yang dilakukan, dapat menyimpulkan bahwa:

- Snort dapat mendeteksi aktivitas torrenting pada jaringan BitTorrent dengan cukup efektif. Snort dapat mengidentifikasi paket data yang mengandung kata “torrent” dan

menyimpannya ke dalam file snort.u2. Jumlah data yang digunakan oleh snort dalam setiap pengujian bervariasi tergantung pada jumlah koneksi BitTorrent yang dilakukan oleh virtual komputer klien. Rata-rata jumlah data yang didapatkan oleh snort dalam 10 pengujian adalah 18025.4 bytes.

- Telegram bot dapat memberikan notifikasi kepada pengguna dengan cukup cepat. Telegram bot dapat mengirimkan pesan teks yang berisi informasi tentang paket data yang dideteksi oleh snort ke channel telegram. Pesan teks tersebut berisi tanggal, waktu, sumber, tujuan, sid dari rule, dan lama penggunaan dengan format jam dan menit.
- Topologi jaringan BitTorrent yang dibuat di GNS3 dapat diterapkan di topologi jaringan fisik. Topologi jaringan BitTorrent yang dibuat di GNS3 menggunakan topologi star dengan node cloud sebagai titik pusat yang merepresentasikan router. Topologi ini dapat merepresentasikan jaringan publik wifi ITTS dengan menggunakan perangkat keras dan perangkat lunak yang sama.
- Torrenting dalam jaringan kampus dapat berdampak negatif pada kualitas layanan internet dan keamanan jaringan. Pengguna jaringan kampus harus bertanggung jawab dan menghormati hak dan kepentingan pengguna lain. Torrenting harus dihindari atau dibatasi agar tidak mengganggu aktivitas online yang penting dan sah.

Ucapan Terima Kasih (Acknowledgement)

Saya ingin mengucapkan terima kasih kepada dosen pembimbing saya, Ibu Oktavia Ayu Permata dan Bapak Rizky Fenaldo Maulana yang telah membimbing saya dalam penulisan karya ilmiah ini. Saya juga ingin mengucapkan terima kasih kepada teman-teman dan keluarga yang telah memberikan dukungan moril dan materil selama proses penulisan karya ilmiah ini.

Daftar Pustaka

Bobo, S. (2021). *ARSITEKTUR DAN SISTEM KEAMANAN JARINGAN MENGGUNAKAN METODE NDLC PADA SEKOLAH MENENGAH KEJURUAN NEGERI 13 LUWU*.

Christianto, N., & Sulistyo, W. (2021). Model Pemantauan Keamanan Jaringan Melalui Aplikasi Telegram Dengan Snort. *Jurnal Teknik Informatika Dan Sistem Informasi*, *7*, 2443–2229. https://doi.org/10.28932/jutisi.v7i1.4088

Fachri, B., & Harahap, F. H. (2020). Simulasi Penggunaan Intrusion Detection System (IDS) Sebagai Keamanan Jaringan dan Komputer. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, *4*(2), 413. https://doi.org/10.30865/mib.v4i2.2037

Hänninen, M. (2019). *Open source intrusion detection systems evaluation for small and medium-sized enterprise environments*.

MEHDI, M. (2019). Interception of P2P Traffic in a Campus Network. *Revista Română de Informatică Şi Automatică*, *29*(2). https://doi.org/10.33436/v29i2y201902

Mulyanto, Y., & Prakoso Budi, P. (2020). *RANCANG BANGUN JARINGAN KOMPUTER MENGGUNAKAN SISTEM MANAJEMEN OMADA CONTROLLER PADA INSPEKTORAT KABUPATEN SUMBAWADENGAN METODE NETWORK DEVELOPMENT LIFE CYCLE (NDLC)*.

Rahadiyan, I., & Rezita Sari, A. (2019). *PELUANG DAN TANTANGAN IMPLEMENTASI FINTECH PEER TO PEER LENDING SEBAGAI SALAH SATU UPAYA PENINGKATAN KESEJAHTERAAN MASYARAKAT INDONESIA* (Vol. 4, Issue 1). http://www.bi.go.id/id/perbankan/keuanganinklus

Sanjaya, T., & Setiyadi, D. (2019). Network Development Life Cycle (NDLC) Dalam Perancangan Jaringan Komputer Pada Rumah Shalom Mahanaim. In *Rawa Panjang Bekasi Timur* (Vol. 4, Issue 1).

Saputra, R. D., & Komputer, F. (2020). *IMPLEMENTASI JARINGAN PEER TO PEER DALAM PROSES TRANSFER DATA DUA PERSONAL COMPUTER MENGGUNAKAN KABEL UTP BERTYPE CROSS*.

Tampubolon, H. R. (2019). SELUK-BELUK PEER TO PEER LENDING SEBAGAI WUJUD BARU KEUANGAN DI INDONESIA. *Jurnal Bina Mulia Hukum*, *3*(2), 188–198. https://doi.org/10.23920/jbmh.v3n2.15