

# PERBANDINGAN ALGORITMA RED, SFQ, DAN AQM PADA JARINGAN ENTERPRISE DENGAN VMWARE ESXI DAN ROUTER OS

*by Himamunanto Agustinus Rudatyo*

---

**Submission date:** 03-Apr-2023 11:07AM (UTC+0700)

**Submission ID:** 2054228620

**File name:** ARH\_-\_INFINTERAKTIF-JANA.pdf (767.66K)

**Word count:** 4910

**Character count:** 29996

Volume 6 Nomor 3 September 2021

# INFORMASI INTERAKTIF

JURNAL INFORMATIKA DAN TEKNOLOGI INFORMASI

**PROGRAM STUDI INFORMATIKA – FAKULTAS TEKNIK -UNIVERSITAS JANABADRA**

RANCANG APLIKASI SISTEM INFORMASI BANK SOAL DAN ANALISIS BUTIR SOAL DI FAKULTAS KEDOKTERAN XYZ

**Dinar Mustofa, Anggit Wirasto, Abdul Ghofur**

PERBANDINGAN ALGORITMA RED, SFQ, DAN AQM PADA JARINGAN *ENTERPRISE* DENGAN *VMWARE ESXI* DAN *ROUTER OS*

**Azriel Christian Nurcahyo, Listra Firgia, Ag. Rudatyo Himamunanto**

IMPLEMENTASI METODE *K-NEAREST NEIGHBOR* BERBASIS *EUCLIDEAN DISTANCE* UNTUK KLASIFIKASI PENERIMAAN VAKSIN COVID-19

**Yumarlin MZ, Rizqi Mirza Fadilla, Indra Pratama**

ANALISIS DATA UNTUK PENGELOMPOKAN MAHASISWA DENGAN METODE K-MEAN (STUDI KASUS : INSTITUT SHANTI BHUANA)

**Santi Thomas, Noviyanti P**

JARINGAN SYARAF TIRUAN MENGGUNAKAN METODE *BACKPROPAGATION* DALAM PREDIKSI PERSEDIAAN BAHAN BAKU (STUDI KASUS : PT. BINTANG TOBA LESTARI)

**Niko Surya Atmaja, Deri Lianda**



INFORMASI  
INTERAKTIF

Vol. 6

No. 3

Hal. 96 - 123

Yogyakarta  
September  
2021

ISSN  
2527-5240

**DEWAN EDITORIAL**

- Penerbit** : Program Studi Informatika Fakultas Teknik Universitas Janabadra
- Ketua Penyunting (Editor in Chief)** : Fatsyahrina Fitriastuti, S.Si., M.T. (Universitas Janabadra)
- Penyunting (Editor)** : 1. Jemmy Edwin B, S.Kom., M.Eng. (Universitas Janabadra)  
2. Ryan Ari Setyawan, S.Kom., M.Eng. (Universitas Janabadra)  
3. Yumarlin MZ, S.Kom., M.Pd., M.Kom. (Universitas Janabadra)
- Alamat Redaksi** : Program Studi Informatika Fakultas Teknik  
Universitas Janabadra  
Jl. Tentara Rakyat Mataram No. 55-57  
Yogyakarta 55231  
Telp./Fax : (0274) 543676  
E-mail: [informasi.interaktif@janabadra.ac.id](mailto:informasi.interaktif@janabadra.ac.id)  
Website : <http://e-journal.janabadra.ac.id/>
- Frekuensi Terbit** : 3 kali setahun

**JURNAL INFORMASI INTERAKTIF** merupakan media komunikasi hasil penelitian, studi kasus, dan ulasan ilmiah bagi ilmuwan dan praktisi dibidang Informatika. Diterbitkan oleh Program Studi Informatika Fakultas Teknik Universitas Janabadra di Yogyakarta, tiga kali setahun pada bulan Januari, Mei dan September.

## DAFTAR ISI

	<i>halaman</i>
Rancang Aplikasi Sistem Informasi Bank Soal Dan Analisis Butir Soal Di Fakultas Kedokteran XYZ <b>Dinar Mustofa, Anggit Wirasto, Abdul Ghofur</b>	96 - 100
Perbandingan Algoritma RED, SFQ, Dan AQM Pada Jaringan <i>Enterprise</i> Dengan <i>Vmware ESXI</i> Dan <i>Router OS</i> <b>Azriel Christian Nurcahyo, Listra Firgia, Ag. Rudatyo Himamunanto</b>	101-110
Implementasi Metode <i>K-Nearst Neighbor</i> Berbasis <i>Euclidean Distance</i> Untuk Klasifikasi Penerimaan Vaksin Covid-19 <b>Yumarlin MZ, Rizqi Mirza Fadilla, Indra Pratama</b>	111-117
Analisis Data Untuk Pengelompokan Mahasiswa Dengan Metode K-MEAN (Studi Kasus : Institut Shanti Bhuna) <b>Santi Thomas, Noviyanti P</b>	118-123
Jaringan Syaraf Tiruan Menggunakan Metode <i>Backpropagation</i> Dalam Prediksi Persediaan Bahan Baku (Studi Kasus : PT. Bintang Toba Lestari) <b>Niko Surya Atmaja, Deri Lianda</b>	124-133

## **PENGANTAR REDAKSI**

Puji syukur kami panjatkan kehadirat Allah Tuhan Yang Maha Kuasa atas terbitnya JURNAL INFORMASI INTERAKTIF Volume 6, Nomor 2, Edisi September 2021. Pada edisi kali ini memuat 5 (lima) tulisan hasil penelitian dalam bidang informatika.

Harapan kami semoga naskah yang tersaji dalam JURNAL INFORMASI INTERAKTIF edisi September tahun 2021 dapat menambah pengetahuan dan wawasan di bidangnya masing-masing dan bagi penulis, jurnal ini diharapkan menjadi salah satu wadah untuk berbagi hasil-hasil penelitian yang telah dilakukan kepada seluruh akademisi maupun masyarakat pada umumnya.

Redaksi

## PERBANDINGAN ALGORITMA RED, SFQ, DAN AQM PADA JARINGAN ENTERPRISE DENGAN VMWARE ESXI DAN ROUTER OS

Azriel Christian Nurcahyo<sup>1</sup>, Listra Firgia<sup>2</sup>, Ag. Rudatyo Himamunanto<sup>3</sup>

<sup>1,2</sup>Teknologi Informasi, Institut Shanti Bhuna, Jl. Bukit Karmel No.1, Bengkayang, Kalimantan Barat

<sup>3</sup>Informatika, Universitas Kristen Immanuel, Jl. Solo Km. 11.1 Kalasan, Yogyakarta

Email : <sup>1</sup>azriel@shantibhuana.ac.id, <sup>2</sup>listrajsc@gmail.com, rudatyo@ukrimuniversity.ac.id

### ABSTRACT

The use of the internet network since the covid-19 pandemic is increasingly needed as the backbone of the implementation of mobile technology, websites, and desktops where all of them utilize network infrastructure and servers. The use of the right enterprise network combined with adequate server management results in optimal access and stable network conditions. The more complex the network that is built, the more server needs are needed and the more complicated the queuing algorithm that is built to produce the appropriate access speed and optimal storage. Behind these needs, there are several algorithms that can be implemented to optimize the application of enterprise networks.

In this study, the implementation of a comparison between the RED, SFQ, and AQM algorithms on RouterOS and VMware ESXi in an enterprise network, in this case a university-scale internet network, is carried out. RouterOS was chosen because most of the networks in the agency, especially using Mikrotik RouterOS for network management and monitoring, while VMware ESXi is used as a virtualization server that can replace many physical servers into one server by implementing virtualization in it to be more practical and efficient. The implementation of the RED, SFQ, and AQM algorithms was tested during March to July and the results showed that the AQM throughput test value was greater than RED and SFQ by 91.4%, the delay and packet loss test values of AQM were better than RED and SFQ by 7% and 8856 ms.

**Keywords:** VMware ESXi, RED, SFQ, AQM, routerOS.

### 1. PENDAHULUAN

Penggunaan jaringan komputer saat ini berfungsi sebagai tulang punggung utama dalam akses sistem untuk aplikasi berbasis web, desktop hingga android [1]. Tidak terkecuali di era pandemi covid-19 tahun 2020 hingga 2021 terutama di sektor pendidikan dan bisnis ketika semua pembelajaran dilaksanakan secara online, hingga sektor bisnis yang melibatkan penggunaan platform jual beli online hingga toko online [2]. Hal ini tentu memerlukan perhatian khusus sebagai pihak IT administrator bagi instansi maupun perusahaan dalam mengelola suatu jaringan dikarenakan dari sisi pengguna tentu menjadi bertambah, serta kebutuhan akan server menjadi penting mengingat banyaknya sistem yang diperlukan untuk hosting hingga mahalnya biaya menyewa VPS yang tentu membutuhkan dana lebih mahal hingga penggunaan cpanel yang hanya dapat dimanfaatkan sebatas web dikarenakan terbatas dari sisi storage [3]. Dari sisi pengguna yang bertambah tentu diperlukan pengelolaan

manajemen bandwidth yang sesuai dengan kebutuhan dari sisi upload dan download baik jalur IX dan IIX, serta dari sisi server tentu diperlukan manajemen pengelolaan lebih lanjut untuk alokasi bandwidth [4]. Tentunya akan menghabiskan anggaran jika membuat 1 server dengan 1 web sistem informasi di dalamnya, oleh karena itu diperlukan manajemen server lebih lanjut selain manajemen bandwidth, hal ini merupakan satu kesatuan yang tidak terpisah sebagai IT administrator ketika dilakukan pengelolaan sistem dan jaringan enterprise [3].

Beberapa penelitian telah membahas manajemen bandwidth dengan model LAN maupun WLAN namun hanya sebatas metode seperti simple queue, queue tree, CBQ, PFIFO, hingga HTB. Serta beberapa penelitian lainnya membahas mengenai manajemen server seperti load balance web server hingga manajemen bandwidth upload pada server. Pada penelitian ini dilakukan integrasi keduanya antara manajemen bandwidth pada server vmware dengan menggunakan beberapa algoritma yang berbeda antara lain algoritma manajemen

bandwidth RED, SFQ, dan AQM, dimana AQM masih sedikit dibahas mengenai implementasinya serta adanya integrasi server menggunakan virtualisasi Vsphere ESXi. Virtualisasi server ditujukan untuk mengalokasikan website dan sistem operasi dalam satu server tanpa harus melibatkan beberapa PC real sebagai server lainnya jika administrator hendak menerapkan lebih dari satu sistem informasi dalam satu server [5]. Hal ini sesuai dengan arah pengembangan jaringan di bidang server dari tahun ke tahun seperti berkembangnya beberapa platform virtualisasi antara lain docker, proxmox, virtual box, hingga dalam hal ini digunakan dalam penelitian yaitu vsphere esxi [6]. Selain server dari sisi jaringan juga dikembangkan routing, maupun implementasi manajerial bandwidth. Pada penelitian ini digunakan routers mikrotik untuk melakukan manajerial bandwidth yang terintegrasi dengan virtualisasi vmware vsphere esxi.

Penelitian ini melakukan perbandingan ketiga algoritma antrian bandwidth yaitu RED (Random Early Drop), SFQ (Stochastic Fairness Queueing), dan AQM (Active Queue Management) pada jaringan enterprise menggunakan perangkat yang ada di laboratorium pemrograman dan jaringan Institut Shanti Bhuna dengan perangkat real yaitu mikrotik routerOS yang terintegrasi dengan Vmware ESXi sehingga dapat dibuktikan algoritma mana yang lebih efisien pada pengujian sesuai standarisasi TIPHON jika digunakan pada skala instansi seperti kampus. Diharapkan penelitian ini dapat menjadi solusi dan contoh bagi para peneliti bidang jaringan komputer dan server, maupun IT administrator yang selama ini masih kesulitan mengimplementasikan serta sedang mencari metode terbaik dan termudah untuk manajemen bandwidth sistem jaringan dan server.

## 2. PENELITIAN TERKAIT

Penelitian mengenai manajemen algoritma bandwidth, server maupun analisa backbone jaringan skala enterprise pada suatu objek instansi maupun secara simulasi pada penelitian ini dibagi menjadi empat penelitian sebagai dasar rujukan, maupun relevansi, dengan penelitian ini. Pertama penelitian pernah dilakukan oleh Try Adrianto Darsono dengan judul Analisis Kinerja Jaringan Penghubung Antar gedung Menggunakan Wifi (Studi Kasus :

Universitas Musamus Merauke) tahun 2017 jurusan S2 Ilmu Komputer UGM. Kesimpulan yang dapat ditarik yaitu pertama setelah dilakukan uji pengukuran, jaringan penghubung antar gedung menggunakan wifi di Universitas Musamus Merauke didapatkan bahwa belum dapat memenuhi kebutuhan kuota. Kedua hasil nilai throughput yang didapat keseluruhan 3579,8 Bps. Ketiga nilai packet loss 15,5 % berdasarkan standar TIPHON nilai delay 398,55 ms [7]. Pada penelitian ini memiliki kekurangan yang dikembangkan pada penelitian ini yaitu melibatkan kondisi jaringan yang lebih kompleks sehingga menggunakan sampel jaringan universitas yang terintegrasi dengan vmware.

Kedua sebagai rujukan penelitian oleh Muhlis Muhallim dengan judul Pengembangan Jaringan Komputer Universitas Andi Djemma Palopo Berdasarkan Perbandingan Jaringan Protokol Routing Statik dan OSPFv2 pada tahun 2017 di S2 Ilmu Komputer UGM. Penelitian ini menghasilkan pernyataan bahwa jaringan dengan menerapkan protokol OSPFv2 sebagai routing protokol yang direkomendasikan dalam pengembangan jaringan komputer. Pada penelitian ini masih memiliki saran dimana perlu dilakukan implementasi langsung pada jaringan tidak hanya sebatas simulasi [8]. Ketiga penelitian oleh Muhammad Diponegoro dengan judul Analisis Quality of Service (QoS) Pada Jaringan Internet Dengan Metode Fixed Daily Measurement Interval (FDMI) dan Non FDMI Studi Kasus : UGM-Hotspot di Program Pascasarjana S2 Ilmu Komputer Universitas Gadjah Mada pada tahun 2015. Pada penelitian ini terdapat kesimpulan pokok yaitu performa layanan jaringan internet UGM-Hotspot Pascasarjana Ilmu Komputer Universitas Gadjah Mada memiliki tingkat kualitas delay sebesar 177 milidetik sehingga dapat diterima dengan ketentuan bahwa administrator mengetahui waktu transmisi dan dampaknya pada kualitas transmisi pengguna aplikasi. Besar throughput sebesar 78 Mbps dan utilisasi sebesar 63% dikategorikan menurut standarisasi TIPHON bagus tetapi memiliki nilai packet loss ratio sebesar 21 % dikategorikan menurut standarisasi TIPHON adalah jelek [9]. Pada penelitian tersebut tidak dijelaskan algoritma yang digunakan pada manajerial bandwidth maupun server sehingga terbatas pada analisis penggunaan tanpa melibatkan konfigurasi. Sehingga pada penelitian ini dilakukan

konfigurasi jalur distribusi *router* dan *server* sebagai bentuk pembeda.

Penelitian keempat oleh Ivan Paulov dengan judul *Routing in a Virtualised Environment with RouterOS Helsinki Metropolia University of Applied Sciences Bachelor of Engineering Information Technology* pada *thesis* tahun 2015. Pada penelitian ini menghasilkan simulasi jaringan menggunakan *Mikrotik Router OS* pada *VM Ware* berhasil dilakukan meliputi *OSPF, Tunneling, Routing* namun masih dapat dikembangkan lagi untuk simulasi *MPLS, IPv6, hingga BGP* [10]. Pada penelitian ini terbatas pada simulasi *vmware* dan *routerOS* tanpa melibatkan kondisi real, sehingga memiliki perbedaan dengan penelitian yang akan dilakukan penulis melibatkan jaringan server enterprise skala universitas.

### 3. LANDASAN TEORI

#### 3.1 Algoritma RED

Algoritma *Random Early Drop* adalah mekanisme antrian yang digunakan untuk menghindari kemacetan jaringan dengan mengontrol ukuran antrian rata-rata. Ukuran antrian rata-rata jika dibandingkan dengan dua ambang batas: ambang minimum (*min th*) dan maksimum (*max th*). Jika ukuran antrian rata-rata (*avg q*) kurang dari ambang batas *minimum*, tidak ada paket yang dijatuhkan. Namun ketika ukuran antrian rata-rata lebih besar dari ambang batas *maksimum*, semua paket yang masuk akan dilakukan penghapusan. Kemudian jika ukuran antrian rata-rata berada di antara ambang *minimum* dan *maksimum*, paket-paket dijatuhkan secara acak dengan probabilitas *P<sub>d</sub>* di mana probabilitas merupakan fungsi dari ukuran antrian rata-rata:

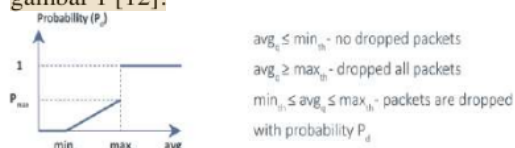
$$P_d = P_{max}(avg_q - min_{th}) / (max_{th} - min_{th})$$

Jika antrian rata-rata selalu bertambah, kemungkinan untuk membuang paket yang masuk juga bertambah [11].

Hal yang memungkinkan dilakukan drop atau tidaknya paket dalam algoritma *RED* terpenuhi pada beberapa kondisi sebagai berikut

1. Jika nilai *AVGq* lebih kecil dari nilai *Min Threshold* maka paket akan tetap diproses
2. Jika nilai *AVGq* lebih besar dari nilai *Max Threshold* maka paket tidak akan diproses atau dilakukan *drop*.

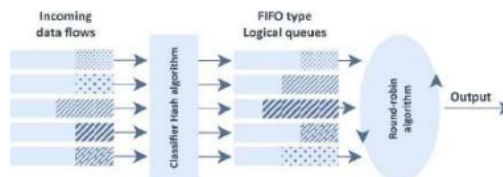
Apabila nilai *AVGq* berada pada antara nilai *Min Threshold* dan nilai *Max Threshold* maka besar kemungkinan pak<sup>18</sup> dilakukan drop. Model skema algoritma *RED* dapat dilihat pada gambar 1 [12].



Gambar 1 Skema Algoritma RED

#### 3.2 Algoritma SFQ

Algoritma *Stochastic Fairness Queuing* (*SFQ*) digunakan untuk menyeimbangkan trafik atau *packet flow* dalam sebuah jaringan [13]. *Packet flow* yang masuk akan dikelompokkan ke dalam beberapa *sub queue* menggunakan algoritma hashing. Ilustrasi mekanisme *sfq* dapat dilihat pada gambar 2.



Gambar 2. Mekanisme SFQ

Cara kerja *sfq* dimulai pada sekumpulan *packet* yang masuk diklasifikasikan ke dalam beberapa *sub queue* dengan menggunakan algoritma *hashing*. *Sub queue* yang dibentuk oleh *sfq* menggunakan mekanisme *first in first out* yang pertama masuk maka diproses *flow* sesuai yang telah masuk. Kemudian *packet* masuk ke dalam *sub queue* maka *traffic* dikeluarkan melalui *output interface*. Agar dapat mengeluarkan *packet-packet* dalam *sub-queue*, maka diterapkan mekanisme *hashing* pada *sfq* untuk menentukan *packet* yang keluar maupun ditahan atau diproses terlebih dahulu. Dalam pemrosesan akhir, terbentuk 1024 *sub queue* dimana tiap *per sub queue* terdiri dari 128 paket.

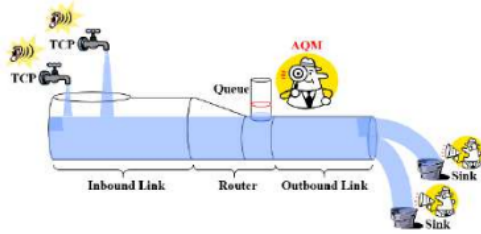
#### 3.3 Algoritma AQM

Algoritma *AQM* masih sangat sedikit dibahas pada konsep manajerial *bandwidth* jaringan server. Model dari konsep algoritma ini dapat dilihat pada gambar 3 [14]. Pada algoritma ini dilakukan penandaan paket



tergantung pada panjang antrian rata-rata paket dengan formula sebagai berikut

$$p = p(x)$$



Gambar 3. Mekanisme AQM

Pada gambar dijelaskan untuk pengendalian kemacetan pada AQM melalui paket TCP dengan cara mengurangi tingkat pengiriman saat kehilangan paket yang tidak bisa terdeteksi, dan dilakukan pembuangan paket untuk paket yang tidak terdaftar, kemudian pada saat terjadi kemacetan dilakukan penandaan paket untuk interaksi antara TCP dan router. Pada AQM dilakukan pembuangan paket yang telah ditandai seperti gambar 5 dan sebagai paket tidak terdaftar dengan rumus *loss rate* sebagai berikut

$$l(t) = B(t-t)*p(t-t)$$

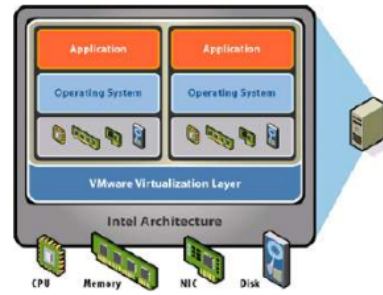


Gambar 4. Pembuangan Paket yang tidak ditandai pada AQM

### 3.4 Virtualisasi

Virtualization <sup>3</sup> atau dikenal dengan virtualisasi merupakan teknik atau cara dalam membuat sesuatu dalam bentuk virtualisasi, namun model yang dihasilkan sesuai dengan kenyataan yang ada [3]. Sebagai contoh windows 10 dapat divirtualisasikan di dalam <sup>3</sup> windows 10 menggunakan *virtual box*. Platform virtualization digambarkan sebagai contoh terdapat sebuah hardware komputer dengan sistem operasinya (*host* dan *control program*) yang menciptakan simulasi dari komputer yang lain dengan sistem operasi miliknya sendiri sebagai *guest software*. Guest software tidak akan selalu dibatasi oleh aplikasi pada pengguna, beberapa *host* dapat mengeksekusi secara lengkap sistem operasinya seperti linux, windows dan sistem operasi lainnya. Model pengoperasian virtualisasi

dengan *vmware* dapat dilihat pada gambar 5 [15].



Gambar 5. Skema Virtualisasi

### 3.5 Router OS

MikroTik RouterOS™ merupakan sistem operasi yang diperuntukkan sebagai *network router*. MikroTik routerOS terdiri dari sistem operasi dan perangkat lunak yang dapat digunakan dari komputer biasa menjadi *router network* mencakup berbagai fitur yang dibuat untuk *ip network* dan jaringan *wireless* [16]. Fitur-fitur *router OS* yaitu *Firewall & Nat*, *Routing*, *Hotspot*, *Point to Point Tunneling Protocol*, *DNS server*, *DHCP server*, *Hotspot*, dan lain-lain. MikroTik routerOS merupakan sistem operasi *Linux base* yang diperuntukkan sebagai *network router* seperti halnya <sup>7</sup> perangkat yang digunakan pada penelitian ini pada gambar 6.



Gambar 6. Penggunaan RouterOS pada penelitian ini.

### 3.6 Jaringan Enterprise Shanti Bhuana

Sistem jaringan kampus Institut Shanti Bhuana dikelola oleh departemen Sistem Informasi dan dibantu oleh beberapa dosen dari prodi Teknologi Informasi. Lingkup jaringan

yang dikelola meliputi *server*, *router backbone*, *router distribusi*, *access point* hingga *maintenance* [16] ngan dimana kondisi jaringan yang dikelola dapat dilihat pada gambar 7.



Gambar 7. Kondisi Jaringan Enterprise ISB

Jaringan kampus ISB menggunakan *vmware* yang terhubung dengan 4 *line* berbeda yang tergabung secara *load balance* pada *RB 1100 Ahx*. *VMWare* terdiri dari beberapa sistem operasi yaitu *ubuntu*, *windows*, dan *windows server*. Pada implementasinya *vmware* terintegrasi dengan laboratorium pemrograman dan jaringan untuk melakukan riset bidang jaringan komputer dan server. Besaran *bandwidth* yang digunakan sebesar 150 *Mbps*, adapun beberapa sistem yang dibuat melalui *vmware* seperti kebutuhan jurnal dengan hosting *journal.shantibhuana.ac.id* di mana server teralokasi pada *vmware* dan beberapa aplikasi sistem informasi lainnya seperti *siakad* yang diakses pada *siakad.shantibhuana.ac.id*, dan *lms.shantibhuana.ac.id* untuk *moodle*.

#### 4. METODE PENELITIAN

Tujuan dari penelitian ini yaitu mengimplementasikan untuk membandingkan hasil dari algoritma *RED*, *SFQ*, dan *AQM* pada jaringan *enterprise* dengan *VMWare EsXI* dan *Router OS*. Metode penelitian yang diterapkan menggunakan pendekatan *NDLC (Network Development Life Cycle)* dengan penjabaran sebagai berikut

##### 1. Analisis (*Analysis*)

Analisis dilakukan dengan mengetahui permasalahan pada penelitian ini yaitu

melakukan perbandingan tiga algoritma antrian untuk jaringan skala besar atau *enterprise*.

##### 2. Desain (*Design*)

Pembuatan desain untuk memetakan kondisi jaringan yang digunakan pada penelitian ini, adapun penelitian ini menggunakan data pada jaringan Institut Shanti Bhuana.

##### 3. Pembuatan Simulasi (*Simulation*)

Simulasi dibangun menggunakan prototipe perangkat seperti *RB 951 UI 2HND*, dan *PC* dengan spek *Core i5* dan *RAM 8GB* untuk membangun rangkaian jaringan dan server sebelum diimplementasikan ke kondisi jaringan real.

##### 4. Implementasi (*Implementation*)

Implementasi dilakukan secara langsung selama 3 bulan yaitu Januari hingga Maret 2021, dengan mengkonfigurasi *vmware* dan *router gateway* utama yaitu *RB 1100 Ahx* untuk menerapkan antrian *RED*, *SFQ*, dan *AQM*.

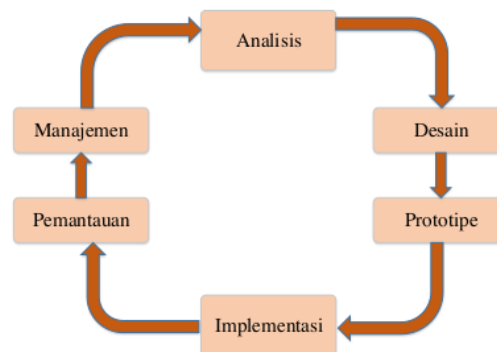
##### 5. Pemantauan (*Monitoring*)

Monitoring hasil ketiga algoritma dilakukan secara berkala selama lima bulan penuh yaitu bulan maret hingga juli 2021 melalui *MRTG* publik pada *gateway*.

##### 6. Manajemen (*Management*)

Setelah dilakukan monitoring dapat diketahui perbedaan dari ketiga algoritma untuk nilai yang paling baik menurut standar *TIPHON* untuk nantinya dapat digunakan sebagai rekomendasi penerapan algoritma antrian pada jaringan dan server. Pada penelitian ini dilakukan enam tahapan selama delapan bulan yang terdiri dari implementasi dan pengujian untuk memperoleh hasil yang optimal agar hasil dapat digunakan sebagai acuan para *network* dan *system administrator*.

Rangkaian model penelitian *NDLC* dibuktikan pada skema gambar 8 berikut.



Gambar 8. Rangkaian Metodologi Penelitian

## 5. ANALISIS DAN PERANCANGAN SISTEM

### 5.1 Analisis Jaringan

Sistem jaringan pada penelitian ini merujuk pada jaringan di Institut Shanti Bhuana yang terdiri dari beberapa jalur antara lain sebagai berikut

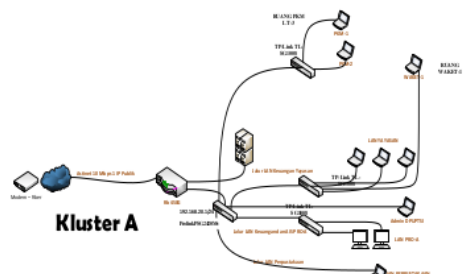
#### 1. Jalur Server

Jalur ini terdiri dari jalur server dengan kapasitas 10 Mbps alokasi dari provider Astinet dan memperoleh ip publik untuk dialokasikan pada vmware yang didistribusikan secara proxy namun tetap dapat akses hingga 250 Mbps dikarenakan load balance untuk port 443 dan 80 dilakukan pemisahan karena keterbatasan ip publik seperti gambar 9.

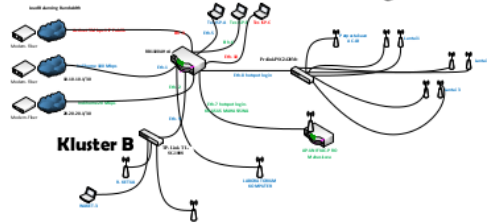
#### 2. Jalur Pengguna

Pada jalur pengguna dibagi menjadi beberapa kategori yaitu dosen dengan kapasitas up to 50 Mbps, mahasiswa kapasitas up to 20 Mbps, lab komputer dengan kapasitas up to 50 Mbps dan CCTV maupun keperluan riset dengan 1 ip publik sebesar 5 Mbps seperti pada gambar 10.

3. Model VMWare menggunakan vmware EsXi dengan kapasitas 2000 TB storage untuk kebutuhan berbagai macam aplikasi seperti sistem informasi akademik pada ubuntu 16.04, proxy, jurnal, moodle, database repositori pada ubuntu 18 maupun storage untuk keperluan riset.



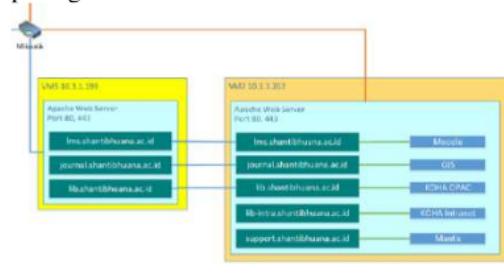
Gambar 9. Kondisi Pemetaan Jaringan Server



Gambar 10. Kondisi Jaringan Pengguna

### 5.2 Arsitektur VMWare

Virtualisasi pada penelitian ini dibagi ke dalam beberapa bagian yaitu VM 1-6 dengan jalur proxy VPS ke hosting internet yang terhubung langsung dengan ip publik seperti pada gambar 11.

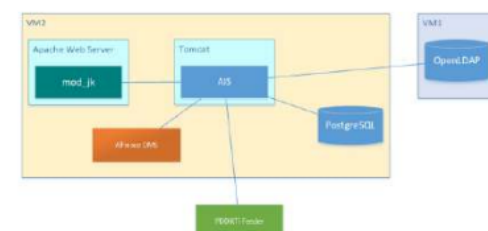


Gambar 11. Model Infrastruktur VMWare EsXI

Pada VMWare dibagi ke dalam beberapa kategori seperti contoh salah satunya yaitu VMWare 1 untuk OpenLDAP menyediakan database akun user terpusat untuk authentication, kemudian VMWare 2 untuk Alfresco DMS yaitu aplikasi berbasis web untuk repositori pengelolaan dokumen-dokumen peraturan, prosedur dan lainnya. Kemudian pada VMWare 6 untuk windows server yang diimplementasikan agar memudahkan pengelolaan server. VM lain pada server ini menggunakan OS Linux, yang cenderung lebih banyak dioperasikan menggunakan command line. Model rancangan untuk pembagian vmware dapat dilihat pada gambar 12 dan pengimplementasian VMWare 1 dapat dilihat pada gambar 13.



Gambar 12. Rancangan alokasi VMWare



Gambar 13. Rancangan alokasi LDAP pada VMWare 1

### 5.3 Algoritma RED

Implementasi algoritma *RED* pada penelitian ini dilakukan menentukan parameter dengan rumus

$$\begin{aligned} &min\_th - \text{minimum threshold} \\ &max\_th - \text{maximum threshold} \\ &avg\_len - \text{average queue length} \\ &-avg\_len = (1-w)*avg\_len + w*sample\_len \end{aligned}$$

Dilakukan untuk mengetahui probabilitas *queue* serta dilakukan implementasi untuk mengetahui packet yang dibuang pada sistem jaringan dengan rumus sebagai berikut

$$\begin{aligned} &If (avg\_len < min\_th) \rightarrow \text{enqueue packet} \\ &If (avg\_len > max\_th) \rightarrow \text{drop packet} \\ &If (avg\_len \geq min\_th \text{ and } avg\_len < max\_th) \\ &\rightarrow \text{enqueue packet with probability } P \end{aligned}$$

### 5.4 Algoritma SFQ

Pada algoritma *SFQ* penelitian ini dilakukan *hashing* untuk menghitung jumlah *hash bin* (*N*), jumlah *level hash* (*L*) dan perkiraan jumlah aliran *TCP* dengan rumus berikut

$$P_h(L, N, B) = \left( \frac{1}{N^{B+1}} \sum_{i=0}^B i \binom{N}{i} \sum_{k=0}^i (-1)^k \binom{i}{k} (i-k)^B \right)^L$$

```

5 hash
p = 1;
for i = 0 to L - 1 do
j = hash(i, packet);
p = min(p, prob[i][j]); /* min drop p seen so
far */ bytes[i][j] = bytes[i][j] +
sizeof(packet);
end for
    
```

$N = C / T_{TCP}(m, RTT_{sys})$  where,  
 $RTT_{sys}$  : Perkiraan waktu rata-rata untuk Round Trip Time

### 5.5 Algoritma AQM

Pada *AQM* dilakukan untuk membagi aliran bandwidth antara kongesti dan antrian seperti hasil rumus berikut

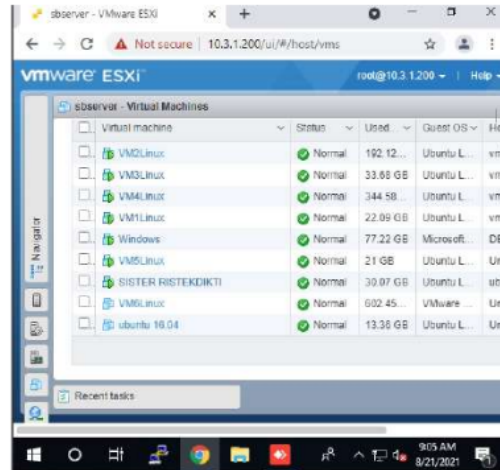
$$\begin{aligned} \Delta &= \alpha d_{avg} \text{ Spare} - \beta \text{ Queue} \\ 0 < \alpha < \frac{\pi}{4\sqrt{2}} \quad \text{and} \quad \beta &= \alpha^2 \sqrt{2} \end{aligned}$$

If  $\Delta > 0 \Rightarrow$  Divide  $\Delta$  equally between flows  
 If  $\Delta < 0 \Rightarrow$  Divide  $\Delta$  between flows

## 6. IMPLEMENTASI DAN PENGUJIAN SISTEM

### 6.1 VMWare

Implementasi *vmware* dilakukan pada *server* dan dikonfigurasi menggunakan *command line interface*. Pada bagian awal dilakukan konfigurasi pembagian disk untuk masing-masing *vmware* seperti pada gambar 14 dan 15.



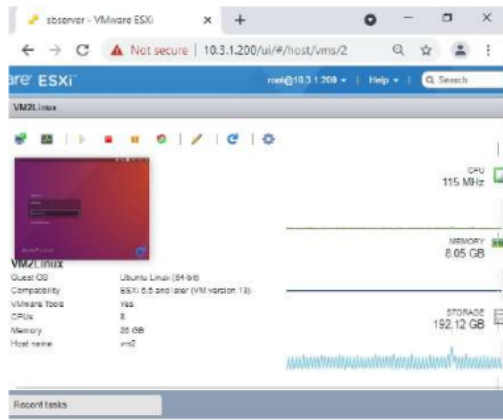
Gambar 14. Pembagian alokasi VMWare



Gambar 15. Model Server yang digunakan VMWare

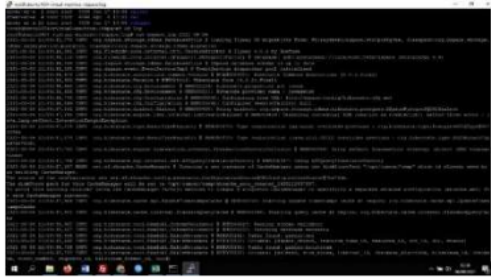
Pada gambar 14 dibuktikan *vmware* dibagi menjadi 6 alokasi masing-masing dengan sistem operasi *Ubuntu* dan *Windows* yang dikonfigurasi *database* dan *web server* aplikasi untuk jurnal, siakad, hingga repositori. Pada gambar 15 dibuktikan penggunaan *CPU load* dan memori pada server yang digunakan sebagai virtualisasi. Kemudian pada dilakukan monitoring penggunaan *resource* untuk masing-masing sistem operasi yang terdapat dalam *VMWare* yang dapat dilihat seperti pada gambar 16. Pada gambar 16 dibuktikan penggunaan *resource CPU Load* dan *memory*

dari masing-masing sistem operasi dalam VMWare.

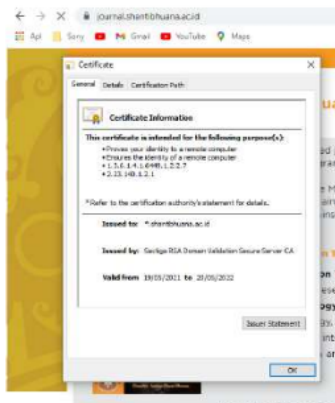


Gambar 16. Monitoring Penggunaan Resource Pada Salah Satu VMWare

Selain konfigurasi masing-masing vmware, dilakukan konfigurasi proxy agar dapat diakses via publik, konfigurasi dilakukan seperti pada gambar 17 dan hasil dari penerapan proxy yang kemudian dilakukan pembaharuan SSL dapat dilihat seperti pada gambar 18 untuk website jurnal kampus.



Gambar 17. Konfigurasi Proxy



Gambar 18. Hasil Proxy dan SSL pada [www.journal.shantibhuana.ac.id](http://www.journal.shantibhuana.ac.id)

## 6.2 Router OS

Konfigurasi algoritma RED, SFQ, dan AQM dilakukan pada RouterOS yang terintegrasi dengan VMWare EsXI. Pada penelitian ini dilakukan permodelan queue untuk server pada algoritma RED dengan sebagian konfigurasi sebagai berikut

```
/queue tree
add limit-at=512k max-limit=728k name=00-ICMP packet-mark=icmp parent=global \
priority=1 queue=red
add comment="118,2 GiB 2-7-20" name=01-DOWNLOAD parent=global priority=2 \
queue=red
add name=02-UPLOAD parent=global priority=2
queue=red
```

Pada antrian SFQ dilakukan sebagian konfigurasi sebagai berikut.

```
/queue tree
add limit-at=512k max-limit=728k name=00-ICMP packet-mark=icmp parent=global \
priority=1 queue=sfq
add comment="118,2 GiB 2-7-20" name=01-DOWNLOAD parent=global priority=2 \
queue=sfq
add name=02-UPLOAD parent=global priority=2
queue=sfq
add bucket-size=2 comment="5,1 GiB 2-7-20"
limit-at=18M max-limit=20M
```

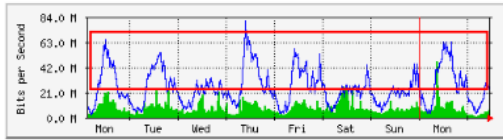
Pada antrian AQM dilakukan sebagian konfigurasi sebagai berikut.

```
/queue tree
add limit-at=512k max-limit=728k name=00-ICMP packet-mark=icmp parent=global \
priority=1 queue=aqm
add comment="118,2 GiB 2-7-20" name=01-DOWNLOAD parent=global priority=2 \
queue=aqm
add name=02-UPLOAD parent=global priority=2
queue=aqm
:foreach a in=$arp do={
:local ip ($a->"172.16.10.1/24");
:local zz ($a->"172.16.10.1/24");
:if ($enabled) do={/queue tree add
name=($zz) target=$ip max-
limit=($150M."/".$150M)
parent=$parentLimit packet-mark=$packetMark
place-before=($parentLimit."-all"); }
```

## 6.3 Analisis dan Pengujian

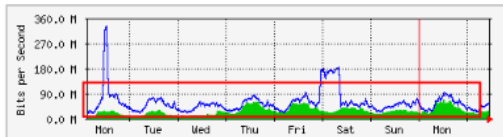
Pengujian dilakukan secara TIPHON meliputi nilai delay, throughput dan packet loss. Hasil dari pengujian secara TIPHON dilakukan analisis pada MRTG selama lima bulan untuk masing masing algoritma antara RED, SFQ, dan AQM secara bergantian setiap minggunya. Pada algoritma RED nilai bandwidth tidak pernah berada di atas 90 Mbps meskipun alokasi bandwidth server dan pengguna pada umumnya dapat hingga 150 Mbps seperti pada gambar 19. Nilai RED pada throughput bandwidth lebih

banyak dilakukan pembuangan packet terutama pada paket *HTTPS* 443.



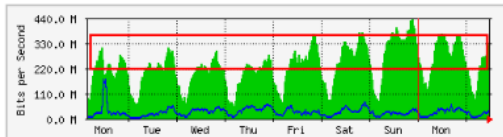
Gambar 19. Pengujian RED dari MRTG

Pada pengujian *SFQ throughput bandwidth* dapat di atas 150 Mbps bahkan di atas alokasi 150 Mbps namun hanya untuk beberapa jam saja, dan cenderung lebih dibawah RED pada hari biasa pada 90 Mbps. Nilai *throughput bandwidth SFQ* cenderung melonjak pada waktu tertentu namun tidak merata dan sering berada di bawah 90 Mbps dikarenakan penggunaan *FIFO* dan pengacakan *packet* dengan *round robin* membuat banyak *packet* cenderung tidak teridentifikasi seperti pada gambar 20.



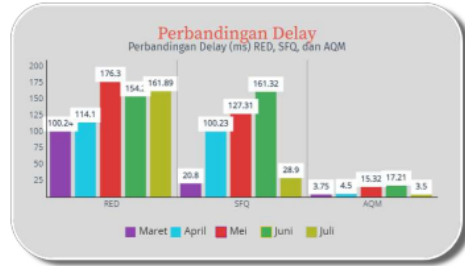
Gambar 20. Pengujian SFQ dari MRTG

Selanjutnya pengujian *AQM* selama 5 bulan diperoleh nilai *MRTG* cenderung lebih besar *throughput bandwidth* yang diperoleh dibandingkan *SFQ* dan *RED* bahkan menembus 250 Mbps sesuai keseluruhan kapasitas *bandwidth* yang ada seperti gambar 21.



Gambar 21. Pengujian AQM dari MRTG

Pengujian *delay* pada algoritma *RED*, *SFQ*, dan *AQM* hasil perbandingan *delay* didapatkan seperti gambar 22. Pada uji *delay* didapatkan nilai *AQM* jauh lebih kecil dibandingkan pada *SFQ* dan *RED* yaitu sebesar 8.856 ms dibandingkan *RED* sebesar 141.346 ms dan *SFQ* sebesar 87.712 ms. Nilai keseluruhan *RED*, *SFQ* dan *AQM* memiliki kapasitas *delay* yang bagus sesuai *TIPHON* namun nilai terkecil dan terbaik berada pada algoritma *AQM* dengan kategori sangat bagus.



Gambar 22. Pengujian Delay

Pada pengujian *throughput* dilakukan *monitoring* selama lima bulan dan diperoleh hasil bahwa prosentasi terbesar berada pada algoritma *AQM* dengan rata-rata 91.4% kemudian algoritma *SFQ* sebesar 81.8% dan algoritma *RED* sebesar 68.4%. Pengujian *throughput* membuktikan pada penelitian ini bahwa nilai uji *AQM* jauh lebih baik sebesar 91.4% dibandingkan metode *SFQ* dan *RED* seperti pada gambar 23.



Gambar 23. Pengujian Throughput

Pengujian *packet loss* dilakukan pada ketiga algoritma dan diperoleh hasil rata-rata *AQM* sebesar 7%, pada *SFQ* sebesar 48% dan *RED* sebesar 65% seperti gambar 24.



Gambar 24. Pengujian Packet Loss

Dari hasil pengujian *packet loss* diperoleh nilai uji terkecil pada algoritma *AQM* sebesar

7% dengan parameter bagus, dan untuk algoritma *SFQ* sebesar 45% serta *RED* sebesar 65% dalam kategori *TIPHON* “buruk atau sangat buruk”.

## 7. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan didapatkan kesimpulan bahwa

1. *VMWare ESXi* dan *Router OS* dapat dikombinasikan dalam sebuah jaringan enterprise.
2. Nilai uji pada algoritma *AQM* jauh lebih baik dibandingkan algoritma *SFQ* dan *RED* dari sisi *delay*, *throughput* dan *packet loss* selama pengujian 5 bulan Maret-Juli 2021.

## 8. SARAN

Diperlukan kombinasi algoritma antrian yang lebih kompleks untuk penelitian lebih lanjut khususnya algoritma *AQM*.

## DAFTAR PUSTAKA

- [1] Greenstein, S. (2020). The Basic Economics of Internet Infrastructure. *Journal of Economic Perspectives*, 34(No.2), 192–214.
- [2] Handayani, P. W., Yazid, S., Bressan, S., and Sampe, A. F. 2020. “Information and Communication Technology Recommendations for the Further Development of a Robust National Electronic Health Strategy for Epidemics and Pandemics,” *Jurnal Sistem Informasi (Journal of Information System)* (16:2), pp. 31–42.
- [3] Sugianto, M. V. (2018). *Panduan Dasar VMWare VSphere 6.7* (M. R. Utomo (ed.); 1st ed.). Excellent Publishing.
- [4] Towidjojo, R. (2016). *Mikrotik Kungfu Kitab 3*. Jasakom.
- [5] Nomnga, P., Scott, M. S., & Nyambi, P. B. (2014). A Technical Cost Effective Network-Domain Hosting through Virtualization: a VMware ESXi and vSphere Client Approach. *International Journal of Computer Applications* (0975 – 8887), 91(10), 39–47.
- [6] Jaradat, Yousef & Jannoud, Ismael & Masoud, Mohammad & Abu Elhaija, Wejdan. (2017). A Roadmap for SDN-based Network Management: Al-Zaytoonah University of Jordan Case. 10.1109/STA.2017.8314900.
- [7] Darsono, T. A. (2017). *Analisis Kinerja Jaringan Penghubung Antar Gedung Menggunakan WiFi (Studi Kasus : Universitas Musamus Merauke)*. Universitas Gadjah Mada Yogyakarta.
- [8] Muhlis Muhallim. (2017). *Pengembangan Jaringan Komputer Universitas Andi Djemma Palopo Berdasarkan Perbandingan Jaringan Protokol Routing Statik dan OSPFv2*. Universitas Gadjah Mada Yogyakarta.
- [9] Muhammad Diponegoro. (2015). *Analisis Quality of Service (QoS) pada Jaringan Internet Dengan Metode Fixed Daily Measurement Interval (FDMI) dan non (FDMI) Studi Kasus : UGM-Hotspot Pascasarjana Ilmu Komputer Universitas Gadjah Mada*. Universitas Gadjah Mada Yogyakarta.
- [10] Paulov, I. (2015). *Routing in a Virtualised Environment with RouterOS* (Issue November). Helsinki Metropolia University of Applied Sciences.
- [11] Domańska J., Domański A. (2009) Adaptive RED in AQM. In: Kwiecień A., Gaj P., Stera P. (eds) *Computer Networks. CN 2009. Communications in Computer and Information Science*, vol 39. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-642-02671-3\\_21](https://doi.org/10.1007/978-3-642-02671-3_21)
- [12] MikroTik. (2019). *Random Early Drop*. <https://help.mikrotik.com/docs/display/ROS/Queues#Queues-RED>
- [13] U. R. Pujeri, V. Palaniswamy, P. Ramanathan, and R. Pujeri, “Comparative analysis and comparison of various AQM algorithm for high speed,” *Indian J. Sci. Technol.*, vol. 8, no. 35, 2015
- [14] Bhatnagar, S., Patel, A., & Karmeshu. (2017). A stochastic approximation approach to active queue management. *Telecommun Syst*, 1. <https://doi.org/DOI 10.1007/s11235-017-0377-1>
- [15] Nomnga, P., Scott, M S., Nyambi and B P. Article: A Technical Cost Effective Network-Domain Hosting through Virtualization: a VMware ESXi and vSphere Client Approach. *International Journal of Computer Applications* 91(10):39-47, April 2014.
- [16] Kartadie, R. (2016). Performance Test of Openflow Agent on Openflow Software-Based Mikrotik RB750 Switch. *Scientific Journal of Informatics*, 3(2), 217-228. doi:<https://doi.org/10.15294/sji.v3i2.7987>

# PERBANDINGAN ALGORITMA RED, SFQ, DAN AQM PADA JARINGAN ENTERPRISE DENGAN VMWARE ESXI DAN ROUTER OS

## ORIGINALITY REPORT

9%

SIMILARITY INDEX

8%

INTERNET SOURCES

5%

PUBLICATIONS

3%

STUDENT PAPERS

## PRIMARY SOURCES

1	<b>Carla Prastika Bandhaso, Yudi Sutanto. "Analisis Dan Implementasi Metode Random Early Detection (Red) Pada Jaringan TCP/IP", Respati, 2022</b> Publication	2%
2	<b>journal.universitasmulia.ac.id</b> Internet Source	1%
3	<b>jurnal.una.ac.id</b> Internet Source	1%
4	<b>akbarmukahfi.blogspot.com</b> Internet Source	1%
5	<b>web.wpi.edu</b> Internet Source	1%
6	<b>Muhammad Diponegoro, Wendy Yuniarto, Rusman Rusman, Sarah Bibi. "Optimasi Kinerja Jaringan Wireless Menggunakan Repeater Berbasis Open DD-WRT Dengan Metode Drive Test Studi Kasus Pada Jaringan Internet Jurusan Teknik Elektro Politeknik Negeri Pontianak.", Jurnal ELIT, 2022</b> Publication	1%
7	<b>journal.uad.ac.id</b> Internet Source	<1%
8	<b>123dok.com</b> Internet Source	<1%

repositori.usu.ac.id



9	Internet Source	<1 %
10	es.scribd.com Internet Source	<1 %
11	Submitted to Fakultas Ekonomi dan Bisnis Universitas Gadjah Mada Student Paper	<1 %
12	www.theseus.fi Internet Source	<1 %
13	database.rizaarifudin.com Internet Source	<1 %
14	eprints.akakom.ac.id Internet Source	<1 %
15	www.springerprofessional.de Internet Source	<1 %
16	fr.scribd.com Internet Source	<1 %
17	repository.its.ac.id Internet Source	<1 %
18	www.slideshare.net Internet Source	<1 %

Exclude quotes Off

Exclude matches Off

Exclude bibliography Off