

IMPLEMENTASI MANAJEMEN *BANDWIDTH* JARINGAN MENGGUNAKAN METODE *QUEUE TREE* PADA SMA NEGERI 1 KEFAMENANU

Nini Trifena uky^{1*}, Yoseph P. K. Kelen², Kristoforus Fallo,³ Ernes J. Blegur⁴

Fakultas Pertanian Sains dan Kesehatan, Program Studi Teknologi Informasi, Universitas Timor, Kefamenanu, Timor Tengah Utara, Nusa Tenggara Timur, Indonesia

Keywords:

3-5 keyword;

Parameter *QoS*: *Queue*

Tree; Manajemen *Bandwidth*;

Correspondent Email:

Ninitrifenuky@gmail.com



Copyright © [JITET](http://www.jitet.org) (Jurnal Informatika dan Teknik Elektro Terapan). This article is an open access article distributed under terms and conditions of the Creative Commons Attribution (CC BY NC)

Abstrak. Teknologi jaringan komputer saat ini telah memasuki hampir keseluruhan segi kehidupan. Sangat sulit pada saat ini menemukan bidang kegiatan yang belum tersentuh oleh teknologi jaringan komputer. Hal ini dapat dilihat dari pengguna fasilitas jaringan komputer, baik untuk keperluan suatu instansi maupun untuk keperluan pribadi. Teknologi jaringan komputer telah menjadi salah satu hasil penting untuk diterapkan dalam era globalisasi dan komputerisasi. Jaringan komputer merupakan satuan komputer yang saling berhubungan satu sama lain dengan menggunakan protokol komunikasi yang melalui media komunikasi agar dapat saling berbagi informasi. Jaringan komputer sendiri bukanlah hal yang baru karena pada setiap instansi telah menggunakan dan memanfaatkan jaringan komputer. [1] Manajemen *bandwidth* merupakan salah satu yang penting dalam jaringan komputer, dimana manajemen *bandwidth* berfungsi sebagai pengatur jaringan *bandwidth* yang setiap pengguna jaringan tersebut memperoleh *bandwidth* yang merata walaupun dengan jumlah pengguna yang banyak. Manajemen *bandwidth* juga merupakan ukuran, lebar dan luas saluran data yang melewati sebuah koneksi yang dikirim melalui jaringan internet. Manajemen *bandwidth* adalah suatu proses pengontrol paket yang dikirim melalui jaringan agar paket yang dikirim tidak terjadi kegagalan diakibatkan koneksi jaringan yang buruk atau kemacetan dalam jaringan. Untuk mengatur manajemen *bandwidth* dibutuhkan metode yang tepat agar permasalahan bisa teratasi sehingga akses jaringan menjadi optimal.

Abstract Computer network technology has now entered almost every aspect of life. It is very difficult today to find a field of activity that has not been touched by computer network technology. This can be seen from the use of computer network facilities, both for institutional purposes and personal needs. Computer network technology has become one of the important outcomes to be applied in the era of globalization and computerization. A computer network is a unit of computers interconnected with one another using communication protocols through communication media so that they can share information. Computer networks themselves are not something new, since every institution has already used and benefited from them (Heryanto, 2019). Bandwidth management is one of the important aspects in computer networks, where bandwidth management functions as a regulator to ensure that every network user receives fair bandwidth distribution even with a large number of users. Bandwidth management is also defined as the measurement, width, and capacity of the data channel passing through a connection that is transmitted over the internet network. Bandwidth management is a process of controlling packets transmitted through the network so that packet delivery does not fail due to poor connections or network congestion.

1. PENDAHULUAN

Teknologi jaringan komputer saat ini telah memasuki hampir keseluruhan segi

kehidupan. Sangat sulit pada saat ini menemukan bidang kegiatan yang belum tersentuh oleh teknologi jaringan komputer. Hal ini dapat dilihat dari pengguna fasilitas jaringan komputer, baik untuk keperluan suatu instansi maupun untuk keperluan pribadi. Teknologi jaringan komputer telah menjadi salah satu hasil penting untuk diterapkan dalam era globalisasi dan komputerisasi. Jaringan komputer merupakan satuan komputer yang saling berhubungan satu sama lain dengan menggunakan protokol komunikasi yang melalui media komunikasi agar dapat saling berbagi informasi. Jaringan komputer sendiri bukanlah hal yang baru karena pada setiap instansi telah menggunakan dan memanfaatkan jaringan komputer.[1]

Manajemen *bandwidth* merupakan salah satu yang penting dalam jaringan komputer, dimana manajemen *bandwidth* berfungsi sebagai pengatur jaringan *bandwidth* yang setiap pengguna jaringan tersebut memperoleh *bandwidth* yang merata walaupun dengan jumlah pengguna yang banyak. Tujuan Implementasi Manajemen *Bandwidth* Jaringan Menggunakan Metode *Queue Tree* Pada SMA Negeri 1 Kefamenanu yaitu membantu Administrator melakukan manajemen *bandwidth* jaringan untuk mengoptimalkan pengguna jaringan, serta Untuk menerapkan metode *Queue Tree* pada SMA Negeri 1 Kefamenanu dan memastikan kinerja yang optimal dan adil bagi semua pengguna.[2]

2. TINJAUAN PUSTAKA

2.1 Jaringan Komputer

Menurut [1] Jaringan Komputer sekumpulan dua atau lebih komputer yang saling terhuung dengan satu sama lain untuk melakukan komunikasi data. *File* dan informasi yang menggunakan *protocol* komunikasi bisa melalui (kabel dan nirkabel), sehingga komputer-komputer bisa berbagi informasi, data program secara bersamaan. Sedangkan menurut

Jaringan komputer yang memungkinkan antar komputer untuk saling berkomunikasi dengan bertukar data, perangkat lunak komputer sampai dengan aplikasi dengan adanya jaringan komputer ini menjadikan penggunaanya bisa berinteraksi atau berkomunikasi dengan pengguna lain. Adapun jenis- jenis jaringan komputer berdasarkan luas areanya atau luas geografisnya

2.2 Manajemen *Bandwidth*

Management *bandwidth* adalah sebuah cara untuk mengatur dan mengontrol *bandwidth* sehingga tidak terjadi kinerja yang buruk pada jaringan komputer. kemudian pendapat dari Management *Bandwidth* adalah suatu cara atau kegiatan untuk mengontrol pengiriman kecepatan paket dari suatu tempat ke tempat lain sesuai dengan kapasitas maksimal *bandwidth* yang telah ditetapkan dalam suatu jaringan internet agar jaringan tersebut menjadi efisien.[3]

2.3 Metode *Queue Tree*

Metode *Queue Tree* adalah proses pelimitan *bandwidth* dengan sangat komplek dimana pelimitan akan digabungkan berdasarkan *protocol*, *port*, dan *ip addrees*. [4] Untuk melakukan pelimitan *bandwidth* pada *queue tree* harus mengaktifkan *fitur mangle* pada mikrotik bertujuan untuk supaya paket data tersebut dikenali oleh *queue tree*. tidak berpengaruh terhadap hasil manajemen *bandwidth* yang diinginkan dikarenakan pada saat konfigurasi tidak bisa memindahkan urutan dari konfigurasi *queue* yang sudah ada. Dengan diprosesnya paket secara stimulant maka penggunaan pada *queue tree* jelas akan lebih mempercepat proses paket.[5]

2.4 Parameter *QOS (Quality of service)*

1. *Bandwidth*

Bandwidth adalah jumlah besaran data yang dapat ditransmisi dari suatu tempat ke tempat lain dalam kurun waktu tertentu. *Bandwidth* merupakan ukuran yang sangat penting dalam jaringan, namun konsep ini

mempunyai kekurangan yang sangat penting dalam jaringan, namun konsep ini mempunyai kekurangan atau keterbatasan karena hukum fisika dan keterbatasan teknologi yang memberikan batasan pada panjang, kecepatan maksimum atau perlakuan khusus terhadap media yang dapat digunakan (Standar yang digunakan untuk mengukur *Bandwith* adalah standar yang dikeluarkan oleh FCC (*Federal Communications Comission*) bahwa untuk pelajar *bandwith* yang dianjurkan untuk unduh adalah 5-25Mbps (FCC, 2022).[6]

2. Throughput

Throughput adalah kesanggupan sebenarnya suatu jaringan dalam mengirimkan data .[7]*Throughtput* dengan kata

lain kecepatan dalam transmisi data ang efektif dan diukur dengan satuan bps. *Throughput* dengan kata lain jumlah total paket yang berhasil yang dikirim selama periode waktu tertentu dibagi dengan durasi kurun waktu transmisi.[6]

3. Jitter

Jitter adalah variasi atau perubahan *latency* dari *delay* atau variasi waktu kedatangan paket *Jitter* juga dapat dikatakan sebagai sebuah gangguan saat pentransmisian data yang disebabkan oleh perubahan sinyal karena referensi posisi waktu. *Jitter* ini juga bisa dapat mengakibatkan hilangnya data atau paket, terutama pada pengiriman dengan menggunakan kecepatan tinggi. Banyak hal yang dapat menyebabkan *jitter*[6]

4. Packet loss

Packet loss merupakan suatu parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang, [8]dapat terjadi karena *collision* dan *congsestion* pada jaringan *Packet loss* merupakan kegagalan paket data mencapai tujuan.[6]

5. Delay (latency)

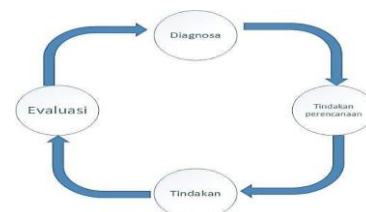
Delay adalah tenggang waktu yang dibutuhkan mulai mengirim data sampai data diterima,[9][10][11] *delay* disebabkan

oleh penghalang, jarak,[12][13][14][15] sehingga nilai *delay* yang dihasilkan akan semakin besar[6]

3. METODE PENELITIAN

3.1 Tipe Penelitian

Dalam penelitian ini, penulis menggunakan jenis penelitian *Action Research*. Metode *Action Research* adalah penelitian tindakan yang menitik beratkan pada kegiatan peneliti untuk menemukan masalah yang diteliti.[5] metode penelitian ini biasa digunakan untuk ada proses



menguji serta ,mengembangkan, dan menciptakan sebuah tindakan baru agar bisa diterapkan

Gambar 3.1 Siklus Metode *Action Research*

3.2 Teknik Pengumpulan Data

Dalam penelitian yang dilakukan ini, penulis menggunakan teknik pengumpulan data yaitu:

1) Studi pustaka

Teknik pengumpulan data awal yang dilakukan oleh peneliti adalah mencari referensi dari jurnal, artikel dari berbagai sumber pendukung yang dapat dijadikan bahan pedoman dalam penulis menyelesaikan penelitian ini

2) Studi Literatur

Mempelajari penelitian terdahulu untuk menemukan teori dasar yang bisa dijadikan pedoman dalam untuk penelitian yang akan dilaksanakan.

3) Wawancara

Untuk mendapatkan hasil yang lebih maksimal dan beragam, penulis juga melakukan interaksi langsung melalui tanya jawab kepada dengan administrator

jaringan di SMA Negeri 1 Kefamenanu terutama tentang masalah-masalah yang berkaitan dengan Manajemen *Bandwith* yang disediakan oleh sekolah sebagai salah satu penunjang pembelajaran di sekolah.

4. HASIL DAN PEMBAHASAN

4.1 Konfigurasi Dasar dan Lingkungan Simulasi

4.1.1 Lingkungan VirtualBox dan Mikrotik

1. Konfigurasi dasar Mikrotik RouterOs pada VirtualBox

Pada tahapan awal konfigurasi, peneliti menggunakan perangkat lunak *VirtualBox* untuk menjalankan *Mikrotik RouterOs* secara *virtual*. Pengaturan dilakukan melalui tab *Network*, dengan menggunakan 2 *adapter* jaringan. Pada *adapter* pertama dan *adapter* kedua, dipilih opsi *Bridged Adapter* agar mesin *virtual* (VM) dapat terhubung langsung dengan jaringan fisik yang digunakan oleh komputer *host*.

4.2.1 Konfigurasi Mikrotik RouterOs pada Winbox

Peneliti melakukan konfigurasi dasar pada *Mikrotik RouterOs* menggunakan aplikasi *Winbox*. Langkah-langkah konfigurasi dasar dilakukan untuk memastikan bahwa *Mikrotik RouterOs* dapat terhubung ke jaringan fisik serta dapat berinteraksi dengan perangkat klien seperti laptop atau *smartphone* secara langsung.

1. Penetapan IP Address

Pada tahap ini, peneliti melakukan konfigurasi alamat *IP* (*Internet Protocol*) pada masing-masing *interface router Mikrotik* menggunakan aplikasi *winbox*. Pengaturan dilakukan melalui menu *IP Address* dengan tujuan untuk memberikan identitas jaringan pada setiap *interface*. Berdasarkan Gambar konfigurasi *interface ethernet 1* yang berfungsi sebagai jalur internet (*WAN*) dikonfigurasi dengan alamat *IP* 192.168.88.1/24, sedangkan *interface ethernet 2* yang digunakan untuk jaringan *WIFI (LAN)* dikonfigurasi dengan

alamat *IP Address* 192.168.88.1/24. Kedua alamat *IP* tersebut berada dalam jaringan 192.168.88.1/24, yang menyediakan rentang alamat cukup besar, yaitu hingga 2046 host sehingga sesuai dengan kebutuhan jaringan sekolah yang memiliki banyak pengguna.

2. Pembuatan DHCP Server

Setelah melakukan pengaturan alamat *IP Address Ethernet 1* dan *Ethernet 2*, langkah selanjutnya adalah membuat *DHCP Server* pada *interface Ethernet 2* *WiFi* menggunakan aplikasi *winbox*, *DHCP Server* ini berfungsi untuk membagikan alamat *IP*, *gateway*, dan *DNS* secara otomatis kepada setiap perangkat yang terhubung ke jaringan *WiFi*, sehingga pengguna tidak perlu mengatur alamat *IP* secara manual di perangkat mereka. Konfigurasi dilakukan melalui menu *IP*, *DHCP Server*, lalu memilih tab *DHCP setup* dan menentukan *interface ether 2* *WiFi* sebagai jalur yang akan digunakan. Setelah dikonfigurasi, sistem secara otomatis membuat *DHCP Server* dengan nama *dhcp1*, menggunakan alamat *IP* dari *dhcp_pool*, dan menetapkan waktu peminjaman alamat *IP* selama 10 menit.

3. Konfigurasi NAT

Langkah selanjutnya adalah mengaktifkan fitur *Network Address Translation (NAT)*. Tujuan dari konfigurasi ini adalah untuk memungkinkan perangkat klien yang berada di jaringan lokal *interface WiFi (ethernet 2)* dapat mengakses jaringan internet melalui koneksi publik yang tersedia di *interface internet* atau *WAN (ethernet 1)*. Konfigurasi *NAT* dilakukan dengan menambahkan aturan pada menu *IP*, *Firewall*, *NAT* di *winbox*, menggunakan *general chain*, *srcnat* dan *out-interface ethernet 1* sebagai jalur keluar ke internet, serta memilih *action masquerade* untuk menerjemahkan *IP* lokal menjadi *IP* publik

4. Uji Koneksi

Setelah seluruh dasar dilakukan,

peneliti melakukan beberapa langkah pengujian koneksi :

1. Uji koneksi ke internet (*Ping*)

Pada *New Termial Mikrotik*, perintah *Ping* 8.8.8.8 digunakan untuk mengecek apakah *Mikrotik RouterOS* sudah terhubung ke internet. Jika *ping* berjalan normal tanpa ada *request time out* berarti konfigurasi berhasil.

2. *DHCP Server – Leases*

Setelah *DHCP Server*, langkah selanjutnya adalah memeriksa daftar perangkat yang berhasil terhubung dan memperoleh alamat *IP* secara otomatis. Pemeriksaan ini dilakukan melalui tab *Leases* pada menu *DHCP Server* di aplikasi winbox. Berdasarkan gambar ini menampilkan informasi seperti alamat *IP*, alamat *MAC*, nama perangkat (*host name*), serta status koneksi. status *bound* menunjukkan bahwa perangkat telah menerima dan sedang menggunakan alamat *IP* dari *DHCP Server*, sedangkan *offered* menunjukkan bahwa alamat *IP* telah ditawarkan, namun belum sepenuhnya digunakan. Data ini membuktikan bahwa *DHCP Server* telah berfungsi dengan baik dalam mendistribusikan alamat *IP* kepada perangkat klien secara otomatis.

4.2 Analisis Data *QOS* Menggunakan *Phyton* di *Google Colab*

Analisis parameter *Quality of Service (QoS)* dalam penelitian ini dilakukan menggunakan bahasa pemrograman *Phyton* melalui *plafrom Google Colab*. Data lalu lintas jaringan diperoleh dari hasil tangkapan (*capture*) menggunakan *Wireshark*, kemudian dikonversi terlebih dahulu ke dalam format file CSV agar dapat diolah secara sistematis. Proses analisis mencakup perhitungan empat parameter *QoS* yaitu *throughput*, *packet loss*, *delay*, dan *jitter*. Keempat parameter ini digunakan untuk mengevaluasi kinerja jaringan sebelum dan sesudah penerapan metode *Queue Tree* dalam manajemen *bandwith*.

1) *Throughput*

Troughput adalah jumlah data yang berhasil dikirimkan dari satu titik ke titik lain dalam jaringan dalam satu satuan waktu, biasanya dinyatakan dalam *Bit per second (bps)*, *Kilobit per second (kbps)*, atau *megabit per second (Mbps)*.

Persamaan *Throghput*

$$Throughput = \frac{\text{paket yang diterima (kb)}}{\text{waktu pengiriman (s)}} \dots \dots \dots (1)$$

Packet loss adalah jumlah paket data yang hilang atau gagal terkirim selama proses pengiriman dari pengirim ke penerima dalam jaringan. Semakin kecil *packet loss* makin kualitas jaringan di nilai semakin bagus. Untuk menghitung nilai *packet loss*, langkah yang dilakukan adalah mengurangi jumlah paket yang dikirim dengan jumlah paket yang diterima, dan terakhir dikalikan dengan 100%.

Persamaan *packet loss*

$$packet\ loss = \frac{(\text{paket dikirim} - \text{paket diterima})}{\text{paket yang dikirim}} \times 100 \dots \dots (2)$$

3) *Delay*

Delay merupakan waktu yang dibutuhkan oleh suatu paket data yang berpindah dari perangkat pengirim ke perangkat penerima dalam jaringan. Untuk menghitung *delay*, dilakukan dengan cara mengambil selisih antara waktu kedatangan paket dan waktu pengirimnya. Hasil perhitungan kemudian dikalikan dengan 1000 untuk mengubah satuan dari detik (*s*) menjadi milidetik (*ml*), sesuai dengan standar satuan yang digunakan dalam penelitian ini. Semakin kecil nilai *delay*, maka menunjukkan bahwa kualitas jaringan semakin bagus.

Persamaan *delay*:

$$delay = \frac{\text{total delay}}{\text{jumlah paket}} \dots \dots \dots (3)$$

variasi waktu (*delay*) dalam pengiriman paket data, yang menunjukkan ketidakberaturan waktu kedatangan paket dari pengirim ke penerima. Nilai *jitter* diperoleh dari selisih antara *delay* paket yang diterima secara berurutan. Hasil perhitungan *jitter* dikalikan dengan 1000 untuk mengubah satuan dari detik(s) ke milidetik(*ml*), karena standar satuan yang digunakan dalam penelitian ini adalah *milisecond*. Maka semakin baik kualitas jaringan, karena menunjukkan kestabilan waktu pengiriman data.

Persamaan *jitter* :

$$Jitter = \frac{\text{Total variasi delay}}{\text{Total paket diterima}} \dots \dots (4)$$

4.3 Kondisi Jaringan Sebelum Penerapan Queue Tree

Sebelum metode *Queue Tree* diterapkan, kualitas layanan jaringan WiFi di SMA Negeri 1 Kefamenanu tergolong kurang stabil, khususnya pada saat jam sibuk. Berdasarkan hasil observasi dan wawancara dengan administrator jaringan, dan guru Tik. Diketahui bahwa sistem jaringan belum menerapkan metode manajemen *bandwith* berbasis prioritas layanan, *Bandwith* masih dibagi secara umum ke seluruh pengguna tanpa mempertimbangkan jenis layanan yang digunakan, sehingga ketika banyak perangkat terhubung secara bersamaan Koneksi menjadi lambat dan tidak merata. Kondisi ini menjadi dasar dilakukannya pengujian terhadap parameter *Quality of Service (QoS)* guna mengetahui performa teknis jaringan sebelum dan sesudah penerapan metode *Queue Tree*. Sebelum proses analisis *QoS* dilakukan, peneliti terlebih dahulu mengumpulkan data mentah jaringan menggunakan aplikasi *Wireshark*. Data tersebut menggunakan hasil tangkapan langsung terhadap lalu lintas jaringan WiFi SMA Negeri 1 Kefamenanu. Pengambilan data dilakukan dalam dua kondisi berbeda, yaitu saat jam sibuk (banyak pengguna aktif) dan saat

jam tidak sibuk (sedikit pengguna aktif), dengan tujuan untuk membandingkan kondisi *trafik* jaringan pada beban tinggi dan rendah sebelum penerapan metode manajemen *Bandwith Queue Tree*.

4.4 Hasil pengukuran QoS sebelum metode Queue Tree

Pengukuran kualitas layanan jaringan dilakukan pada jaringan WiFi sekolah sebelum penerapan metode *Queue Tree*. Pengujian dilakukan dalam dua kondisi yang berbeda, yaitu saat jam sibuk dan saat jam tidak sibuk, yang keduanya dilakukan pada hari yang sama untuk menjaga konsistensi pengujian. Selama proses pengukuran, peneliti menjalankan dua jenis aktivitas secara bersamaan, yaitu *browsing* dan *streaming youtube*, dengan durasi pengujian selama 5 menit. Data lalu lintas jaringan dikumpulkan melalui aplikasi *Wireshark*, kemudian dianalisis lebih lanjut menggunakan *python jupyter* melalui *platform google collab*. Analisis dilakukan terhadap empat parameter utama *Quality of Service (QoS)* yaitu *throughput*, *packet loss*, *delay* dan *jitter*.

1. Jam Sibuk

Pada tahap awal, penelitian dilakukan saat jam sibuk sebelum penerapan metode *Queue Tree*. Tabel 4.1 menampilkan hasil perhitungan parameter *QoS*, yaitu *throughput*, *packet loss*, *delay*, dan *jitter*.

Parameter	Nilai Rata – Rata
<i>Throughput (kbps)</i>	358 kbps
<i>Packet loss (0%)</i>	0 %
<i>Delay (ms)</i>	11,98 ms
<i>Jitter (ms)</i>	12,08 ms

Hasil pengukuran menunjukkan bahwa nilai *throughput* yang diperoleh pada kondisi ini adalah sebesar 358 kbps. *Throughput* menunjukkan seberapa besar data yang berhasil dikirim dari sumber ke tujuan dalam satu detik. Nilai ini tergolong rendah, yang menandakan bahwa saat jam sibuk, kapasitas jaringan tidak dimanfaatkan secara optimal. Sementara

itu, parameter *packet loss* menunjukkan presentase data yang hilang selama proses pengiriman. Hasil pengujian menunjukkan bahwa nilai *packet loss* adalah 0 % yang berarti seluruh paket data berhasil dikirim tanpa kehilangan, Nilai *delay* yang tercatat adalah sebesar 11,98 milidetik(*ms*). *Delay* atau waktu tunda menggambarkan lainnya waktu yang dibutuhkan sebuah paket untuk berpindah dari pengirim ke penerima, Adapun nilai *jitter* yang tercatat adalah 12,08 *ms jitter* menunjukkan seberapa besar variasi waktu antar kedatangan paket data. Nilai ini mencerminkan bahwa waktu antar pengiriman paket tidak konsisten dan mengalami fluktuasi yang signifikan. Dalam praktinya, *jitter* yang tinggi dapat menurunkan kualitas layanan berbasis waktu nyata (*real-time*), seperti suara yang tersendat dalam panggilan video atau gangguan dalam pemutaran video *streaming*.

2. Jam Tidak Sibuk

Selanjutnya pengukuran *QOS* dilakukan kembali pada saat jumlah pengguna jaringan lebih sedikit (jam tidak sibuk) sebelum penerapan metode *Queue Tree* Tabel 4.2 menampilkan hasil perhitungan parameter *QOS* yaitu *throughput*, *packet loss*, *dellay*, dan *jitter*.

Parameter	Nilai Rata – Rata
<i>Troughput (kbps)</i>	1,273 <i>kbps</i>
<i>Packet Loss (%)</i>	0 %
<i>Dellay (ms)</i>	4,88 <i>ms</i>
<i>Jitter (ms)</i>	4,88 <i>ms</i>

Nilai *throughput* tercatat sebesar 1,27*kbps*, yang menunjukkan jumlah data yang berhasil dikirimkan tiap detik saat jaringan dalam keadaan tidak padat. Angka ini menunjukkan peningkatan signifikan dibandingkan kondisi jam sibuk sebelumnya yang hanya mencapai 358 *kbps*. Nilai *throughput* yang tinggi ini menunjukkan bahwa kapasitas *bandwith* jaringan dapat dimanfaatkan lebih optimal saat sedikit pengguna yang aktif. Sementara itu nilai *packet loss* pada

kondisi ini tetap tercatat 0 % yang berarti tidak ada paket data yang hilang selama proses pengiriman. Konsistensi ini menunjukkan bahwa keandalan jaringan tetap tinggi, meskipun *throughput* meningkat tidak adanya *packet loss* menjamin bahwa seluruh data yang dikirim diterima utuh oleh penerima, *Dellay* pada jam tidak sibuk tercatat 4,88 milidetik, yang menunjukkan waktu tunda pengiriman data dari pengirim ke penerima relatif cepat. Nilai ini menunjukkan adanya penurunan yang signifikan dibandingkan dengan kondisi saat responsif dan efisien, sehingga akses ke halaman web menjadi lebih cepat, Adapun *jitter* atau variasi waktu antar kedatangan paket tercatat sebesar 4,88 *ms*, lebih rendah dibandingkan nilai *jitter* saat jam, sibuk yang mencapai 12,08 *ms* penurunan ini menunjukkan peningkatan kestabilan jaringan, dimana waktu antar kedatangan paket menjadi lebih konsisten. *Jitter* yang rendah sangat penting untuk layanan waktu nyata (*real time*) seperti *video conference* atau *streaming youtube*, karena menghindari jeda tiba –tiba penurunan kualitas video, atau audio tersendat.

4.5 Penerapan Metode *Queue Tree* Pada *Winbox*

Metode *Queue Tree* diterapkan untuk mengelola total *Bandwith* sebesar 80 *Mbps*. Adapun langkah – langkah konfigurasi di *Mikrotik* melalui *winbox* adalah sebagai berikut:

1. Konfigurasi *Mangle Rule*

Dalam penelitian ini konfigurasi *mangle* digunakan untuk menandai koneksi dan paket data berdasarkan jenis *trafik*, khususnya *trafik streaming* dan *browsing*. Penandaan ini bertujuan agar sistem dapat mengelompokkan *trafik* sesuai jenis layanannya dan mengalokasikan *bandwith* secara proposional melalui konfigurasi *Queue Tree*. *Trafik browsing* dibagi menjadi dua *download* ditandai pada *chain postrouting* dengan *packet*

markk browsing packet download dan *upload* ditandai pada *chain prerouting* dengan *packet mark browsing packet upload*. Hal ini memungkinkan sistem untuk mengatur alokasi *bandwith* berdasarkan prioritas layanan, sehingga manajemen *bandwith* dapat berjalan secara optimal dan mendukung peningkatan kualitas layanan jaringan *Quality of Service*.

2. Konfigurasi Metode *Queue Tree* di *Winbox*

Setelah konfigurasi *Mangle Rule*, langkah selanjutnya adalah konfigurasi *Queue Tree* dengan metode *Queue Tree* pada perangkat *Mikrotik*. Tujuan dari konfigurasi ini adalah untuk mengelola alokasi *bandwith* secara hierarkis, berdasarkan arus lalu lintas *upload* dan *download* serta jenis layanan yang digunakan, seperti *browsing* dan *streaming Youtube*.

Tujuan dari konfigurasi ini adalah agar pembagian *Bandwith* dapat dilakukan secara efisien dan adil, dengan layanan seperti *browsing* (prioritas tinggi) mendapatkan *bandwith* yang lebih stabil dan terjamin, terutama pada kondisi jaringan yang padat. Sebaiknya, layanan dengan prioritas lebih rendah seperti *streaming youtube* tetap mendapatkan *bandwith*, namun dalam batasan yang lebih kecil.

4.6 Pengujian Metode *Queue Tree*

Setelah melakukan konfigurasi, Peneliti melakukan pengujian untuk memastikan bahwa konfigurasi yang telah diterapkan dapat berfungsi dengan baik. Pengujian dilakukan dengan menjalankan dua jenis aktivitas jaringan, yaitu *download* dan *upload*, yang bertujuan untuk mengamati apakah aturan penandaan *trafik* (melalui *mangle Rule*) dan pembagian *bandwith* (melalui *queue tree*) bekerja sesuai dengan konfigurasi yang telah dibuat. Gambar berikut menampilkan kondisi jaringan saat aktivitas *download* dan *upload* sebagai

bagian dari proses pengujian *queue tree*.

1. *Mangle Rule*

Setelah konfigurasi *Mangle Rule* selesai diterapkan, dilakukan pengujian untuk memastikan bahwa siste penandaan *trafik* berjalan sesuai dengan konfigurasi yang telah ditentukan. Pengujian dilakukan dengan menjalankan dua jenis aktivitas yaitu *download* dan *upload*.

2. *Queue Tree* saat melakukan aktivitas *download*

Setelah konfigurasi *Mangle Rule* dinyatakan berfungsi, pengujian dilakukan dengan menjalankan aktivitas *download* untuk memastikan bahwa metode *Queue Tree* bekerja sesuai konfigurasi pada *Queue Tree*. Dengan konfigurasi *Limit At* sebesar 2 *Mbps* dan 30 *Mbps* yang menunjukkan bahwa *trafik* tersebut diproses sesuai dengan prioritas yang telah ditentukan.

3. *Queue Tree* saat melakukan aktivitas *Upload*

Setelah pengujian pada aktivitas *download*, tahap selanjutnya adalah pengujian pada aktivitas *upload*. Tujuan dari pengujian ini adalah untuk memastikan bahwa *trafik upload* yang telah ditandai melalui *Mangle Rule* dapat diarahkan ke antrian yang sesuai dalam *Queue Tree*. Antrian *youtube upload* berada dibawah *parent upload* global pada *Ethernet 2 Wifi*, dengan konfigurasi *Limit At* sebesar 2 *Mbps* dan *Max Limit* 5*Mbps*, yang menunjukkan *trafik* tersebut diproses sesuai dengan prioritas yang telah ditentukan.

4.7 Hasil Pengukuran *QoS* Setelah Metode *Queue Tree*

Setelah metode *Queue Tree* dikonfigurasi pada perangkat *Mikrotik*, pengukuran kualitas layanan jaringan (*Qos*) dilakukan kembali untuk mengetahui sejauh mana perubahan perfoma jaringan dibandingkan dengan kondisi sebelum penerapan metode *Queue Tree*. Pengukuran dilakukan pada dua kondisi, yaitu saat jam sibuk dan saat jam tidak

sibuk, yaitu kedua dilakukan pada hari yang sama untuk menjaga konsistensi hasil. Selama proses pengukuran peneliti melakukan aktivitas *browsing* dan *upload* secara bersamaan dengan durasi pengujian selama 5 menit. Data lalu lintas jaringan dikumpulkan menggunakan aplikasi *Wireshark* kemudian analisis menggunakan bahasa pemrograman *Python* melalui *Platform Google Colab*. Analisa dilakukan terhadap empat parameter utama *Quality of Service (QoS)*, yaitu *throughput*, *packet loss*, *delay*, dan *jitter*. Sebelum dilakukan proses analisis lebih lanjut, berikut disajikan cuplikan data mentah hasil tangkapan lalu lintas jaringan setelah penerapan metode *Queue Tree*.

Tabel 4.3 data mentah jam sibuk hasil *capture* di *Wireshark* sebelum *Queue Tree*

No	Time Span	Jumlah Bytes
1	0	1051
2	0	1051
3	0	1051
4	0	1051

Tabel 4.4 Data mentah jam tidak sibuk hasil *capture* di *Wireshark* sebelum *Queue Tree*

No	Time Span	Jumlah Bytes
1	0	54
2	0.029434	64
3	0.076878	54
4	1.887232	64

Tabel diatas menunjukkan contoh data mentah hasil tangkapan paket yang diperoleh menggunakan *Wireshark* pada kondisi jaringan jam sibuk dan jam tidak sibuk. Data tersebut digunakan sebagai dasar perhitungan parameter *QoS* seperti *throughput*, *packet loss*, *delay*, dan *jitter* menggunakan *Python* pada *Google Collab*. Kolom *Time Span* menunjukkan selisih waktu kedatangan antar paket dalam satuan detik. Nilai ini digunakan untuk menghitung *delay* dan *jitter* pada Tabel 4.3 (jam sibuk), semua paket memiliki *time span* sebesar 0 detik, yang artinya paket tiba secara stabil dan tidak ada jeda waktu

antar paket. Hal ini menunjukkan bahwa metode *Queue Tree* berhasil menjaga kestabilan pengiriman data meskipun jaringan sangat padat. Sementara itu, pada Tabel 4.4 (jam tidak sibuk), *time span* bervariasi dari 0.029434 hingga 0.076878 detik. Ini menunjukan bahwa saat kondisi jaringan lebih ringan, pengiriman data mengikuti pola lalu lintas yang lebih besar dan teratur.

Kolom jumlah Bytes menunjukkan ukuran data tiap paket dalam satuan *byte*. Nilai ini digunakan untuk menghitung *throughput*, yaitu kecepatan data yang berhasil dikirim dalam satu satuan waktu. Pada Tabel 4.3, semua paket berukuran 1051 *byte*, yang mencerminkan alokasi *bandwidth* yang konsisten saat jam sibuk. Sedangkan pada Tabel 4.4 ukuran paket berkisar antara 54 hingga 64 *byte*, menunjukkan aktivitas jaringan yang lebih ringan pada jam tidak sibuk.

1. Kondisi Jam Sibuk

Pengukuran dilakukan kembali saat jam sibuk, setelah penerapan metode *Queue Tree*.

Tabel 4.5 hasil perhitungan *QoS* jam sibuk setelah penerapan *Queue Tree*

No	Time Span	Jumlah Bytes
1	0	54
2	0.029434	64
3	0.076878	54
4	1.887232	64

Parameter	Nilai Rata-rata
<i>Throughput (kbps)</i>	288 kbps
<i>Packet Loss (%)</i>	0 %
<i>Delay (ms)</i>	2,86 ms
<i>Jitter (ms)</i>	9,93 ms

Pada tabel diatas nilai *throughput* yang diperoleh pada kondisi ini adalah sebesar 288 kbps, mengalami peningkatan yang signifikan dibandingkan nilai sebelumnya (220 kbps sebelum *Queue Tree*). *Throughput* menunjukkan bahwa *Queue Tree* mampu mengelola dan mendistribusikan

bandwith secara lebih efisien, bahkan dalam kondisi jaringan padat. Dalam praktiknya, *throughput* yang tinggi berkontribusi langsung terhadap kelancaran akses internet, seperti *browsing* yang cepat dan pemutaran video tanpa *buffering*. Sementara itu *packet loss* tercatat sebesar 0 % yang berarti tidak ada paket data yang hilang selama proses transmisi dari pengiriman ke penerima. Konsistensi nilai ini menunjukkan bahwa metode *queue tree* tidak hanya meningkatkan kecepatan pengiriman data, tetapi juga tetap mempertahankan keandalan jaringan, *Delay* atau waktu tunda pengiriman data tercatat sebesar 2,88 *milidetik*, yang menunjukkan peningkatan efisien dibandingkan kondisi sebelum *queue tree* saat jam sibuk (11,98 *ms*). Nilai *delay* ini mencerminkan kecepatan respon jaringan terhadap permintaan data, seperti saat pengguna membuka halaman *web* atau mengakses aplikasi daring. Adapun *jitter* yang mengukur variasi waktu antar kedatangan paket, juga tercatat sebesar 9,93 *ms*, menurun dari nilai sebelumnya (12,08 *ms*). Penurunan ini menunjukkan bahwa interval waktu antar paket menjadi lebih stabil, sehingga pengiriman data berlangsung secara lebih teratur dan tidak fluktuatif.

2. Kondisi jam tidak sibuk Selanjutnya, pengukuran *QoS* dilakukan kembali pada saat jam tidak sibuk. Setelah penerapan metode *Queue Tree* Tabel 4.6 menampilkan hasil perhitungan parameter *QoS* yaitu : *throughput*, *packet loss*, *delay*, dan *jitter*.

Tabel 4.6 hasil pengukuran *QoS* jam tidak sibuk setelah penerapan *Queue Tree*

Parameter	Nilai Rata-rata
<i>Throughput (kbps)</i>	16,950 <i>kbps</i>
<i>Packet Loss (%)</i>	0 %
<i>Delay (ms)</i>	181 <i>ms</i>
<i>Jitter (ms)</i>	137 <i>ms</i>

Pada tabel diatas nilai *throughput* yang diperoleh pada kondisi ini adalah sebesar 16,950 *kbps*, menunjukkan bahwa jaringan

mampu mengirimkan data dalam jumlah besar setiap detik. Ini merupakan peningkatan signifikan dibandingkan kondisi sebelum *Queue Tree* (1,273 *kbps*), meskipun beban trafik tergolong ringan. Nilai *throughput* yang tinggi ini menunjukkan bahwa metode *queue tree* tetap memberikan manfaat dalam mengoptimalkan penggunaan *bandwith*, bahkan ketika jumlah pengguna sedikit. Sementara itu nilai *packet loss* tetap sebesar 0% menandakan bahwa seluruh paket data berhasil dikirim dan diterima dengan sempurna tanpa kehilangan selama proses pengiriman (transmisi). *Delay* atau waktu tunda pengiriman data sebesar 1,81 *milidetik*, yang menunjukkan respon jaringan yang sangat cepat. Nilai ini jauh lebih rendah dibandingkan kondisi sebelum *Queue Tree* diterapkan (4,88 *ms*), dan mencerminkan bahwa jaringan dapat segera merespon permintaan pengguna tanpa jeda yang berarti *delay* rendah sangat berpengaruh pada pengalaman pengguna. Adapun *jitter* yang mengukur variasi waktu antar kedatangan paket, juga tercatat sebesar 137 *milidetik*, mengalami penurunan dari nilai sebelumnya (9,93*ms*). Penurunan ini menunjukkan peningkatan kestabilan jaringan, dimana waktu antar paket menjadi lebih konsisten dan tidak mengalami fluktuasi besar. *Jitter* yang rendah sangat penting bagi layanan multimedia berbasis waktu (*real-time*) seperti *video streaming*, panggilan video, dan kelas daring. Nilai *jitter* yang stabil dapat mencegah gangguan masalah seperti *buffering*, suara terputus-putus, atau perubahan kualitas gambar secara tiba-tiba.

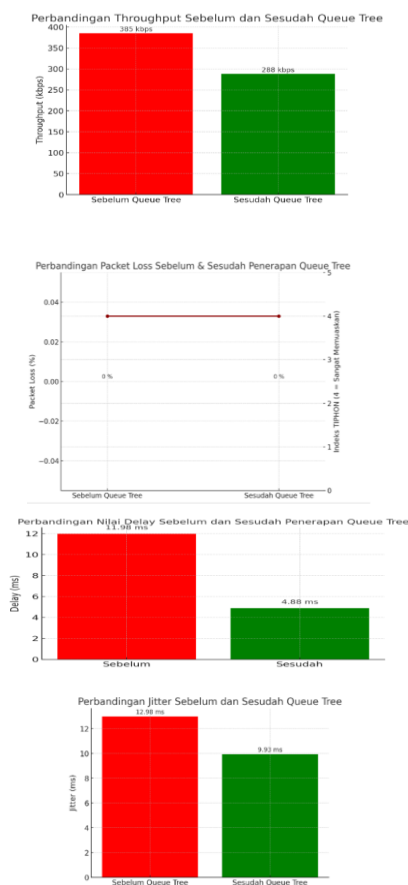
4.8 Evaluasi

4.8.1 Perbandingan *QoS* pada jam sibuk berdasarkan Standar *Tiphon*

Perbandingan parameter *QoS* pada jam sibuk sebelum *Queue Tree* dan sesudah *Queue Tree* berdasarkan hasil pengukuran yang dilakukan ditunjukkan pada Tabel 4.7 dan gambar 4.15 dibawah ini

Tabel 4.7 Perbandingan Parameter *QoS* berdasarkan Standar *Tiphon*

Parameter	Jam sibuk sebelum <i>queue tree</i>	Kategori <i>Tiphon</i>	Indeks	Jam sibuk sesudah <i>queue tree</i>	Kategori <i>Tiphon</i>	Indeks
<i>Troughput (kbps)</i>	385 kbps	Kurang memuaskan	1	288 kbps	Memuaskan	3
<i>Packet loss (%)</i>	0%	Sangat memuaskan	4	0%	Sangat memuaskan	4
<i>Delay (ms)</i>	11,98 ms	Sangat memuaskan	4	2,86 ms	Sangat memuaskan	4
<i>Jiiter</i>	12,08 ms	Memuaskan	3	9,93 ms	memuaskan	3

Gambar 4.15 perbandingan parameter *QoS* sebelum dan sesudah *Queue Tree* pada jam sibukGambar 4.15 Grafik Perbandingan *QoS* Berdasarkan Standar *Tiphon*

Berdasarkan Tabel 4.7 dan Gambar 4.15 diperoleh perbandingan parameter *Quality of Service (QoS)* sebelum dan sesudah penerapan metode *Queue Tree*. Pada parameter *troughput*, nilai meningkat dari 385 kbps sebelum penerapan *queue tree* menjadi 288 kbps setelah penerapan. Berdasarkan standar *tiphon*, nilai sebelum penerapan *queue tree* dikategorikan kurang memuaskan dengan indeks 1

sedangkan penerapan *queue tree* meningkat menjadi

kategori memuaskan dengan indeks 3. Peningkatan ini menunjukkan bandwa manajemen *bandwith* menggunsksn metode *queue tree* mampu meningkatkan alokasi kecepatan transfer data secara signifikan.

Sementara itu untuk parameter *packet loss*, pengujian menunjukan nilai tetap sebesar 0% baik sebelum maupun sesudah penerapan *queue tree*. Berdasarkan standar *tiphon* kondisi ini tergolong sangat memuaskan dengan indeks 4 yang menunjukkan bahwa jaringan tetap stabil tanpa adanya kehilangan atau kegagalan pengiriman paket, baik sebelum maupun sesudah penerapan *queue tree*. Parameter *delay* juga mengalami perbaikan dengan penurunan dari 11,98 ms menjadi 4,88 ms setelah penerapan *queue tree*. Kedua nilai tersebut berada dalam kategori sangat memuaskan dengan indeks 4, dimana penurunan ini mencerminkan peningkatan efesiensi waktu respon jaringan yang lebih cepat setelah penerapan metode *queue tree*.

Selanjutnya parameter *jitter* menunjukkan penurunan dari 12,98 ms menjadi 9,93 ms, meskipun kedua niali tersebut tetap berada dalam kategori memuaskan dengan indeks 3, penurunan ini menunjukkan bahwa variasu waktu antar kedatangan paket menjadi lebih stabil setelah penerapan metode *queue tree*.

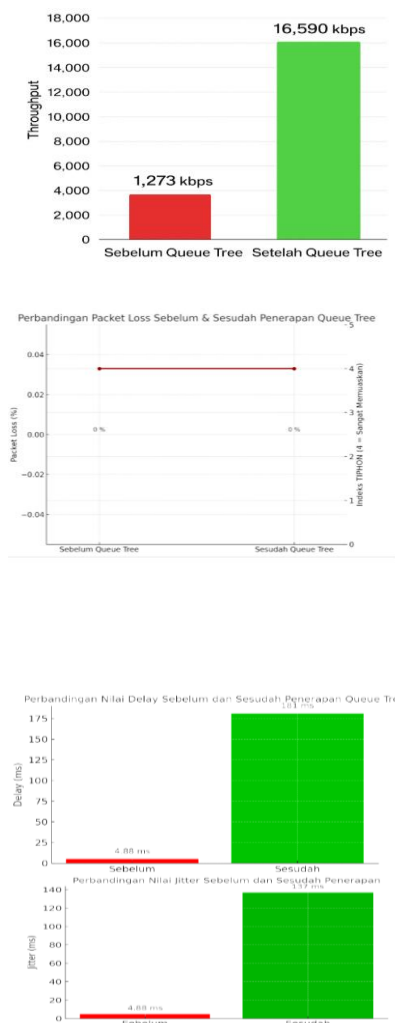
4.8.2 Perbandingan *QoS* pada jam tidak sibuk berdasarkan standar *Tiphon*

Perbandingan parameter *QoS* pada jam tidak sibuk sebelum *queue tree* dan sesudah *queue tree* berdasarkan hasil pengukuran yang dilakukan ditunjukkan pada Tabel 4.8 dan gambar 4.16 dibawah ini

Tabel 4.8 Perbandingan parameter *QoS* berdasarkan standar *Tiphon*

Parameter	Jam sibuk sebelum <i>queue tree</i>	Kategori <i>Tiphon</i>	Indeks	Jam sibuk sesudah <i>queue tree</i>	Kategori <i>Tiphon</i>	Indeks
<i>Troughput</i> (kbps)	1,273kbps	Sangat memuaskan	4	16,950kbps	Sangat memuaskan	4
<i>Packet loss</i> (%)	0 %	Sangat memuaskan	4	0 %	Sangat memuaskan	4
<i>Delay</i> (ms)	181 ms	Sangat memuaskan	4	4,88 ms	Sangat memuaskan	4
<i>Jitter</i>	4,88 ms	Memuaskan	3	137 ms	Memuaskan	3

Gambar 4.16 perbandingan parameter *QoS* sebelum dan sesudah *Queue Tree* pada jam sibuk



Gambar 4.16 Grafik Perbandingan *QoS* Berdasarkan Standar *Tiphon*

Berdasarkan Tabel 4.7 dan Gambar 4.15 diperoleh perbandingan parameter *Quality of Service (QoS)* sebelum dan sesudah penerapan metode *Queue Tree*. Pada parameter *troughput*, nilai meningkat dari 1,273 kbps sebelum penerapan *queue tree* menjadi 16,590 kbps setelah

penerapan. Berdasarkan standar *Tiphon*, nilai sebelum penerapan *queue tree* dikategorikan sangat memuaskan dengan indeks 4 sedangkan penerapan *queue tree* meningkat menjadi kategori sangat memuaskan dengan indeks 4. Peningkatan ini menunjukkan bandwa manajemen *bandwith* menggunakan metode *queue tree* mampu meningkatkan alokasi kecepatan transfer data secara signifikan.

Sementara itu untuk parameter *packet loss*, pengujian menunjukan nilai tetap sebesar 0% baik sebelum maupun sesudah penerapan *queue tree*. Berdasarkan standar *tiphon* kondisi ini tergolong sangat memuaskan dengan indeks 4 yang menunjukkan bahwa jaringan tetap stabil tanpa adanya kehilangan atau kegagalan pengiriman paket, baik sebelum maupun sesudah penerapan *queue tree*.

Parameter *delay* juga mengalami perbaikan dengan penurunan dari 181 ms menjadi 4,88 ms setelah penerapan *queue tree*. Kedua nilai tersebut berada dalam kategori sangat memuaskan dengan indeks 4, dimana penurunan ini mencerminkan peningkatan efesiensi waktu respon jaringan yang lebih cepat setelah penerapan metode *queue tree*.

Selanjutnya parameter *jitter* menunjukkan penurunan dari 4,88 ms menjadi 137 ms, meskipun kedua nilai tersebut tetap berada dalam kategori memuaskan dengan indeks 3, penurunan ini menunjukkan adanya peningkatan kestabilan waktu kedatangan antar paket secara keseluruhan.

4.9 Pembahasan

Berdasarkan hasil penelitian terhadap parameter *Quality of Service (QoS)*, yaitu *troughput*, *packet loss*, *delay*, dan *jitter*, diketahui bahwa penerapan metode *queue tree* pada jaringan *WiFi* di SMA Negeri 1 Kefamenanu mampu meningkatkan kualitas layanan jaringan secara signifikan. Peningkatan ini terlihat baik saat jam sibuk maupun jam tidak sibuk, yang menunjukan

bahwa metode *Queue Tree* efektif dalam mengatur penggunaan *bandwith* dan menjaga kinerja layanan yang dianggap penting.

Sebelum metode *Queue Tree* diterapkan, hasil pengukuran menunjukkan bahwa jaringan cenderung tidak stabil, terutama saat jam sibuk. Nilai *troughput* hanya sebesar 385 *kbps*, yang termasuk dalam kategori kurang memuaskan menurut standar *TIPHON*. Meskipun nilai *delay* mencapai 11,98 *ms* (kategori sangat memuaskan), *jitter* sebesar 12,08 *ms* yang termasuk dalam kategori kurang memuaskan menurut standar *TIPHON*. Meskipun nilai *delay* mencapai 11,98 *ms* (kategori sangat memuaskan), *jitter* sebesar 12,08 *ms* (kategori memuaskan), dan *packet loss* sebesar 0 % (kategori sangat memuaskan), rendahnya *troughput* menandakan bahwa *bandwith* tidak terbagi secara optimal ketika banyak perangkat aktif secara bersamaan. Pada kondisi jaringan dengan *trafik* rendah sebelum penerapan *Queue Tree*, performa jaringan terlihat lebih baik. *Troughput* tercatat sebesar 385 *kbps* (kategori cukup memuaskan), dengan *delay* dan *jitter* masing-masing sebesar 181 *ms*, serta *packet loss* tetap 0%. Hal ini menunjukkan bahwa jaringan sudah berjalan cukup stabil ketika tidak banyak perangkat yang menggunakan layanan secara bersamaan.

Setelah *Queue Tree* diterapkan melalui pengaturan *mangle* dan *queue tree* pada perangkat *Mikrotk*, terjadi peningkatan kualitas jaringan yang signifikan. Pada jam sibuk, *troughput* meningkat menjadi 288 *kbps* (kategori memuaskan), *delay* menurut menjadi 2,86 *ms* (kategori sangat memuaskan), *jitter* juga turun menjadi 9,93 *ms* (kategori memuaskan), dan *packet loss* tetap stabil diangka 0% ini menunjukkan bahwa metode *Queue Tree* mampu mengatur antrian lalu lintas dan membagi *bandwith* secara lebih adil, sehingga layanan dengan prioritas tinggi tetap berjalan lancar meskipun jaringan sedang

padat. Peningkatan yang lebih besar juga terlihat pada saat kondisi *trafik* rendah. *Troughput* meningkat menjadi 1,237 *kbps* (kategori sangat memuaskan), *delay* dan *jitter* menurun menjadi 137 *ms*, dan *packet loss* tetap 0% penurunan nilai *delay* dan *jitter* meskipun dalam kondisi jaringan tidak padat membuktikan *Queue Tree* tidak hanya bermanfaat saat *trafik* tinggi, tetapi juga dapat menjaga efesiensi dan kestabilan jaringan saat hanya sedikit perangkat yang aktif.

Temuan ini didukung oleh penelitian yang dilakukan oleh (Yuda Irawan dkk, 2022) yang menunjukkan bahwa *Queue Tree* dapat meningkatkan *troughput*, menurunkan *delay* dan *jitter*, serta menjaga *packet loss* tetap 0% penelitian lain (bagus dkk., 2024) juga menunjukkan hasil serupa, dimana *Queue Tree* terbukti meningkatkan *troughput*, menjaga *packet loss* tetap 0% dan menurunkan *delay*.

Secara keseluruhan, hasil ini menunjukkan bahwa *Queue Tree* mampu mengelola lalu lintas seperti *browsing* tetap mendapatkan alokasi *bandwith* yang optimal. Penerapan metode *Queue Tree* melalui penandaan *trafik* (*mangle*) dan pengelolaan antrian (*Queue Tree*) terbukti dapat meningkatkan efesiensi, kestabilan.

1. KESIMPULAN

a. Dalam manajemen *bandwith* jaringan disekolah mampu meningkatkan kualitas layanan jaringan. Hal ini terlihat dari penurunan *delay* dan *jitter* serta peningkatan *troughput*, yang berdampak pada akses internet yang lebih stabil dan merata bagi seluruh pengguna.

b. Penerapan manajemen *bandwith* menggunakan metode *Queue Tree* memungkinkan adanya pembagian *bandwith* sesuai prioritas kebutuhan, sehingga pemanfaatan jaringan dapat lebih efektif untuk mendukung pembelajaran digital, administrasi sekolah, maupun kebutuhan lainnya.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada pihak-pihak terkait yang telah memberi dukungan terhadap penelitian ini

DAFTAR PUSTAKA

- [1] Sene, E., Nababan, D., & Kelen, Y. P. (2023). Analisis Dan Perancangan Jaringan Hotspot Server Berbasis Mikrotik Pada Xpresso Café Kefamenanu. *JUTEKBIDIK: Jurnal Teknologi, Bisnis & Pendidikan*, 1(1), 10-21.
- [2] Bano, R. M. E., Kelen, Y. P., & Rema, Y. O. (2023). Perancangan Jaringan Hotspot Dengan System Voucher Berbasis Mikrotik Pada Warnet B-Net Menggunakan Metode WDS (Wireless Distribution System). *JUTEKBIDIK: Jurnal Teknologi, Bisnis & Pendidikan*, 1(1), 22-37.
- [3] M., & Suharyanto, C. E. (2020a). Implementasi Manajemen Bandwidth Menggunakan Metode Queue Tree Pada Jaringan Internet. *Innovation in Research of Informatics (INNOVATICS)*, 2(2). <https://doi.org/10.37058/innovatics.v2i2.1482>
- [4] Azizah, U., & Veritawati, I. (2021). Implementasi Management Bandwidth Menggunakan Metode Queue Tree Dengan PCQ (Per Connection Queue). *Journal of Informatics and Advanced Computing (JIAC)*, 2(1), 2021. <https://journal.univpancasila.ac.id/index.php/jiac/article/view/2687>
- [5] T Eriyadi, S. (2023). Implementasi Manajemen Bandwidth Menggunakan Routerboard 941 Di Smk Muhammadiyah Belik. *Jurnal Teknik Informatika Dan Sistem Informasi (JURTISI)*, 3(1), 33–38.
- [6] Firmansyah, B. (2020). Dengan Metode Queue Tree Pada Pt . Cipta Banuata Anugrah Jakarta. *Jurnal Ilmiah Teknik Informatika (TEKINFO)*, 21(1), 94–103. <https://journals.upi-yai.ac.id/index.php/TEKINFO/article/view/1144>
- [7] Heryanto, A. (2019). *Manajemen Pemakaian Bandwidth Dan Akses Jaringan Komputer Menggunakan Htb Mikrotik Oleh : Wira Kusuma Bangsa*.
- [8] Manao, M. F., Sinaga, A., & Aripin, S. (2022). Penerapan Metode Queue Tree Dalam Manajemen Bandwidth Berbasis Mikrotik Untuk Memudahkan Pembelajaran Dalam Menghadapi Dampak Covid-19. *Journal Global Technology Computer*, 1(3), 115–121. <https://doi.org/10.47065/jogtc.v1i3.2111>
- [9] Rismawan, E. (2012). Jurnal Informatika dan Teknik Elektro Terapan. *Jurusan Teknik Elektro Fakultas Teknik Universitas Lampung*.
- [10] Manao, M. F., Sinaga, A., & Aripin, S. (2022). Penerapan Metode Queue Tree Dalam Manajemen Bandwidth Berbasis Mikrotik Untuk Memudahkan Pembelajaran Dalam Menghadapi Dampak Covid-19. *Journal Global Technology Computer*, 1(3), 115–121. <https://doi.org/10.47065/jogtc.v1i3.2111>
- [11] Musmuharam, & Eko Suharyanto, C. (2020). Implementasi Manajemen Bandwidth Menggunakan Metode Queue Tree Pada Jaringan Internet. *Innovation in Research of Informatics (INNOVATICS)*, 2(2), 69–79.
- [12] S. Raharjo, A. N. Putra, and L. Hidayat, “Optimasi Manajemen Bandwidth Menggunakan Metode Queue Tree Berbasis Mikrotik,” **Jurnal Informatika dan Teknik Elektro Terapan (JITET)**, vol. 12, no. 2, pp. 145–152, 2025. <https://doi.org/10.23960/jitet.v12i2.5012>
- [13] Azizah, U., & Veritawati, I. (2021). Implementasi Management Bandwidth Menggunakan Metode Queue Tree Dengan PCQ (Per Connection Queue). *Journal of Informatics and Advanced Computing (JIAC)*, 2(1), 2021. <https://journal.univpancasila.ac.id/index.php/jiac/article/view/2687>
- [14] Firmansyah, B. (2020). Dengan Metode Queue Tree Pada Pt . Cipta Banuata Anugrah Jakarta. *Jurnal Ilmiah Teknik Informatika (TEKINFO)*, 21(1), 94–103. <https://journals.upi-yai.ac.id/index.php/TEKINFO/article/view/1144>
- [15] Rismawan, E. (2012). Jurnal Informatika dan Teknik Elektro Terapan. *Jurusan Teknik Elektro Fakultas Teknik Universitas Lampung*.