

OPTIMASI JARINGAN DENGAN COMMON APPLICATION KEPT ENHANCED PADA MIKROTIK ROUTEROS 7 UNTUK MENGATASI BUFFERBLOAT

Muhammad Aziz¹, Imam Suharjo²

Universitas Mercu Buana Yogyakarta;

Jl. Jembatan Merah 84C Gejayan Yogyakarta 55283; (0274) 6647444 / (0274) 6498213

Keywords:

Quality of Service (QoS), RouterOS, Active Queue Management (AQM), Mikrotik, Common Application Kept Enhanced (CAKE), Bufferbloat

Correspondent Email:

muhaziz0987@gmail.com



Copyright © [JITET](#) (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. Penelitian ini bertujuan untuk mengoptimalkan kinerja jaringan dengan menerapkan algoritma Common Application Kept Enhanced (CAKE) pada MikroTik RouterOS 7 dalam mengatasi permasalahan bufferbloat. Bufferbloat sendiri merupakan penumpukan paket data pada buffer router yang menyebabkan peningkatan latensi dan penurunan QoS. Penelitian ini akan menggunakan metode eksperimental dengan membandingkan kinerja jaringan sebelum dan sesudah implementasi CAKE. Parameter meliputi latensi, jitter, throughput, dan pengalaman pengguna jaringan (video streaming dan bermain game). Hasil penelitian menunjukkan bahwa implementasi CAKE pada MikroTik RouterOS 7 secara signifikan mampu mengurangi latensi dan jitter. Pada pengujian bufferbloat test, Dari yang sebelumnya rata-rata latensi download 29 ms dan upload 6,66 ms setelah optimasi rata-rata latensi download menjadi 0,66 ms dan upload 1 ms. Pada pengujian streaming video terdapat perubahan signifikan pada rata-rata jitter dimana sebelum optimasi mendapatkan 0,444 ms, setelah optimasi menjadi 0,017 ms. Sedangkan untuk bermain game online tidak ada peningkatan, tidak ada perbedaan pengalaman bermain dan hanya ada peningkatan kualitas koneksi jaringan sebesar 1%. Setelah optimasi terdapat dampak pada pengujian bufferbloat test yakni penurunan rata-rata throughput sebesar 9,78 Mbps, peningkatan QoS yang signifikan membuktikan CAKE merupakan solusi yang efektif untuk mengatasi bufferbloat pada jaringan yang menggunakan MikroTik RouterOS 7.

1. PENDAHULUAN

Pada saat ini internet sangat berkembang pesat dari mulai sebagai sarana komunikasi global, sumber informasi yang tak terbatas, platform pertukaran data yang efisien, hingga media hiburan yang telah menjadi bagian dari aktivitas sehari-hari manusia. Hal ini tentu saja mendorong perkembangan teknologi yang

semakin maju pesat sehingga menimbulkan konsekuensi terhadap infrastruktur jaringan yang ada, yang harus terus ditingkatkan agar mampu memenuhi tuntutan kebutuhan pengguna yang semakin kompleks. Salah satu tantangan utama dalam pengelolaan jaringan modern adalah mengatasi bufferbloat. Bufferbloat sendiri merupakan fenomena yang

terjadi dalam jaringan packet-switched, di mana akumulasi paket dalam buffer menyebabkan peningkatan latensi dan variasi penundaan paket (dikenal juga sebagai jitter), serta menurunkan throughput jaringan secara keseluruhan [1].

Dalam bahasa yang lebih mudah, fenomena ini timbul karena perangkat jaringan mencoba untuk mengatasi kemacetan dengan memperbesar ukuran buffer, yang justru memperburuk masalah dengan menambahkan penundaan yang tidak perlu. Bufferbloat dapat mempengaruhi pengalaman pengguna secara signifikan, terutama pada aplikasi yang sensitif terhadap latensi seperti game online, video conference, dan VoIP.

Permasalahan ini membuat perusahaan perangkat jaringan asal Latvia yaitu Mikrotik membuat sebuah solusi untuk mengatasi masalah ini yaitu dengan mengembangkan fitur bernama Common Application Kept Enhanced atau dikenal dengan CAKE pada upgrade RouterOS versi 7. Fitur ini di klaim mampu mengatasi masalah bufferbloat yang ada saat ini dengan cara yang lebih efisien dan tidak menimbulkan dampak berlebihan pada jaringan itu sendiri.

Namun untuk mengimplementasikan fitur ini tentu tidak bisa asal karena harus dilakukan riset lebih lanjut agar bisa diimplementasikan secara efektif dan efisien karena perbedaan kondisi jaringan yang bervariasi membutuhkan konfigurasi CAKE yang sesuai pula. Dengan latar belakang tersebut saya memutuskan untuk melakukan sebuah penelitian yang berjudul "Optimasi Jaringan Dengan Common Application Kept Enhanced pada Mikrotik RouterOS 7 untuk Mengatasi Bufferbloat".

Adanya penelitian ini untuk menjawab beberapa pertanyaan dari peneliti, yakni untuk menganalisis apa dampak penerapan CAKE terhadap jaringan pada MikroTik RouterOS, bagaimana cara kerja algoritma Common Application Kept Enhanced (CAKE) dalam mengatasi bufferbloat, dan apa tantangan yang dihadapi dalam implementasi CAKE pada MikroTik RouterOS 7 dalam jaringan. Adapun tujuan dari penelitian ini adalah untuk menganalisis efektivitas Common Application Kept Enhanced dalam mengurangi bufferbloat pada lingkungan jaringan yang berbeda-beda, mengidentifikasi parameter konfigurasi yang optimal untuk mencapai kinerja terbaik, dan

memberikan panduan praktis bagi network administrator dalam mengimplementasikan solusi ini.

2. TINJAUAN PUSTAKA

Dalam penelitian terdahulu oleh M. I. R. Febriansah yang membandingkan kinerja tiga algoritma AQM yang sudah ada (Droptail, RED, dan SFQ) dalam menangani bottleneck dan bufferbloat pada transmisi TCP NewReno. Simulasi dengan variasi bandwidth dan ukuran buffer menunjukkan bahwa RED dan SFQ secara umum lebih efektif dalam mengurangi packet drop dibandingkan Droptail [1].

Dalam penelitian lain oleh H. Wang memperkenalkan algoritma AQM adaptif baru, yakni TODU. Algoritma ini dirancang untuk menyeimbangkan queuing delay dan link utilization. Strategi TODU dalam meniadakan dampak buruk bufferbloat adalah dengan secara dinamis menyesuaikan probabilitas packet drop untuk mencegah penumpukan paket yang berlebihan. Hasil simulasi menunjukkan bahwa TODU memberikan trade-off yang lebih baik antara latensi rendah dan utilisasi link tinggi dibandingkan dengan algoritma AQM konvensional, khususnya dalam lingkungan dinamis yang rentan terhadap bufferbloat [2].

Penelitian lain yang saya kaji berfokus pada mitigasi bufferbloat untuk optimasi streaming video. Penulis yakni V. Vimaladevi dan M. Bala mengusulkan algoritma Hybrid FQ-CoDel, yang menggabungkan CoDel dan Adaptive CoDel dengan fair queuing. Dengan menggabungkan mekanisme ini, algoritma Hybrid FQ-CoDel secara efektif mengurangi queuing delay dan meningkatkan utilisasi bandwidth untuk streaming video, sehingga meminimalisir dampak negatif bufferbloat terhadap kualitas video. Adaptive CoDel, yang dikembangkan khusus untuk lalu lintas video, terbukti efektif dalam mengontrol queuing delay yang dinamis, sehingga mencegah terjadinya bufferbloat [3].

Penelitian yang lebih mendalam oleh J. Ye, K. C. Leung, dan S. H. Low memberikan kerangka kerja teoritis untuk mengatasi bufferbloat dalam skenario jaringan multi-bottleneck, yang lebih realistis. Penelitian ini menganalisis keseimbangan dan stabilitas sistem TCP/AQM multi-bottleneck, dan menghasilkan kondisi yang cukup untuk

menjamin keunikan titik keseimbangan dan stabilitas asimtotik local [4]. Berdasarkan analisis ini, algoritma Self-Tuning CoDel dikembangkan untuk secara dinamis menyesuaikan parameter CoDel berdasarkan kondisi jaringan, sehingga secara proaktif mencegah terjadinya bufferbloat. Simulasi pada penelitian menunjukkan bahwa Self-Tuning CoDel secara signifikan menstabilkan queuing delay dan mengurangi dampak bufferbloat dalam jaringan multi-bottleneck.

Perbedaan mendasar antara penelitian-penelitian yang sudah ada dengan penelitian ini tertera secara garis besar pada tabel E.1 berikut ini.

3. LANDASAN TEORI

3.1. Bufferbloat

Menurut M. I. R. Febriansah dalam penelitiannya yang berjudul "Analisis Bottleneck dan Bufferbloat pada AQM Droptail, RED dan SFQ di Komunikasi Data TCP Newreno". Bufferbloat adalah fenomena yang terjadi dalam jaringan packet-switched, di mana akumulasi paket dalam buffer menyebabkan peningkatan latensi dan variasi penundaan paket (dikenal juga sebagai jitter) [1], serta menurunkan throughput jaringan secara keseluruhan. Latensi ini terdiri dari tiga jenis penundaan yang berbeda yang terkait dengan keterlambatan transmisi paket melalui jaringan:

- a. Delay Propagasi, yang mengacu pada waktu yang dibutuhkan sebuah paket untuk bergerak dari satu titik ke titik lain dalam jaringan.
- b. Delay Transmisi, yang mencerminkan waktu yang dibutuhkan untuk mendorong semua bit paket ke dalam media transmisi.
- c. Delay Antrian, yang mengacu pada waktu yang dihabiskan paket menunggu dalam antrian buffer sebelum diproses dan dikirimkan [5].

Dalam penelitian yang lain oleh D.B. Pratama, Bufferbloat terjadi ketika router atau switch memiliki buffer yang terlalu besar, yang menyebabkan paket data tertunda secara signifikan karena harus menunggu dalam antrian yang panjang. Hal ini dapat mengakibatkan latensi yang tinggi dan kinerja jaringan yang buruk, terutama untuk aplikasi

yang membutuhkan respons cepat seperti game online atau video streaming [6].

3.2. Active Queue Management (AQM)

Masih dalam penelitian yang sama oleh M. I. R. Febriansah menurutnya AQM atau Active Queue Management adalah teknik yang digunakan router untuk mengontrol antrian paket data dan mencegah kemacetan jaringan, bukan hanya dengan menyimpan paket, tetapi juga dengan strategi aktif seperti menjatuhkan atau menandai paket [1]. Algoritma AQM memantau kedalaman antrian dan lalu lintas jaringan, menentukan kapan harus menjatuhkan paket atau menandai paket. Terdapat beberapa jenis algoritma AQM yang umum digunakan, antara lain:

- a. Random Early Detection (RED): Algoritma ini secara acak menjatuhkan paket sebelum antrian penuh, dengan tujuan memberikan sinyal kepada pengirim untuk mengurangi laju pengiriman data [7].
- b. Controlled Delay (CoDel): Algoritma ini akan mencoba menjaga penundaan antrian tetap rendah dengan menjatuhkan paket secara agresif ketika penundaan melebihi ambang batas tertentu [8].
- c. FlowQueue-CoDel (FQ-CoDel): Algoritma ini memadukan antara mekanisme control delay dan flow queueing. Dimana perbedaan dari CoDel biasanya adalah setiap paket yang masuk akan diklasifikasikan ke dalam berbagai aliran berdasarkan IP sumber dan tujuan, nomor port sumber dan tujuan, dan protokol paket [9].

Menurut penelitian lain berjudul "Non-Stationary Characteristics of AQM Based on the Queue Length" oleh Andrzej Chydzinski, Active Queue Management (AQM) bekerja dengan cara melakukan drop packet secara acak atau dengan ketentuan paket tertentu, proses ini bertujuan untuk mencegah antrian paket berlebihan yang jika dibiarkan akan menimbulkan bufferbloat [10].

3.3. Quality of Service (QoS)

Quality of Service (QoS) adalah suatu metode untuk menentukan dan menjamin kualitas layanan jaringan, dengan mendefinisikan karakteristik dan sifat layanan yang diberikan [11]. Parameter QoS meliputi bandwidth, latency, jitter, dan packet loss. Dengan menerapkan QoS, seorang network

administrator dapat memprioritaskan lalu lintas tertentu, memastikan akses aplikasi atau website penting akan mendapatkan prioritas dalam jaringan dan memberikan pengalaman pengguna yang lebih baik [12].

3.4. Common Application Kept Enhanced (CAKE)

Menurut website komunitas jaringan "www.bufferbloat.net", CAKE atau yang dikenal dengan Common Application Kept Enhanced adalah algoritma manajemen antrian modern canggih yang dirancang untuk menangani beragam kondisi dan jenis lalu lintas jaringan [13]. CAKE merupakan peningkatan dari algoritma manajemen antrian sebelumnya, seperti CoDel dan FQ_CoDel. CAKE menggabungkan beberapa teknik untuk mencapai kinerja yang lebih baik, termasuk shaping, fair queuing, dan penghindaran kemacetan adaptif.

Salah satu fitur utama CAKE yakni kemampuannya untuk melakukan shaping lalu lintas, yang memungkinkan seorang network administrator untuk membatasi laju lalu lintas yang masuk atau keluar dari jaringan. Hal ini berguna untuk mencegah satu aplikasi atau pengguna menghabiskan semua bandwidth yang tersedia. Selain itu, CAKE juga menggunakan fair queuing untuk memastikan bahwa semua aliran lalu lintas menerima bagian bandwidth yang adil, dan CAKE juga memiliki mekanisme penghindaran kemacetan adaptif yang secara otomatis menyesuaikan parameter antrian berdasarkan kondisi jaringan. Dengan menggunakan CAKE, seorang network administrator dapat meningkatkan kinerja jaringan, mengurangi latensi, dan memitigasi gangguan jaringan seperti bufferbloat.

3.5. Mikrotik RouterOS

Mikrotik RouterOS merupakan sistem operasi yang dikembangkan perusahaan penyedia layanan internet (ISP) asal Latvia sejak 1996, yang memungkinkan komputer difungsikan sebagai router jaringan berbasis linux [14]. MikroTik RouterOS menyediakan berbagai fitur jaringan, termasuk routing, firewall, VPN, dan QoS. RouterOS juga dikenal karena fleksibilitas, kemampuan konfigurasi yang luas, dan harga yang terjangkau [15].

Dalam penelitian ini, fitur Common Application Kept Enhanced yang merupakan bagian dari fitur queue type Mikrotik akan

dimanfaatkan untuk mengatasi bufferbloat. Queue type CAKE pada Mikrotik hanya ada pada RouterOS 7 bersamaan dengan CoDel dan FQ-CoDel. Ketiga queue type ini merupakan pengembangan dari fitur yang sebelumnya sudah ada pada RouterOS 6 yakni RED, SFQ, dan PCQ.

4. METODE PENELITIAN

Penelitian ini mengadopsi pendekatan metodologis yang dirancang untuk mengeksplorasi efektivitas optimasi jaringan melalui implementasi Common Application Kept Enhanced pada MikroTik RouterOS 7. Adapun tahapan metodologi yang akan dilakukan pada penelitian ini sebagai berikut.



Gambar 1. Tahapan Penelitian

4.1. Pengumpulan Data

Dalam penelitian ini, metode pengumpulan data yang diterapkan adalah pendekatan kualitatif. Data kualitatif akan diperoleh melalui serangkaian eksperimen yang dilakukan sebelum dan sesudah penerapan optimasi. Parameter yang akan diukur meliputi grade bufferbloat (diukur menggunakan website waveform), Delay, Jitter, Throughput, dan Packet Loss selama streaming video (diukur menggunakan Wireshark). Selain pengujian streaming video dan website, penelitian ini juga akan melibatkan pengujian menggunakan game

online, dengan Counter-Strike 2 sebagai platform untuk mengumpulkan data.

Dari ketiga data yang diambil pada website, video streaming, dan game online akan diolah menggunakan metode tabel perbandingan guna melihat perbedaan antara kondisi jaringan sebelum dan sesudah optimasi. Berikut adalah tabel data pengujian yang akan digunakan pada penelitian ini.

Tabel 1. Daftar Pengujian

No.	Pengujian	Data	Keterangan
1	Speedtest & Bufferbloat Grade	Latency, Speed dan Grade Bufferbloat	Data diambil dengan pengujian jaringan pada website waveform.com dengan hasil Bufferbloat Grade (E - A+)
2	Streaming Video	Troughput, Packet Loss, Delay dan Jitter	Data diambil menggunakan Wireshark
3	Game Online	Persentase Buffer	Data diambil dengan console developer mode dalam game "Counter-Strike 2" setelah menyelesaikan 1 permainan pada server SEA (Singapura)

Selain ketiga data tersebut, data konfigurasi router sebelum dan sesudah implementasi CAKE akan didokumentasikan secara rinci. Meliputi topologi jaringan, versi firmware mikrotik dan parameter QoS yang digunakan. Selanjutnya pada pengujian streaming video akan dilakukan perhitungan delay dan jitter secara manual dengan data yang diambil dari Wireshark menggunakan rumus sederhana sebagai berikut.

$$\text{Delay} = \text{timespan 2} - \text{timespan 1} \quad (1)$$

$$\text{Rata - rata delay} = \frac{\text{total delay}}{\text{banyak paket}} \quad (2)$$

$$\text{jitter} = \text{delay 2} - \text{delay 1} \quad (3)$$

$$\text{Rata - rata jitter} = \frac{\text{total jitter}}{\text{banyak paket}} \quad (4)$$

4.2. Alat Penelitian

Pada penelitian ini akan digunakan beberapa alat penelitian yang terbagi menjadi hardware, software dan jaringan internet untuk mendukung penelitian ini. Alat yang disebutkan disini merujuk kepada peralatan yang langsung digunakan dalam penelitian dengan spesifikasi yang memadai untuk melakukan penelitian.

4.2.1. Hardware

Penelitian ini menggunakan hardware dengan detail spesifikasi sebagai berikut.

Tabel 2. Hardware

Hardware	Merk / Seri	Spesifikasi
Laptop	Advan PixelWar Gaming Laptop	AMD Ryzen 5 6600H (12 CPUs) @3.3GHz AMD Radeon RX 6500M Laptop GPU 4 GB VRAM RAM 16 GB LPDDR5 1 Terabyte SSD NVME Gen 3 Wireless Mediatek RZ608 Wi-Fi 6E 80MHz
AP Router	Advan Revealink AC1200 V1.0	1 Gigabit WAN Up to 867 Mbps 5Ghz Wireless Up to 300 Mbps 2.4Ghz Wireless
Router	Mikrotik CCR1016-12G	Architecture TILE CPU TLR4-01680 (16 CPUs) @1.2 GHz RAM 2 GB Storage 512 GB (NAND)

4.2.2. Software

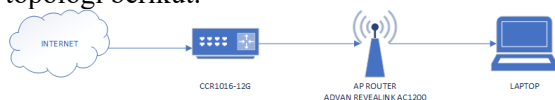
Penelitian ini menggunakan software dengan detail spesifikasi sebagai berikut.

Tabel 3. Software

Software / Sistem Operasi	Seri / Versi
Windows 11	Versi 24H2 (OS Build 26100.4202)
AP Router Firmware	S01.43.01.001
Mikrotik OS & Firmware	Router OS 6 / 6.49.10 Router OS 7 / 7.19.1
Wireshark	Versi 4.4.6
Winbox	V3.42
Chrome Browser	Versi 137.0.7151.56
Counter-Strike 2	<i>Latest Patch</i> 18.6.6593.3

4.2.3. Topologi Jaringan

Adapun topologi jaringan lokal pada penelitian ini akan dijabarkan pada gambar topologi berikut.



Gambar 2. Topologi Jaringan

4.2.4. Jaringan Internet

Penelitian ini menggunakan jaringan internet dengan detail spesifikasi sebagai berikut.

Tabel 4. Jaringan Internet

Layanan	Provider	Spesifikasi
Internet	PT. Wahana	100 Mbps
	Lintas Nusa	Dedicated
	Persada	(Metro
	(WLAN)	Telkom)

4.3. Implementasi

Tahapan implementasi sangatlah penting. Karena tahapan ini akan menentukan hasil dari penelitian, sedikit perubahan konfigurasi dapat berpengaruh secara signifikan ke hasil penelitian ini. Implementasi pada penelitian ini akan berfokus dari proses upgrade router dari RouterOS 6 ke RouterOS 7 dilanjutkan perubahan konfigurasi QoS dengan queue type RED ke CAKE. Seluruh proses upgrade dan

konfigurasi akan dilakukan dengan menggunakan antarmuka aplikasi winbox.

Berikut adalah tahapan implementasi atau proses optimasi dalam penelitian ini.

1. Backup konfigurasi router mikrotik.

Pada tahap pertama akan dilakukan backup konfigurasi, tahapan ini penting karena penelitian ini dilakukan pada jaringan internet aktif dan akan ada batasan waktu upgrade dimana jika terjadi kendala pada saat proses implementasi dapat dilakukan restore konfigurasi router dengan singkat

2. Upgrade RouterOS dari RouterOS 6 diupgrade ke RouterOS 7.

Proses upgrade dilakukan tepat setelah proses backup selesai karena proses upgrade masih membutuhkan koneksi internet. Proses upgrade akan menggunakan antarmuka Winbox.

3. Konfigurasi Simple Queue dengan Queue Type CAKE.

Setelah melakukan konfigurasi jaringan secara keseluruhan aru dapat dilakukan implementasi QoS pada Queue Tree dengan Queue Type CAKE yang akan disesuaikan dengan kebutuhan jaringan.

5. HASIL DAN PEMBAHASAN

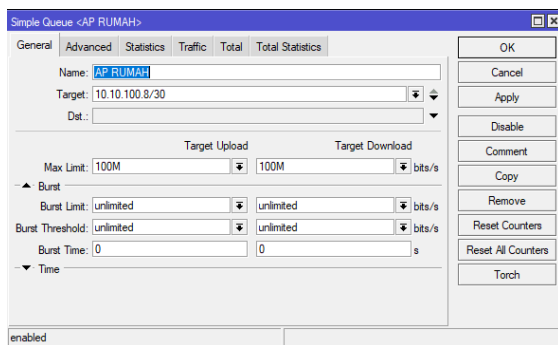
Hasil dari penelitian ini adalah perbandingan bufferbloat jaringan sebelum optimasi dengan sesudah optimasi menggunakan algoritma CAKE pada router Mikrotik dengan RouterOS versi 7.

5.1. Analisis dan Pengumpulan Data

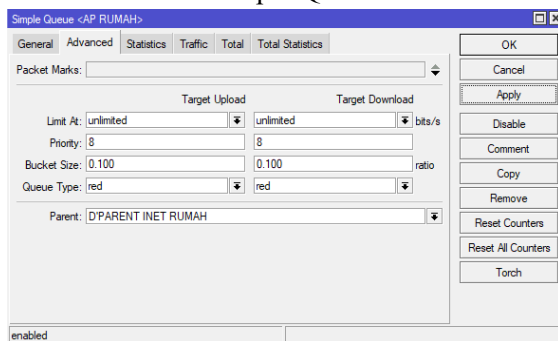
Tahap analisis dilakukan dengan mengumpulkan data konfigurasi dan hasil pengujian dari sebelum optimasi menggunakan algoritma CAKE, kemudian pengambilan data meliputi konfigurasi mikrotik, pengujian bufferbloat, pengujian video streaming dan pengujian game.

5.1.1. Topologi Jaringan

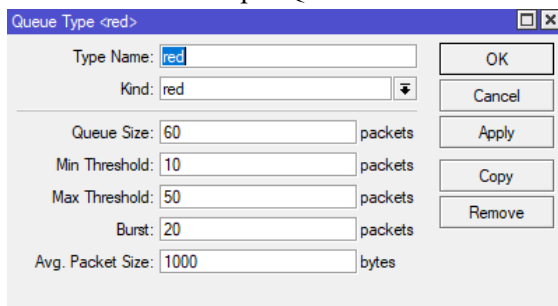
- a. Berikut konfigurasi QoS pada router mikrotik sebelum dilakukan optimasi.



Gambar 3. Simple Queue - General



Gambar 4. Simple Queue – Advanced



Gambar 5. Queue Type – RED

Dari ketiga gambar diatas dapat dipahami jika konfigurasi QoS pada AP Router saat ini memiliki limit sebesar 100 Mbps dan menggunakan queue type – RED dengan detail konfigurasi sebagai berikut.

1. Queue Size

Pada kolom ini menunjukkan maksimal antrian di angka 60 paket.

2. Min Threshold

Kemudian pada kolom ini menunjukkan ambang batas bawah algoritma RED. Dengan asumsi jika antrian sudah lebih dari 10 paket maka algoritma RED mulai memonitor dan menandai paket.

3. Max Threshold 50 paket

Sedangkan pada kolom max treshold menunjukkan ambang batas atas algoritma RED. Dengan asumsi jika paket sudah menyentuh angka 50

maka probabilitas paket drop meningkat secara signifikan

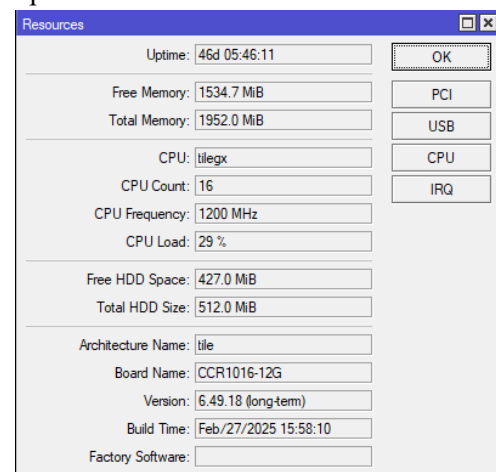
4. Burst

Pada kolom burst menunjukkan batas toleransi ketika terjadi lonjakan antrian dimana batasan ada pada angka 20 paket.

5. Average Packet Size

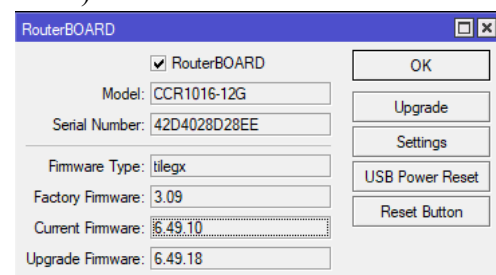
Pada kolom Average Packet Size menunjukkan angka rata-rata besaran setiap paket. Dengan asumsi router harus menjaga besar setiap paket diangka 1000 byte.

b. Berikut adalah versi RouterOS 6 dan firmware yang digunakan sebelum optimasi



Gambar 5. Versi RouterOS 6

Pada gambar 4.4 dapat dilihat jika RouterOS mikrotik masih menggunakan RouterOS 6 dengan versi 6.49.18 (long-terms).



Gambar 6. RouterBOARD

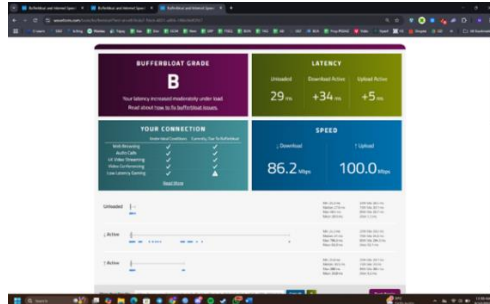
Pada Gambar 4.5 menunjukkan jika versi firmware mikrotik masih menggunakan versi 6.49.10.

5.1.2. Pengujian Sebelum Optimasi

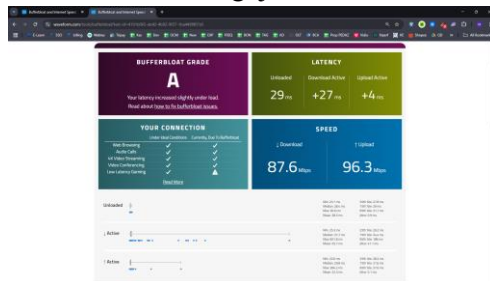
Pada tahap ini akan ada 3 pengujian sebelum dilakukan optimasi menggunakan algoritma CAKE sebagai pembanding pengujian yang

akan dilakukan kembali setelah optimasi. Berikut adalah ketiga pengujian sebelum optimasi.

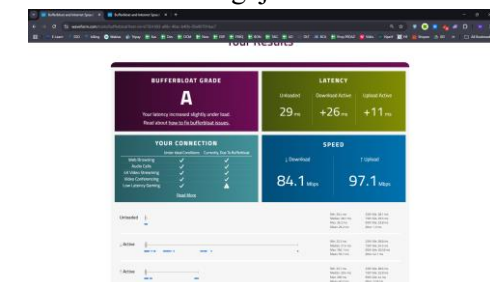
1. Pengujian Bufferbloat



Gambar 6. Pengujian Bufferbloat 1



Gambar 7. Pengujian Bufferbloat 2



Gambar 8. Pengujian Bufferbloat 3

Pada pengujian bufferbloat dilakukan 3 kali pengujian untuk mendapatkan data yang lebih valid. Pada ketiga pengujian didapatkan hasil pengujian pertama dengan grade B, pengujian kedua dengan grade A dan pengujian terakhir mendapatkan grade A. Sehingga dapat disimpulkan jika jaringan menggunakan queue type RED terkadang masih mengalami sedikit bufferbloat. Berikut tabel hasil pengujian bufferbloat pada pengujian sebelum optimasi.

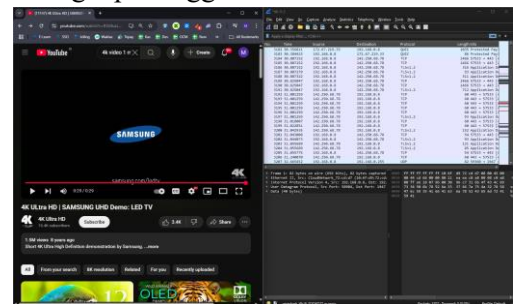
Tabel 5. Tabel Hasil Pengujian Bufferbloat

No	Latensi Download	Latensi Upload	Kecepatan Download	Kecepatan Upload	Nilai
1	34 ms	5 ms	87,6 Mbps	96,3 Mbps	B

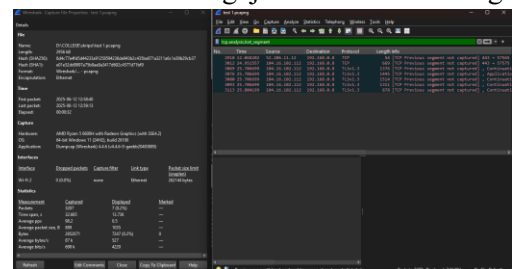
2	27 ms	4 ms	87,6 Mbps	96,3 Mbps	A
3	26 ms	11 ms	84,1 Mbps	97,1 Mbps	A

2. Pengujian Video Streaming

Pada tahap pengujian video streaming akan dilakukan pengujian dengan memutar video youtube pada resolusi 4k berdurasi 29 detik. Yang kemudian data traffic akan ditangkap menggunakan wireshark.



Gambar 10. Pengujian Video Streaming



Gambar 11. Analisis Wireshark

Pada pengujian video streaming sebelum dilakukan optimasi diperoleh detail pengujian sebagai berikut.

Tabel 6. Tabel Hasil Pengujian streaming video

Rata-rata Throughput	Paket loss	Rata-rata Delay	Rata-rata Jitter	Keterangan
698 Kbps	7 paket dari 3.207 paket (0,2%)	10,185 ms	0,444 ms	Selama pengujian video berjalan lancar, tidak ada <i>buffering</i> sama sekali

3. Pengujian Game Online (Counter-Strike 2)

```

CONSOLE
[WARNING] Est avail bandwidth: 788.0M/s
[WARNING] Bytes buffered: 0
[WARNING] Lifetime stats:
[WARNING] Totals:
[WARNING] Rcv: 117,832 bytes 44,985,426 bytes
[WARNING] Rcv: 126,787 bytes 46,193,218 bytes
[WARNING]
[WARNING] Rcv: w seq: 124,780 bytes
[WARNING] Dropped: 1 60 pcts 0.00%
[WARNING] Offender: 1 pcts 0.00%
[WARNING] Dropped: 38 pcts 0.00%
[WARNING] Duplicate: 0 pcts 0.00%
[WARNING] Retrans: 0 pcts 0.00%
[WARNING]
[WARNING] Ping histogram: (10001 total samples)
[WARNING] 0-50 50-100 100-150 150-200 200-250 250-300 300-350
[WARNING] 10889 93897 44 13 0 0 0
[WARNING] 13.8% 85.1% 0.0% 0.0% 0.0% 0.0% 0.0%
[WARNING]
[WARNING] Ping distribution:
[WARNING] 6th 50th 75th 95th 100th
[WARNING] 6th 50th 75th 95th 100th
[WARNING]
[WARNING] Connection quality histogram: (374 measurement intervals)
[WARNING] perfect 6% 97.9% 89.9% 86.6% 75.0% 50.0% +10 dead
[WARNING] 245 23 6 2 0 0 0
[WARNING] 92.2% 6.1% 1.8% 0.0% 0.0% 0.0% 0.0% 0.0%
[WARNING]
[WARNING] Connection quality distribution:
[WARNING] 50th 25th 6th 2nd
[WARNING] 100% 100% 100% 100%
[WARNING]
[WARNING] Latency variance histogram: (100008 total measurements)
[WARNING] <= 1% 1% 1% 1% 1% 1% 1%
[WARNING] 84021 22168 11773 1061 397 10 0
[WARNING] 79.5% 18.9% 3.8% 0.5% 0.3% 0.0%
[WARNING]
[WARNING] Rate stats collected from 1000000 samples:
[WARNING] 64.6 kpts/s 12.7 k/sps
[WARNING] Rcv: 62.0 kpts/s 12.6 k/sps
[WARNING]
[WARNING] Ping 25ms Max latency variance: 4.8ms
[WARNING] Quality: 99.0% (Dropped: 0.00% Warnings: 1.00%)
[WARNING] Bytes buffered: 0

```

Gambar 12. Developer Console



Gambar 4.13 Game Network Monitor

Pada pengujian game dilakukan 1 kali permainan berdurasi sekitar 45 menit berikut adalah tahapan menghitung rata-rata ping pada pengujian game.

$$\begin{aligned} \text{Rata-rata ping} &= \frac{((12,5 * 14.999) + (37,5 * 93.097) + (62,5 * 46) + (87,5 * 15))}{(14.999 + 93.097 + 46 + 15)} \\ &= \frac{(187.487,5) + (3.491.137,5) + (2.875) + (1.312,5)}{108.157} \\ &= \frac{3.682.812,5}{108.157} = 34,05 \text{ ms} \end{aligned}$$

Tabel 7. Tabel Hasil Pengujian Game Online

Rata-rata ping / latensi	Paket Drop	Kualitas Jaringan	% Buffer	Keterangan
34,05 ms	60 Paket dari 242.717 Paket	99%	0%	Selama pengujian berlangsung ada beberapa kali lonjakan latensi akan tetapi tidak berpengaruh dengan pengalaman bermain.

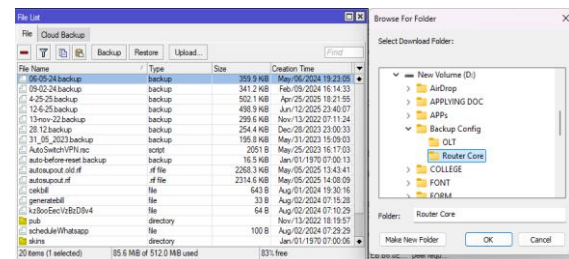
Dari ketiga pengujian dapat disimpulkan jika kondisi jaringan sebelum dilakukan optimasi sudah baik akan tetapi masih ada sedikit bufferbloat.

5.1.3. Upgrade RouterOS dan Optimasi Jaringan

Tahapan ini merupakan tahapan paling penting pada penelitian ini karena akan menentukan berhasil atau tidak penelitian ini. Pada tahapan ini akan dilakukan proses upgrade routerOS dari RouterOS 6 ke Router OS 7 menggunakan antarmuka winbox.

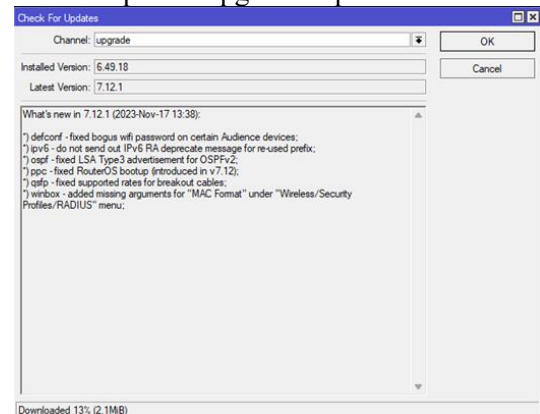
4.1.3.1 Upgrade RouterOS

Sebelum melakukan proses upgrade akan dilakukan proses backup konfigurasi hal ini dimaksudkan untuk keamanan konfigurasi jika terjadi hal yang tidak diinginkan.



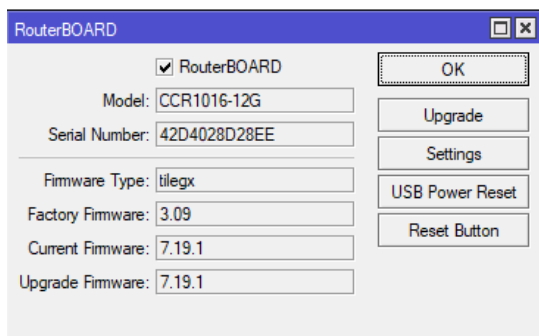
Gambar 14. Backup Mikrotik

Setelah backup konfigurasi selesai, kemudian proses upgrade dapat dilakukan.



Gambar 15. Upgrade

Dapat dilihat jika Router sebelum diupgrade berada pada versi 6.49.18 kemudian jika diupgrade maka akan menjadi versi 7.12.1.

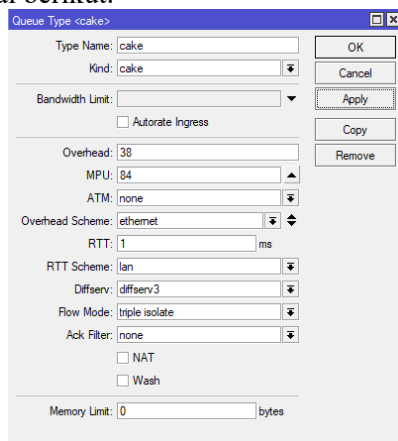


Gambar 16. Selesai Upgrade

Pada gambar 4.15 terdapat data dari RouterBOARD sudah menjadi 7.19.1 hal ini terjadi karena setelah dilakukan upgrade ke versi 7.12.1 terdapat update ke versi 7.19.1 jadi saya putuskan untuk sekaligus melakukan upgrade ke versi 7.19.1.

4.1.3.2 Upgrade RouterOS

Tahap optimasi ini akan berfokus kepada konfigurasi pada QoS Mikrotik, pada sistem Mikrotik sendiri QoS ada pada fitur Queue sebagai berikut.



Gambar 17. Queue Type – CAKE

Langkah pertama, menambahkan queue type dengan algoritma CAKE dengan detail konfigurasi sebagai berikut.

1. Type Name

Type name merupakan nama dari queue type yang akan dibuat, disini queue type diberi nama “cake”.

2. Kind

Pada kolom kind berisikan algoritma atau queue type yang akan dipakai, untuk optimasi menggunakan CAKE maka disesuaikan pada kolom ini diisi dengan “cake”.

3. Bandwidth Limit

Bandwidth limit disini sama dengan fitur rate pada queue type PCQ. Pada kolom ini

dapat diisi dengan batas kecepatan bandwidth. Pada penelitian ini tidak ada batasan limit bandwidth jadi kolom dikosongkan.

4. Autorate Ingress

Pada check box Autorate Ingress ini digunakan untuk menyesuaikan apakah jaringan yang dipakai fluktuatif atau tidak. Sebagai contoh jaringan nirkabel atau wireless merupakan jaringan yang fluktuatif atau tidak stabil, jika jaringan yang dipakai tidak stabil fitur Autorate Ingress ini dapat dipakai karena memungkinkan sistem untuk otomatis menyesuaikan batas bandwidth berdasarkan kondisi jaringan. Karena penelitian ini menggunakan jaringan yang stabil atau kabel maka fitur ini tidak perlu diaktifkan.

5. Overhead

Kolom ini digunakan untuk menentukan penambahan paket header pada jaringan, pada optimasi ini kolom sudah diisi otomatis karena menggunakan overhead scheme dengan angka 38.

6. MPU (Minimum Packet Unit)

Kolom MPU digunakan untuk menentukan minimum paket unit. Jika banyak paket sudah melebihi minimum paket unit maka algoritma akan bekerja dengan melakukan control delay pada paket berikutnya. Pada penelitian ini diperoleh angka 84 sesuai dengan overhead scheme.

7. ATM (Asynchronous Transfer Mode)

Pada kolom ATM digunakan untuk jaringan khusus yang menggunakan jaringan ADSL atau VPN. Karena jaringan yang dipakai untuk penelitian bukan merupakan jaringan ADSL atau VPN jadi kolom ini kosong.

8. Overhead Scheme

Kolom overhead scheme digunakan untuk menentukan skema jaringan. Pada penelitian ini digunakan skema “ethernet” menyesuaikan kondisi jaringan yang merupakan jaringan lokal dan menggunakan ethernet. Dengan ditentukannya skema pada kolom ini akan otomatis mengisi nilai pada kolom Overhead dan MPU.

9. RTT (Round Trip Time)

RTT adalah perkiraan waktu perjalanan suatu paket dari pengiriman hingga kembali

lagi. Pada penelitian ini RTT berada pada angka 1 ms. Kolom ini diisi otomatis dengan menentukan RTT Scheme.

10. RTT Scheme

RTT scheme digunakan untuk menentukan skema RTT menyesuaikan kondisi jaringan yang digunakan. Karena penelitian menggunakan jaringan LAN maka RTT scheme menggunakan “lan”.

11. Diffserv (Different Service)

Kolom ini menentukan klasifikasi layanan sesuai dengan lalu lintas jaringan. Pada penelitian ini digunakan layanan default yakni “diffserv3”.

12. Flow Mode

Pada kolom konfigurasi ini digunakan untuk mengatur mode aliran mana yang dilakukan isolasi ataupun dibuka. Pada penelitian ini menggunakan pengaturan default “triple isolate”.

13. Ack Filter (Acknowledge Filter)

Ack Filter digunakan untuk menambahkan filter khusus yang berhubungan dengan algoritma CAKE dengan pilihan agresive, filter dan none. Menyesuaikan kebutuhan jaringan yang digunakan pada penelitian jadi pada kolom ini dipilih “none”.

14. NAT

Opsi NAT digunakan untuk mengontrol bagaimana algoritma CAKE menangani lalu lintas NAT. Kontrol ini akan membiarkan layanan NAT dari router yang ada dibawah router, untuk lewat tanpa harus dilakukan delay ataupun mengisolasi paket dari layanan tersebut.

15. Wash

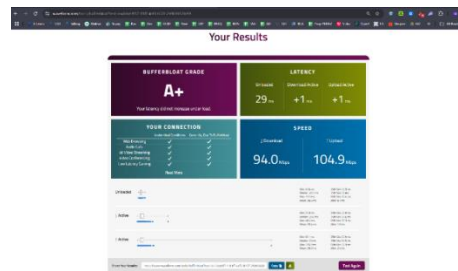
Jika fitur Wash ini diaktifkan maka sistem akan memeriksa ulang paket yang melewati router. Setiap paket memiliki tingkat prioritas yang ditandai dengan label DSCP (Differentiated Service Code Point) pada setiap paket. Jika label tersebut hilang atau tidak sesuai maka sistem akan memberikan label DSCP kembali yang sesuai dengan jenis paket tersebut.

16. Memory Limit

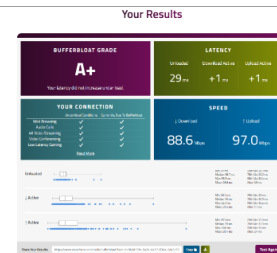
Kolom ini digunakan untuk membatasi kapasitas memory yang digunakan pada router. Karena pada penelitian ini memory tidak dibatasi maka dimasukan angka 0.

5.1.4. Pengujian Setelah Optimasi

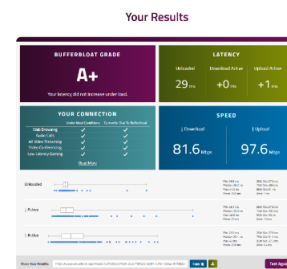
1. Pengujian Bufferbloat



Gambar 18. Pengujian Bufferbloat 1



Gambar 19. Pengujian Bufferbloat 2



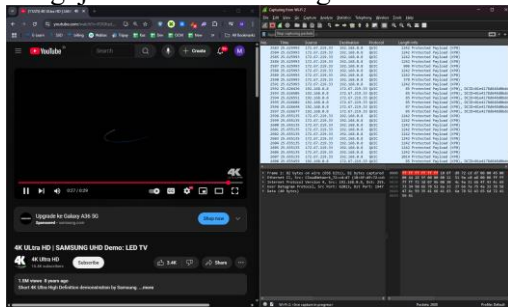
Gambar 20. Pengujian Bufferbloat 3

Pada pengujian bufferbloat dengan menggunakan Waveform setelah dilakukan optimasi terjadi peningkatan hasil yang sangat signifikan dengan ketiga hasil pengujian mendapatkan grade A+ dan latensi tertinggi hanya 1 ms. Berikut adalah detail hasil pengujian bufferbloat.

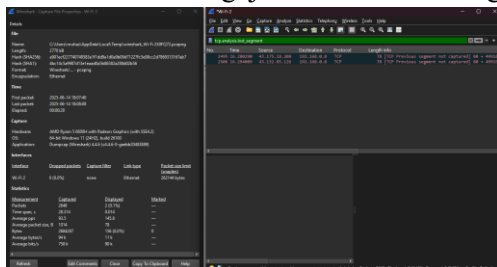
Tabel 7. Tabel Hasil Pengujian Bufferbloat

No	Latensi Download	Latensi Upload	Kecepatan Download	Kecepatan Upload	Nilai
1	1 ms	1 ms	94 Mbps	104,9 Mbps	A+
2	1 ms	1 ms	88,6 Mbps	97 Mbps	A+
3	0 ms	1 ms	81,6 Mbps	97,6 Mbps	A+

2. Pengujian Video Streaming



Gambar 4. 21 Pengujian Video Streaming

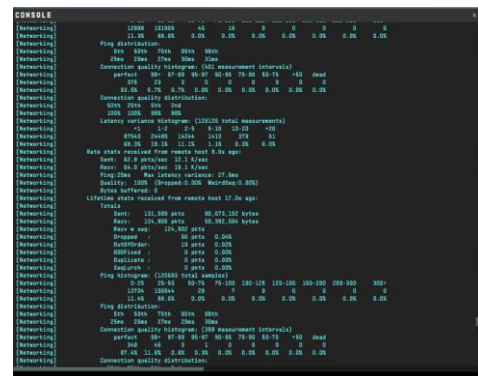


Gambar 4. 22 Analisis Wireshark
Pada pengujian video streaming setelah dilakukan optimasi diperoleh data dengan detail sebagai berikut.

Tabel 8. Tabel Hasil Pengujian Video Streaming

Rata-rata Troughput	Paket loss	Rata-rata Delay	Rata-rata Jitter	Keterangan
758 Kbps	2 paket dari 2.648 paket (0,1%)	10,692 ms	0,017 ms	Selama pengujian video berjalan lancar, tidak ada buffering sama sekali. Terdapat perubahan signifikan pada rata-rata jitter.

3. Pengujian Game Online (Counter-Strike 2)



Gambar 4. 23 Developer Console



Gambar 4. 23 Game Network Monitor

Berikut adalah tahapan menghitung rata-rata ping pada pengujian setelah dilakukan optimasi.

Rata-rata ping

$$= \frac{((12,5 * 13.704) + (37,5 * 106.944) + (62,5 * 28) + (87,5 * 7))}{(13.704 + 106.944 + 28 + 7)}$$

$$= \frac{(171.300) + (4.010.400) + (1.750) + (612,5)}{120.683}$$

$$= \frac{4.184.062,5}{120.683} = 34,67 \text{ ms}$$

Tabel 9. Hasil Pengujian Game Online

Rata-rata ping / latensi	Paket Drop	Kualitas Jaringan	% Buffer	Keterangan
34,67 ms	56 Paket dari 256.495 Paket	100%	0%	Selama pengujian berlangsung ada beberapa lonjakan latensi akan tetapi tidak berpengaruh dengan pengalaman

				bermain dan setelah dilakukan optimasi tidak ada perubahan signifikan apapun.
--	--	--	--	---

Pengujian game setelah optimasi dilakukan dalam 1 permainan dengan durasi sekitar 50 menit. Dapat dilihat jika tidak perubahan pada pengujian game. Setelah pengujian sesudah optimasi selesai diperoleh data perbandingan.

Tabel 10. Tabel Perbandingan Hasil Pengujian

Pengujian	Sebelum Optimasi	Sesudah Optimasi
Waveform Bufferbloat Test	-Grade Pengujian = B, A dan A -Rata-rata download latensi = 29 ms -Rata-rata upload latensi = 6,66 ms -Rata-rata download = 97,8 Mbps -Rata-rata upload = 85,96 Mbps	- Grade Pengujian = A+, A+ dan A+ - Rata-rata download latensi = 0,66 ms - Rata-rata upload latensi = 1 ms - Rata-rata download = 88,06 Mbps - Rata-rata upload = 99,83 Mbps
Video Streaming	- Rata-rata troughput = 698 Kb/s - Paket loss = 7 paket dari 3.207 paket (0,2%) - Rata-rata delay = 10,185 ms - Rata-rata jitter = 0,444 ms	- Rata-rata troughput = 758 Kb/s - Paket loss = 2 paket dari 2648 paket (0,1%) - Rata-rata delay = 10,692 ms - Rata-rata jitter = 0,017 ms
Game Online (Counter-Strike 2)	- Total paket = 242.717 paket - Paket drop = 60 paket (0,05%)	- Total paket = 256.495 paket

	- Kualitas Jaringan = 99,0% - Persentase buffer = 0%	-Paket drop = 56 paket (0,04%) -Kualitas Jaringan = 100,0% -Persentase buffer = 0%
--	---	--

Dari seluruh proses pengujian dan optimasi terdapat penurunan rata-rata latensi pada pengujian bufferbloat senilai 97,46% (dari 26ms menjadi 0,66 ms) untuk download dan 84,98% (dari 6,66 ms menjadi 1 ms) untuk upload. Sedangkan pada pengujian Video Streaming terjadi 96,17% peningkatan pada rata-rata jitter (dari 0,444 ms menjadi 0,017 ms). Terakhir pada pengujian Game Online tidak ada perubahan signifikan pada hasil pengujian.

6. KESIMPULAN

Berdasarkan dari hasil penelitian dan optimasi jaringan menggunakan Common Application Kept Enhanced, dapat disimpulkan jika fitur CAKE pada router Mikrotik bekerja efektif dalam mengatasi bufferbloat. Hasil pengujian menunjukkan perubahan pada 3 percobaan pengujian bufferbloat. Dengan detail pada pengujian bufferbloat menggunakan Waveform mengalami perubahan signifikan terutama pada rata-rata latensi download dan upload hingga menjadi 1 ms. Akan tetapi terjadi penurunan rata-rata bandwidth download sebesar 9,74 Mbps. Selanjutnya, untuk pengujian video streaming dapat disimpulkan bahwa perubahan paling signifikan terjadi pada rata-rata jitter yang tadinya menyentuh angka 0,444 ms setelah dilakukan optimasi hanya menjadi 0,017 ms. Terjadi penurunan jitter hingga 0,427 ms, 26 kali lipat dari sebelumnya. Terakhir untuk pengujian game online menggunakan Counter-Strike 2 tidak terjadi perubahan signifikan pada pengalaman bermain dan juga hasil pengujian jaringan. Berdasarkan hasil penelitian ini disarankan untuk dilakukan pengujian lebih lanjut dengan skenario dan aplikasi yang lebih beragam untuk lebih memvalidasi efektivitas optimasi jaringan dengan CAKE secara menyeluruh. Seperti pengujian dengan banyak pengguna jaringan atau pengujian menggunakan tools atau aplikasi lainnya.

1. UCAPAN TERIMA KASIH

Sebagai penulis, saya ingin menyampaikan rasa syukur dan terima kasih tulus kepada Dosen Pembimbing, Keluarga, Teman-Teman dan Seluruh pihak yang berkontribusi dalam penelitian saya. Setiap bentuk dukungannya, sekecil apapun sangat berarti untuk saya.

2. DAFTAR PUSTAKA

- [1] M. I. R. Febriansah, "Analisis Bottleneck dan Bufferbloat pada AQM Droptail, RED dan SFQ di Komunikasi Data TCP Newreno," *J. Repos.*, vol. 2, no. 9, pp. 1213–1224, 2020, doi: 10.22219/repositor.v2i9.748.
- [2] H. Wang, "Trade-off queuing delay and link utilization for solving bufferbloat," *ICT Express*, vol. 6, no. 4, pp. 269–272, 2020, doi: 10.1016/j.ict.2020.05.008.
- [3] V. V. and M. B. P., "Optimization of Video Streaming through Bufferbloat Mitigation using Hybrid FQ-CoDel Algorithm," *Int. J. Comput. Appl.*, vol. 142, no. 12, pp. 32–36, 2016, doi: 10.5120/ijca2016909957.
- [4] J. Ye, K.-C. Leung, and S. H. Low, "Combating Bufferbloat in Multi-Bottleneck Networks: Theory and Algorithms," *IEEE/ACM Trans. Netw.*, vol. 29, no. 4, pp. 477–1493, 2021, doi: 10.1109/TNET.2021.3066505.
- [5] A. Roy, J. L. Pachua, and A. K. Saha, "An overview of queuing delay and various delay based algorithms in networks," *Computing*, vol. 103, no. 10, pp. 2361–2399, 2021, doi: 10.1007/s00607-021-00973-3.
- [6] D. B. Pratama, D. B. Santoso, and R. Rahmadewi, "Analisis Performa Jaringan Software Defined Networking dengan Algoritma Random Early Detection," *Power Elektron. J. Orang Elektro*, vol. 12, no. 1, 2023, doi: 10.30591/polektro.v12i1.4309.
- [7] Y. Y. Manuputty and I. R. Widiyari, "Analisis Performa Jaringan Software Defined Networking dengan Algoritma Random Early Detection," *J. JTIK (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 8, no. 2, pp. 324–332, 2024, doi: 10.35870/jtik.v8i2.1745.
- [8] V. Bothra, A. Peer, V. K. Singh, M. Maity, and R. Shah, "CoDel-ACT: Realizing CoDel AQM for Programmable Switch ASIC," *2024 IFIP Netw. Conf. (IFIP Networking)*, pp. 468–474, 2024, doi: 10.23919/IFIPNetworking62109.2024.10619729.
- [9] F. A. Johansson, "Applying FQ-CoDel For Packet Schedulers In Tunneled Transport Layer Access Bundling," *Karlstads Univ. Institutionen för Mat. och datavetenskap (from 2013)*, 2020.
- [10] A. Chydzinski, "Non-Stationary Characteristics of AQM Based on the Queue Length," *Sensors*, vol. 23, no. 1, 2023, doi: 10.3390/s23010485.
- [11] T. Tiawan *et al.*, "Implementasi Quality Of Service (Qos) dengan Sistem Wifi Hotspot Voucher pada Jaringan," *Sentinel*, 2024, doi: doi.org/10.56622/sentineljournal.v5i1.47.
- [12] S. Folorunsho, O. A. Adenekan, C. Ezeigweneme, and I. C. Somadina, "Optimizing network performance and quality of service with AI-driven solutions for future telecommunications," *Int. J. Front. Eng. Technol. Res.*, vol. 7, no. 1, pp. 073–092, 2024, doi: 10.53294/ijfetr.2024.7.1.0041.
- [13] T. Høiland-Jørgensen, D. Täht, and J. Morton, "Cake - Common Applications Kept Enhanced," *2018 IEEE Int. Symp. Local Metrop. Area Networks*, pp. 37–42, 2018, doi: 10.1109/LANMAN.2018.8475045.
- [14] F. H. I. Pamungkas, R. Satra, and E. I. Alwi, "Buletin Sistem Informasi dan Teknologi Islam Perbandingan Manajemen Bandwidth Menggunakan Metode PCQ (Per Connection Queue) dan HTB (Hirarchical Token Bucket)," *Bul. Sist. Inf. dan Teknol. Islam*, vol. 2, no. 3, 2021, doi: doi.org/10.33096/busiti.v2i3.888.
- [15] S. Adrian and R. Cahyana, "Perancangan Arsitektur Jaringan Kampus Menggunakan Metode Network Development Life Cycle," *J. Algoritma*, vol. 18, no. 2, pp. 545–552, 2022, doi: 10.33364/algoritma/v.18-2.845.