

Pengendalian Kepadatan dalam *Multimedia Streaming* dengan *Active Queue Management*

Ilham Fahmi Kurniawan, Abdullah Alkaff, Nurlita Gamayanti

Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS)

Kampus ITS Sukolilo, Surabaya 60111

e-mail: ilham.fahmik94gmail.com, abealkaff@gmail.com, nurlita.gamayanti@gmail.com

Abstrak-- *Multimedia streaming* yang semakin hari semakin banyak digunakan oleh pengguna internet menyebabkan terjadinya kepadatan pada jaringan internet. Umumnya, algoritma *Drop-Tail Queue Management* digunakan untuk mengatur sistem antrian pada *buffer* pada kasus *multimedia streaming*. Hal ini menyebabkan distorsi yang cukup signifikan. *Active Queue Management* (AQM) adalah suatu algoritma yang mengatur masuknya paket data ke dalam *buffer* sebagai sistem antrian dalam jaringan komputer. Dalam tugas akhir ini, AQM digunakan sebagai solusi untuk mengatur panjang antrian dan waktu tunda dalam *buffer*, dalam kasus ini adalah paket data. Dalam *buffer*, sistem antrian akan diatur pemrosesan paket data tersebut dengan menciptakan antrian virtual sebagai media pemrosesan sementara di dalam *buffer*. Penelitian dilakukan dengan menggunakan proses *multimedia streaming* yang telah dimodelkan menjadi model matematik dan kemudian membandingkan *Quality-of-Service* (QoS) berupa rata-rata waktu tunggu dalam sistem dan nilai *packet-loss* dari algoritma AQM dengan algoritma drop-tail untuk menemukan nilai QoS terbaik. Berdasarkan penelitian tersebut, penulis menyimpulkan bahwa sistem dengan AQM lebih baik dari sistem dengan drop-tail, dibuktikan dengan nilai *packet-loss* tidak di bawah kategori sedang (15%) dan *delay* yang lebih cepat.

Kata kunci: *Active Queue Management, Buffer, Drop-Tail, Multimedia streaming, Quality-of-Service, Sistem antrian*

I. PENDAHULUAN

Meningkatnya popularitas layanan real-time video menyebabkan peningkatan volume transmisi data. Hal tersebut menyebabkan kepadatan jalur jaringan internet yang menghasilkan beberapa permasalahan, seperti panjangnya antrian dan waktu tunda dalam *buffer*. Secara bersamaan, banyak pengguna layanan video seperti YouTube, Google video, dan Netflix merasakan lamanya waktu tunda ketika menonton video yang mana menjadi tantangan yang dihadapi oleh penyedia layanan video. Dengan meningkatnya permintaan waktu tunda yang kecil oleh pengguna internet, waktu tunda yang relatif lambat meningkatkan kesan buruk dari pengguna internet tersebut.

Sampai dengan hari ini, sistem antrian pada *buffer* masih banyak yang menerapkan algoritma *Drop-Tail Queue Management*. Algoritma *Drop-Tail* merupakan algoritma yang sederhana yang mengatur antrian dengan menggunakan router dan switch sebagai penentu paket data yang akan ditiadakan [1]. Algoritma *Drop-Tail* ini menyetarakan setiap paket data tanpa pertimbangan penting tidaknya paket data tersebut, sehingga dapat menyebabkan distorsi yang cukup signifikan. Oleh karena itu, banyak dilakukan penelitian yang membahas solusi dan alternatif untuk mengatasi permasalahan tersebut.

Cukup banyak literatur yang membahas perihal algoritma *Active Queue Management* (AQM) yang dipublikasikan. Sistem AQM telah dikenal sebagai solusi dari permasalahan diatas dengan cara mengatur rata-rata panjang

antrian dan waktu tunda dalam antrian [1]. Sebagian besar ide yang ditawarkan adalah dengan meniadakan secara acak paket data yang masuk ketika jaringan menjadi padat. Oleh karena perbedaan karakteristik dan spesifikasi dari jalur transmisi video dengan jaringan lainnya, dirumuskan sebagai suatu kebutuhan untuk AQM dalam kasus peningkatan *Quality-of-Service* dari *scalable video streaming*.

II. SISTEM ANTRIAN PADA MULTIMEDIA STREAMING

A. Sistem Antrian

Sistem antrian adalah teori yang membahas mengenai apakah terdapat waktu tunda atau tidak dari antrian yang diamati. Pengamatan dapat dilakukan secara langsung maupun tidak langsung untuk menghasilkan data sebagai sumber yang akan dianalisis. Analisis tersebut akan berupa perhitungan yang menunjukkan performa dari sistem antrian tersebut. Tujuan penggunaan teori antrian adalah untuk merancang fasilitas pelayanan, untuk mengatasi permintaan pelayanan yang berfluktuasi secara random dan menjaga keseimbangan antara biaya (waktu menganggur) pelayanan dan biaya (waktu) yang diperlukan selama antri.

Adapun, *Little's Formula* merupakan salah satu teori antrian yang paling berhubungan dekat dengan antrian yang dikembangkan oleh John D. C. Little pada tahun 1960-an [2]. Formula ini berhubungan dengan steady-state rata-rata ukuran sistem dengan *steady-state* rata-rata lama waktu menunggu pelanggan. Dengan T_q merepresentasikan lama waktu pelanggan menunggu dalam sistem dan T merepresentasikan lama waktu pelanggan di dalam sistem keseluruhan ($T = T_q + S$, dimana S adalah waktu pelayanan, dan T , T_q , dan S adalah variabel acak). $W_q = E[T_q]$ dan $W = E[T]$, masing-masing biasa digunakan untuk menunjukkan rata-rata lama waktu menunggu di dalam antrian dan rata-rata lama waktu menunggu di dalam sistem. *Little's Formula* adalah sebagai berikut.

$$L = \lambda W \quad (1)$$

Serta,

$$L_q = \lambda W_q \quad (2)$$

Keterangan:

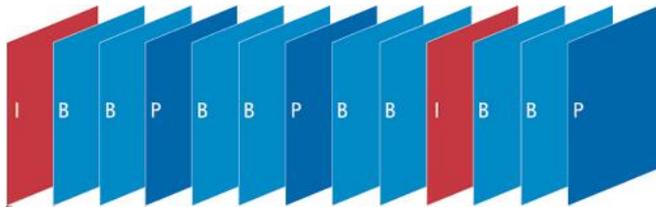
- L adalah panjang antrian dalam sistem
- L_q adalah panjang antrian dalam antrian
- W adalah lama waktu dalam sistem
- W_q adalah lama waktu dalam antrian
- λ adalah laju kedatangan

B. *Multimedia Streaming*

Proses *multimedia streaming* ini dapat dipecah menjadi berbagai jenis, mulai dari teks, gambar, audio, video, sampai dengan animasi. Masing-masing jenis *file* tersebut dapat

dilakukan proses *streaming*. Proses *streaming* jenis *file* tersebut pada dasarnya adalah sama, *video streaming* dibandingkan dengan *audio streaming* melalui proses yang sama. Namun yang membedakan adalah beban yang diterima oleh *server* berbeda-beda tergantung seberapa banyak tipe data yang diolah dan kompleksitas jenis *file* tersebut.

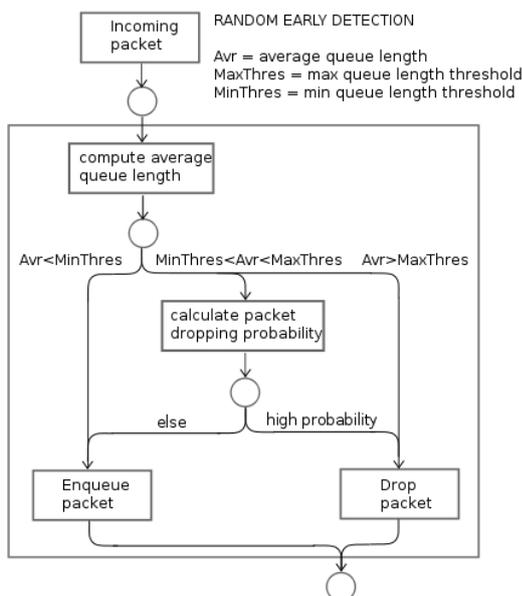
Video streaming dapat diartikan transmisi file video secara berkelanjutan yang memungkinkan video tersebut diputar tanpa menunggu file video tersebut tersampaikan secara keseluruhan [3]. Secara sederhana, *video streaming* adalah layanan internet untuk menonton video tanpa mengunduh video tersebut untuk memutarinya.



Gambar 1. Ilustrasi pengiriman data *video streaming*

C. Active Queue Management

Pendekatan *Active Queue Management* (AQM) secara umum dibagi menjadi dua kategori, yaitu AQM berbasis antrian dan AQM berbasis beban [4]. AQM berbasis antrian, seperti RED dan BLUE, menggunakan panjang antrian sebagai indikator kepadatan dan mengatur laju peniadaan paket data berdasarkan panjang antrian. AQM berbasis beban, seperti REM dan AVQ, menggunakan faktor beban sebagai indikator kepadatan. AQM berbasis antrian tergolong sulit dalam mengkonfigurasi parameter-parameter dan bergantung pada rata-rata panjang antrian dalam kurun waktu yang lama. AQM berbasis antrian tidak sensitif terhadap laju kedatangan paket data dan laju peniadaan paket data. Sedangkan, AQM berbasis beban dapat mempercepat respon, namun dapat menimbulkan ketidakstabilan sistem.



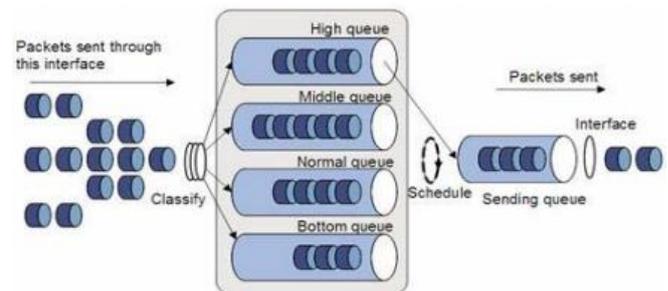
Gambar 2. Algoritma *Random Early Detection* (RED)

D. Drop-Tail Queue Management

Drop-Tail Queue Management adalah algoritma manajemen antrian sederhana yang digunakan pada *router*, yang mana akan meniadakan paket berdasarkan pada panjang antrian. Pada penggunaan algoritma Drop-Tail, semua paket dianggap identik, dalam artian tidak ada prioritas pengiriman

paket data [5]. Dengan demikian, ketika kapasitas antrian telah dipenuhi, maka paket data yang datang akan ditiadakan sampai kapasitas antrian memiliki kapasitas yang cukup untuk menampung paket data yang berdatangan.

Drop-Tail memiliki dua kelemahan, yaitu *lock-out* dan antrian penuh [5]. *Lock-out* adalah kemungkinan satu atau lebih stream dapat memonopoli antrian pada *router*. Antrian penuh maksudnya adalah *router* akan meniadakan paket hanya ketika kapasitas antrian sedang penuh. Solusi untuk permasalahan antrian penuh adalah dengan *router* meniadakan paket sebelum kapasitas antrian terpenuhi seluruhnya.



Gambar 3. Ilustrasi pengimplementasian algoritma Drop-Tail

E. Quality-of-Service (QoS)

Quality-of-Service (QoS) merupakan metode pengukuran tentang seberapa baik jaringan dan merupakan suatu usaha untuk mendefinisikan karakteristik dan sifat dari suatu layanan [6]. QoS digunakan untuk mengukur sekumpulan atribut kinerja yang telah dispesifikasikan dan diasosiasikan dengan suatu layanan.

QoS didesain untuk membantu *user* menjadi lebih produktif dengan memastikan bahwa *user* mendapatkan kinerja yang handal dari aplikasi-aplikasi berbasis jaringan. Komponen-komponen dari QoS adalah *throughput*, *jitter*, *delay*, *loss packet*.

a) Jitter

Jitter merupakan variasi *delay* antar paket yang terjadi pada jaringan. Besarnya nilai *jitter* akan sangat dipengaruhi oleh variasi beban trafik dan besarnya kepadatan antar paket yang ada dalam jaringan tersebut. Semakin besar beban trafik di dalam jaringan akan menyebabkan semakin besar pula peluang terjadinya kepadatan, dengan demikian nilai *jitter*-nya akan semakin besar. Semakin besar nilai *jitter* akan menyebabkan nilai QoS semakin turun. Kategori kinerja jaringan versi *Telecommunications and Internet Protocol Harmonization Over Networks* (TIPHON) mengelompokkan menjadi empat kategori kinerja jaringan berdasarkan nilai *jitter*.

Tabel 1. Tabel performa jaringan berdasarkan *Jitter*

Kategori Degradasi	Peak Jitter
Sangat Bagus	0 ms
Bagus	75 ms
Sedang	125 ms
Jelek	225 ms

b) Latency

Latency dalam jaringan adalah hasil dari dua arah atau *round-trip* (bolak-balik). *Latency* dapat diilustrasikan ketika komputer A terhubung ke komputer B, komputer A mengirimkan data ke komputer B, kemudian

komputer B akan memberikan respon. Jumlah waktu yang dibutuhkan paket berangkat dari komputer A lalu kembali ke komputer A inilah yang dinamakan *latency*. Salah satu contoh kasusnya yaitu, apabila salah satu komputer mengirimkan data sebesar 3 Mbyte pada saat jaringan sepi waktu yang dihabiskan sebesar 5 menit, tetapi pada saat jaringan padat dapat menghabiskan waktu sebesar 15 menit. *Latency* ketika jaringan sibuk berkisar 50-70 msec.

Tabel 2. Tabel performa jaringan berdasarkan *delay*

Kategori <i>Latency</i>	Besar <i>Delay</i>
Sangat Bagus	< 150 ms
Bagus	150 s/d 300 ms
Sedang	300 s/d 450 ms
Jelek	> 450 ms

c) *Packet Loss*

Packet loss merupakan jumlah paket yang hilang saat pengiriman paket data ke tujuan. Paket yang hilang dapat menyebabkan *file* menjadi rusak dan tidak dapat dibaca. Sehingga nilai *packet loss* ini diharapkan minimum. Secara umum terdapat empat kategori penurunan performansi jaringan dengan versi TIPHON and *internet protocol harmonization over networks*, yaitu sebagai berikut.

Tabel 3. Tabel performa jaringan berdasarkan *packet loss*

Kategori Degradasi	<i>Packet Loss</i>
Sangat bagus	0 %
Bagus	3 %
Sedang	15 %
Jelek	25 %

III. PERMODELAN SISTEM ANTRIAN

A. Gambaran Umum Sistem

Pada tugas akhir ini digunakan sebuah sistem yang telah dirancang sendiri dan dimodelkan berdasarkan referensi peneliti-peneliti sistem yang sebelumnya. *Plant* utama yang digunakan pada tugas akhir ini merupakan jaringan komputer. Jaringan komputer ini meliputi *server* dan workstation atau *user* yang mengakses jaringan komputer tersebut. Service yang difokuskan pada penelitian ini adalah *video streaming*, dimana video yang digunakan adalah yang telah tersimpan pada *database server*. Proses *video streaming* yang dapat disimulasikan dimulai pada proses *read file*. Proses awal sampai dengan proses terhubungnya jaringan untuk *streaming* merupakan tahapan proses yang sulit untuk dapat dimodelkan.



Gambar 3. Proses *streaming* yang dimodelkan

Berdasarkan gambar diatas, sistem antrian dimulai setelah jaringan telah terhubung, sehingga proses yang dijalankan hanya melalui 3 tahapan, yaitu *read file*, *write buffer*, dan *send buffer*. Peran dan fungsi masing-masing proses akan dijelaskan pada sub-bab selanjutnya. Adapun hal yang penting untuk diperhatikan adalah pelanggan atau *customer* dalam sistem antrian yang dimodelkan ini akan disebut sebagai paket data dan roses *multi stream* yang terjadi pada *buffer* akan dimodelkan menjadi sistem antrian dengan *n user* dan *c server*.

B. Perancangan Model Matematis Sistem

Model matematis sistem berfungsi untuk menginterpretasikan sistem dan permasalahannya dalam bentuk numerik yang lebih detail. Model matematis menjelaskan fungsi objektif dan fungsi kendala dari sistem tersebut. Adapun notasi sistem antrian yang dirancang ini yaitu M/M/c/K, yang berarti sistem antrian *multi server* dengan kapasitas terbatas.

a) Fungsi objektif

Fungsi objektif dapat dianggap pula sebagai tujuan dari sistem. Fungsi objektif dari sistem antrian adalah meminimalkan lama waktu paket data di dalam sistem. Berdasarkan *Little's Formula* terkait performa sistem antrian *multi server* dengan kapasitas terbatas (M/M/c/K), serta mempertimbangkan algoritma AQM berbasis antrian. Dapat dirumuskan fungsi objektif sebagai berikut.

$$\min \sum_{n=1}^c \bar{W}_{q_n} + 1/\bar{\mu}_n \tag{3}$$

Keterangan:

- \bar{W}_q adalah rata-rata waktu paket di dalam antrian
- $\bar{\mu}$ adalah rata-rata laju pelayanan *server*
- *c* adalah jumlah *server*

Persamaan (3) dapat digunakan untuk mencari *delay* selama proses *read file* sampai dengan proses *send buffer*. Semakin kecil nilai fungsi objektif maka semakin kecil pula nilai *delay*. Sehingga performa sistem antrian akan menjadi lebih baik.

b) Fungsi kendala

Fungsi kendala digunakan sebagai syarat-syarat yang harus dipenuhi sebelum mencari nilai dari fungsi objektif. Dalam kasus ini syarat-syarat yang harus dipenuhi merupakan syarat-syarat yang berhubungan dengan waktu *time-out* dan kapasitas fisik *buffer*.

$$\bar{W}_n \leq \bar{W}_{max} \tag{4}$$

$$\frac{1}{\bar{\mu}_n} \leq \frac{1}{\bar{\mu}_{max}} \tag{5}$$

$$\sum_{n=1}^c K_n \leq K_{max} \tag{6}$$

Keterangan:

- \bar{W} = Rata-rata waktu paket data di dalam sistem
- \bar{W}_{max} = Rata-rata waktu *time-out* dalam sistem
- $\bar{\mu}$ = Rata-rata laju pelayanan *server*
- $\bar{\mu}_{max}$ = Rata-rata laju paket yang *time-out*
- K* = Kapasitas antrian masing-masing *server*
- K_{max} = Kapasitas fisik *buffer*
- c* = Jumlah *server*

Persamaan (4) digunakan untuk memastikan waktu paket data dalam sistem yang melalui antrian virtual tidak melebihi waktu *time-out* yang akan terpasang sesuai kebijakan *time-out* masing-masing *server*. Apabila waktu yang dihabiskan oleh paket data melebihi waktu *time-out* maka paket data akan ditiadakan. Paket data yang telah ditiadakan ini akan diproses oleh *server* sesuai kebijakan *time-out* dengan pilihan paket data akan diabaikan atau dilakukan *request* ulang terkait paket data tersebut.

Persamaan (5) digunakan untuk memastikan bahwa waktu pelayanan tidak melebihi waktu *time-out* yang diatur untuk mengontrol kepadatan. Pengontrolan kepadatan melalui waktu *time-out* pada proses *write buffer* dapat dilakukan dengan memperpanjang waktu *time-out* atau mempersingkat waktu *time-out*. Semakin singkat waktu *time-*

out maka akan semakin singkat pula *delay* dari sistem antrian. Namun, hal ini akan berimbas pada *packet loss* yang memperburuk kualitas video.

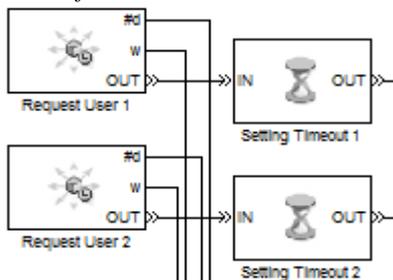
Sedangkan persamaan (6) mempertegas eksistensi *buffer*. Diketahui *buffer* merupakan *memory* yang bersifat sementara. Nilai *memory buffer* yang dapat berubah-ubah membuat *buffer* menjadi sistem antrian dengan kemampuan beradaptasi sesuai permintaan (*demand*). Meski demikian, kapasitas masing-masing antrian virtual pada *buffer* tidak dapat melebihi kapasitas fisik *buffer*, yang direpresentasikan dengan nilai *memory* dari *temporary buffer*.

C. Perancangan Simulasi

Dalam merancang simulasi sistem antrian yang didasari oleh model matematis dan mempertimbangkan sistem antrian. Maka harus didapatkan data awal sebagai input dari simulasi sistem tersebut. Data-data yang penting didapat adalah rata-rata jumlah *request* dari para *user* yang menggunakan sistem tersebut sebagai laju kedatangan, rata-rata nilai *memory temporary buffer* sebagai nilai maksimal kapasitas antrian, rata-rata nilai *delay* pada pemrosesan *request* sebagai waktu pelayanan, nilai *delay* terbesar yang dapat ditoleransi sistem sebagai waktu *time-out*, dan jumlah server yang didapat setelah menentukan beban kerja antrian.

- a) Waktu antar kedatangan
Berdasarkan *incoming request* dari satu atau lebih user yang diolah maka dapat diketahui besarnya waktu antar kedatangan.
- b) Kapasitas antrian
Rata-rata nilai *memory temporary buffer* pada tahap peng-ambilan data yang kemudian diolah dan dikonversi maka didapatkan nilai kapasitas antrian.
- c) Waktu pelayanan
Idealnya waktu pelayanan diambil dari nilai *delay* pada saat pemrosesan *request*.
- d) Waktu *time-out* paket data di dalam sistem
Nilai ini didapatkan berdasarkan *delay* terbesar yang ditoleransi ketika proses pengambilan data. Data tersebut dijadikan data inisial setelah dikonversi berdasarkan *simulation time* pada SimuLink.
- e) Jumlah server
Jumlah server dapat ditentukan dengan menghitung beban kerja antrian dan menentukan bilangan yang membulatkan beban kerja tersebut. Jumlah server menentukan proses simulasi sistem antrian.
- f) Proses multimedia streaming pada simulasi
Dengan meninjau gambar 3, diketahui bahwa proses multimedia streaming memiliki peran dalam simulasi sistem antrian ini. Hubungan antara proses *read file*, *write buffer*, dan *send buffer* adalah sebagai berikut.

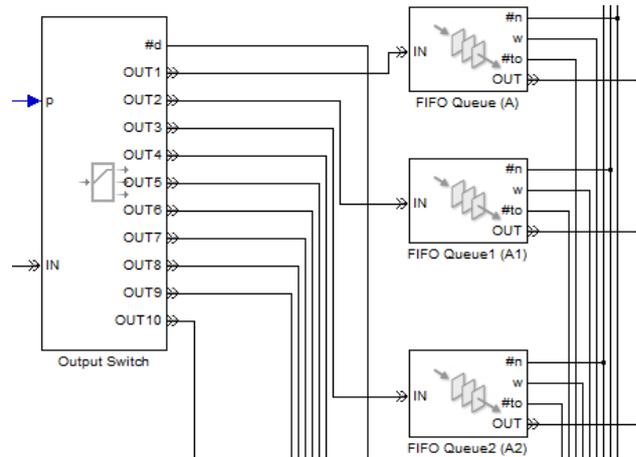
a. *Read file*



Gambar 4. Peran read file pada simulasi sistem

Proses *read file* merupakan proses membaca *file* sebelum pengiriman. Dalam simulasi proses ini berperan sebagai tahap kedatangan paket data dan pengaturan *time-out*.

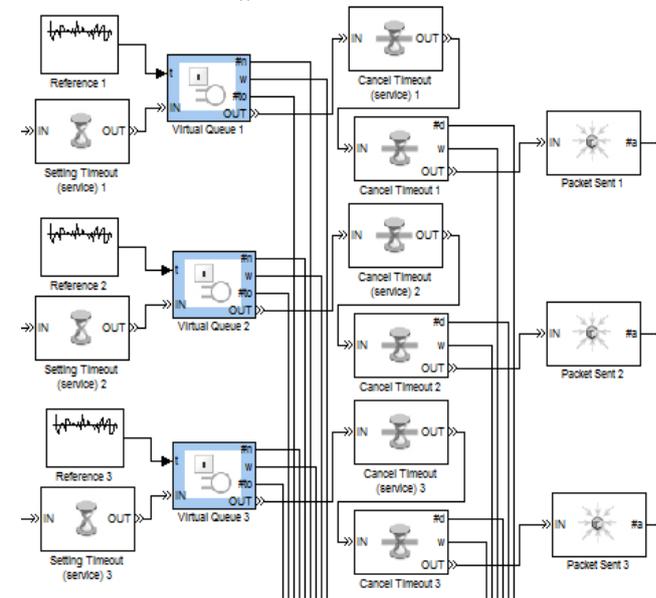
b. *Write buffer*



Gambar 5. Peran write buffer pada simulasi sistem

Proses *write buffer* merupakan proses *encoding* dan penyimpanan data yang telah di-*encode*. Oleh karena itu, *memory buffer* bersifat sementara karena dapat berubah-ubah sesuai dengan banyak dan kompleksitas paket data.

c. *Send buffer*



Gambar 6. Peran send buffer pada simulasi sistem

Proses ini merupakan proses pengiriman paket data dari *buffer* menuju komputer *user*. Pada proses inilah terjadi pelayanan yang bermaksud untuk merepresentasikan proses transmisi.

IV. SIMULASI DAN ANALISIS DATA

Pengujian ini dilakukan bertujuan untuk mengetahui QoS dari sistem antrian yang telah disimulasikan di SimuLink sebagai bentuk representatif dari sistem antrian di *buffer*. Selain itu juga untuk mengetahui kelebihan dan kekurangan dari sistem antrian yang telah dirancang. Dari pengujian tersebut kemudian dapat dibandingkan dan dianalisis kekurangan, kelebihan, dan batasan-batasan dari masing-masing algoritma.

Pengujian dilakukan dengan memberikan asumsi besarnya masing-masing paket data. Hal ini dilakukan untuk mendapatkan nilai kapasitas antrian (K). Diasumsikan pula nilai paket data yang berdatangan sebesar 25 KB dan 45 KB.

Nilai tersebut dijadikan acuan besar masing-masing kapasitas antrian virtual (K), yaitu 108 dan 60. Adapun simulasi dilakukan dengan waktu simulasi (*simulation stop-time*) senilai 50000. Waktu simulasi tersebut ditentukan agar dapat mengamati kepadatan yang terjadi, dalam hal ini disebut sebagai *volume requests*.

Hasil pengujian yang dilakukan pada simulasi sistem antrian dengan menggunakan algoritma AQM dan algoritma drop-tail adalah sebagai berikut.

Tabel 4. Hasil simulasi sistem antrian AQM (K = 60)

Volume Requests	Packet-loss (%)	Total Delay (ms)	Jitter (ms)
100	3.65606039	367.671217	65.19304
200	4.93768165	613.271845	66.67624
300	6.0140374	1047.13556	68.81581
400	5.84334568	1557.99444	67.85365
500	9.54797071	2160.2805	66.21581

Tabel 5. Hasil simulasi sistem antrian AQM (K = 108)

Volume Requests	Packet-loss (%)	Total Delay (ms)	Jitter (ms)
100	5.63985513	419.775343	65.77412
200	5.39914921	613.271845	66.67624
300	6.03885607	1040.40272	68.93648
400	5.83466294	1557.71953	67.8214
500	6.15122882	1787.71267	68.08448

Tabel 6. Hasil simulasi sistem antrian drop-tail (K = 60)

Volume Requests	Packet-loss (%)	Total Delay (ms)	Jitter (ms)
100	0	318.418706	58.33254
200	0	423.638057	60.00564
300	0	714.767344	68.64688
400	6.48584906	1283.7189	76.34984
500	22.308064	2781.1245	69.24308

Tabel 7. Hasil simulasi sistem antrian drop-tail (K = 108)

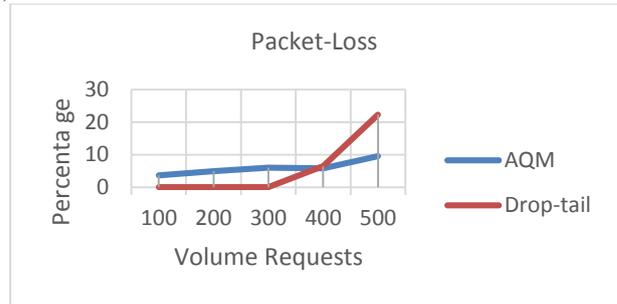
Volume Requests	Packet-loss (%)	Total Delay (ms)	Jitter (ms)
100	0	340.494951	51.21627
200	0	451.646844	57.76633
300	0	667.709829	54.34265
400	5.02422201	823.726267	71.52556
500	41.1002635	2169.33842	68.2211

Nilai dari *packet-loss*, *total delay*, dan *jitter* pada tabel 4, tabel 5, tabel 6, dan tabel 7 didapatkan dari nilai rata-rata hasil simulasi yang dilakukan. Nilai *packet-loss* merupakan persentase hasil dari nilai paket yang ditiadakan (*drop*) ditambah nilai paket yang *time-out* (TO) yang kemudian hasil penjumlahan tersebut dibagi oleh jumlah paket data yang telah dan sedang berada di dalam sistem antrian (IN).

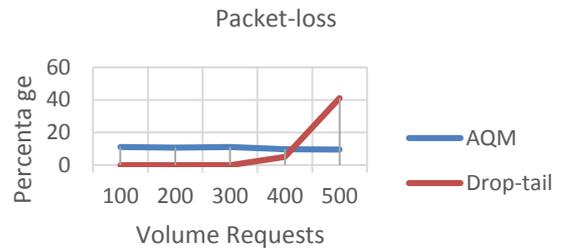
$$Packetloss = \frac{drop + TO}{IN} \times 100$$

Dengan demikian, nilai QoS pada tabel-tabel diatas terhadap *volume requests* menunjukkan kualitas layanan dari model sistem antrian yang dirancang.

1) *Packet-loss*



Gambar 7. Grafik perbandingan packet-loss (K=60)



Gambar 8. Grafik perbandingan packet-loss (K=108)

Packet-loss dari sistem antrian algoritma drop-tail pada saat nilai *volume requests* bernilai 100, 200, 300, dan 400 bernilai lebih kecil dibandingkan dengan *packet-loss* dari sistem antrian algoritma AQM.

2) *Total delay*



Gambar 9. Grafik perbandingan total delay (K=60)

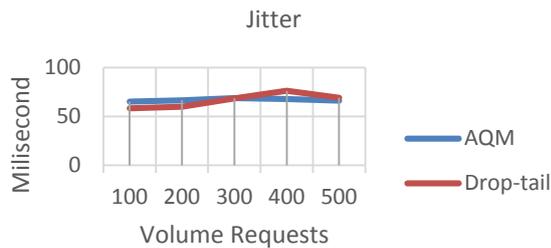


Gambar 10. Grafik perbandingan total delay (K=108)

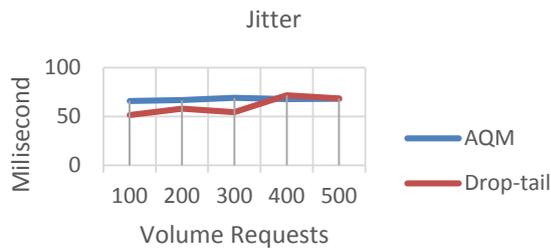
Kondisi pada grafik perbandingan *packet-loss* Total *delay* dari sistem antrian algoritma drop-tail pada saat nilai *volume requests* bernilai 100, 200, 300, dan 400 bernilai lebih

kecil dibandingkan dengan total *delay* dari sistem antrian algoritma AQM.

3) Jitter



Gambar 11. Grafik perbandingan jitter (K=60)



Gambar 12. Grafik perbandingan jitter (K=108)

Nilai *jitter* pada kedua sistem antrian dapat dianggap stabil. Peningkatan dan penurunan nilai *jitter* tidak berubah secara signifikan.

Pada saat *volume requests* mencapai nilai 100, simulasi algoritma drop-tail baru berjalan selama waktu 700-800 sedangkan simulasi algoritma AQM telah berjalan dalam selang waktu 3 sampai 4 kali lipat waktu simulasi algoritma drop-tail. Hal ini kemudian menyebabkan jumlah paket data yang telah dan sedang berada dalam sistem terjadi perbedaan yang cukup signifikan (terlampir pada lampiran C). Sehingga paket data yang terakumulasi pada sistem antrian algoritma AQM menumpuk dan sebaliknya pada sistem antrian algoritma drop-tail belum terjadi hal serupa. *Trend* ini juga terjadi pada *volume requests* 200, 300, dan 400. Sehingga, ketika *volume requests* mencapai nilai 500 maka pada sistem antrian algoritma drop-tail baru terjadi penumpukan paket yang hampir sama dengan sistem antrian algoritma AQM.

Analisis tersebut menunjukkan bahwa algoritma AQM dapat mengatur kepadatan atau *volume requests* lebih baik dibandingkan dengan drop-tail, yang ditunjukkan berdasarkan waktu simulasi terhadap *volume requests*. Sehingga, dapat ditentukan apabila kondisi kedua sistem antrian adalah sama maka sistem antrian dengan algoritma AQM berbasis antrian lebih baik dibandingkan sistem antrian dengan algoritma drop-tail.

V. KESIMPULAN

Berdasarkan penelitian dan simulasi yang telah dilakukan maka dapat disimpulkan sebagai berikut.

1. Performa sistem antrian pada simulasi sistem antrian dengan algoritma AQM berbasis antrian tergolong sedang yang cenderung baik. Hal ini dibuktikan dengan nilai *packet-loss* tidak di bawah kategori sedang (15%) dan nilai *jitter* tergolong baik.
2. Performa sistem antrian pada simulasi sistem antrian dengan algoritma Drop-Tail tergolong sedang yang cenderung buruk. Hal ini dibuktikan dengan nilai

packet-loss berada di bawah kategori sedang (15%), namun nilai *jitter* tergolong baik.

3. Rata-rata waktu yang dihabiskan paket-paket data di dalam sistem lebih baik pada sistem antrian dengan algoritma AQM berbasis antrian dibandingkan dengan sistem antrian dengan algoritma Drop-Tail jika dinilai pada kondisi yang sama.
4. Kualitas video yang direpresentasikan dengan hasil simulasi sistem antrian, algoritma AQM berbasis antrian lebih baik dibandingkan dengan algoritma Drop-Tail.
5. Simulasi sistem yang dirancang kurang ideal untuk menghasilkan performa sistem antrian yang diharapkan. Hal ini dikarenakan jumlah server yang dirancang pada simulasi sistem antrian kurang memenuhi syarat sistem antrian yang baik.

DAFTAR PUSTAKA

- [1] S. E. Ghoreishi, A. H. Aghvami and H. Saki, "Active Queue Management for Congestion Avoidance in Multimedia Streaming," *European Conference on Networks and Communications (EuCNC)*, 2015.
- [2] O. C. IBE, *Fundamentals of Stochastic Networks*, Lowell: John Wiley & Sons, Inc, 2011.
- [3] M. A. Saleh, H. Hashim and N. M. Tahir, "A Low Computational Method of Secure Video Streaming in Mobile System," *IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, 2014.
- [4] H. Luo, "Improve Delay Performance of Wireless Video Streaming with Active Queue Management," *IEEE*, 2012.
- [5] O. Almomani, O. Ghazali dan S. Hassan, "Performance Study of Large Block FEC with Drop Tail for Video Streaming over the Internet," dalam *First International Conference on Networks & Communications*, 2009.
- [6] M. D. A. Wibowo, "Analisis dan Implementasi Quality of Service (QoS) Menggunakan IPCOP di SMK Muhammadiyah Imogiri," *Naskah Publikasi AMIKOM Yogyakarta*, 2014.